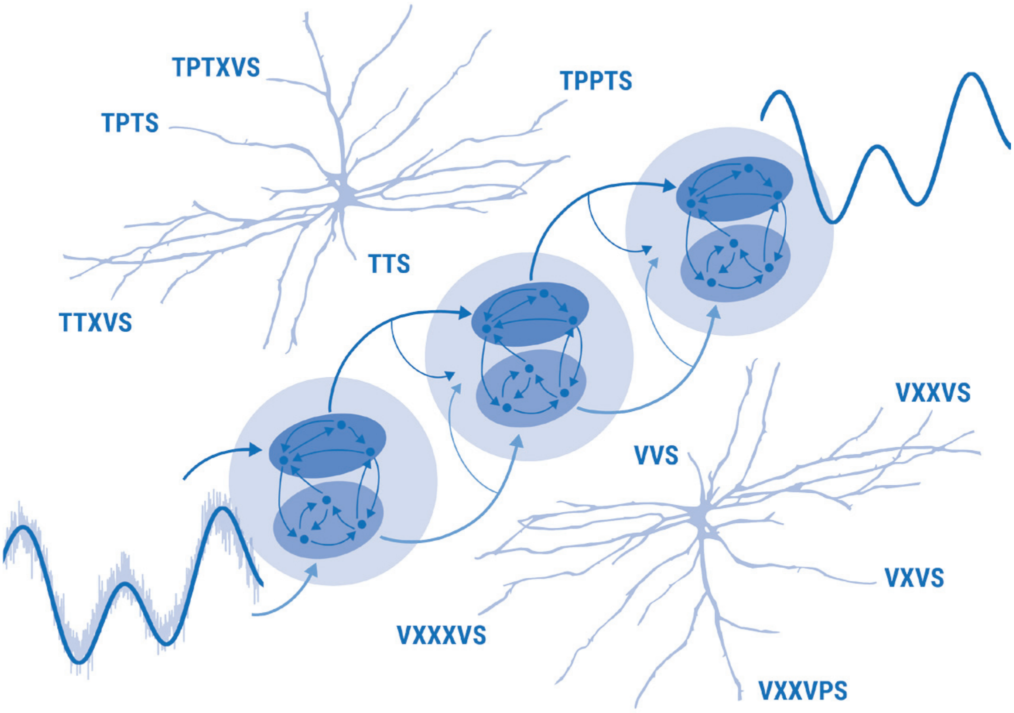
SE RWTH AACHEN UNIVERSITY

Barna Zajzon

Sequential information processing in modular spiking networks



Aachener Informatik-Berichte,
Software Engineering

Band 56

Hrsg: Prof. Dr. rer. nat. Bernhard Rumpe
Prof. Dr. rer. nat. Abigail Morrison

Sequential information processing in modular spiking networks

Von der Fakultät für Mathematik, Informatik und Naturwissenschaften der
RWTH Aachen University zur Erlangung des akademischen Grades
eines Doktors der Naturwissenschaften genehmigte Dissertation

vorgelegt von

M.Sc. RWTH
Barna Zajzon
aus Câmpia Turzii, Rumänien

Berichter: Univ.-Prof. Dr. rer. nat. Abigail Morrison
Jun.-Prof. Michael Schaub, Ph.D.

Tag der mündlichen Prüfung: 08. Mai 2024

Aachener Informatik-Berichte, Software Engineering

herausgegeben von
Prof. Dr. rer. nat. Bernhard Rumpe
Prof. Dr. rer. nat. Abigail Morrison
Software Engineering
RWTH Aachen University

Band 56

Barna Zajzon
RWTH Aachen University

Sequential information processing in modular spiking networks

Shaker Verlag
Düren 2025

Bibliographic information published by the Deutsche Nationalbibliothek
The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available in the Internet at http://dnb.d-nb.de.

Zugl.: D 82 (Diss. RWTH Aachen University, 2024)

Printed in Germany.

Print-ISBN	978-3-8440-9957-7
PDF-ISBN	978-3-8191-0037-6
ISSN	1869-9170
eISSN	2944-6910

Shaker Verlag GmbH • Am Langen Graben 15a • 52353 Düren
Phone: 0049/2421/99011-0 • Telefax: 0049/2421/99011-9
Internet: www.shaker.de • e-mail: info@shaker.de

Eidesstattliche Erklärung

Ich, Barna Zajzon,
erkläre hiermit, dass diese Dissertation und die darin dargelegten Inhalte die eigenen sind und selbstständig, als Ergebnis der eigenen originären Forschung, generiert wurden.

Hiermit erkläre ich an Eides statt

1. Diese Arbeit wurde vollständig oder größtenteils in der Phase als Doktorand dieser Fakultät und Universität angefertigt;
2. Sofern irgendein Bestandteil dieser Dissertation zuvor für einen akademischen Abschluss oder eine andere Qualifikation an dieser oder einer anderen Institution verwendet wurde, wurde dies klar angezeigt;
3. Wenn immer andere eigene- oder Veröffentlichungen Dritter herangezogen wurden, wurden diese klar benannt;
4. Wenn aus anderen eigenen- oder Veröffentlichungen Dritter zitiert wurde, wurde stets die Quelle hierfür angegeben.
5. Diese Dissertation ist vollständig meine eigene Arbeit, mit der Ausnahme solcher Zitate;
6. Alle wesentlichen Quellen von Unterstützung wurden benannt;
7. Wenn immer ein Teil dieser Dissertation auf der Zusammenarbeit mit anderen basiert, wurde von mir klar gekennzeichnet, was von anderen und was von mir selbst erarbeitet wurde;
8. Teile dieser Arbeit wurden zuvor veröffentlicht, ersichtlich im Abschnitt *Publications and contributions.*

Köln, August 2023

Abstract

Molded by evolutionary processes to cope with the statistical regularities in the world, the symbiotic relation between the structure, dynamics and function of the neural machinery underlies all behavioral and cognitive processes. Established paradigms formulate these processes in terms that involve the manipulation of sequentially organized time-discrete (symbolic) representations. This underscores two basic functional requirements that cortical circuits must fulfill: the ability to create suitable representations from a highly volatile and noisy environment; and the capacity to process, and learn from, their spatio-temporal structure.

While the precise mechanisms are largely unknown, these processes must be implemented in the biophysical substrate of the brain, where the complex interactions of neuronal populations can leverage a hierarchical and modular architecture in order to process information on multiple spatial and temporal scales. From a modeler's perspective, one can tackle these problems from two complementary angles: identify some fundamental organizing principles, such as modularity, and try to elucidate their role (bottom-up); or focus on a specific functionality, like sequence processing, and devise possible, biophysically plausible models for it (top-down). Combining software tools, simulation studies and theoretical analysis, this thesis touches upon both approaches over the course of a series of research projects, with the shared goal of disentangling how modular structures enable neural circuits to learn and process sequential information in an efficient and reliable manner.

The first part analyses the characteristics of state representations in modular spiking networks and the architectural and dynamical constraints that influence the system's ability to retain, transfer and integrate stimulus information in the presence of noise. It explores the novel hypothesis that modular topographic maps, a pervasive anatomical feature of the cortex, may provide a structural scaffold for sequential denoising of stimulus representations. By combining modeling with network theory, this thesis demonstrates that the sharpness of topographic projections acts as a bifurcation parameter, controlling the macroscopic dynamics and representational precision of the system. In-depth theoretical analysis unravels the dynamical principles underlying the mechanism, and suggests a robust and generic structural feature that enables a broad range of behaviorally-relevant operating regimes.

The second part of this work is dedicated to investigating existing, biologically detailed models of sequence processing. If we are to harvest the knowledge within these models and arrive at a deeper mechanistic understanding of the involved phenomena, it is critical that the models and their findings are accessible, reproducible, and quantitatively comparable. First, the importance of these aspects are illustrated through a replication study. Building on this, the study lays the initial steps towards a conceptual and practical, theoretically-grounded framework for benchmarking and comparison of

sequence learning models. Through such a meta-analysis study, it aims not only to provide critical evaluation of current models, but also to synthesize their insights into a set of functional and neurobiological features that could be corroborated with experimental data and guide future studies.

Kurzfassung

Durch evolutionäre Progresse geformt um mit den statistischen Regelmäßigkeiten der Welt zurecht zu kommen, unterliegen alle Verhaltens- und Erkenntnisprozessen der symbiotischen Beziehung zwischen Struktur, Dynamik und Funktion.

Etablierte Paradigmen formulieren diese Prozesse in Begriffen, die die Manipulation von sequentiell organisierten zeitdiskreten (symbolischen) Repräsentationen beinhalten. Dies unterstreicht zwei grundlegende funktionelle Anforderungen, die kortikale Schaltkreise erfüllen müssen: die Fähigkeit, geeignete Repräsentationen aus einer hochgradig unbeständigen und verrauschten Umgebung zu erzeugen, und die Fähigkeit, ihre räumlich-zeitliche Struktur zu verarbeiten und daraus zu lernen.

Während die genauen Mechanismen weitgehend unbekannt sind, müssen diese Prozesse im biophysikalischen Substrat des Gehirns implementiert werden, wo die komplexen Interaktionen von Neuronenpopulationen eine hierarchische und modulare Architektur nutzen können, um Informationen auf mehreren räumlichen und zeitlichen Ebenen zu verarbeiten. Aus der Sicht eines Modellierers kann man diese Probleme aus zwei komplementären Blickwinkeln angehen: man kann einige grundlegende Organisationsprinzipien, wie z. B. die Modularität, identifizieren und versuchen, ihre Rolle zu ergründen (Bottom-up); oder man kann sich auf eine spezifische Funktionalität, wie z. B. die Sequenzverarbeitung, konzentrieren und mögliche, biophysikalisch plausible Modelle dafür entwickeln (Top-down). Durch die Kombination von Software-Tools, Simulationsstudien und theoretischen Analysen werden in dieser Arbeit beide Ansätze im Rahmen einer Reihe von Forschungsprojekten verfolgt, mit dem gemeinsamen Ziel, zu entschlüsseln, wie modulare Strukturen neuronale Schaltkreise in die Lage versetzen, sequenzielle Informationen effizient und zuverlässig zu lernen und zu verarbeiten.

Im ersten Teil werden die Eigenschaften von Zustandsrepräsentationen in modularen Spiking-Netzwerken und die architektonischen und dynamischen Beschränkungen analysiert, die die Fähigkeit des Systems beeinflussen, Reizinformationen im Beisein von Rauschen zu behalten, zu übertragen und zu integrieren. Es wird die neuartige Hypothese untersucht, dass modulare topografische Karten, ein weit verbreitetes anatomisches Merkmal des Kortex, ein strukturelles Gerüst für die sequentielle Entrauschung von Stimulusrepräsentationen bieten können. Durch die Kombination von Modellierung und Netzwerktheorie zeigt diese Arbeit, dass die Schärfe der topographischen Projektionen als Bifurkationsparameter fungiert und die makroskopische Dynamik und Repräsentationsgenauigkeit des Systems kontrolliert. Eine eingehende theoretische Analyse entschlüsselt die dynamischen Prinzipien, die dem Mechanismus zugrunde liegen, und legt ein robustes und generisches Strukturmerkmal nahe, das eine breite Palette von verhaltensrelevanten Betriebsregimen ermöglicht.

Der zweite Teil dieser Arbeit widmet sich der Untersuchung bestehender, biologisch detaillierter Modelle der Sequenzverarbeitung. Wenn wir das in diesen Modellen enthal-

tene Wissen nutzen und zu einem tieferen mechanistischen Verständnis der beteiligten Phänomene gelangen wollen, ist es von entscheidender Bedeutung, dass die Modelle und ihre Ergebnisse zugänglich, reproduzierbar und quantitativ vergleichbar sind. Wie wichtig diese Aspekte sind, wird zunächst anhand einer Replikationsstudie veranschaulicht. Darauf aufbauend legt die Studie die ersten Schritte zu einem konzeptionellen und praktischen, theoretisch fundierten Rahmen für das Benchmarking und den Vergleich von Sequenzlernmodellen fest. Durch eine solche Meta-Analyse-Studie soll nicht nur eine kritische Bewertung aktueller Modelle vorgenommen werden, sondern auch eine Synthese ihrer Erkenntnisse in Form einer Reihe funktioneller und neurobiologischer Merkmale, die mit experimentellen Daten untermauert werden können und als Leitfaden für künftige Studien dienen.

Acknowledgments

Pursuing research is a rewarding but also grinding process at times, which can turn into an increasingly solitary affair during the last stretches of writing a thesis. Making it this far, I wish to express my gratitude to the wonderful people who enabled, supported, and towards the end, tolerated me.

First and foremost, I would like to thank Abigail Morrison for providing a jumping board and making it possible to pursue a long-standing dream of, crudely put, "just being near brain research". Your guidance, endless patience, and practical approach to all things related to science, academia and beyond have been of immense value throughout the years. The freedom in shaping my own projects, which perhaps I have not fully taken advantage of, has nevertheless allowed me to grow and explore just about anything I found interesting. Crucial to this was the nurturing yet laid-back and witty atmosphere in the group, where we found not only a space for scientific and technical questions and answers, but were also reminded that there's life outside our bubble.

Renato Duarte, to whom I am deeply grateful for being a tremendous source of inspiration and mentor throughout this journey. I still remember the day, many years ago, when I toured the institute you explained your work in a way that hooked me instantly and led me down this path. Since then, your contagious enthusiasm for and fascination about neuroscience has been a constant source of addiction, and helped keep perspective even in darker times. While the rigor and standards you held me up to were sometimes difficult to attain, it deeply shaped my view and attitude towards science and research in general. In our countless discussions over the years I have always appreciated your honesty and critical thinking, along with the simple but invaluable act of being there whenever I needed advice.

Special thanks goes to my collaborators, without whom none of this would have been possible. David Dahmen, for diving deep into his tricks-of-the-trade bag and helping out with the theory, demonstrating incredible levels of knowledge, pragmatism and efficiency — all this with a smiling and positive attitude, I'm not sure how you pull this off. Aitor Morales-Gregorio, for the short-lived but intense and fruitful collaboration. Younes Bouhadjar, for the many wandering discussions, endless debugging sessions no matter how random the hour, and enduring support right up until the end. Tobias "Tobi" Schulte to Brinke, for being an amazing friend, always managing to hide any stress and ready to cheer up everyone around — I wish our joint project was less agonizing, but, you know, despite all our experiences, spiking networks must be good for *something.*

Everyone else, who, not unlike as glial cells, work and contribute "in the shadows" to maintain one's sanity and keep the machinery running smoothly. Prof. Michael Schaub, for agreeing to act as a reviewer on this thesis. The whole INM-6, NEST and NESTML (particularly Charl Linssen) team, for keeping us busy and perfectly balancing the gained computational efficiency with the time required for tweaking and debugging when implementing new functionality. The old guard, Tom, Philipp, Jyotika, Claudia and old Robin, and the new one, Regi, Anno, Simon, younger Robin, Peter, Moritz L, and of course, the evergreen and chatty maté lovers who are always a treat to be around, Mick and Joshua.

On a final and personal note, I'd like to thank the people whom I care the most about. To Erika Ábrahám and Nils Jansen, for taking me under their wings when I started at RWTH and casting me into the deep waters of research early on. To my closest friends, near and far, Sharan, Robi and Konstantinos, and the ones home, Timi, Emese, Dani, Lori, Isti. To my parents and my Édesmami, who have always encouraged me and my sister to strive for more and be the best version of ourselves, even if this meant leaving home more than a decade ago. For keeping us fed despite being half a continent apart, the art of packaging, and your unconditional love and sacrifices. My best friend and confidant, Nocikám, whom I had the unique pleasure to share the highs and lows of a PhD with — thanks for taking care of me since forever and correcting my graphics, every, single, time, with the latest being the cover illustration of this work. And to my muse, *Keleti Hajnal*. You have stood by me ever since that silly encounter at INCAS and supported me unreservedly through some arduous years — while I can't make up for the lost time, I do hold you responsible for the most beautiful moments in my life, including that coldest December day. This is for you, Gergő.

Köszönöm, hogy vagytok nekem!

Köln, August 2023
Barna Zajzon

Publications and contributions

The work presented in this thesis is in parts based on the following publications of the author:

Transferring state representations in hierarchical spiking neural networks
Barna Zajzon, Renato Duarte, and Abigail Morrison
2018 International Joint Conference on Neural Networks (IJCNN) (2018), pp. 1-9. IEEE.

Parts of this publication enter Chapter 5.
Contribution: Under the supervision of Renato Duarte and Abigail Morrison, the author performed all parts of the above publication. All authors contributed to the design of the numerical experiments and the writing of the manuscript.

Passing the Message: Representation Transfer in Modular Balanced Networks
Barna Zajzon, Sepehr Mahmoudian, Abigail Morrison, and Renato Duarte
Frontiers in Computational Neuroscience 13 (2019)

Chapter 5 is based on this publication, with excerpts entering Chapter 9.
Contribution: Under the supervision of Renato Duarte and Abigail Morrison, the author performed all parts of the above publication. All authors contributed to the design of the numerical experiments and the writing of the manuscript. Preliminary results in this publication were obtained as part of the author's Master's thesiso.

Signal denoising through topographic modularity of neural circuits
Barna Zajzon, David Dahmen, Abigail Morrison, Renato Duarte
Elife 12 (2023): e77009

Chapter 6 is based this publication, with parts of it entering Chapter 9.
Contribution: Under the supervision of Renato Duarte and Abigail Morrison, the author performed all network simulations and contributed to the derivation and implementation of the theory. David Dahmen derived the analytical results and contributed to their implementation and visualization. All authors contributed to the design and analysis of the numerical experiments, and to the writing of the manuscript.

Towards reproducible models of sequence learning: replication and analysis of a modular spiking network with reward-based learning
Barna Zajzon, Renato Duarte, and Abigail Morrison
Frontiers in Integrative Neuroscience (2022)

Chapter 7 is based this publication, with parts of it entering Chapter 9.

Contribution: Under the supervision of Renato Duarte and Abigail Morrison, the author performed all parts of the above publication. All authors contributed to the design of the numerical experiments and the writing of the manuscript.

Sequence learning under biophysical constraints
Barna Zajzon (BZ), Younes Bouhadjar (YB), Jette Oberlaender (JO), Simon Michau (SM), Tom Tetzlaff (TT), Renato Duarte (RD) and Abigail Morrison (AM)
In preparation...

Chapter 8 will constitute the basis of this publication.

Contribution: BZ, YB, TT, RD and AM conceptualized the study. BZ re-implemented the *CS* model, and extended the *KM* model based on an initial version by SM, while YB provided the implemention for *spkTM*. RD implemented the grammar generation and related metrics, and BZ and YB jointly developed the APIs and analysis for the different models. BZ performed all experiments on the *CS* and *KM* models, while YB was responsible for *spkTM*. BZ created all visualizations. All authors have jointly analyzed the presented results. BZ is the sole author of the text in Chapter 8, with RD contributing minor excerpts.

Trans-thalamic Pathways: Strong Candidates for Supporting Communication between Functionally Distinct Cortical Areas
Barna Zajzon, Aitor Morales-Gregorio
Journal of Neuroscience (2019)

Ideas from this opinion paper are briefly touched upon in Chapter 9.

Contribution: Barna Zajzon conceptualized the study. Both authors wrote and reviewed the manuscript. Sacha van Albada provided feedback on the manuscript.

Contents

Chapter 1

Introduction

> *It is the pervading law of all things organic and inorganic, of all things physical and metaphysical, of all things human and all things superhuman, of all true manifestations of the head, of the heart, of the soul, that the life is recognizable in its expression, that form ever follows function. This is the law.*
>
> – Louis H. Sullivan

1.1 Structure, function and the cerebral cortex

"Form follows function" — this adage, coined by Louis H. Sullivan at the turn of the 19th century, has profoundly shaped the following generations of architects and designers, culminating in the Bauhaus movement and giving rise to the design principle that form (structure) should be/is the expression of the purpose (function) (Charles Eames, 1972). This view was challenged early on, with Frank Lloyd Wright suggesting a more holistic perspective in that "form and function should be one, joined in a spiritual union" (1908). Such debates about the intimate relationship between form and purpose, or structure and function, have deep roots in philosophy and science, involving a multitude of disciplines from physics to software engineering. The relevance of the issue is perhaps nowhere more pertaining than in the context of biological systems: if we see a certain structure or organizational principle, most will automatically wonder what function it may serve.

If the goal is to simply understand the functional principles of a particular biological organism or its components (units), determining the causal relation between its structure and its function may not even be necessary. Instead, taking a pragmatic approach in light of available scientific evidence may be more fruitful. Structure, that is the arrangement of elements in a unit, such as the folded structure of proteins or the network of neuronal cells and synapses in the nervous system, mechanistically determines function insofar as it constrains the possible processes within and activity of the unit. Such arrangements permeate all levels of biological organization. In turn, the function, which arises as a consequence of the physical characteristics of the structure and interactions of its elements (Herman et al., 2021), shapes, to some extent, the underlying structure either directly (Henry et al., 2013) or through slower evolutionary processes (Abbot, 1916).

Such reciprocal influences between structure and function are well illustrated by various sensory systems: for instance, the cochlea determines which sound frequencies we can distinguish, but there is indication that in humans, the development of speech strongly shaped the evolution of frequency selectivity patterns (Manley, 2016).

The tight coupling between the two concepts suggests that the complexity of function correlates with the complexity of structure (Spencer, 1866). This relation is perhaps best exemplified by the mammalian brain, where billions of neurons connect through remarkably complex synaptic structures, forming intricate networks that generate behavior, and, ultimately, cognition. All mental processes, from simple motor primitives to language and decision-making, arise from the mosaic of biophysical processes in the brain (Searle, 1980). The concerted interactions of these processes, collectively referred to as neural activity, have been theorized to represent a physical implementation of computation (Searle, 1980; Piccinini and Bahar, 2012). While the exact nature of cognition remains heavily disputed (Globus, 1992; Piccinini and Scarantino, 2011), the "symbiotic" relation between the structural organization and functional aspects of the neuronal machinery is well established (Rubinov et al., 2009). This mutual influence occurs at multiple spatial and temporal scales, not just during development but as a continuous, ongoing process (Sporns, 2011; Pan and Monje, 2020). Neural activity is thus shaped and constrained by the underlying (connectivity) structure (Honey et al., 2007), and in turn, may mold the synaptic structure through activity-dependent mechanisms (Fauth and Tetzlaff, 2016). And, as in all complex systems in the natural domain (Ball, 2009) which "find their place in one or more of four intertwined hierarchic sequences" (Simon, 1977), the connectivity and activity of neural circuits also compose into diverse and characteristic patterns.

One such overarching organization principle in the cerebral cortex is *modular hierarchy*, both at an anatomical (structural) and a functional level (Felleman and Van Essen, 1991; Bullmore and Sporns, 2009; Meunier et al., 2010; Markov and Kennedy, 2013; Park and Friston, 2013). From a graph theory perspective, topological modularity (Newman, 2003) in the cortex is expressed through stereotypical connection motifs between populations of neurons, with neurons connecting densely within and more sparsely outside their module (Sporns, 2011; Hilgetag and Goulas, 2015). While the definition of a *module* in the cortex remains ambiguous, *modularity* permeates all scales of organization with a varying degree of associated functionality. One (although contested, see Horton and Adams, 2005) such unit is the cortical column (Mountcastle, 1997). Its ubiquity plays into the intriguing idea that neural circuits are variations on a common "canonical microcircuit" (Harris and Shepherd, 2015), and may use similar computation principles for very different tasks despite their segregation into functionally specialized cortical areas. Zoom out to this level of larger areas, and the hierarchical character of the neocortex begins to crystalize, most evidently in the configuration of the sensory regions and their processing (Hilgetag and Goulas, 2020). These are characterized by a (global) direction in the laminar projection patterns at a structural level (Felleman and Van Essen, 1991), as well as a progression of spatial and temporal processing scales (Murray et al., 2014) and gradient of feature representations (Sharpee et al., 2011) at a functional level.

This raises the question of what advantages, if any, modularity and hierarchy confer a complex biological system. While both features may simply be a by-product of random processes (Corominas-Murtra et al., 2013), modularity has been suggested to emerge naturally in dynamical networks when driven by growth (Lorenz et al., 2011) or when connections involve a cost (e.g., metabolic costs, see Mengistu et al., 2016). From an evolutionary perspective, a modular composition may enable a more granular, selective variation of individual components while avoiding interference with other, already optimized subsystems (Hansen, 2003; Gerhart and Kirschner, 2007). Similarly, a hierarchical organization of modules is thought to endow the system with greater robustness, adaptability and evolvability (Meunier et al., 2009; Kaiser, 2010a). These general traits apply to any complex system that evolves and learns through time, including neural networks.

1.2 Modularity and hierarchy for neural computation

Neural networks are the foundational blocks of connectionist and neurocomputational theories of cognition. Early connectionist models (Rumelhart et al., 1986a; McLeod et al., 1998) drew on the neuron doctrine of Santiago Ramón y Cajal (1888), and used simplified neuron models, such as the binary neuron of (McCulloch and Pitts, 1943) and the perceptron (Rosenblatt, 1958), to build functional networks that could perform simple computations.

While these models are constrained only by behavioral data, some modern neurocomputational theories rely on networks that additionally include detailed neurophysiological information, such as some models used in *computational neuroscience*. A third view is represented by the classical theories of cognition (Newell and Simon, 1975; Fodor and Pylyshyn, 1988), which draw strong parallels between cognition and digital computers and formulate (artificial) intelligence and cognitive processes in terms of symbolic algorithms that can be replicated by idealized computing devices like Turing machines (Turing, 1937, 1952).

Unlike such abstract devices, recurrent neural networks are much better suited for mechanistic models of cortical circuits, while preserving the full computational power of a universal Turing machine (Garzon and Franklin, 1991; Siegelmann and Sontag, 1991). Given that, to a first approximation, cortical circuits are recurrent networks of spiking neurons (Lorente de Nó, 1933), insights into the representational and learning characteristics of neural networks can drive and inform modeling studies that seek to understand the behavioral and functional properties of biological circuits. For instance, works on deep networks demonstrated how a hierarchy of concepts and representations can emerge through a layered architecture, fostering generalization capacity (LeCun et al., 2015); how recycling more compact representations from lower layers allows efficient use of limited resources (Bengio and LeCun, 2007); or how, in practice, solving complex tasks is close to impossible without some form of hierarchy (Bengio and LeCun, 2007).

The generality of these computational benefits means that, in principle, they also apply to similarly organized biological neural circuits. When sophisticated functionality arises from simple building blocks (modules), unfavorable evolutionary changes or brief perturbations can be limited to a small part of the network, thereby lowering the impact on the overall system operation. Compartmentalization of (learned) functionality through modularity helps cortical networks to avoid detrimental interferences between specialized modules, which may lead to catastrophic forgetting of previously acquired skills (Ellefsen et al., 2015).

At the very least, modularity and hierarchy in cortical circuits regulate how information is distributed through the processing stages and constrain/shape the dynamics of the neural activity in every module. Computational studies demonstrated that compared to random networks, modular circuits facilitate a wider range of complex activity patterns (Sporns and Betzel, 2016) characterized by long transients and high information transfer (Rubinov et al., 2011). Additionally, a hierarchical arrangement increases the range of stable dynamics in recurrent networks (Jarvis et al., 2010), is preferred by evolutionary algorithms that seek to maximize information transmission (Yamaguti and Tsuda, 2015), and represents an optimal structure for stable, but limited sustained neural activity (Kaiser et al., 2007; Kaiser, 2010b). Modular hierarchy thus enables rich patterns of communication and may be a natural way to balance network segregation and integration to maximize functionally relevant interactions of individual modules (Tononi et al., 1994).

1.3 Modular sensory pathways and topographic maps

Segregation and integration are key computational strategies enabling a hierarchical aggregation of stimulus features. This implies a progression of scales along which information is processed, which in the cortex arises from the combination of local features (Duarte et al., 2017a) and inputs from other modules and areas mediated through structured pathways. The division of labor concerning specific features of the input is particularly evident in the early sensory cortices and is determined by the projection of signals from the peripheral nervous system.

Many sensory streams are mapped onto the neocortex in a highly non-random or *topographic* manner, mostly reflecting the spatial layout of the sensory receptors. These mappings determine the receptive field (Sherrington, 1906) properties of the cells in the early cortical regions, roughly defined as the part of sensory space that evokes a neuronal response when stimulated. Neighboring cells in the somatosensory cortex receive input from and encode adjacent locations on the body surface (somatotopy, Kaas, 1997), whereas tonotopic maps originating from the cochlea ensure that similar frequencies are processed in spatially proximal regions of the auditory cortex (Weisz et al., 2004). In the early visual system, the receptive fields of the neurons can be traced back to particular positions on the retina and thus to specific portions of the visual field, with nearby neurons “looking at” nearby points in the input (retinotopy; Holmes, 1918; McLaughlin and O’Leary, 2005).

This organization is similar to the concept of receptive field in convolutional neural networks (CNNs), which is defined as the region in the input (image) that yields the feature (Araujo et al., 2019). Much like cortical cells tuned to parts of the sensory input space, each unit in a convolutional layer processes data only from a restricted area, inducing a segmentation (tiling) of the input. These projections are maintained, to a variable degree, through many processing modules and layers. The effective receptive field size then grows with network depth, allowing each module to learn features at increasing scales and take full computational advantage of the hierarchical structure. This is important for size-invariant object recognition, among other functions, because it allows deeper modules to integrate stimuli over a greater spatial range.

In the mammalian brain, such projection patterns preserve the spatial relationship between neurons in different areas and collectively give rise to topographic maps (Hagler and Sereno, 2006; Harris and Mrsic-Flogel, 2013). Although many studies demonstrated their existence, there is considerable uncertainty regarding their formation and functional role(s). They are often regarded as a consequence of some optimization during developmental processes, in particular the optimization (reduction) of axonal wiring length (Buzsáki et al., 2004). This theory underlies some models of orientation and motion tuning development based on self-organization (Swindale and Bauer, 1998), and in non-biological networks it could explain the emergence of topographic connectivity that mirrored realistic spatial organization Lee et al. (2020). Beyond representing an efficient wiring pattern, the presence of topography in the associative and frontal areas suggests an important role in higher cognitive processes (Thivierge and Marcus, 2007). These range from facilitating rapid local computations (Hilgetag et al., 2000) and enabling accurate maintenance of basic spatial information (Friston, 2002) to being involved in object selectivity (Silver and Kastner, 2009) and multisensory attention (Anderson et al., 2010), as well as working memory (Kastner et al., 2007).

Thus, the preservation of topographic organization through large portions of the cortical hierarchy may facilitate a dynamic link between perception and more complex processes (Mackey et al., 2017). Given the robustness of most sensory systems to variations in the input, we should perhaps not be surprised if certain stimulus features are maintained, repeated or amplified across the cortex. Take for instance the aforementioned retinotopic maps, which conserve the spatial structure of an image. These maps occur through most of the visual hierarchy, often arranged into multiple, spatially continuous clusters spanning different areas (Wandell et al., 2005). Although individual maps within a cluster may implement distinct computations (e.g., related to color or orientation), sharing a common visual field representation can facilitate their integration in a spatially local manner. This suggests that the computations performed within a cluster serve related perceptual functions, taking advantage of efficient resource sharing (Wandell et al., 2007).

1.4 Robust state representations from noisy sensory inputs

It is intriguing to consider topographic maps as a ubiquitous structural feature, whereby certain (sensory) input properties are maintained and transmitted across different cortical areas efficiently. In principle, this information could then be integrated with various local computations and serve as a basis for perception. In order to understand the functional requirements involved, it is helpful to first take a step back and examine some more fundamental aspects of sensory signals and their processing.

Our everyday experience of the chaotic, continuous flux of information from the world seems to be divided into discrete events that exhibit some systematic relation (Zacks and Tversky, 2001; Zacks, 2020). When we listen to music or learn a new dance, we naturally partition the continuous streams into temporally extended events based on some shared attributes. The boundaries, which may be spatially and temporally imprecise or even discontinuous, are created through a process of segmentation that links stimuli, based on similar context or spatiotemporal features, into "coherent and bounded sub-sequences" (Schapiro et al., 2013). Often, such segmentations involve a higher level of abstraction and association of inputs over prolonged time intervals (e.g., switching between different types of dances). VanRullen and Koch (2003) considered this as a kind top-down information integration, and contrasted it with the process of "discrete perception". In this view, continuous representations are created from individual snapshots of the sensory input. When these occur within a perceptual timeframe that is determined by the intrinsic timescales of neurophysiological processes (less than 100 ms), they are grouped and interpreted as a single event. Importantly, this interpretation validates computational models that approximate sensory stimuli as discrete inputs with various degrees of abstraction.

Irrespective of the nature and level of discretization of such perceptual events, their representation must be extracted and constructed from a highly dynamic and noisy environment. Therefore, as a first step towards engaging with the world, cortical networks must create reliable and meaningful representations from sensory inputs that are often ambiguous, incomplete or corrupt (Renart and Machens, 2014). Sources of volatility can be extrinsic (e.g., blurred image) or intrinsic to the system, such as imperfect signal transduction at the peripheral receptors or cellular and synaptic noise, which can accumulate and lead to a deterioration of the input representation (Faisal et al., 2008). From these noisy inputs, cortical circuits must then distill the relevant features to forge a ground truth against which internally generated signals from inferential processes can be evaluated (Młynarski and Hermundstad, 2018; Parr et al., 2019).

For this, signals coming from the sensory periphery must be routed through the modules on the lower levels of the cortical circuitry, their information content represented and integrated with ongoing processes (Macaluso, 2006; Keller et al., 2012) that depend on both local and long-range interactions (Duarte, 2015). Since information that fails to permeate the cortical hierarchy can not influence sensory perception, the encoding and transmission process must minimize signal degradation. This is all the more important considering that at each subsequent stage, information can only be lost and not gained (see data-processing inequality theorem; Cover and Thomas, 1991).

The precise mechanisms through which neuronal circuits can overcome the detrimental effects of noise are still actively debated. Previous studies have focused on single-neuron behavior (Marrufo-Pérez et al., 2020), circuit-level plasticity mechanisms (Turrigiano, 2011), and optimal tuning (Seriès et al., 2004) and connectivity profiles (Renart and van Rossum, 2012). However, it remains unclear whether the modular topographic organization of the sensory pathways also contributes to alleviating this problem.

1.5 From discrete representations to sequence processing

As discussed above, discretization of the continuous information flow may be an integral part of sensory processing, but at the same time, we can perceive, learn and generalize from the temporal dependencies within the input stream. Perceiving causality, anticipating events, understanding language and generating movement — all facets of complex behavior are anchored in time and rely on the ability to encode, recognize and express temporally patterned sequences. Since (Lashley, 1951) noted an innate proclivity of humans and other animals towards acquiring multi-item sequential structure in the input, a long line of research, spanning a variety of disciplines, has been devoted toward formalizing the problem and deciphering the underlying neural mechanisms (Chomsky, 1956; Reber, 1967; Dehaene et al., 2015). Although experimental and theoretical approaches vary significantly depending on the aspects they aim to capture, most of them can be roughly aligned along the taxonomy of five mechanisms for sequence processing formulated by Dehaene et al. (2015): transition and timing knowledge, chunking or segmentation, knowledge of order, algebraic patterns, and nested tree structures.

Sensitivity to order and timing is reflected in the neural activity throughout the brain. Responses to simple sequences of images or tones indicate the learning of both transition probabilities and duration of stimuli, which can be faithfully recalled (replayed) upon a cued input (Gavornik and Bear, 2014). Sequence elements are anticipated even if omitted, whereas items that violate the learned structure generate detectable mismatch responses (Garrido et al., 2009). To a certain degree, these mechanisms are semi-automatic and occur through mere exposure, even in the absence of attention. Representations are item-specific and detailed, encoding both their duration and the time interval between them. While there may be several neural processes underlying timing (Tsao et al., 2022), evidence indicates that it is (just) one dimension along which the brain computes. On the one hand, elapsed time can be quantified and used for decision-making (Balci et al., 2009), and sequences such as a song or movement can be learned at one speed and replayed (rescaled) flexibly at different ones. On the other hand, time can also be abstracted from during sequence recognition, most evidently during language processing (Miller et al., 1984).

Language also illustrates the ability to segment a continuous input stream into distinct groups or "chunks" based on certain properties (e.g., phonemes into words). More broadly, this process refers to recognizing frequently recurring elements (embedded into longer sequences) as one chunk and concatenating them into single-item representations for further manipulation as a whole (Gobet et al., 2001). In humans and some primates, this capability is very general and underlies a range of cognitive functions, from visuomotor processing (Orbán et al., 2008) to episodic memory (Schapiro et al., 2013).

The remaining three mechanisms in the taxonomy are more complex and involve an increasing degree of abstraction. Ordinal knowledge implies the ability to retain and recall the order of elements in a sequence, independently of their timing. Abstracting further away from the stimulus identities, representations of algebraic patterns refer to the capacity to internalize motifs such as ABB, and recognize any sequence of that form (Marcus, 2003). Lastly, in addition to the extraction of such abstract rules and generalizing them to new inputs, processing language requires the ability to also handle nested tree structures generated by symbolic rules, possibly involving recursion (Chomsky, 1956). As we will see later on, Chomsky's work on formal languages and symbolic processing ties in deeply with the theory of computation, and can serve as a framework for evaluating the computational power of neural networks.

Contingent on the architectural details, the power of artificial neural networks spans the entire hierarchy of complexity, from recognizing very simple sequences like ABCD to Turing-completeness (Delétang et al., 2022). In contrast, biologically more detailed models, such as recurrent networks of spiking neurons in conjunction with non-supervised learning rules, are limited to sequences of significantly lower complexity. Most models obeying these feature restrictions focus on the transition and timing aspects (Murray and Escola, 2017; Bouhadjar et al., 2022; Cone and Shouval, 2021), with few propositions to solve chunking (Asabuki et al., 2022). These typically focus on individual features of sequence processing (e.g., acquisition of order or temporal rescaling), and often employ architectures and narrow sets of biophysical properties that can specifically support the functionality in question. In many cases, they are intended as a proof-of-concept that demonstrates the behavior of the system for a few prototypical sequences, without examining its limitations in more detail. For instance, a model may learn and replay sequences where elements are presented in direct succession, but fail as soon as small gaps (e.g., 100 ms) are introduced.

A deeper understanding of the neural basis of sequence processing therefore requires models that can not only account for more than single features but are also robust to variations in the input properties that characterize naturalistic signals. One difficulty towards such general models of biological sequence processing is the lack of a unified computational framework that allows systematic benchmarking and comparison on a set of well-defined tasks.

1.6 Scope and structure of the thesis

The previous sections took the reader on an expansive journey, following an imaginary gradient from structural features (modularity, hierarchy and topography) to basic functional requirements (discrete representations of noisy stimuli) underlying more complex cognitive processes (learning structured sequences). Along the way, two aspects stood out as insufficiently explored, which also constitute two of the three main thematic threads of this work: the possible role of topographic maps in supporting robust representations from noisy inputs; and the limited scope and shortcomings of biologically inspired models of sequence learning, including the absence of a rigorous framework for their evaluation.

By attempting to bridge these gaps, the overall aim of this thesis is to advance our understanding of how neural circuits can leverage modular structures to learn and process sequential information efficiently and reliably. It follows the cognitive science paradigm according to which all mental processes, from sensory perception to cognition, involve the manipulation of sequentially organized time-discrete (symbolic) representations.

To investigate possible neural mechanisms underlying such processing in a functional context, we first introduce a toolkit for benchmarking neuronal networks on sequence learning tasks using the reservoir computing (RC) paradigm (Chapter 4). This touches on the third thematic line of this thesis, which will become more relevant towards the end: reproducibility in computational studies.

Equipped with this framework, we then embark on a series of studies that exhibit an increasing gradient in both the complexity of network models and utilized tasks. Throughout this journey, we strive to relate architectural features to functional properties, with a strong emphasis on modular connectivity patterns and their possible computational benefits.

The thesis is organized as follows:

Chapter 2 introduces some basics from neurobiology to the unfamiliar reader, and provides a brief overview of the modeling concepts and analytical methods used throughout the thesis. Beginning with the properties of biological neurons and mathematical models thereof, network-level phenomena are described along with statistical tools (mean-field) for approximating the behavior of large neuronal populations. To set up a foundation for the sequence learning models studied in the second part of this work, it additionally describes the principle synaptic plasticity mechanisms and contrasts them to established machine learning algorithms and more non-traditional approaches such as reservoir computing (RC).

Chapter 3 defines and formalizes the problem of sequence processing, along with the artificial grammar learning paradigm and related complexity measures for quantitative analysis. It then gives a brief survey of existing artificial and biologically-inspired models, with a focus on their core properties and the nature of the tasks they are intended to solve. This extended literature review, fundamental for the future meta-analysis character of Chapter 8, is meant to help in positioning the models along the axis of biological plausibility – computational capabilities – cognitive relevance.

Chapter 4 presents *Functional Neural Architectures* (FNA), a Python library for generating complex symbolic sequence processing tasks, flexible creation of (spiking) network architectures and their benchmarking using the RC principle. This tool represents the computational backbone in all subsequent chapters.

Chapter 5 addresses the question of how the (feedforward) connectivity profile between multiple, recurrently connected populations of spiking neurons impacts stimulus representations, their integration and transmission across cortical circuits. In a first step, we contrast networks with random and structured projections (inspired by cortical topographic maps), and use the RC paradigm to assess their capability to produce and maintain distinguishable representations of the input in each stage. The performances and properties of the two network types are then further characterized in terms of response variability, robustness to noise and memory capacity. Given that cortical circuits typically handle information from multiple sources simultaneously, in a second step, the model is extended to include two input streams. This allows us to study different integration schemes and their influence on the networks' strategy to solve nonlinear tasks.

In Chapter 6 we build on the findings from Chapter 5 and investigate the hypothesis that the modularity of topographic projections plays a key role in enabling accurate stimulus representations and their reliable transmission across different processing modules. By taking a more systematic approach and combining network simulations with theoretical analysis, we show that modularity acts as a bifurcation parameter and identify a critical level beyond which the signal-to-noise ratio improves during propagation. Performance is evaluated by probing the network's ability to reconstruct continuous signals of varying complexity, corresponding to stimuli presented sequentially but without any temporal structure. We reproduce this novel denoising effect in multiple spiking and continuous-time network models and demonstrate that it is purely a structural effect. Finally, we argue that during an input integration task, modularity acts as a control parameter for network dynamics, reproducing activity patterns that are associated with a variety of behavioral effects.

Chapter 7 transitions from sequential, but independent stimuli to studying models that learn and process sequences containing temporal regularities. We begin by re-implementing a biologically plausible model recently published by Cone and Shouval

(2021), using the NEST simulator and the FNA tool presented in Chapter 4. After replicating the main results from the original study, the model's robustness to parameter settings and underlying assumptions are evaluated, highlighting its strengths and weaknesses. We demonstrate a limitation of the model consisting in the hard-wired sequence order in the connectivity patterns and suggest possible solutions. The chapter closes with a discussion on reproducibility issues given that an important outcome of this work, in addition to providing an open-source implementation of the model for the computational neuroscience community, is the revised and corrected model description in the original publication and accompanying source code.

Chapter 8 presents the first steps towards a conceptual and computational framework for benchmarking and comparison of biologically-constrained models of sequence processing. Expanding upon the formal methods outlined in Chapter 3, we propose a set of concrete tasks aimed at assessing a diverse set of capabilities expected from a general sequence processing model. Although the framework represents only a small step towards elucidating the biological implementations of sequence processing, the presented concepts and analysis tools, illustrated on a few selected models, are aimed at deriving a set of functional and biophysical principles that can guide future studies on this topic.

We conclude with a short discussion of our contributions and potential future work in Chapter 9.

Chapter 2

Concepts and tools from computational neuroscience

2.1 Mathematical models of biological neurons

When McCulloch and Pitts described the first computational model of a neuron in 1943, the *Threshold Logic Unit* (TLU McCulloch and Pitts, 1943), they formalized an already long-standing view that considered neurons to be the structural and functional units of the nervous system. In their model and subsequent connectionist approaches, neurons were simple computational units that receive inputs, integrate information and produce output. An LTU receives only binary input (excitatory and inhibitory), outputs 1 if the sum of all excitatory inputs exceeds a certain threshold and no inhibitory inputs are active, and emits a 0 otherwise. This simple threshold operation limits the LTU to computing boolean functions, LTU, rendering it essentially a logical operator. Many of these properties were improved upon in the following decades beginning with Rosenblatt's *perceptron* (Figure 2.1A), which included weighted inputs and was later generalized to continuous values. However, LTUs conceptually already resembled the three major functional components of biological neurons: dendrites, as an input device; the soma as a nonlinear integrator; and the axon, as an output device.

More specifically, neurons are excitable cells in which the lipid membrane acts as an electrical insulator and diffusion barrier, creating a difference in electrical potential between the intra- and extracellular space called the *membrane potential.* At terminals located on the dendrites, incoming signals (at chemical synapses through the release of neurotransmitters) cause a positive (negative) deflection of the membrane potential for excitatory (inhibitory) connections, which is propagated through the dendritic tree towards the soma. By integrating many such events, the membrane may become sufficiently depolarized such that, above some critical voltage threshold, the cell generates an *action potential* or *spike* (Figure 2.1B) which is transmitted along the axon and delivered to other neurons at *synapses* (points of contact). These electrical pulses, representing the primary means of communication in the cortex, are brief events with stereotypical waveforms (Fee et al., 1996). For this reason, one often assumes that information is contained in the presence or absence of a spike rather than its shape, similar to the on/off

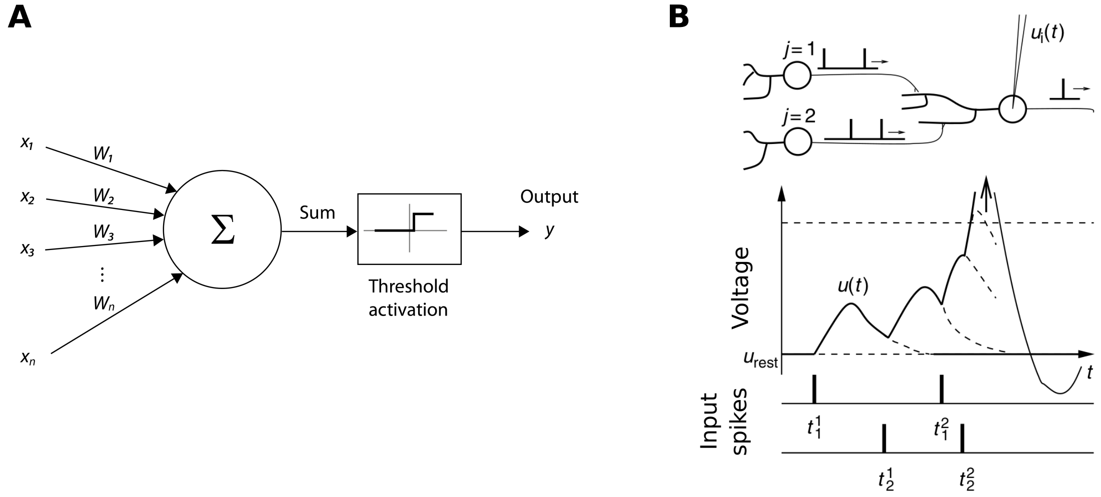


Figure 2.1: **Principles of artificial and biological neurons. (A):** Schematic of a perceptron. The inputs x_i, multiplied by the corresponding weights w_i, are summed and passed through the threshold activation function to yield the binary output y. **(B):** Example for the input integration and response of a biological (spiking) neuron. A neuron receives spikes from two afferents, each spike evoking a positive deflection in its membrane potential $u_i(t)$ (excitatory postsynaptic potential). If $u_i(t)$ crosses a certain threshold, an action potential is generated whereby the neuron is depolarized rapidly after which the potential relaxes back to its resting value. Adapted from Gerstner et al. (2014), with permission.

states of an LTU. Neuron models that abstract from the biophysics of spike generation and describe spikes as discrete events are called *leaky integrate-and-fire* (LIF) models, and trace their origin to the work of Lapicque (1907). The neuron is treated as an electric circuit composed of a parallel capacitor and resistor, representing the capacitance and leakage resistance of the cell membrane.

In these models, the dynamics of the membrane potential is described by a linear differential equation, and spikes are elicited when the potential crosses a threshold from below. Following a spike, the membrane potential is reset and often remains clamped to a fixed value below threshold (refractory period) to mimic the reduced neuronal excitability during hyperpolarization, after which it relaxes back to the resting potential. With these ingredients, the LIF model can capture the spatial and temporal integration of synaptic inputs, spike generation, refractory period and passive membrane properties of a real neuron. The input integration depends on the chosen synaptic model, typically representing synaptic inputs as changes in currents or conductances (see Burkitt, 2006). In the first case, synaptic currents are independent of the membrane potential, have a fixed temporal kernel and are summed linearly, making this approach more convenient for detailed analytical studies. In biologically more accurate conductance-based models (Destexhe et al., 2003), the synaptic current does depend on the membrane potential, introducing a nonlinearity in the input integration.

Other neuron models have additional layers of complexity to better approximate the nonlinearity of neuronal dynamics (see Izhikevich, 2004, for a review), either by including biophysical detail via multiple (voltage-gated) ion channels (Hodgkin and Huxley, 1952), incorporating additional variables (e.g., describing adaptation) that can be fitted to experimental data (eg., generalized linear models, see Mihalas and Niebur, 2010), or taking into account the dendritic (spatial) morphology as done in compartmental models (Hay et al., 2011). While these features may allow capturing a more diverse spiking behavior (but not always, see Gerstner and Naud, 2009), they reduce the analytical tractability of the models and can increase the computational cost prohibitively, particularly for large-scale network simulations. In contrast, point-neuron models such as the LIF are not only computationally more efficient and mathematically more accessible, but they are also sufficiently complex to capture a wide range of dynamical states and firing pattern statistics observed in cortical recordings, including at the network level (Shadlen and Newsome, 1998; Renart et al., 2010; Gerstner et al., 2012; Brunel, 2013).

2.2 Network dynamics and mean-field approximations

In the context of cortical circuits, the description level of "networks" refers to populations of coupled excitatory and inhibitory neurons, typically through recurrent connections. Thus, individidual neurons, nonlinear dynamical systems themselves owing to their complex dendritic structure and a variety of intrinsic adaptation mechanisms, organize into recurrent networks that process information in a highly nonlinear, time-dependent manner. This processing is state-dependent (Buonomano and Maass, 2009), meaning that the dynamics of cortical networks and their responsiveness to external stimuli depends on the behavioral context (Harris and Thiele, 2011) and the current internal state of the system (molded by past inputs). Given that changes in the environment may require slow integration (e.g., during learning) or fast reactions (e.g., during flight), these networks must maintain a ground state that allows rapid switching between different operating regimes (Tsodyks and Sejnowski, 1995; McGinley et al., 2015). Depending on whether the system is simply idling (e.g., during sleep or quiet wakefullness) or engaged in active processing, these operating regimes may be characterized by globally synchronized (oscillatory) spiking (Steriade et al., 2001) or by a state where individual neurons fire asynchronously and irregularly at low rates (Matsumura et al., 1988; Softky and Koch, 1993), often referred to as the *asynchronous-irregular* (AI) state (Brunel, 2000; Renart et al., 2010).

Pioneering modeling work by van Vreeswijk and Sompolinsky (1996) predicted that such AI activity emerges naturally in large, but sparsely connected networks of excitatory and inhibitory neurons due to an approximate balance between excitation and inhibition (E/I balance). Subsequent experiments *in-vivo* confirmed this dynamic balance (Haider et al., 2006; Dehghani et al., 2016), and the model became a popular/ an essential building block of later computational studies investigating computation in cortical networks (Shadlen and Newsome, 1998; Brunel, 2000; Vogels et al., 2005; Destexhe, 2009). In the *balanced state*, each neuron receives approximately the same amount of excitatory and inhibitory input that cancel out on average, pushing the mean membrane potential just below threshold. Spikes are caused by external stimuli or spontaneous fluctuations in the activity, the timescale of which is influenced by the synaptic strength (Ostojic, 2014), homeostatic plasticity mechanisms (Vogels et al., 2011; Froemke, 2015) and connectivity structure (Litwin-Kumar and Doiron, 2012). This dynamical state allows the network to quickly respond to external perturbations through a transient adaptation of the E/I balance and has been suggested to represent a stable regime for neural communication (Vogels and Abbott, 2009; Joglekar et al., 2018) and advantageous operating point for complex computations (Buonomano and Maass, 2009; Duarte and Morrison, 2014; Denève and Machens, 2016), while its disruption is associated with a variety of pathological conditions (Yizhar et al., 2011; Vecchia and Pietrobon, 2012).

The stereotypical *balanced network model* consists of two interacting populations of excitatory and inhibitory neurons driven by external input, typically in a ratio of $4:1$ that reflects the distribution of cells in local cortical circuits (Braitenberg and Schüz, 1991). Brunel (2000) demonstrated that this simple model can not only reproduce asynchronous irregular dynamics but also diverse firing patterns of varying synchrony and regularity, depending on the choice of parameters. For a broad range of these, the balanced state emerges in a self-consistent (stable) manner (van Vreeswijk and Sompolinsky, 1996, 1998) if the network is sufficiently large and is inhibition-dominated. The first condition is necessary to ensure that while each neuron receives enough input to evoke spiking, the number of inputs it shares with other cells is relatively small such that their firing patterns are only weakly correlated. The second condition, a hallmark feature of the cortex (Sanzeni et al., 2020), is required to stabilize the dynamics and prevent runaway excitatory activity. These network-level effects can be captured by a variety of neuron models, including the LIF neuron described in the previous section. And for analytically tractable models such as the LIF, in many cases it is possible to estimate and predict the collective dynamics of the neurons through mean-field approximations (Fourcaud and Brunel, 2002; Helias et al., 2013).

These mathematical tools offer a statistical description of the activity (e.g., firing rate or pairwise correlations) at the level of (homogeneous) neuronal populations, where the single-unit and population-averaged activity are taken to be equivalent. For spiking neurons such as the LIF, the stochastic behavior of the subthreshold dynamics can be described using the Fokker-Planck (or diffusion approximation) formalism (Risken, 1996), assuming that each neuron receives a large number of small amplitude inputs in every timestep. If we further assume that these spike trains are independent and can be described by uncorrelated Poisson processes whose mean represents the firing rate — a good approximation in the AI state —, the total synaptic drive (summed

Poissonian input) to a neuron can be replaced by a Gaussian process with mean $\mu(t)$ and variance $\sigma^2(t)$. In the stationary state, these quantities, along with the firing rates of each afferent, can be well approximated by their average value over time. This leads to a self-consistent expression for the mean firing rate of a single neuron (and equivalently of the population), that takes into account not only the mean but also the fluctuations in the input (Amit and Brunel, 1997; Fourcaud and Brunel, 2002). Thus, we can obtain a mean-field-like description of the macroscopic dynamics in a recurrently connected network, in which the input and output statistics of a neuron are related through a closed set of equations. Such descriptions can be extremely valuable for characterizing the network behavior in a computationally efficient manner, for instance, to determine whether the activity converges to a stable point over time.

2.3 Neuroplasticity and learning

Although mean-field approximations are often sufficient for describing the stationary activity of populations, the interactions within and between cortical circuits involve a variety of adaptation mechanisms at the microscopic level. Such experience-driven changes, in particular at the neuronal and synaptic level, have long been associated with learning and memory, and therefore ought to be taken into account by any computational model seeking to go beyond simple dynamics. For instance, neurons can regulate their excitability in an activity-dependent manner, altering their intrinsic firing properties and thus potentially impacting circuit dynamics and computations (Turrigiano et al., 1994; Destexhe and Marder, 2004). Although the functional implications of such intrinsic plasticity are only slowly emerging, it appears to play an important and synergistic role in memory formation, along with synaptic plasticity (Titley et al., 2017; Lisman et al., 2018; Debanne et al., 2019).

2.3.1 Correlation-based (Hebbian) plasticity

Synaptic plasticity, on the other hand, has been considered the biological substrate of learning since Donald Hebb's seminal work in 1949. Colloquially, but imprecisely summarized as "neurons that fire together, wire together" (Shatz, 1992), Hebb's theory postulated that the connection strength between two neurons increases if the presynaptic neuron regularly takes part in the firing of the postsynaptic neuron, a mechanism that does involve temporal causality. *Hebbian learning*, as it is known today, has since been refined and generalized to modifications of the synaptic transmission efficacy that are mediated by correlations in the firing activity of pre- and postsynaptic neurons. In a very general formulation, the synaptic weight w_{ij} between the neurons i and j can be written as

$$\frac{d}{dt}w_{ij} = F(A_i, A_j), \tag{2.1}$$

where F is an arbitrary function (often product), and A_i and A_j are the activations of the corresponding neurons. These predictions heavily influenced the first learning rules in ANNs based on coincident activation (Rosenblatt, 1958), which mostly represented the activity as continuous functions (rate-based Hebbian) and calculated the weight changes proportionally to the product of pre- and postsynaptic activity, thereby ignoring any causality. While this simplification allowed for elementary associative-learning of input patterns (Hopfield, 1982) and self-organization (von der Malsburg, 1973; Kohonen, 1982), experimental findings indicated that long-term potentiation (LTP) and depression (LTD) of synapses (Bliss and Lomo, 1973; Levy and Steward, 1983) depended, among others, on the precise timing of pre- and postsynaptic spikes (Markram et al., 1997; Sjöström et al., 2001). Such long-lasting synaptic changes have been formalized as spike-timing dependent plasticity (STDP) rules (Bi and Poo, 2001; Song et al., 2000). With STDP, the synaptic weight is modified if the two neurons spike within a short interval called the STDP window. The weight is potentiated if the presynaptic spike arrives shortly before a postsynaptic action potential and depressed otherwise (although other STDP variants also exist).

2.3.2 Neuromodulation, eligibility traces and three-factor learning rules

In its classic form, STDP is an asymmetric form of unsupervised Hebbian learning, but over the years a variety of phenomenological models were developed to include additional factors to address some of its limitations (e.g., different pairing schemes and windows, voltage dependence; Morrison et al., 2008; Pfister and Gerstner, 2006; Clopath et al., 2010). In particular, its relation to phenomena on behavioral timescales is unclear due to its local nature, dependence on millisecond precision and the fact that it ignores, by design, any high-level information about novelty, reward or punishment. Such signals are thought to be mediated by (more global) modulatory processes involving dopamine, acetylcholine and other neurotransmitters, which can strongly impact synaptic activity including STDP (Marder, 2012; Brzosko et al., 2019). To account for these influences, models of neuromodulated STDP (Frémaux and Gerstner, 2016) incorporate a third factor in addition to the pre- and postsynaptic activity, which can be formally written as

$$\frac{d}{dt}w_{ij} = F(M, A_i, A_j), \tag{2.2}$$

In such *three-factor learning rules*, M is an arbitrary (global) modulatory signal, which can be available at all or only a subset of synapses given that neuromodulators typically diffuse over larger cortical areas. For instance, M can be a time-dependent reward signal that gates (or scales) the Hebbian component, so that the weight is only updated in specific instances (Cone and Shouval, 2021). Although this generalization can partially link certain behavioral or environmental events to local plasticity processes, there remains a large temporal gap between synaptic modifications on the scale of a few tens of milliseconds and possible behavioral consequences of that change occurring after minutes or hours.

To solve this issue, known as the *temporal credit assignment problem*, theoretical considerations (Frémaux et al., 2010) proposed that there should be some signal at the synapse, like a tag or a trace, that decays on a much slower timescale and would thus allow bridging the temporal gap to the modulatory signal. While the concept of synaptic tagging is not new (Frey and Morris, 1997), experimental support for such "eligibility traces" and their involvement in LTP and LTD as formulated in three-factor learning rules is rather recent (He et al., 2015; Gerstner et al., 2018). These traces are synapse-specific markers that are activated through coincidental (Hebbian) pre- and postsynaptic activity as in STDP. However, plasticity is only induced if the neuromodulatory signal arrives at the tagged synapse before the eligibility trace has decayed. Formally, this means that for neuromodulated STDP the activations A_i and A_j in Eq. 2.1 will denote the eligibility traces instead of the spike times or firing rates. Such models of three-factor learning rules, in combination with eligibility traces, can learn many supervised (Zenke and Ganguli, 2018), reinforcement (Vasilaki et al., 2009; Huertas et al., 2016) and sequence learning tasks (Cone and Shouval, 2021).

2.3.3 Short-term plasticity

In addition to STDP and other forms of more persistent modifications, synapses exhibit multiple types of short-term plasticity (STP) which evolve on a scale of hundreds of milliseconds to seconds and minutes (see e.g., Fioravante and Regehr, 2011, for a review). In contrast to STDP, these depend almost exclusively on the history of the presynaptic activity, with a common set of mechanisms able to cause both depression and enhancement of the postsynaptic responses at different synapses. Short-term depression is linked to the depletion of neurotransmitter vesicles following presynaptic spiking, whereas facilitation occurs as a result of calcium influx into the presynaptic axon, increasing the subsequent release probability of neurotransmitters and thereby enhancing synaptic efficacy.

These are also the features that are captured by many phenomenological models of STP (Markram et al., 1998; Tsodyks et al., 1998; Mongillo et al., 2008), which typically model short-term dynamics using two dynamics variables describing the fraction of resources that remain available after neurotransmitter depletion, and available resources which are ready for use (release probability), respectively. The interplay of these two variables, controlled by the specific time constants, determines whether the synapse is depressed or facilitated in response to repeated presynaptic spiking. Given that these rapid but transient dynamics operate on the timescale of stimulus processing, working memory and behavior, STP can directly influence the computations performed by neural circuits (Abbott and Regehr, 2004). Experimental and theoretical work demonstrated that STP endows recurrent circuits with rich dynamics, implementing transient functions such as gain control (Abbott et al., 1997) and network stabilization (Wu and Zenke, 2021), temporal filtering (Bourjaily and Miller, 2012), working memory (Mongillo et al., 2008), but is also crucial for the consolidation of long-term memories (Zenke et al., 2015).

2.4 Functional benchmarking using reservoir computing

While the local and unsupervised plasticity mechanisms described above represent a part of the processes underlying learning in cortical circuits, they raise an important challenge for modelers: how to evaluate whether such systems can solve a given task?

2.4.1 End-to-end training

The classical supervised learning methods in current machine learning approaches involve defining an objective (loss/cost) function that measures the error between the input and some specified target mappings, and optimize the model parameters to minimize this error (LeCun et al., 2015). In practice, the optimization procedure typically relies on a variant of gradient descent, whereby the network weights are gradually adjusted in the direction that reduces the error most rapidly. For multilayered, feedforward architectures the gradients can be computed efficiently through the whole network via *backpropagation* (Rumelhart et al., 1986b). This algorithm can also be adapted for training recurrent networks by unfolding them in time — via *backpropagation through time* (BPTT, Werbos, 1990) —, but it nevertheless remains relatively difficult because the gradients tend to vanish or explode over time (Bengio et al., 1994; Hochreiter, 1991). Although solutions to overcome these problems exist, for instance through specific architectures such as Long-Short Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997), BPTT and other training approaches (e.g., Real-Time Recurrent Learning Williams and Zipser, 1989) remain computationally very expensive.

From a neurobiological perspective, backpropagation is generally considered to be implausible (Crick, 1989) since many of its aspects are difficult to reconcile with learning in cortical circuits (Bengio et al., 2015). These range from technical aspects (such as weight symmetry and local computations) to more basic principles concerning the very nature of learning (supervised or not, learning from a few examples). In recent years, considerable effort was directed toward exploring how the brain may implement backpropagation-like with some moderate success at tackling individual differences (see Whittington and Bogacz, 2019; Lillicrap et al., 2020, for reviews), but fundamental questions remain along with efficiency considerations.

2.4.2 Transient dynamics and reservoir computing

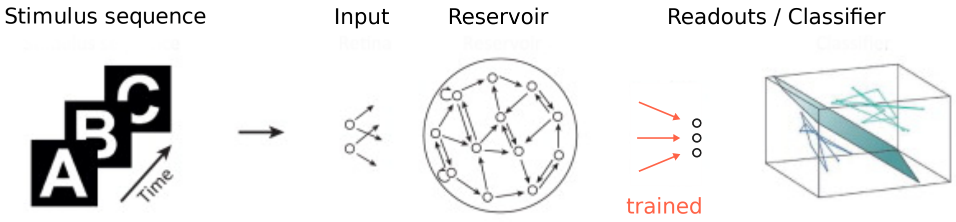


Figure 2.2: **Reservoir Computing.** In a typical experiment, a sequence of spatiotemporal stimuli is encoded and used to drive a randomly connected network (reservoir). This is connected to a set of readout units, which are trained to classify the high-dimensional state according to the task specifications. Figure modified with permission from Singer (2013).

One can take a step back from training systems end-to-end, and instead examine the computations performed by a single RNN. Recurrence allows such networks to represent time explicitly and retain information from long context windows, endowing them with a form of dynamic memory - sustained through transient reverberation of activity - that gradually fades in time. Provided they are sufficiently high-dimensional, such systems with fading memory can exhibit rich dynamics (Jaeger, 2001) where multiple signals are maintained and mixed in a nonlinear fashion, making them a powerful substrate for context- and state-dependent computations (Buonomano and Maass, 2009; Sussillo, 2014). Under the hood, the RNN performs a nonlinear temporal expansion of the (typically) low-dimensional input, projecting it to a higher-dimensional feature space where representations are encoded in the transient response trajectories (Rabinovich et al., 2008). In many cases, the transformation brings a practical computational advantage because the complex representations in the input space may become linearly separable in the feature space without actually training the recurrent weights, enabling, for instance, to calculate simpler decision boundaries for classification.

These observations represent the core principle behind a computing paradigm collectively known as *reservoir computing* (RC; Jaeger, 2001; Maass et al., 2002; Verstraeten et al., 2007; Lukoševičius and Jaeger, 2009), which took a novel approach to training RNNs. RC separates conceptually between the RNN, treated as an excitable *reservoir* or *liquid* and harnessed in its role as a nonlinear temporal expansion function, and decoding/extracting information from it through a “memory-less” *readout* mechanism (Maass and Markram, 2004). Instead of adjusting the recurrent weights, the network responses are combined (typically) linearly to approximate a target output (see Figure 2.2). By modifying only the connections from the reservoir to the readout, RC bypasses the challenging training problem without curtailing, in theory, the computational power of the system. Under moderate conditions, reservoirs with linear readouts can approximate any time-invariant function with fading memory (Maass and Markram, 2004). The principle is similar to kernel-based methods such as Support Vector Machines (Vapnik, 1998; Schölkopf and Smola, 2002) since each node in the reservoir effectively computes a random nonlinear function of the input, which can be weighted to learn any target function in the limit of $N \to \infty$. In practice, this ability depends on the internal properties driving the reservoir dynamics, with the nature of their fading memory often limiting their ability to handle long-term dependencies (Enel et al., 2016; Inubushi and Yoshimura, 2017). Nevertheless, the conceptual separation between an input-driven dynamic reservoir and trained readout enables studying a variety of recurrent circuits, without placing any constraints on the network architecture or its dynamics.

2.4.3 Benchmarking biological neural circuits

Here we leverage this flexibility and use RC to investigate the computational capabilities of biologically plausible circuits of spiking neurons. Inspired by the structural similarities between reservoirs and cortical networks (particularly recurrence), a number of studies have drawn parallels at the functional level, including how information is integrated and processed in the primary visual (Nikolic et al., 2009) and auditory (Klampfl et al., 2012) cortex, cerebellum (Yamazaki and Tanaka, 2007) and prefrontal cortex (Enel et al., 2016). While some underlying computational principles, such as mixed selectivity (Rigotti et al., 2013), transient dynamics and state-dependent processing are likely shared (Singer, 2013; Hinaut and Dominey, 2013), one can also employ the RC paradigm to simply probe the ability of an arbitrarily complex system to solve particular tasks (Haeusler and Maass, 2007; Duarte and Morrison, 2019).

Adopting this latter approach, we are interested not so much in designing / engineering computationally powerful reservoirs, but whether specific computations are supported by systems with biologically constrained architecture, dynamics and learning rules. In this sense, the readout mechanism, often criticized due to the lack of an obvious biological counterpart, becomes only a tool to evaluate if the circuits contain sufficient task-specific information in a useful manner that is, at minimum, linearly separable. Although the readouts are trained in a supervised fashion, the task-relevant computations are performed entirely by the reservoir, the structure of which may be fixed or involve various learning (plasticity) processes. There are two immediate advantages to this approach: it allows investigating and comparing existing system models, with possibly very different characteristics, through a unified metric; and it enables measuring their performance on several tasks simultaneously (using multiple readouts), on tasks that may go well-beyond the models' original purpose. In particular, we will use this framework to evaluate and compare different, biologically-inspired models of sequence learning.

Chapter 3

Computational models of sequence processing

"Sequence processing" is a generic term defined variably in the domains of cognitive science, psycholinguistics and computer science, but it generally encompasses one or more aspects of processing temporally structured information, including learning, recognition, prediction and replay. The core problem can be formulated as to whether a system (artificial or biological) can derive and learn a general rule - the underlying grammar - from a finite set of specific input instances and use this rule to make predictions.

Although we have a fairly good theoretical and practical understanding of how different neural network model classes perform on such tasks, a comprehensive overview of existing models of biological (specifically spiking) networks that use biophysically realistic learning rules is still lacking. Given that this is the gap we aim to bridge in the second half of the thesis, we begin this chapter by briefly formalizing relevant experimental designs, after which we briefly review models of sequence processing in both artificial and biological networks. Because our primary objective is to understand how cortical circuits implement these tasks, we restrict our review to recurrent networks as a minimal requirement for biological plausibility. We focus on the class of tasks and sequence complexity they can master, both from a theoretical (where applicable) and practical perspective and touch upon their defining properties.

3.1 Sequence complexity and experimental designs

To help formalize the task specifications and enable a theoretical investigation under controlled complexity, models and experiments of sequence processing often consider computations on a set of discrete symbols. Symbols can be considered abstract representations of any naturalistic or artificial stimuli, which allows investigating computations and learning independently of the input properties. Symbolic processing is rooted in seminal works on the theory of computation and sequential processing in the 1950s and 1960s (Turing, 1937; Lashley, 1951; Miller, 1967; Reber, 1967), and was formalized in the context of cognitive linguistics as by Chomsky (1956, 1959).

3.1.1 Formal language theory and the Chomsky hierarchy

In laying the groundwork of *formal language theory* (FLT), Chomsky sought to understand the syntactic regularities of natural languages by analyzing the underlying generative rules or grammars. A formal language $\mathcal{L}$ can be defined as a set of words or strings, each of which is a finite sequence of symbols drawn from a finite alphabet Σ. An example of a language over the alphabet $\Sigma = \{\mathrm{A}, \mathrm{B}\}$ could be $L = \{\mathrm{A}^n\mathrm{B} \mid n > 0\}$, which contains all the strings where A occurs $n > 0$ times and is followed by B, such as "AAAB". The method for constructing valid strings is typically defined by a *grammar* (formally defined in the next section), which is a set of finite rules for creating strings of possibly infinite cardinality in both length and number. This generative property, i.e., the ability to generate infinite sequences from finite means, is a fundamental characteristic of natural languages.

One of the simplest systems exhibiting such generative capabilities is the *finite-state automaton* (FSA; Kleene, 1956; Hopcroft and Ullman, 1979). An FSA is an abstract computational "machine" containing a set of finite states, including start and end states, as well as labeled transitions between them that are realized by reading a symbol from an input sequence drawn from a finite alphabet. Due to their tight link to formal grammars, FSAs are often referred to as "acceptors" of a language - iff by reading a string from the language, the automaton transitions from an initial to an end state (Keller, 2001). Whether the FSA is able to recognize sequences from a specific grammar depends on the language complexity, which Chomsky categorized into four classes: regular, context-free, context-sensitive and recursively enumerable or Turing-complete (see Figure 3.1). These can be arranged into a hierarchy of increasing complexity, capturing one or several key grammar properties such as counting, repetition, long-distance dependencies, and (nested) hierarchy (Chomsky, 1956). Although in their original formulation FSAs can only recognize regular grammars, corresponding to the lowest complexity, augmenting them with a form of additional memory such as a stack, linear tape or infinite tape can extend their computational capacity to cover languages of all complexity (see equivalence in Figure 3.3).

3.1.2 Artificial grammar learning

A formal (artificial) grammar (AG) consists of a finite set of terminal symbols, a finite set of non-terminal symbols, a distinctive start symbol, and a finite set of production rules that map from non-terminal symbols to other symbols. AGs are typically expressed through state transition diagrams to visualize the (possibly probabilistic) sequential string generation process, with multiple styles common in different fields (see Figure 3.2). These diagrams can be easily transformed into more compact (probabilistic) transition matrices, which capture all rules but also permit quantitative analysis using methods developed for studying Markov processes. We elaborate on and employ these techniques

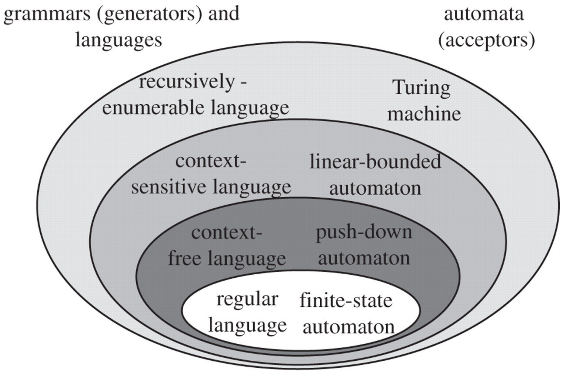


Figure 3.1: **The Chomsky hierarchy.** Formal languages were initially categorized into four, nested levels of complexity depending on the properties of the grammars that can generate them (left). For each class, there is a corresponding formal machine or automata of equivalent computational power (right). Figure reproduced with permission from Fitch et al. (2012).

in Chapter 8.

An experimental paradigm that builds on these is *artificial grammar learning* (AGL; Reber, 1967; Pothos, 2007; Fitch et al., 2012), which allows to systematically study how temporal regularities and the underlying rules can be learned by a system. Initially conceived for investigating language processing by Reber, an AGL experiment involves two phases: in the first stage, participants (or a model) are presented with a set of sequences (typically letters) generated by a formal grammar. During a second test phase, they must decide whether certain (new) sequences belong to the grammar, evaluating their ability to have learned its rules.

AGL is a powerful paradigm because it bridges empirical rule learning across disciplines (neuroscience, psycholinguistics) and provides a formal link to theoretical frameworks on computation through FLT (mathematics, computer science; see Uddén and Männel (2018) and overview in Fitch and Friederici (2012)). It enables studying rule learning both in a passive (implicit) way and testing explicit hypotheses. In addition, it allows for a wide range of sequence complexity that can be used for studying pattern and rule learning. More importantly for our purposes, AGL is an ideal methodology for comparing the computational capabilities of different (arbitrary) systems in a rigorous manner and through a unified protocol.

3.1.3 Measures of sequence complexity

In AGL, and more generally any sequence learning experiment, it is important to estimate how simple or difficult each input sequence is to process. Although the Chomsky hierarchy represents an abstract and rather coarse-grained method for this, sequences

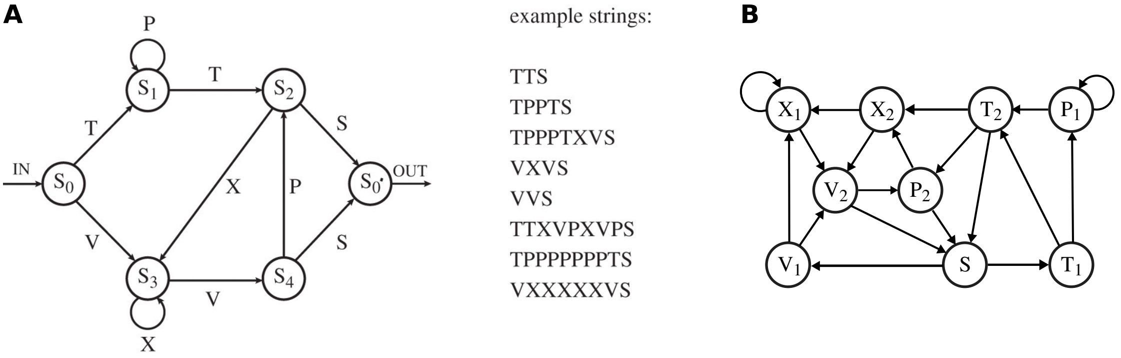


Figure 3.2: **The Reber grammar, illustrated by its corresponding finite-state machine (Reber, 1967).** **(A)** The automaton starts in the initial state S_0, and updates its state by reading input letters sequentially and moving along the corresponding labeled, directed transitions. A string is accepted if the machine is in the end state $S_{0\cdot}$ after processing the complete input. For instance, the language defined by this grammar includes strings of the form $\mathrm{T}^{2n}\mathrm{S}$ with $n > 0$, but none that does not terminate with the S. This grammar contains ambiguities because some labels like P or X appear multiple times in different contexts. Such representations are more common in cognitive sciences. Figure modified with permission from Fitch et al. (2012). **(B)** Equivalent Markov-style diagram, where labels are at the nodes and ambiguous states are indexed. Note that the index is ignored for symbol (output) generation. This style is more frequent in Markov modeling and mathematical analyses. Figure based on Warren and Schroeder (2015). Although transition probabilities are assumed to be equal and are not depicted here, generally they are specified along the edges.

and grammars on the same hierarchical level can still have vastly differing complexity. Intuitively, a grammar with fewer symbols and restrictions on the transitions is easier to learn than one with many symbols and rules. Straightforward measures of "grammatical complexity" include simply counting the number of rules in it (Shanks and Johnstone, 1999) or patterns of substrings and the frequency of their occurrence (Perruchet and Vinter, 1998). For a more fine-grained distinction between sequences within the same complexity class (but not only), one can use information-theoretic and other string metrics, which ought to reflect the richness of the language the sequence belongs to.

A simple approach for this is based on the similarity of the strings the language can generate. These can be roughly categorized into edit distance, token-based and sequence-based metrics. Edit distance metrics, such as the Levenshtein or Hamming measure, compute the minimum number of changes required to transform one string into another. Averaged across all string pairs of a language, this is suggestive of its mean complexity: more difficult ones will have a larger average distance. Token-based algorithms such as the Jaccard index, where token refers to parts of a sequence like n-grams (not just one symbol), operate by evaluating the number of common tokens in the strings of the language. Sequence-based metrics, such as *complexity index* (Janson et al., 2004), look at the number of shared or distinct substrings within each or between different sequences. Intuitively, the more repetitions a string contains, the lower its complexity.

This is intimately related to the concept of sequence *compressibility* as a proxy for complexity. Intuitively, one can better compress a sequence that contains some redundant information (e.g., repetitive substrings) than one where each symbol is informative. This is formalized by the Kolmogorov complexity (Kolmogorov, 1963), which in this context can be informally described as the length of the shortest program (or description) that produces a particular sequence. For instance, the string ABABAB can be compactly described as "$3 * \mathrm{AB}$", while the string of the same length ABCBAD, which cannot be so easily reduced, is considered more complex. In general, the Kolmogorov complexity can be only approximated by measures such as the Lempel-Ziv, which serves as the basis for the Lempel-Ziv-Welch (LZW; Welch, 1984) lossless compression algorithm. LZW and similar algorithms work by encoding substrings into efficient codes that are stored in a dictionary, and can be adapted to generate strings of a target compression rate (Cahuantzi et al., 2021).

Instead of measuring the complexity of individual strings, a more comprehensive approach is to characterize the complexity of an entire grammar. Such global metrics can be grounded in Shannon's information entropy (Shannon, 1948; Jamieson and Mewhort, 2005, e.g., predictable sequences have low-entropy;), or based on the transition matrix of the grammar. An example of the latter is *topological entropy* (TE), introduced as a complexity measure for dynamical systems by Adler et al. (1965), and refined and popularized in AGL by Bollt and Jones (2000). Applied to a formal grammar, TE measures the growth rate of distinct sequences the grammar can generate (which can be therefore seen as a dynamical system). The method proposed by Bollt and Jones relies on a "lifting technique", which modifies the transition table to a first-order (memoryless) Markov representation in a larger matrix, on which TE can be computed using the eigenvalues (Robinson, 1998). However, performing the lifting is cumbersome and error-prone, so instead we will use a method put forward by Warren and Schroeder (2015) that circumvents the lifting and computes TE using a direct approach based on symbol subscripts.

In Chapter 8, we will use the method by Warren and Schroeder to control for the complexity of grammars and sequences in a systematic fashion. This is particularly important as a meta-analysis by Schiff and Katan (2014) revealed that, in virtually all experimental conditions, the subject performance in AGL tasks is strongly correlated with the complexity of the underlying grammar.

3.2 Recurrent neural networks and formal languages

Since Elman (1990) proposed RNNs as a natural system for discovering structure in temporal sequences, for reasons previously discussed in Chapter 2, a long and productive line of research sought to evaluate their computational (or expressive) power through formal analysis and empirical testing in the context of FLT (see e.g., Elman, 1991; Siegelmann and Sontag, 1991; Ackerman and Cybenko, 2020; Delétang et al., 2022). Theoretical results demonstrated early on that Elman-RNNs (or simple SRNNs) are Turing-complete under the assumption of unbounded computation time and arbitrary precision (Siegelmann and Sontag, 1992). In addition, the proposed architecture stored the whole input in its internal memory and performed the relevant computation (and produced output) only after reading a terminal token. These results extend to modern RNN architectures with hidden gating units, such as Gated Recurrent Units (GRU; Cho et al., 2014) and Long Short-Term Memory (LSTM; Hochreiter and Schmidhuber, 1997) networks, which are supersets of SRNNs. In practice, however, the computation steps and precision are limited, and the input is typically only available sequentially. Depending on how many of these practical constraints are taken into account, the complexity of languages RNNs can express is gradually restricted (Korsky and Berwick, 2019; Merrill et al., 2020; Ackerman and Cybenko, 2020).

If only linear computation steps and logarithmic precision are allowed, SRNNs and GRUs become equivalent to FSAs and can theoretically recognize all regular languages (Merrill et al., 2020), such as $\mathrm{AB}^*\mathrm{D}$ (B occurs arbitrarily often) or $\mathrm{A}^n\mathrm{B}^m$. In addition, they can learn some palindrome (Rodriguez and Wiles, 1997) or simple counting languages like $\mathrm{A}^n\mathrm{B}^n$ where the system needs to first count the number of As to generate a matching number of Bs (Wiles and Elman, 1995), which are context-free. LSTMs, which are *k-counter machines*, can additionally learn well-balanced parenthesis (counting) languages such as (shuffled) Dyck-1 using a dedicated counting mechanism (Suzgun et al., 2019). Compared to GRUs, which perform better on low-complexity sequences, LSTMs are better at predicting more difficult sequences (Cahuantzi et al., 2021). Moreover, LSTMs can handle some simple context-sensitive languages such as $\mathrm{A}^n\mathrm{B}^n\mathrm{C}^n$, but struggle to generalize to large n (Weiss et al., 2018). These results highlight the importance of distinguishing between theoretical bounds that consider mathematical limits and practical implementations and training algorithms of the model. Although a formal upper bound on the expressiveness in the context of FLT means that the system can (in theory) recognize all languages within and below that class, limitations of resources and gradient-based training methods typically restrict the ability of these models to generalize to sequence lengths beyond those seen during training (Gers and Schmidhuber, 2001; Weiss et al., 2018).

Learning more advanced languages requires exponential memory in terms of the input length, as illustrated by the failure of LSTMs to learn the Dyck-2 language (equivalent to all context-free languages; Sennhauser and Berwick, 2018; Suzgun et al., 2019). In

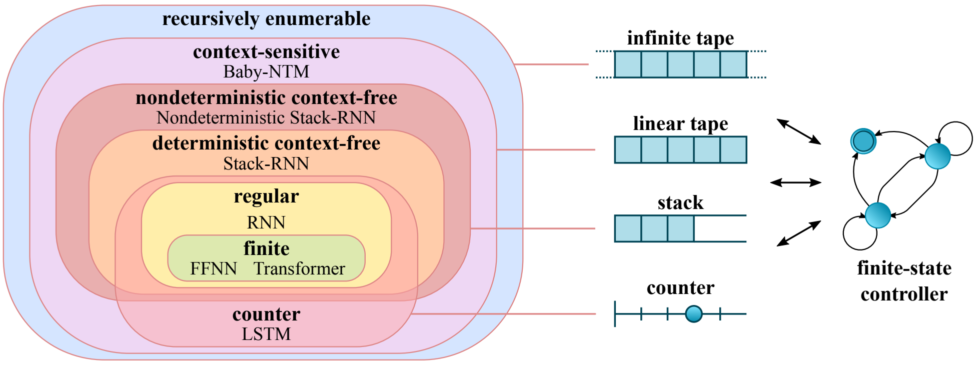


Figure 3.3: **RNNs and the Chomsky hierarchy.** Under non-idealized conditions, simple RNNs can process only regular languages, whereas supra-granular languages require additional storage for memory. Figure reproduced from Delétang et al. (2022).

a thorough study that compared different network architectures on a battery of tasks using the Chomsky hierarchy, Delétang et al. (2022) demonstrated that in order to generalize on sequences from context-free and more complex languages, dedicated (external) memory structures are necessary (see Figure 3.3). Such memory-augmented networks (Joulin and Mikolov, 2015) can copy and reverse strings, perform binary addition or learn cross-serial dependencies. These empirical results are indicative of the systems' theoretical computational power, and in practice, many of these achieve state-of-the-art performance on a variety of sequential tasks, including language processing, time series prediction and robotics (see e.g., Lipton et al., 2015, for a review).

From a neuroscientific perspective, perhaps more relevant than the actual performance is the nature of representations these RNN-based models learn, as well as the biological plausibility of the architectures and learning rules. Most models are trained with variants of backpropagation, which differs significantly from cortical mechanisms as discussed in Section 2.3, and often incorporate engineered solutions for specific functionality (e.g., forget gates in LSTMs). More importantly, their remarkable performance on many complex tasks may be attributed to efficiently sorting out irrelevant information and learning approximations to statistical regularities from large datasets, without actually learning the underlying (hierarchical) grammatical rules (Sennhauser and Berwick, 2018). This paradox is further confirmed by the inability of non-recurrent transformer architectures (Vaswani et al., 2017) to learn simple context-free languages (Delétang et al., 2022), despite achieving impressive results on natural language tasks that in principle correspond to mildly context-sensitive grammars (Joshi et al., 1990).

3.3 Biologically-constrained models

The previous section highlighted the direct relation between memory and the ability of a model to learn complex sequences. Given that achieving stable memories on long timescales in biologically plausible (spiking networks) is a notoriously hard problem, most of these models can only investigate relatively simple sequences from regular languages. Instead, they generally focus on evaluating the impact of various biophysical details of the neural circuitry on the computational capabilities of the system, as well as on their ability to capture and reproduce certain observations from behavioral and neural data. Typical tasks include learning a sequence and replaying it upon a cue, distinguishing and predicting context-dependent sequences with shared elements, such as "ABCD" and "EBCF", or learning the duration (timing) of a sequence and possibly replaying it at different speeds (generalization).

Here we attempt a brief survey of existing models, pointing out their strengths and weaknesses with respect to computational abilities and biological faithfulness. These can be categorized, albeit somewhat arbitrarily, into different classes of models depending on their architectural properties and type of neural dynamics they rely on. As such, one can differentiate between spiking and continuous firing rate networks, fully unsupervised and some form of supervised or reward-based learning, and the dynamical principles they leverage (e.g., transient dynamics). In addition, one ought to distinguish between models focusing simply on the generation and replay of sequential activation patterns observed in specific (sub-)cortical structures, and systems that consider more general and complex symbolic processing tasks.

3.3.1 Reproducing sequential activation patterns

Stereotypical sequential activation of neural cells has been observed during a variety of behaviors in many areas of the brain, including parts of the cortex (Pulvermüller and Shtyrov, 2009; Ikegaya et al., 2004), the hippocampus (Nádasdy et al., 1999), basal ganglia (Barnes et al., 2005) or the HVC motor nucleus in songbirds (Hahnloser et al., 2002). These manifest in strong but brief activations of some neurons, which propagate through the network and sequentially engage distinct cell assemblies. To capture such dynamics, many models assume that the connectivity contains a feedforward (asymmetric) structure along which activity flows (unidirectionally) from one group of neurons to another. Depending on whether the activity relies on elevated firing rates or rather precise (synchronous) spike timing, one typically distinguishes between rate-based propagation and *synfire chains* (Abeles, 1991; Diesmann et al., 1999; Kumar et al., 2010a).

Such networks with specific feedforward projections, in particular synfire chains, were suggested as a viable model for producing sequential activity in the specialized nucleus HVC (Jun and Jin, 2007; Fiete et al., 2010; Long et al., 2010). The chain-like structural scaffold can be hard-wired or learned via various forms of STDP and structural plasticity, and can additionally be embedded into a larger background network. Localized inhibition may improve the chain stability (Cannon et al., 2015), while accounting for neuronal specificities (in a model of hippocampal CA3 place cells) and a symmetric STDP rule facilitates the emergence of a bidirectional pathway upon which sequential activity can propagate in both directions (Ecker et al., 2022).

Given that most of these studies restricted the analysis to the generation of sequential activity, they can be classified as works on activity propagation and are therefore of limited expressive power with respect to learning multiple, structured sequences. Nevertheless, these models learn fully unsupervised and exhibit interesting properties such as spontaneous or cued replay of acquired sequences (also in reverse), the ability to store multiple (although a very limited number of) sequences, or reproduce experimental observations like sharp waves and ripple oscillations.

3.3.2 Static chains and dynamical trajectories for sequence processing

Before considering symbolic sequence processing per se, we ought to clarify a confusing and somewhat misleading categorization into “chain-like” and RNN-based models, prevalent in a number of studies on the topic (Cone and Shouval, 2021; Calderon et al., 2022; Murray and Escola, 2017). At a conceptual level, sequence processing involves representing elements (possibly also their patterned duration) and learning their order by establishing transitions between them. In the brain, sequences can be expressed both in a compressed manner and on a behavioral scale, depending on whether only the order or also the duration is represented (Foster and Wilson, 2006; Gavornik and Bear, 2014). As such, sequences can be replayed either on a circuit-specific, intrinsic timescale or close to behavioral timespans. Underlying these abilities is neural activity that unfolds in time, in some cases clearly sequentially (as discussed in the previous section) but also in many different modes (Babloyantz et al., 1985; Singer, 2013; Hardy and Buonomano, 2016).

To leverage sequential responses for sequence processing, one can attribute meaning to segments of the activity by associating them with particular symbols (stimuli). Replays of the engrained activity can then be considered to recall the learned sequence. In the case of synfire chains, which are among the simplest networks that could account for sequential activity, an element can be mapped onto one or multiple subsequently active groups, with the feedforward connectivity ensuring transitions between elements. Time is then an explicit (spatial) dimension represented by a strict feedforward propagation of activity through the network, underscoring the chain-like characteristic both at the architectural and dynamical levels. However, in the classical formulation of synfire chains, features like precise and synchronous spike timing, purely feedforward connectivity, the absence of inhibition and each neuron participating in a single chain are major limitations that make the rigid architecture biologically implausible.

Feedforward structures embedded in recurrent networks allow for neural variability that is more consistent with experimental data, while also providing long-lasting activations that are suitable for sequential processing (Goldman, 2009; Buonomano and Laje, 2010). Such networks are often referred to as “recurrent in architecture, but feedforward in function”. Although chain-like connectivity patterns can be embedded a-priori in the network, in many cases an “effectively feedforward functionality” emerges through learning from initially unorganized recurrent projections (Rajan et al., 2016; Hardy and Buonomano, 2018; Liu and Buonomano, 2009). While in some cases this translates to actual feedforward connectivity emerging at the synaptic level (corresponding to synfire chains as in Fiete et al., 2010), sequential (bump-like) activity may also arise dynamically through shifts in the E/I balance, without the need for hard-wired projections (Hardy and Buonomano, 2018).

Because the feedforward propagation of activity in some (Liu and Buonomano, 2009), but certainly not all of these networks resembles synfire transmission - in the sense that groups of neurons are active (once) sequentially and silent otherwise -, they are sometimes included in the same class of chain-like models. However, the underlying dynamics and emerging architecture are fundamentally different. In RNN-based models, each stimulus elicits a specific transient trajectory in the state-space, which may sometimes involve predominantly sequential but also significantly more complex activity.

In a key difference to synfire chains, such trajectories do not rely on strictly feedforward pathways but arise from complex interactions between the input and recurrent dynamics (Rajan et al., 2016; Goudar and Buonomano, 2018). Under realistic synaptic plasticity rules and homeostatic mechanisms, recurrent networks tend to build small clusters that exhibit slow dynamics through reverberating activity, where the stimulus-specific trajectory involves dynamical switching between multiple clusters (Litwin-Kumar and Doiron, 2014). Such (cyclic) clusters may play an important role in generating complex yet stable neural trajectories, but, crucially, the overall activity is jointly shaped by the input, recurrent structure and dynamical processes at single neuron/synapse (e.g., adaptation) and network level (e.g., E/I balance) (Hardy and Buonomano, 2018; Rajan et al., 2016).

Moreover, each spatiotemporal trajectory may involve many or even all neurons in the network, possibly activating them several times, whereas in synfire chains each neuron is part of a single path. This is also more consistent with the mixed stimulus selectivity displayed by many cortical neurons (Rigotti et al., 2013). Time is then represented implicitly through the evolution of the trajectory, which can be flexibly warped while maintaining encoding stability (Hardy and Buonomano, 2018). In addition, these may depend on the current and previous inputs, thus endowing the system with a form of natural dynamic memory through recurrence (see also Section 2.4). Computationally, this is more powerful than low-dimensional synfire chains because the system can exploit the available state-space dimensionality more efficiently.

Leveraging such transient dynamics, recurrent models prove particularly adept at learning and producing multiple timed motor patterns, which may last up to several seconds (Rajan et al., 2016; Mante et al., 2013; Laje and Buonomano, 2013). On the

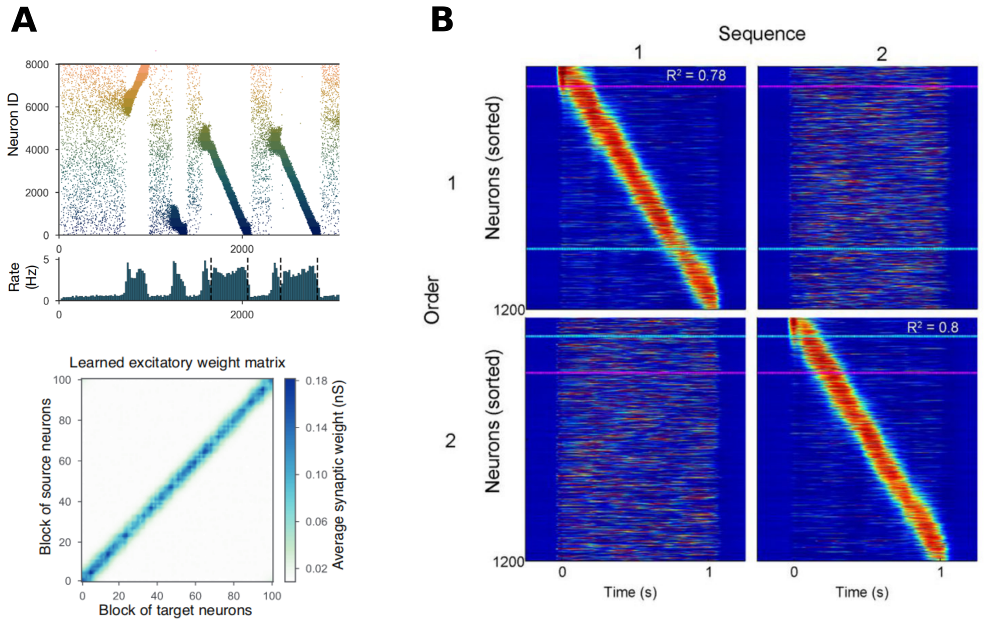


Figure 3.4: **Static chains and dynamics trajectories. (A)** Synfire-like sequential activations patterns (top) and weight matrix (bottom) after learning in the hippocampal model of Ecker et al. (2022). Learning induces a diagonal but bidirectional weight matrix that underlies feedforward and feedback replay. **(B)** Activity in a recurrent network that learned to encode multiple feedforward trajectories involving overlapping sets of neurons (Hardy and Buonomano, 2018). Which trajectory is replayed depends on the input stimulus.

downside, most of these are based on continuous rate networks, often disregard biology by allowing both negative and positive firing rates, and typically employ some kind of supervised (BPTT) or explicit feedback-based learning (FORCE; Sussillo and Abbott, 2009) to train the recurrent weights. Finally, these models typically aim at learning one or several patterns in isolation, without considering temporal relations between them. From this perspective, they can still be qualified as sequential activity generators, similar to synfire chains, even if the underlying mechanism differs.

In light of these observations, it becomes less obvious which characteristics qualify a model as “chain-like”. No matter how individual elements of a sequence are represented, temporal links (transitions) between them must be established somehow. At an abstract level, this requires a “chaining” ability from all sequence processors. For models like synfire chains that rely mainly on feedforward connectivity, whether pre-wired or learned or embedded in a larger network, one can argue that all aspects are chain-like: underlying structure, resulting activity, representation of time, as well as any possible temporal and contextual relations between elements must be carved in explicit synaptic pathways.

Moving toward more dynamical representations, one can assume a system where individual stimuli are encoded in specific trajectories (e.g., in segregated populations with graded response), and only the transitions between elements are learned through dedicated projections. In this case, the chain property could refer only to these transitions, irrespective of whether the trajectories rely on sequential activations (as long as they do not require exclusively feedforward connectivity). A completely dynamical approach, on the other hand, would involve representing both elements and the links in the transient dynamics. This means that stimulus representations must be dynamic and context-dependent, and all relevant history must be encoded along with the current representation. Although this is theoretically possible if the network possesses sufficiently long memory, in practice such systems can only learn short and relatively simple sequences. As we will see in the next section, this (memory) is also the reason why many models either use a combination of asymmetric connectivity and transient dynamics or involve external/dedicated components to handle contextual complexity.

3.3.3 Processing complex stimulus sequences

Illustrative examples of models based on sequential activity are two related studies by Maes et al. 2020; 2021. These models start from an initially random network and first learn a feedforward but circular pathway, through cyclical stimulation of cell assemblies and STDP, to produce a "clock-like" sequential activity (akin to but strictly speaking not synfire chains). A separate set of readout populations, each representing a specific element, are then mapped onto the circular activity through Hebbian co-activation. By decoupling the driving network from the readout layer, the model becomes very flexible and can learn arbitrary inputs, even sequences of sequences when multiple clocks on different timescales (hierarchical) are employed (Maes et al., 2021). However, this approach ignores a significant problem of sequence learning by outsourcing the token representations to some external dimension. In addition to the scarce experimental evidence of such clock-like or pattern generator dynamics in the cortex (except the HVC and possibly hippocampus), the ongoing activity also means that each and all sequences are replayed continuously, and selecting between them requires an external inhibitory input for suppressing all unwanted sequences. Moreover, the timing of sequences and their elements are fixed to the clock speed and cannot be changed, and multiple sequences cannot share the same element unless they are represented by distinct readouts.

In a model that can be considered a conceptual reversion of the above clock, Murray and Escola (2017) demonstrate that a purely inhibitory network based on striatal circuits can learn sequential activity patterns at one speed and replay them at variable rates. The mechanism relies on first acquiring a feedforward, circular pathway (essentially a clock) through the network along which the activity may propagate, with transitions between cell groups occurring due to short-term depression and winner-take-all dynamics. Targeted excitatory inputs from the cortex (seen as a tutor/controller) to specific sub-populations can then select which (possibly overlapping) subsequences are expressed, with the input intensity determining the replay speed. Although in principle this allows flexible selection and combination of particular subsequences, the processes governing these selections and their temporal relations, representing the core of the sequence learning problem, are assumed to happen in the cortex and are abstracted from. As such, the striatal circuit can be considered simply as an output system or manifestation of (symbolic) sequence processing in the cortex, with a more fluid mapping to and between element representations.

Closing this pathway with a thalamic module, Calderon et al. (2022) proposed a cortical - basal ganglia - thalamus loop in which token orders are encoded in a clustered RNN (cortex), while the transitions between elements are mediated through active gating and ramping activity along the basal ganglia - thalamus projections. Although the sequence order was essentially pre-wired, the model was able to learn the timing intervals and could replay the full sequence or just specific elements at various speeds and starting times. However, simplified and specific features of the rate-based model, such as orthogonal input and thalamocortical projections, single-neuron representations of the basal ganglia and thalamus, as well as ambiguity about how to store more than one sequence raise questions about its (details on) biological plausibility and computational effectiveness.

The above models achieved a particular type of flexibility through a conceptual separation between a controller and execution/representation. In other words, either the order of the elements, their duration or the selection of the active subsequence required input from an external, often abstracted system. However, both the duration and order of a sequence can be learned within the same network with an appropriate modular architecture (Cone and Shouval, 2021). Exploiting the columnar organization of specialized cell types, this model could learn and replay simple but long sequences. Stimulus duration is encoded by ramping recurrent activity of token-specific “Timer” populations, while chain-like projections through “Messenger” cells encoded the transitions. Despite using a plausible, reward-modulated Hebbian plasticity rule based on synaptic eligibility traces, the capacity of the network is determined in advance and there is no flexibility (temporal scaling) in the replayed durations. In a rate-based version, the addition of an external reservoir for extended memory allowed the network to also learn simple context-dependent sequences, but it remains unclear whether the approach would also work in spiking networks due to the necessary long time constants.

Maintaining contextual information on longer timescales may be supported by neuronal plasticity mechanisms such spike-rate adaptation Fitz et al. (2020). In this work, the authors demonstrated that intrinsic adaptation, even without recurrent connectivity, can yield sufficiently long memory (a few seconds) and sensitivity to serial order for a semantic labeling task in sentence comprehension. Given the recognition nature of the task, the model focused rather on the working memory aspect and was not designed to replay learned (or generate any kind of) sequences. Similar reservoir computing approaches for various language processing tasks were also investigated in discrete echo state networks, which typically achieve better performance than spiking models on significantly more complex grammatical sequences (see e.g. the work of Hinaut et al. 2013; 2015 or Dominey (2013) for a comprehensive review). Compared to static reservoirs, endowing the system with local, homeostatic synaptic and neuronal plasticity mechanisms significantly improves the performance on counting and occluder tasks, as demonstrated by the Self-Organizing Recurrent Network (SORN; Lazar, 2009) model. History is encoded in the form of stable, dynamic trajectories that allow the system to learn and make context-dependent predictions from simple artificial grammars (Duarte et al., 2014). An adaptation of the model to LIF neurons was also evaluated on a simple sequence replay task, with the principal goal and outcome being the reproduction of data from a visual experiment (Klos et al., 2018). Thus, it is not obvious whether the power of the discrete SORN can fully translate to spiking networks.

Past inputs can also be encoded in sparse, context-dependent neuronal activations that are learned in conjunction with dedicated, history-dependent synaptic pathways (Bouhadjar et al., 2022, 2023). Using structural Hebbian synaptic plasticity and rate-based homeostatic control, the spiking temporal memory model leverages nonlinear dendritic processing for (probabilistic) sequence prediction, mismatch detection and cued replay. Learning induces the maturation (growth) of an explicit, sequence-dependent pathway in an initially random network, along which activity propagates similarly to synfire chains. It can learn higher-order Markovian sequences with shared elements and exhibits relevant features such as prediction of the upcoming element through dendritic action potentials and generation of mismatch signals. On the downside, it also displays some weaknesses typical to synfire chains, including the reliance on strongly correlated input and activity, sensitivity to noise and little trial-to-trial variability, and inability to represent duration or handle longer input signals.

More generally, dendritic processing may play an active role in detecting temporal features (Leugering et al., 2023). Even simple models with two dendritic compartments can perform chunking of sequences, i.e.,g segmentation (recognition) of distinct sequences presented as a continuous stream (Asabuki and Fukai, 2020). Through a rather uncommon learning rule, whereby the dendrites try to predict the activity of the soma and thus attempt to minimize their response differences, chunk-specific cell assemblies emerged that exhibited stereotypical, sequential activation as a response to the preferred sequence. In the absence of recurrent connectivity, memory arose through history-dependent adaptation of the gain and threshold of the somatic transfer function. In a spiking version of this model (Asabuki et al., 2022) that did include recurrent projections, these served to gate the information flow from the dendrite to the soma in a context-dependent manner, acting as a sequence-specific filter from redundant/overlapping token representations learned by the dendrites. This yielded invariance to time-warped, as well as time-reversed input patterns. However, as the chunks were presented with varying gaps between them, it remains somewhat unclear whether the spiking model performs chunking or simply learns to represent higher-order sequences.

The emerging population activity in these networks is very similar to the model proposed by Klampfl and Maass (2013), which consisted of a recurrently coupled network of more abstract, winner-take-all circuits. Such competition-based dynamics allowed the system to encode noisy spatiotemporal patterns through sequential activity that is robust to temporal warping. Brief context windows led to distinct representations, whereas concatenation of multiple patterns induced a (sequential) interlinking of the individual representations, which could be replayed upon a cue or spontaneously. Given that the model was not explicitly designed for sequence processing, this ability was only demonstrated for two patterns. Despite plausible network activity and unsupervised learning, the network relied on engineered features such as symbolic inhibition (through firing rate normalization) and rigid WTA circuits with predefined mean activity. These raise doubts about its capacity for processing multiple and longer sequences in a more realistic setting, for instance with explicit inhibitory cells. Similar results were obtained in a related model based on WTA circuits (Mostafa and Indiveri, 2014), but this work relied heavily on a pre-wired feedforward architecture and exhibited highly implausible spiking activity.

The above models represent only a small, but representative subset of recurrent network models for sequence learning that are, at least to some degree, constrained by biophysical features. From a computational perspective, there are certainly other relevant models that provide insight into possible mechanisms of cortical processing. One example is the hierarchical Lotka-Volterra model for chunking based on winnerless competition dynamics proposed by Fonollosa et al. (2015). Leveraging a two-level network and Hebbian plasticity, the first layer learned representations of single tokens while the second one developed distinct, dynamical encodings for combinations of tokens (subsequences). At each level of the hierarchy, sequential memory was encoded as dynamic trajectories along a chain of metastable fixed points (heteroclinic channel). Effectively, the higher layer contained multiple metastable states, each associated with a heteroclinic sequence in the lower layer that encoded multiple tokens. Such hierarchical dynamics and representations are undoubtedly desirable traits of complex sequence processors, but it is unclear how multiple stable heteroclinic channels can emerge in biologically more detailed systems.

Moreover, these models are typically designed for and evaluated on only a few tasks, making a comparison of their computational abilities difficult. To address this issue, the second part of this thesis develops a benchmarking framework that enables such comparison systematically. As a proof-of-concept, we will demonstrate its features through a detailed investigation of three models.

Chapter 4

Functional Neural Architectures:

A toolkit for functional neural network benchmarking, analysis and comparison

Many of the models discussed in the previous chapter were implemented in custom Python or MATLAB code, or using dedicated neural simulators such as Brian (Stimberg et al., 2019) or NEST (Diesmann and Gewaltig, 2002). More generally, spiking networks can be simulated using a variety of tools, ranging from well-known machine learning libraries to neural simulators and full-fledged integrated simulation environments. Each of these tools has particular advantages, but, importantly, they cater to very different needs. Libraries such as TensorFlow (Abadi et al., 2016) or PyTorch (Paszke et al., 2019) are perfectly suited for training (simpler) networks using supervised learning algorithms, and are therefore often used for neuromorphic applications or spiking versions of deep learning models. However, they require manual specification of the model dynamics (possibly affecting reproducibility) and are less flexible about the model and architecture complexity. While dedicated spiking neural simulators overcome both of these limitations and are widely used in the computational neuroscience community, they are only that: simulators. All the functionality (e.g., task specification and analysis) must still be implemented by the user. Although this is addressed by integrated tools focused on functional networks (e.g., Nengo; Bekolay et al., 2013), these added features often come at the price of efficiency or supported level of biological detail. In the following, we introduce a toolkit that tackles all of the above aspects, providing a flexible tool for creating, simulating and analyzing functional spiking networks.

4.1 Description of the toolkit

Functional Neural Architectures (FNA) is a Python library for neuronal network benchmarking and analysis. It covers all components of a functional experiment involving neuronal networks, which can be roughly grouped into three distinct parts: task definition and input generation; model specification, instantiation and simulation; performance evaluation and a collection of detailed analysis scripts. FNA maintains a conceptual and practical separation between these parts, with interactions occurring through well-defined interfaces. This design is intended to make FNA as simulator-agnostic as possible, allowing it to leverage the vast repertoire of models and their efficient implementation available in existing tools. In other words, FNA provides standalone methods for input management and results analysis, but it relies on third-party simulation engines for handling the actual models. This also means that these simulators determine the types and properties of architectures (neuron and synapse models, circuit topology and connectivity) supported by FNA. Integration of such an engine requires only the extension of an abstract, high-level network object (wrapper), which represents the communication bridge between input and analysis. Currently, FNA includes wrappers for PyNEST, TensorFlow and a custom implementation of continuous rate neurons. Here we will focus more on the PyNEST interface, which was developed as part of and used in all projects of this thesis.

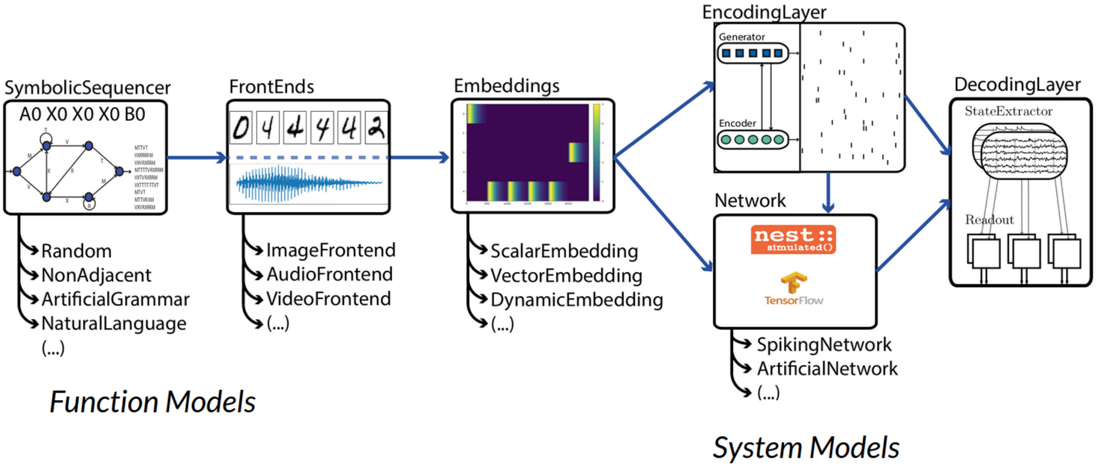


Figure 4.1: **Conceptual overview of Functional Neural Architectures (FNA) and its components.** *Function Models* refers to the task constraints and their functional significance, which are designed to probe the different systems' ability to deal with complex temporally patterned input sequences. FNA employs real-world computational tasks, common benchmark tasks used in the domain of computer science and machine learning as well as explicit experimental paradigms employed in the domain of cognitive science and psychology. *System Models* comprises all architectural and biophysical constraints imposed on the system, from the input encoding to the neuron model and optimization processes. Image adapted from Duarte (2021), with permission.

Intended not only as a benchmarking but also as a comparison tool, the modular architecture of FNA allows users to specify numerical experiments and perform them on a variety of networks. The experiments are typically formulated as computational tasks involving a functional mapping between the input and output, described using a common framework. It exploits the fact that critical features of many experiments, ranging from standard benchmark tasks in computer science and machine learning (e.g., classification) to experimental paradigms in cognitive sciences (e.g., working memory tasks), are sufficiently universal such that the same type of measurements can be used to evaluate performance. In particular, the tool includes all the ingredients of a RC framework (see Section 2.4.2), in which the computational capacity of circuits is probed using arbitrarily complex input stimuli and signals, and performance is measured independently of the system's specificities.

Making use of standardized routines and established simulation engines, FNA also addresses a major issue in computational neuroscience: reproducibility. The workflow ensures consistency of results across repeated runs and computing platforms (contingent on the simulation core providing this feature). With FNA, the same experiment can be executed just as easily on a local machine as on a compute cluster, taking advantage of all available resources.

4.2 Components

The schematic in Figure 4.1 illustrates the main components of the toolkit, which can be functionally grouped into three parts, mirroring the logical flow of a typical experiment: input and preprocessing, model setup and simulation, and postprocessing. Creating an experiment begins by specifying the computational task, i.e., defining the expected input/output mapping, and establishing how the input should be embedded, encoded and delivered to the network. In a second step, the network model is created and the input is processed. Finally, the network output can be used to evaluate the circuit's performance on the task. While each of these stages may depend strongly on the type of input and model properties, FNA aims to overcome these differences through common formalisms, specification and function interfaces in order to allow carrying out the same set of measurements across different circuits. For instance, in traditional ML models, there is no conceptual separation between the last two steps, running (training/testing) the model and decoding its performance.

In addition, there are three auxiliary but essential components of the toolkit: parameter definition, analysis and visualization. These are integrated into the individual parts of the main components to enable fine-grained control of and access to the elements of the experiments. For maximum flexibility, the parameter file is a single Python script containing separate dictionaries for each component. We will now go through each one in more detail.

4.2.1 Task definitions as symbolic sequences

Given that most temporal and non-temporal computational tasks can be formulated as operations on symbolic items or *tokens*, we specify the input/output relation as functions of these elements. Note that in the remainder of this chapter, we will use token, symbol or stimulus interchangeably. As we will see, this formulation leaves room for handling functions defined in both discrete and continuous time. For the moment, let's consider the case where both the input and target output are defined in discrete time. We borrow conventions from formal language theory as introduced in Chapter 2, and define the input $\mathbf{u}$ as a sequence $S_{\mathbf{u}} = \sigma_1, \dots, \sigma_T$ containing T discrete tokens. This is simply a layer of abstraction from the underlying data, the only constraint being that it can be transformed into a form accepted by the system.

At the lowest level, `SymbolicSequencer` provides a base class for managing such symbolic sequences. It maintains data structures for the alphabet and generated strings, along with functions for basic string-level operations, computing simple metrics and creating task-specific target outputs. Although this base class is meant to be derived when implementing more complex tasks, it already provides a set of default, generic mappings that can be used to probe some interesting functional properties. These include determining the token identity in the current step (classification), with the target output defined as $\hat{\mathbf{z}}(t) = \mathbf{u}(t)$, or k steps back (memorization task), formulated as $\hat{\mathbf{z}}(t) = \mathbf{u}(t-k)$, $\forall t \in T$. More generally, for each input sequence $\mathbf{u}$ we can specify a set of task-relevant outputs by defining multiple target functions

$$\hat{\mathbf{z}}_i(t) = f_i(\mathbf{u}(t)) \tag{4.1}$$

and use train individual linear readouts to learn these mappings using the RC approach (more on this shortly, see Section 4.2.5). Table 4.1 includes a list of tasks currently implemented in FNA, categorized based on the temporal relation of the tokens (random vs structured), and whether the output signal is analog or discrete (symbolic). These range from artificial grammar learning (AGL) to analog time series prediction. Analog signals defined in continuous time can be either generated externally and loaded as vector embeddings (see next section), or they can be represented as sequences of T distinct tokens (with T depending on the signal sub-sampling and/or input resolution).

Available tasks in FNA	
Task type	**Task name**
Structured analog	Dynamical system emulation (Mackey-Glass)
	Chaotic time series prediction*
Random analog	n-bit flip-flop
	NARMA
	Analog fading memory
	Continuous integration*
Random symbolic	Symbolic fading memory
	Temporal XOR*
Structured symbolic	Sequence recognition, memory, prediction, and chunking
	Non-adjacent dependencies
	Cross-modal generalization (real-world embeddings)
	Natural language processing
	Delayed-match to sample*
	1-2-AX working memory task*
	Deviant detection (odd-ball paradigm)*

Table 4.1: **Analog and symbolic tasks available in FNA.** The architecture and baseline data structures allow for uncomplicated implementation of new tasks. Examples for these are marked with * (currently not available as dedicated tasks).

4.2.2 Embeddings

At this point, the input $\mathbf{u}(t)$ consists of a series of abstract symbols, with minimal constraints on the type of underlying data. For the system to understand these inputs, they must first be transformed into consistent numerical representations (typically in vector space) through a process known as *embedding*. This is a common technique to map discrete variables to (relatively low-dimensional) vectors of continuous numbers. In machine learning applications, (learned) vector embeddings often involve taking into account semantic (or categorical) meaning, such as in word embeddings, where words with similar meanings are also expected to be mapped to points closer in vector space.

However, vector embeddings can be much more general, and FNA includes a variety of methods such as one-hot, scalar or random mappings. These are implemented as `VectorEmbeddings` objects. If the target network operates in continuous time, any such vector embedding can be further converted into a continuous one by unfolding it in time. Such `DynamicEmbeddings` are obtained either by applying a temporal mask (kernel) to generate a continuous signal, or by converting them to spatiotemporal spike patterns (frozen noise). In some cases, it may be useful to introduce variability during the unfolding process, for instance in the stimulus duration and amplitude, or by injecting temporal jitter (noise).

In addition, FNA provides an interface to use multimodal stimuli from real-world data, such as image (MNIST and Cifar-10) or audio (Heidelberg digits; Cramer et al., 2022) datasets. These sensory frontends can transform the data into the aforementioned embeddings, allowing the user a simple way to create complex symbolic processing tasks using naturalistic stimuli.

Embeddings can be visualized and analyzed using several standard plotting functions and metrics. These include plotting the input space (or a projection of it using one of the available dimensionality reduction methods) or evaluating the embedding quality using any distance measure.

4.2.3 Encoding

For networks operating in discrete time, the embedded input (as described above) can be delivered directly to the network. However, certain models, in particular spiking networks, require an additional *encoding* step that is often specific to the simulation engine used. For spiking models implemented in NEST, the tool includes a `NESTEncoder` object that converts the input to a form accepted by the simulator (current or spikes). Under the hood, continuous signals (embeddings) are translated into Poisson spike train using NEST generator devices, whereas spike patterns can be drawn according to specific stochastic processes (e.g., Ornstein-Uhlenbeck) using functions from the NeuroTools[1] library.

[1] https://github.com/NeuralEnsemble/NeuroTools

4.2.4 Network models

Due to the considerable differences in circuit properties, abstraction levels and implementation specificities across the various simulation engines, FNA requires writing a custom, high-level wrapper for each model type. This can be done by deriving the `Network` abstract class, which exposes a common API for critical function calls such as training and prediction. To bridge over conceptual differences between traditional deep learning and other optimization approaches, we follow the batch-based training and testing/predicting convention. Depending on the model, this stage involves learning the recurrent (internal) weights or simply simulating (evolving) the network dynamics. At the same time, the network state is sampled using predefined methods to train the task-specific readouts composing the decoder (described in detail below). There are three network models currently available: ANNs (via TensorFlow), continuous rate and spiking networks.

The spiking SNN model, central to this thesis, required a more careful treatment to obey the above convention. Biologically plausible models of spiking networks typically involve large, modular circuit architectures and complex wiring schemes, consisting of a variety of neuron types and synaptic connections. Moreover, it is often desirable to compute activity statistics not only for single cells but also on a coarser, population level. To this end, FNA provides `Population` objects to group individual neurons into homogeneous populations, storing their parameters as well as various state variables through recording devices. The `SpikingNetwork` object then acts as a wrapper for the collection of populations, providing functionality for creation, connection and activity management.

4.2.5 Decoder: state extraction and readouts

As the name suggests, the `Decoder` object is responsible for deciphering the ability of the system to perform the specified tasks. Following the RC approach, decoding requires two ingredients: a matrix with the (sampled) population responses and a set of readouts, one for each task. Population responses are gathered by `Extractor` objects, which can require model-specific implementations for extracting the activity from specific state variables. For spiking networks, one can use various sampling strategies (e.g., at specific time intervals or at stimulus offset) to record from continuous variables like the membrane potential of filtered spike trains. Such sampling options offer a large degree of flexibility, for instance, to evaluate how much information is retained about a stimulus at particular times after its offset. From the sampled activity we construct a state matrix $\mathbf{X} \in \mathbb{R}^{N \times T}$, with N representing the population size and T being the size of the input sequence. For discrete-time systems, the state matrix can be created directly from the unit outputs in each time step.

The state matrices are used to train multiple linear readouts in parallel, implemented

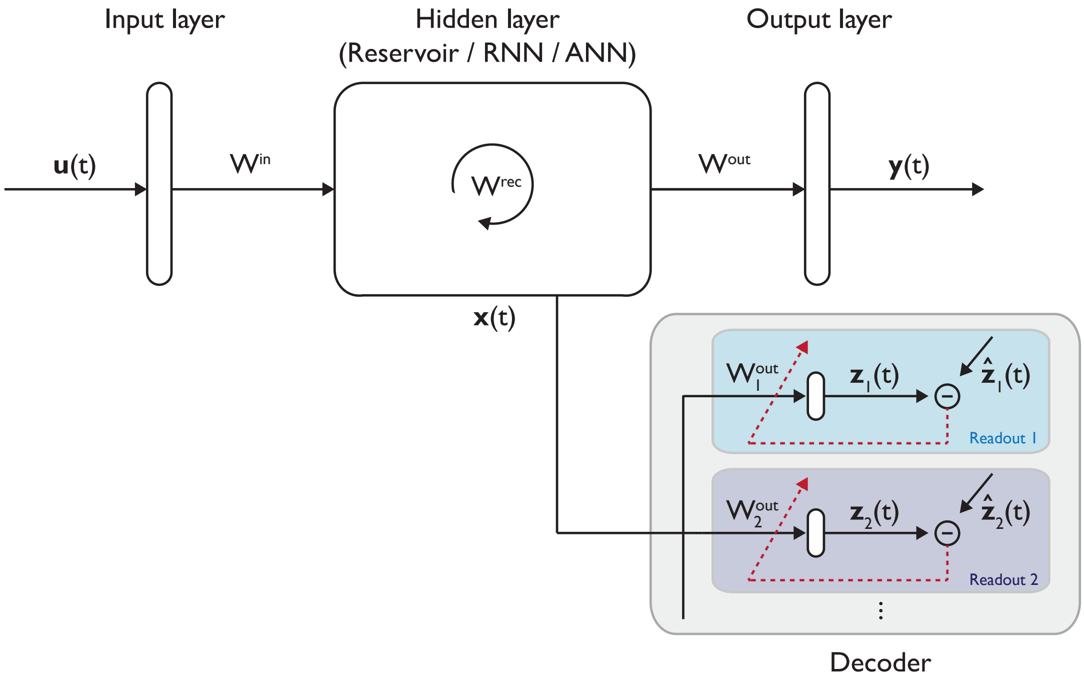


Figure 4.2: **Schematic overview of the decoding process.** The Decoder contains a number of task-specific but independent readouts, which are trained on the system's responses.

as `Readout` objects within the decoder. This is done by linearly combining the population responses using task-specific output weight matrices $\mathbf{W}_i^{\text{out}}$:

$$\mathbf{z}_i = g_i(\mathbf{W}_i^{\text{out}} \times \mathbf{X}), \tag{4.2}$$

where g_i is a suitable nonlinear function that can, for example, normalize the outputs (softmax) or take the maximum (hardmax or k-WTA). To optimize these weights according to some loss function $\mathcal{L}(\hat{\mathbf{z}}_i, \mathbf{z}_i)$, users can choose from a wide range of learning algorithms, including regression methods and gradient-descent based approaches such as ridge regression, SVM with RBF kernel, or linear classifiers with SGD training. The (trained) readout coefficients are held in a separate `OutputMapper` class, which stores the weights and also contains some analysis and plotting routines. Note that these target outputs are all functions of the same input and the solutions use the same population states, so what we are measuring is the ability of the network to create sufficiently rich input-driven dynamical states that allow multiple tasks to be decoded simultaneously.

In standard RC studies, training is a one-step procedure and takes place after simulation on the complete state matrix. However, for large networks with many (and densely) recorded variables, this poses a serious challenge on the computational resources. To reduce memory consumption, FNA supports batch processing in combination with an online training algorithm (e.g., stochastic gradient descent).

4.2.6 Analysis and visualization

FNA ships with a comprehensive set of analysis and visualization methods. In addition to routines included in the individual components as methods of the respective classes, there are separate modules for handling more generic data. The `analysis.metrics` module contains functions for, among others, characterizing the population activity, evaluating the state space complexity and reducing dimensionality. Similarly, the `visualization` package provides functions for visualizing results from the simulation and subsequent analysis, including three-dimensional projections of the state space and trajectories and plotting various performance metrics of readouts.

4.3 Example use case: simulation, analysis and visualization of a spiking network model

To illustrate some of the built-in features of FNA, we will now go through a simple use-case scenario. We test the ability of a spiking recurrent network to classify three digits from the MNIST dataset, and simultaneously probe whether its recurrent dynamics can retain the identity of the previous stimulus. Obviously, the scientific insight from such an experiment is limited, but it will allow us to touch on many important aspects of the tool.

To make things a bit more interesting, the stimuli are not presented in random order but instead obey the transition rules of a simple grammar (Figure 4.3A). As such, the first step is to create an `ArtificialGrammar` object encoding said transitions, from which we can then draw valid sequences (strings) such as ABBBC. Using the image frontend helper, FNA facilitates loading the dataset and automatically creates the mappings between the symbols in the grammar's vocabulary and the different image labels. In this case, A, B and C correspond to 0, 2 and 7. Because the dataset is loaded into a `VectorEmbeddings` object, we can already compute some metrics on it (e.g., distance measures) or simply plot a low-dimensional projection for a quick peek at the data (Figure 4.3B). At this point, each discrete stimulus is represented by a vector, so we need to unfold these into continuous signals that can be fed into our spiking network. Vector embeddings can be unfolded directly into spatiotemporal spike patterns of specific rate and duration, but here we will convert them into analog signals using an alpha-shaped

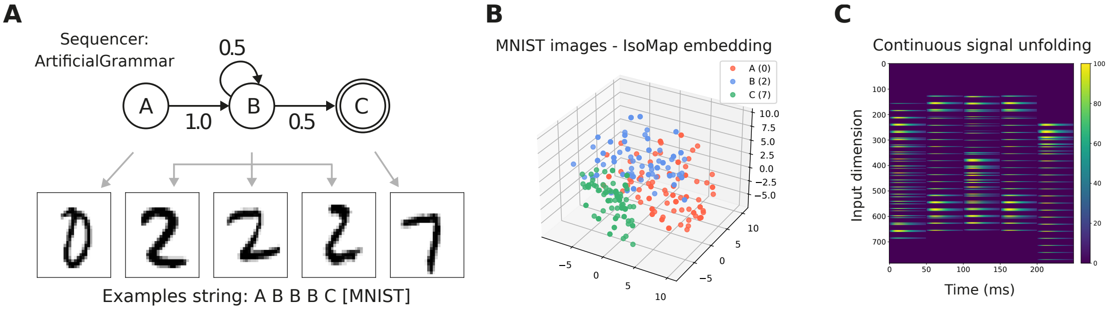


Figure 4.3: **Generating input for a simple task.** **(A)** Input sequences (strings) are drawn from the depicted grammar, with a one-to-one mapping between symbols and some numbers (labels) from the MNIST dataset. The images, presented in an ordered manner, will serve as stimuli to the network after proper encoding. **(B)** Low-dimensional projection of 300 random images of the 3 selected digits from the dataset. **(C)** Each image, embedded as 784-dimensional vector, is unfolded into a 50 ms continuous signal of the same dimensionality using an alpha-shaped kernel with an amplitude of 100 and a time constant of 12. This signal can be then encoded into either spike trains or current and injected into the network.

kernel (Figure 4.3C) in just a few lines:

```
image_mnist = ImageFrontend(path='../data/', label='mnist', vocabulary=['A', 'B', 'C'])
signal_pars = {
    'duration': 50.,
    'amplitude': 100.,
    'kernel': ('alpha', {'tau': 12})
    'dt': 1.,
}
image_mnist.unfold(to_signal=True, **signal_pars)
```

The tool offers a lot of options here, including specifying distributions for the amplitude and duration, different kernels, as well as adding noise (jitter) on top of the signal. As a last step of the input preparation, we create a NEST specific encoder that generates Poissonian spike trains with rate given by the signal amplitude.

Having specified the input, we can concentrate on the network itself and the tasks. For simplicity, we will use one of the example networks included in FNA, a balanced random network with LIF neurons. Setting up the decoding process involves two steps: specifying a state extractor to sample the responses and linking it to a decoder that will hold the readouts. Note that it is possible to create multiple extractors (with various sampling strategies and from different variables), and connect each of these to several decoders (using different training algorithms). Here we will sample the membrane potentials at stimulus offset. Next, we generate some sequences for training and testing (for brevity, only one batch), and specify classification and 1-step memory from the set of default tasks to generate the target labels. The network object's member functions `snn.train` and `snn.test` perform the simulation and also optimize the weights for each readout. After this step, the network and decoder contain all the relevant information which can be used for subsequent analysis and visualization. As shown in Figure 4.4, our simple network can distinguish between the stimuli (classification) but has no sufficient memory. Interestingly, the structure of our sequence is reflected in the higher recall accuracy for label B, which is seen more often during training.

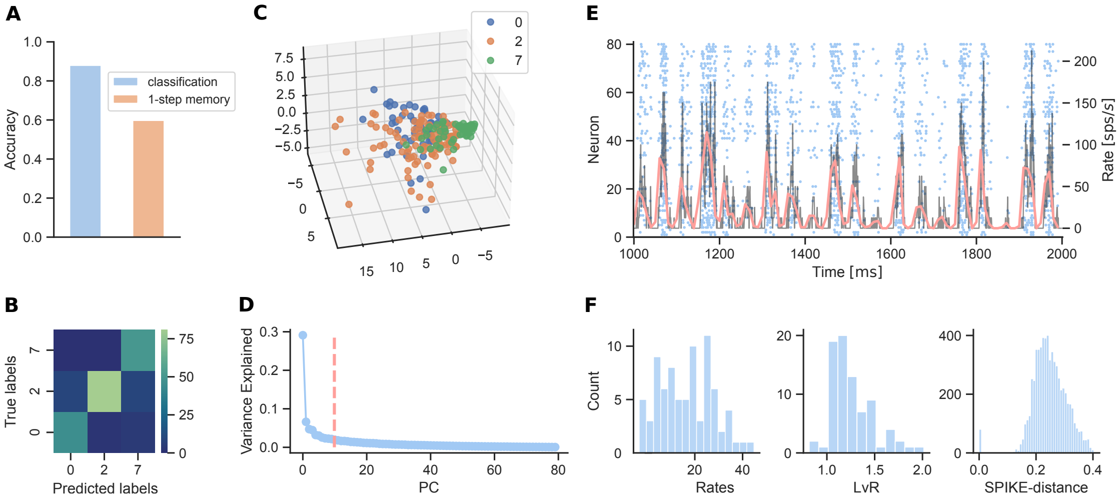


Figure 4.4: **Evaluating a functional experiment using FNA. (A)** Classification and 1-step memory performance of a small Brunel network ($N = 100$) on the task described in Figure 4.3. For each of the three MNIST digits, (the same) 5 images were used in both training and testing, which consisted of 1000 and 200 presentations, respectively. For the state variables, we used the low-pass filtered spike trains with $\tau_f = 20$ ms. **(B)** Confusion matrix for the three classes. **(C)** PCA projection of the 200 data points from the test phase. **(D)** Variance explained by the individual PCs. Red, dashed vertical line denotes the effective dimensionality of the system (see Mazzucato et al., 2015). **(E)** Raster plot and corresponding firing rates for the excitatory population. **(F)** Example activity statistics, showing (from left to right) the distributions of the mean firing rates, revised local variation of coefficient as a measure of irregularity (see Shinomoto et al., 2009), and SPIKE-distance for measuring synchrony (see Kreuz et al., 2013).

By walking through the different stages of this simple functional experiment, we managed to highlight only a small subset of the tool's capabilities. The published package includes detailed documentation and additional examples, while the next chapters of the thesis provide more complex and concrete models that can be built and analyzed with

FNA.

4.4 Conclusions and scope of the toolkit

In this chapter, we presented a modular and versatile toolkit for functional benchmarking and comparing neuronal networks. We provided a thorough description of the tool's components and features, and demonstrated its functionality through an example classification task. While there are no explicit constraints on the system properties, the tool is primarily meant to study the computational principles in recurrent networks. Decoupling the output from the system itself, it exploits the nonlinear and high-dimensional dynamics of such networks to probe them on multiple tasks in parallel using the RC paradigm, beyond what they were trained to solve. For certain tasks, this represents an efficient method to compare the performance of very different systems. On both these fronts, the currently implemented features reflect the origins of this project in computational neuroscience, with a strong focus on symbolic processing and biologically constrained network models.

Due to its modular architecture, the tool can be used in a variety of ways depending on the research objectives and software needs of the users. The core components, input generation and performance evaluation may be used individually and separately from the rest of the tool. For instance, tasks can be defined and input (embeddings) generated and plugged in into custom models and code. Similarly, one can take full advantage of the decoder and analysis simply by providing an output (state matrix) from different simulations.

The tool is part of a larger collaborative effort and is being continuously developed. It builds on and extends the scope of a previously published codebase, Neural Microcircuit Simulation and Analysis Toolkit (NMSAT; Duarte et al., 2017b), which I contributed to during my Master's thesis. The work undertaken as part of this dissertation focused on overhauling and extending the spiking model implementation and related components, updating to NEST 3.x, the decoding process, as well as practical aspects such as reproducibility issues and continuous integration tests. Additional, substantial changes include the modification of the code architecture to be used as a library, the creation of abstraction layers to facilitate custom network models, and the extension of the analysis methods. All subsequent chapters present modeling work that was performed using different versions of the tool or its precursor, NMSAT. In the remainder of this section, we compare FNA to other related tools and highlight some of its limitations.

4.4.1 Related work

As described above, the main advantage of FNA consists in the complete workflow that covers most aspects of functional neuronal network experiments. However, there are a number of tools that offer similar features as the individual components of FNA.

Neural network simulators Although FNA currently includes TensorFlow and NEST as simulation backends, in principle any other simulation engine could be integrated. For modeling detailed multi-compartment neurons, NEURON (Hines and Carnevale, 1997) or Arbor (Akar et al., 2019) are established and powerful options. However, these often do not scale well for large networks and long simulations. As a step towards large-scale models of single (but simpler) neurons, Brian2 (Stimberg et al., 2019) represents a flexible choice for intermediate network sizes. It offers a simple and intuitive Python interface for implementing new neuron and synapse models by specifying the differential equations, as well as dynamic interaction during runtime through network operations. Similarly to NEST, efforts are underway to provide GPU acceleration for Brian2 (Stimberg et al., 2020). Auryn (Zenke and Gerstner, 2014) is another spiking network simulator that is especially well-suited for investigating models with synaptic plasticity that require extended simulation times.

Other tools aim at a more abstract, population-level description and analysis of the network dynamics. The Brain Dynamics Toolbox (Heitmann et al., 2018) and DynaSim (Sherfey et al., 2018), both dynamical systems simulators in MATLAB, come with a wide range of functions (e.g., bifurcation analysis) and visualization routines (e.g., plotting phase portraits), with DynaSim providing methods for simplifying batch simulations and large parameter explorations. Going from population-level to whole-brain models, The Virtual Brain (TVB, Sanz Leon et al., 2013) enables the investigation of the macroscopic dynamics of the whole brain by considering each cortical region (node) as one neural population with realistic, local response behavior. Models in TVB are built using a vast array of structural (connectome) and neuroimaging information (EEG, fMRI), and the included analysis methods allow a direct comparison with experimental observations. All three tools also include graphical interfaces, making them more accessible to a broader user base.

Functional tools The tools described above are able to numerically solve the dynamical equations for concrete instances of various neural network models, but they rarely include components necessary for the functional benchmarking of biological networks. A tool that is specifically designed for investigating how cognitive functions can be implemented in neural circuits, and thus more related to FNA in scope, is Nengo (Bekolay et al., 2013). It builds on the Neural Engineering Framework (NEF, Eliasmith and Anderson, 2003) and the Semantic Pointer Architecture (SPA, Eliasmith, 2013) as theoretical underpinnings, which serve as guiding principles (and constraints) for model construction.

NEF requires defining tuning curves for each neuron and assumes that every population computes a user-specified target function through a linear combination of the responses, which is optimized for (essentially a linear readout). These learned outputs also represent the inputs to any connected target population. Clearly, these abstract assumptions constrain the biological plausibility of the models and may interfere with research objectives investigating more detailed neural mechanisms and complex, emerging dynamics. On the other hand, by relying on abstract symbol processing, SPA provides an interesting approach to tackling cognitive functions and forms the basis of the largest functional brain model, Spaun (Eliasmith et al., 2012).

From an application perspective, Nengo provides an ecosystem for creating functional spiking and non-spiking models, a GUI for interactive analysis, and a number of simulation backends that can be run on different physical hardware, including neuromorphic chips. One major difference to FNA lies in the abstraction level of the model specification: in Nengo, these are defined using a common formalism and the model object is entirely decoupled from the simulator, whereas in many cases FNA expects simulation-specific description while the network object, which exposes common function APIs, is essentially a wrapper around the simulator itself.

Using Nengo as a framework for RC is straightforward, but there are also other dedicated libraries such as ReservoirPy (Trouvain et al., 2020) or EchoTorch (Schaetti, 2018), focusing on Echo State Networks, or PeleNet (Michaelis, 2020), which targets neuromorphic devices. Rather a collection of different tools but with a more general and far-reaching scope, Computation Through Dynamics[2] is a valuable repository for analyzing computation and dynamics in recurrent - but, as of yet not spiking - networks. In addition, there are a number of tools that build on deep learning frameworks to implement spiking models and also include features to generate input signals, such as Rockpool (Muir et al., 2022), but these often involve non-local learning algorithms and are therefore less suitable from a neuroscientific perspective.

Tools for symbolic processing Given that the problem of symbolic sequence processing, as formulated here and implemented in FNA, has deep roots in linguistics and the "Good, Old-Fashioned AI (GOFAI)" prevalent up until the 1980s, it is perhaps not surprising that most available software resources are variants of cognitive models or tools employed in computational linguistics studies. Large cognitive architectures such as COGENT (Cooper and Fox, 1998), SOAR (Laird, 2012) or MIIND (de Kamps et al., 2008) can simulate cognitive phenomena and symbolic processing with various levels of biological plausibility, but the high degree of abstraction often makes them unsuitable for studying more detailed neural mechanisms. Focusing more on the input generation, artificial grammar learning tools such as AGL StimSelect (Bailey and Pothos, 2008), AGSuite (Cook et al., 2017) or pyagl (Beckers) can be used to create and analyze structured sequences similarly to FNA, but these are either no longer maintained or difficult to integrate with other systems. RNNExploration4SymbolicTS[3], a more recent library that allows the generation of sequences of tunable complexity, is offered as part of a

[2] https://github.com/google-research/computation-thru-dynamics

[3] https://github.com/robcah/RNNExploration4SymbolicTS

study comparing artificial RNNs for learning symbolic sequences (Cahuantzi et al., 2021). However, to the best of our knowledge, there are currently no tools that integrate such a wide range of cognitive-inspired tasks as FNA.

4.4.2 Limitations

These comparisons illustrate the key similarities and differences between FNA and other tools, which mostly capture only particular subsets of FNA's capabilities. At the same time, the flexibility offered by FNA is also one of its major flaws. The parameter specification often requires writing a lot of code compared to other tools, with nested dictionary structures that may be difficult for new users to get accustomed to. Moreover, some of these structures are still model- and simulator specific and could be made more consistent. Although FNA supports reproducible runs on both laptops and compute clusters and includes some basic features for parameter space exploration, these could be improved by facilitating the integration of established workflow management systems such as Snakemake (Mölder et al., 2021).

From a software engineering perspective, a critical aspect is the relatively tight coupling between different components, such as the sequence creation, task definition and embedding/encoding, or the model creation and the underlying simulator. Although the models are specified in a separate parameter file, the current design treats networks and simulators as one object and only exposes some standard functions. An alternative approach would be the decouple the network and model instance completely from the simulator, each defined as separate objects as in Nengo, and simply pass a model instance to the simulator object for training or running the network. Similarly, functional encapsulation could be improved by moving many analysis and visualization functions closer to the components they are related to. As a final note, future versions of FNA will also include options for saving and reloading networks, which can be particularly useful for large-scale simulations, as well as an extended set of tasks for benchmarking.

Chapter 5

Random and structured connectivity for representation transfer

5.1 Introduction

One of the advantages of the FNA toolkit presented in the previous chapter is the ease of creation and manipulation of large populations of spiking neurons embedded in a functional context. As discussed in Chapter 1, cortical information processing relies on a distributed functional architecture comprising multiple, specialized modules of spiking neurons that are arranged in complex, but stereotyped networks. Structural organizational principles are noticeable at different scales and impose strong constraints on the systems' functionality, while simultaneously suggest a certain degree of uniformity and a close relation between structure and function. In Section 1.4, we argued that a prerequisite for processing across such large distributed systems is the ability to suitably represent relevant features of spatiotemporal stimuli, and transfer these representations in a reliable and efficient manner through various processing modules. The majority of previous studies on spatiotemporal processing with spiking neural networks have either focused on local information processing without considering the role of, or mechanisms for, modular specialization (e.g. Maass et al. 2004), or on the properties of signal transmission within one or across multiple neuronal populations regardless of their functional context (Kumar et al. 2008a, 2010b; Diesmann et al. 1999; van Rossum et al. 2002; Shadlen and Newsome 1998; Joglekar et al. 2018, but see, e.g. Vogels and Abbott 2005, 2009 for counter-examples).

In order to quantify transmission accuracy and, implicitly, information content, these studies generally look either at the stable propagation of synchronous spiking activity (Diesmann et al., 1999) or asynchronous firing rates (van Rossum et al., 2002). The former involves the temporally precise transmission of pulse packets (or spike volleys) aided by increasingly synchronous responses in multi-layered feed-forward networks (synfire chains, see also Section 3.3.1); the latter refers to the propagation of asynchronous activity and assumes that information is contained and forwarded in the fidelity of the firing rates of individual neurons or certain sub-populations. An alternative approach was recently taken by (Joglekar et al., 2018), in which signal propagation was analyzed in a large-scale cortical model and elevated firing rates across areas were considered a signature of successful information transmission. However, no transformations on the input signals were carried out. Thus, a systematic analysis that considers both computation within a module and the transmission of computational results to downstream modules remains to be established.

In this chapter, we hypothesize that biophysically-based architectural features (modularity and topography) impose critical functional constraints on the reliability of information transmission, aggregation and processing. To address some of the issues and limitations highlighted above, we consider a system composed of multiple interconnected modules, each of which is realized as a recurrently coupled network of spiking neurons, acting as a state-dependent processing reservoir whose high-dimensional transient dynamics supports online computation with fading memory, allowing simple readouts such as linear classifiers to learn a large set of input-output relations (see Section 2.4.2 and Maass et al., 2002). Through the effect of the nonlinear nodes and their recurrent interactions, each module projects its inputs to a high dimensional feature space retaining time course information in the transient network responses. By connecting such spiking neural network modules, we uncover the architectural constraints necessary to enable a reliable transfer of stimulus representations from one module to the next. Using this RC approach (see Section 2.4.3 and Lukoševičius and Jaeger, 2009), the transmitted signals are conferred functional meaning and the circuits' information processing capabilities can be probed in various computational contexts.

5.2 Network architecture and analysis methods

5.2.1 Network architecture

We model systems composed of multiple, sequentially connected modules or sub-networks, abbreviated from here on as SSNs. Each SSN is a balanced random network (see, e.g. Brunel 2000), i.e., a sparsely and randomly connected recurrent network containing $N = 10000$ leaky integrate-and-fire neurons (described below), sub-divided into $N^{\mathrm{E}} = 0.8N$ excitatory and $N^{\mathrm{I}} = 0.2N$ inhibitory populations. Neurons make random recurrent connections within an SSN with a fixed probability common for all sub-networks, $\epsilon = 0.1$, such that on average each neuron in every SSN receives recurrent input from $K_{\mathrm{E}} = \epsilon N^{\mathrm{E}}$ excitatory and $K_{\mathrm{I}} = \epsilon N^{\mathrm{I}}$ inhibitory local synapses.

For simplicity, all projections between the sub-networks are considered to be purely feedforward and excitatory. Specifically, population E_i in SSN_i connects, with probability p_{ff}, to both populations E_{i+1} and I_{i+1} in subsequent sub-network SSN_{i+1}. This way, every neuron in SSN_{i+1} receives an additional source of excitatory input, mediated via $K_{\mathrm{SSN}_{i+1}} = p_{\mathrm{ff}} N^{\mathrm{E}}$ synapses (see Figure 5.1).

To place the system in a responsive regime, all neurons in each SSN further receive stochastic external input (background noise) from $K_{\mathrm{x}} = p_{\mathrm{x}} N^{\mathrm{x}}$ synapses. We set $N^{\mathrm{x}} = N^{\mathrm{E}}$, as it is commonly assumed that the number of background input synapses modeling local and distant cortical input is in the same range as the number of recurrent excitatory connections (Kumar et al., 2008a; Kremkow et al., 2010; Brunel, 2000).

In order to preserve the operating point of the different sub-networks, we scale the total input from sources external to each SSN to ensure that all neurons (regardless of their position in the network) receive, on average, the same amount of excitatory drive. Whereas $p_{\mathrm{x}} = \epsilon$ holds in the first (input) sub-network, SSN_0, the connection densities for deeper sub-networks are chosen such that $p_{\mathrm{ff}} + p_{\mathrm{x}} = \epsilon$, with $p_{\mathrm{ff}} = 0.75\epsilon$ and $p_{\mathrm{x}} = 0.25\epsilon$, yielding a ratio of 3:1 between the number of feedforward and background synapses.

For a complete, tabular description of the models and model parameters used throughout this study, see Supplementary Table A.1 and Supplementary Table A.2.

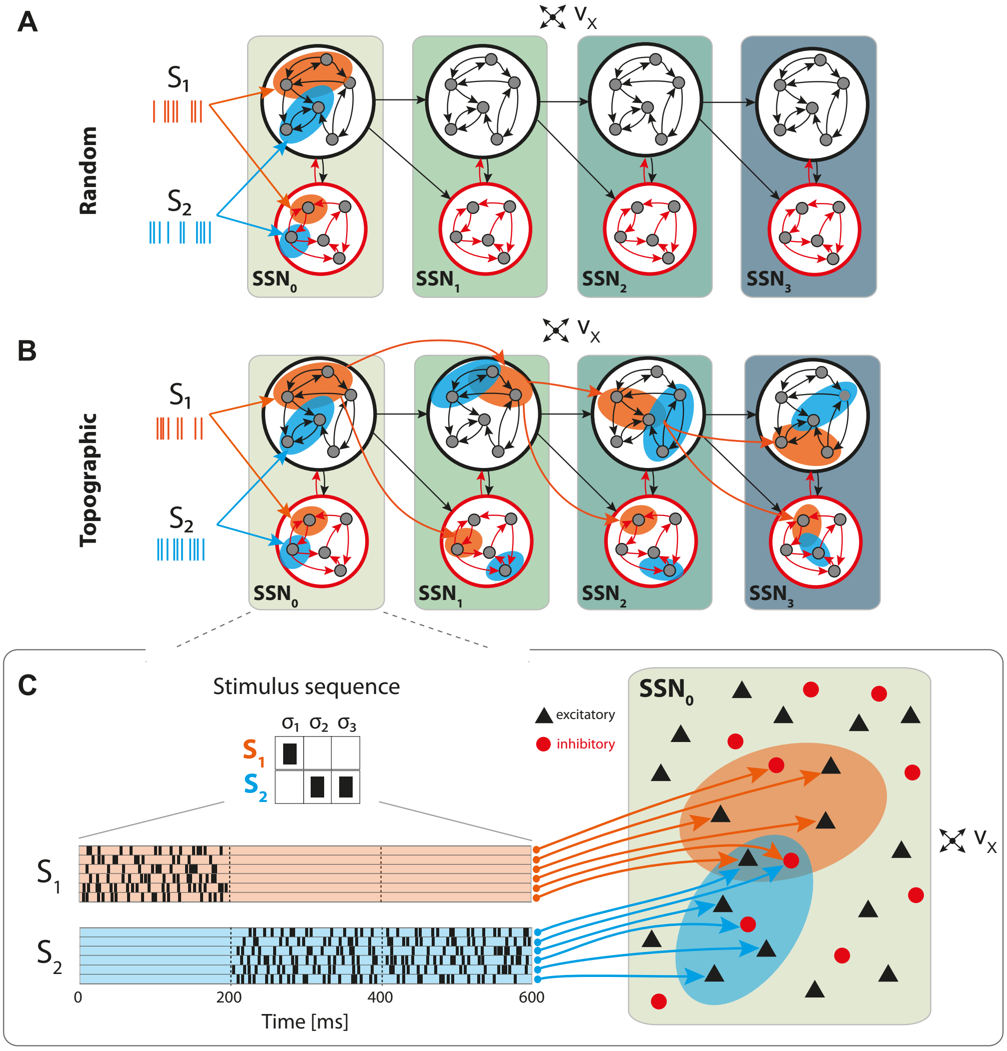


Figure 5.1: **Schematic overview of the sequential setup and input stimuli.** Networks are composed of four sub-networks with identical internal structure, with random (**A**) or topographically structured (**B**) feedforward projections. Structured stimuli drive specific, randomly selected sub-populations in SSN$_0$. For stimulus S_1, the topographic projections (**B**, orange arrows) between the sub-networks are represented explicitly in addition to the corresponding stimulus-specific sub-populations (orange ellipses), whereas for S_2 only the sub-populations are depicted (blue ellipses). The black feed-forward arrows depict the remaining sparse random connections from neurons that are not part of any stimulus-specific cluster. **C**: Input encoding scheme illustration: a symbolic input sequence of length T (3 here), containing $|S|$ different, randomly ordered stimuli ($S = \{S_1, S_2\}$), is encoded into a $|S| \times T$ binary matrix. Each stimulus is then converted into 800 Poissonian spike trains of fixed duration (200 ms) and rate ν_{stim} and delivered to a subset of ϵN^{E} excitatory and ϵN^{I} inhibitory neurons.

5.2.2 Structured feedforward connectivity

We explore the functional role of long-range connectivity profiles by investigating and comparing networks with random (Figure 5.1A) and topographically structured feedforward projections (Figure 5.1B).

To build systems with topographic projections in a principled, but simple, manner, a network with random recurrent and feedforward connectivity (as described in the previous section) is modified by systematically assigning sub-groups of stimulus-specific neurons in each sub-network. Each of these then connects only to the corresponding sub-group across the different SSNs. More specifically, each stimulus S_k projects onto a randomly chosen subset of 800 excitatory and 200 inhibitory neurons in SSN_0, denoted E_0^k and I_0^k. The connections from E_0^k to SSN_1 are then rewired such that neurons in E_0^k project, with probability p_{ff}, exclusively to similarly chosen stimulus-specific neurons E_1^k and I_1^k. These sub-populations in SSN_1 thus extend the topographic map associated with stimulus S_k. By repeating these steps throughout the system, we ensure that each stimulus is propagated through a specific pathway while projections between sub-networks from neurons not belonging to any topographic map remain unchanged (random). This connectivity scheme is illustrated for stimulus S_1 in Figure 5.1B.

It is worth noting that, as the stimulus-specific sub-populations are randomly chosen, overlaps occur (depending on the total number of stimuli). By allowing multiple feedforward synaptic connections between neurons that are part of different clusters, the effective connection density along the topographic maps (p_{ff}) is slightly increased compared with the random case (from 0.075 to 0.081). Any given neuron belongs to at most three different maps, ensuring that information transmission is not heavily biased by only a few strong connections. The average overlap between maps, measured as the mean fraction of neurons shared between any two maps, was 0.61. These values are representative for all sequential setups, unless stated otherwise.

5.2.3 Neuron and synapse model

The networks are composed of leaky integrate-and-fire (LIF) neurons, with fixed voltage threshold and conductance-based, static synapses. The dynamics of the membrane potential V_i for neuron i follows:

$$C_{\mathrm{m}}\frac{dV_i}{dt} = g_{\mathrm{leak}}(V_{\mathrm{rest}} - V_i(t)) + I_i^{\mathrm{E}}(t) + I_i^{\mathrm{I}}(t) + I_i^{\mathrm{x}}(t) \tag{5.1}$$

where the leak-conductance is given by g_{leak}, and I_i^{E} and I_i^{I} represent the total excitatory and inhibitory synaptic input currents, respectively. We assume the external background input, denoted by I_i^{x}, to be excitatory (all parameters equal to recurrent excitatory synapses), unspecific and stochastic, modeled as a homogeneous Poisson process with constant intensity $\nu_{\mathrm{x}} = 5$ Hz. Spike-triggered synaptic conductances are modeled as exponential functions, with fixed and equal conduction delays for all synapse types. The equations of the model dynamics, along with the numerical values for all parameters are summarized in Supplementary Table A.1 and Supplementary Table A.2.

Following Duarte and Morrison (2014), the peak conductances were chosen such that the populations operate in a balanced, low-rate asynchronous irregular regime when driven solely by background input. For this purpose, we set $\bar{g}^{\mathrm{E}} = 1\,\mathrm{nS}$ and $\bar{g}^{\mathrm{I}} = 16\,\mathrm{nS}$, giving rise to average firing rates of ~3 Hz, $\mathrm{CV_{ISI}} \in [1.0, 1.5]$ and $\mathrm{CC} \leq 0.01$ in the first two sub-networks, as described in the previous sections.

5.2.4 Stimulus input and computational tasks

We evaluate the information processing capabilities of the different networks on simple linear and nonlinear computational tasks. For this purpose, the systems are presented with a sequence of stimuli $\{S_1, S_2, ...\} \in S$, of finite total length T and comprising $|S|$ different stimuli.

Each stimulus consists of a set of 800 Poisson processes at a fixed rate $\nu_{\mathrm{stim}} = \lambda * \nu_{\mathrm{x}}$ and fixed duration of 200 ms, mimicking sparse input from an external population of size N^{E} (Figure 5.1C). These input neurons are mapped to randomly chosen, but stimulus-specific sub-populations of ϵN^{E} excitatory and ϵN^{I} inhibitory neurons in the first sub-network SSN_1, which we denote the *input sub-network*. Unless otherwise stated, we set $\lambda = 3$, resulting in mean firing rates ranging between 2-8 spikes/s across the network.

To sample the population responses for each stimulus in the sequence, we collect the responses of the excitatory population in each sub-network SSN_i at fixed time points t^*, relative to stimulus onset (with $t^* = 200$ ms, unless otherwise stated). These activity vectors are then gathered in a state matrix $X_{\mathrm{SSN}_i} \in \mathbb{R}^{N^{\mathrm{E}} \times T}$. In some cases, the measured responses are quantified using the low-pass filtered spike trains of the individual neurons, obtained by convolving them with an exponential kernel with $\tau = 20$ ms and temporal resolution equal to the simulation resolution, 0.1 ms. However, for most of the analyses, we consider the membrane potential V_{m} as the primary state variable, as it is parameter-free and constitutes a more natural choice (van den Broek et al., 2017; Duarte et al., 2018).

Unless otherwise stated, all results are averaged over multiple trials. Each trial consists of a single simulation of a particular network realization, driven by the relevant input stream(s). For each trial, the input-driven network responses are used to evaluate performance on a given task. In the case of the classification and XOR tasks described below, the performances within a single trial are always averaged over all stimuli.

Classification of stimulus identity

In the simplest task, the population responses are used to decode the identity of the input stimuli. The classification accuracy is determined by the capacity to linearly combine the input-driven population responses to approximate a target output (Lukoševičius and Jaeger, 2009):

$$\hat{Y} = W_{\text{out}}^{\intercal} X \tag{5.2}$$

where $\hat{Y} \in \mathbb{R}^{r \times T}$ and $X \in \mathbb{R}^{N^{\text{E}} \times T}$ are the collection of all readout outputs and corresponding states over all time steps T respectively, and W_{out} is the $N^{\text{E}} \times r$ matrix of output weights from the excitatory populations in each sub-network to their dedicated readout units. We use 80% of the input data for training a set of r linear readouts to correctly classify the sequence of stimulus patterns in each sub-network, where $r = |S|$ is the number of different stimuli to be classified. Training is performed using ridge regression (L_2 regularization), with the regularization parameter chosen by leave-one-out cross-validation on the training dataset. In the test phase, we obtain the predicted stimulus labels for the remaining 20% of the input sequence by applying the winner-takes-all (WTA) operation on the readout outputs $\hat{Y}$. Average classification performance is then measured as the fraction of correctly classified patterns.

Nonlinear exclusive-or (XOR)

We also investigate the more complex XOR task, involving two parallel stimulus sources S and S' injected into either the same or two separate input sub-networks. Given stimulus sets $S = \{S_0, S_1\}$ and $S' = \{S'_0, S'_1\}$, the task is to compute the XOR on the stimulus labels, i.e., the target output is 1 for input combinations $\{S_0, S'_1\}$ and $\{S_1, S'_0\}$, and 0 otherwise. In this case, computational performance is quantified using the point-biserial correlation coefficient (PBCC), which is suitable for determining the correlation between a binary and a continuous variable (Haeusler and Maass, 2007; Klampfl and Maass, 2013; Duarte and Morrison, 2014). The coefficient is computed between the binary target variable and the analog (raw) readout output $\hat{Y}(t)$, taking values in the $[-1, 1]$ interval, with any significantly positive value reflecting a performance above chance.

5.2.5 State space analysis

For a compact visualization and interpretation of the geometric arrangement of the population response vectors in the network's state-space, we analyze the characteristics of a low-dimensional projection of the population state vectors (membrane potentials) obtained through principal component analysis (PCA). More specifically, each N^{E}-dimensional state vector $x_i \in X_{\text{SSN}_i}$ is first mapped onto the sub-space spanned by the first three principal components (PCs), yielding a cloud of data points i which we label by their corresponding stimulus id.

In this lower-dimensional representation of the neuronal activity, we then evaluate how similar each data point in one stimulus-specific cluster is to its own cluster compared to neighboring clusters. This is done by assigning a silhouette coefficient $s(i)$ (Rousseeuw, 1987) to each sample i, computed during a single trial as:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}. \tag{5.3}$$

$a(i)$ represents the average distance between i and all other data points in the same cluster (same stimulus label), while $b(i)$ is the mean distance of i to all points in the nearest cluster, i.e., corresponding to a different stimulus label. The coefficients $s(i)$ take values between $[-1, 1]$, with a value close to 1 indicating that the data point lies well within its assigned cluster (correct stimulus label), whereas values close to -1 imply an incorrect cluster assignment and therefore indicate overlapping stimulus representations in the network activity.

To get a single value that is representative of the overall clustering quality in one specific trial, we computed the silhouette score by averaging over all the silhouette coefficients $s(i)$. Note that for the results presented in Figure 5.4B, the silhouette scores were computed using projections onto the first ten PCs, and were further averaged across ten different trials.

In addition to the cluster separation, we also quantify the dimensionality of the subspace where the neuronal activity predominantly lies, using the method introduced in Abbott et al. (2011) and Mazzucato et al. (2016). After performing a standard Principal Component Analysis on the firing rate vectors (average neuronal activity during a single stimulus presentation), we calculated the effective dimensionality as:

$$d = \left(\sum_{i=1}^{N} \widetilde{\lambda}_i^{\,2} \right)^{-1}, \tag{5.4}$$

where N is the real dimensionality of the network's state-space, i.e., the total number of neurons, and $\widetilde{\lambda}_i$ represents the fraction of the variance explained by the corresponding principal component, i.e., the normalized eigenvalues of the covariance matrix of the firing rates. For the analysis in Figure 4C, D and Figure 8, the number of PCs considered was limited to 500.

5.2.6 Numerical simulations and analysis

All numerical simulations were conducted using the Neural Microcircuit Simulation and Analysis Toolkit (NMSAT) v0.2 (Duarte et al., 2017b), the precursor of FNA (see Chapter 4). To ensure the reproduction of all the numerical experiments and figures presented in this study, and abide by the recommendations proposed in (Pauli et al., 2018), we provide a complete code package at https://osf.io/nywc2/, based on NEST 2.12.0 (Kunkel and Schenck, 2017), along with the complete set of parameters in Appendix A.

5.3 Simulation results

Distributed information processing across multiple neural circuits requires, in a first instance, an accurate representation of the stimulus identity and a reliable propagation of this information throughout the network. In the following section, we assess these capabilities using a linear classification task in a sequential setup (illustrated in Figure 5.1), and analyze the characteristics of population responses in the different sub-networks. Subsequently, we look at how different network setups handle information from two concurrent input streams by examining their ability to perform nonlinear transformations on the inputs.

5.3.1 Sequential transmission of stimulus representation

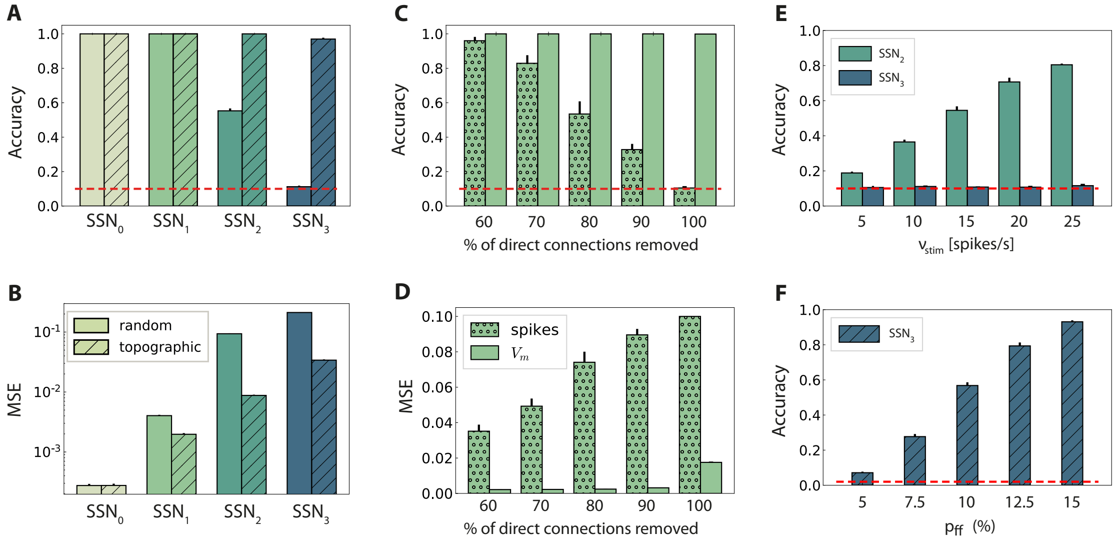


Figure 5.2: **Stimulus classification in sequentially connected modular networks. A, B**: Mean classification accuracy over $|S| = 10$ stimuli and corresponding mean squared error in each of the four sub-networks in the random (plain bars) and topographic (hatched bars) conditions. **C, D**: Mean classification accuracy and corresponding mean squared error in SSN_1 as a function of the number of direct projections (from neurons receiving direct stimulus input in SSN_0 to neurons in SSN_1) when decoding stimulus information from the low-pass filtered spike trains (stippled bars) and the membrane potential (plain bars). **E**: Classification accuracy in SSN_2 and SSN_3 decoded from the membrane potential as a function of the input intensity. **F**: Classification accuracy over $|S| = 50$ stimuli in SSN_3 as a function of the connection density within the topographic projections. All panels show the mean and standard deviations obtained from ten simulations per condition.

In networks with fully random projections (Figure 5.1A), stimulus information can be accurately decoded up to a maximum depth of 3, i.e. the first three SSNs in the sequential setup contain sufficient information to classify (significantly beyond chance level) which of the ten stimuli had been presented to the input sub-network (see Section 5.2 for details of the stimulus generation and classification assessment). Whereas the first two sub-networks, SSN_0 and SSN_1, achieve maximum classification performance with virtually no variance across trials (Figure 5.2A, plain bars), the accuracy of ≈ 0.55 observed in SSN_2 indicates that the stimulus representations have become degraded. These results suggest that while random connectivity between the sub-networks allows the input signal to reach SSN_2, the population responses at this depth are already insufficiently discernible to propagate further downstream, with SSN_3 entirely unable to distinctly represent the different stimuli.

Including structured projections in the system (Figure 5.1B) counteracts these effects, allowing stimulus information to be accurately transferred to the deeper sub-networks (Figure 5.2A, hatched bars). This indicates that stimulus-specific topographic maps, whereby the neurons receiving direct stimulation at SSN_i connect exclusively to another set of stimulus-specific neurons in the subsequent SSN (see Section 5.2), play a critical role in the successful propagation of signals across multiple interacting sub-networks.

As computing the accuracy scores involves a nonlinear post-processing step (winner-takes-all, see Section 5.2), we additionally verify whether this operation significantly biases the results by evaluating the mean squared error (MSE) between the raw readout outputs $\hat{Y}$ and the binary targets Y. These MSE values, depicted in Figure 5.2B, are consistent: performance decays with depth for both network setups, with topography leading to significant computational benefits for all sub-networks beyond the SSN_0. In the following two sections, we investigate the factors influencing stimulus propagation and uncover the relationships between the underlying population dynamics and the system's task performance.

Modulating stimulus propagation

Since random networks provide no clearly structured feedforward pathways to facilitate signal propagation, it is unclear how stimulus information can be read out as far as SSN_2 (Figure 5.2A), considering the nonlinear transformations at each processing stage. However, by construction, some neurons in SSN_0 that receive input stimulus directly also project (randomly) to SSN_1. To assess the importance of these directed projections for information transmission, we gradually remove them and measure the impact on the performance in SSN_1 (Figure 5.2C, D). The system shows substantial robustness with respect to the loss of such direct feedforward projections, as the onset of the decline in performance only occurs after removing half of the direct synapses. Furthermore, this decay is observed almost exclusively in the low-pass filtered responses, while the accuracy of state representations at the level of membrane potentials remains maximal. This

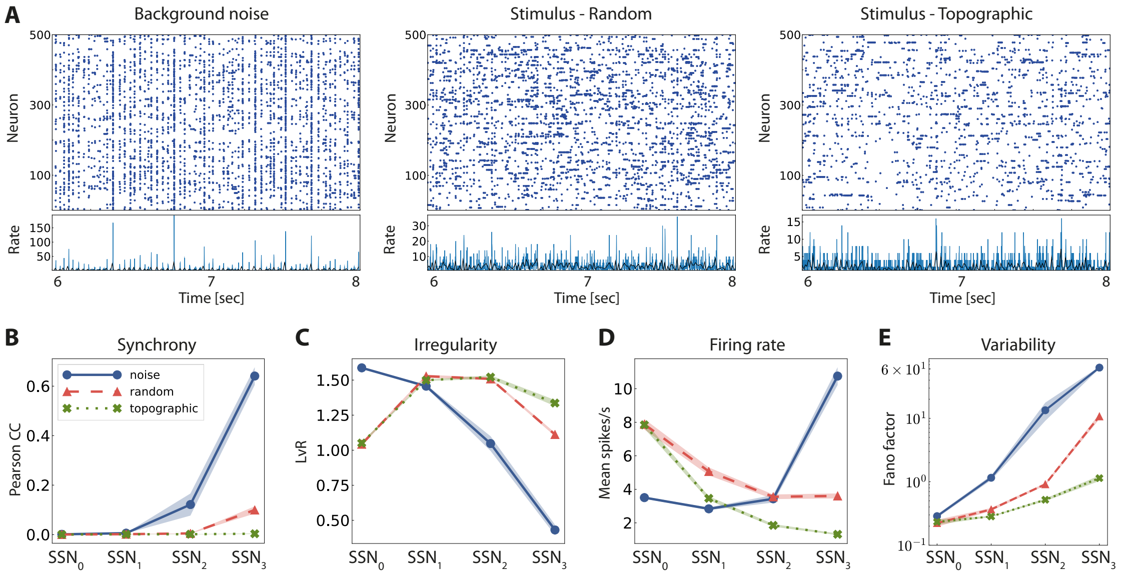


Figure 5.3: **Network activity in three different scenarios: purely noise-driven (no stimulus); stimulus-driven with random feed-forward connections and stimulus-driven with structured, topographic projections.** Top row (**A**) shows 2 seconds of spiking activity and the corresponding firing rates of 500 randomly chosen excitatory neurons in SSN_2. The corresponding statistics across the excitatory populations are shown in the bottom row (**B-E**), for every SNN: **B** — synchrony (Pearson's correlation coefficient, computed pairwise over spikes binned into 2 ms bins and averaged across 500 pairs); **C** — irregularity (measured as the revised local variation, LvR; Shinomoto et al., 2009); **D** — mean firing rate across the excitatory populations; and **E** response variability as measured by the Fano factor (FF) on the population-averaged firing rates (bin width 10 ms). All depicted statistics were averaged over ten simulations, each lasting 10 seconds, with ten input stimuli.

suggests that the populations in SSN_0 are not only able to create internal representations of the stimuli through their recurrent connections, but also transfer these to the next SSN in a suitable manner. The different results obtained when considering spiking activity and sub-threshold dynamics indicate that the functional impact of recurrence is much more evident in the population membrane potentials.

It is reasonable to assume that the transmission quality in the two networks, as presented above, is susceptible to variations in the input intensity. For random networks, one might expect that increasing the stimulus intensity would enable its decoding in all four sub-networks. Although stronger input does improve the classification performance in SSN_2 (Figure 5.2E), this improvement is not visible in the last sub-network. When varying the input rates between 5 and 25 spks/sec, the accuracy increases linearly with the stimulus intensity in SSN_2. However, the signal does not propagate to the last sub-network in a decipherable manner (results remain at chance level), regardless of the input rate and, surprisingly, regardless of the representational accuracy in SSN_2.

Previous studies have shown that, when structured feedforward connections are introduced, the spiking activity propagation generally depends on both the synaptic strength and connection density along the structures, with higher values increasing the transmission success (Vogels and Abbott, 2005; Kumar et al., 2010b). To evaluate this in our model without altering the synaptic parameters, we increase the task difficulty and test the ability of the last sub-network, SSN_3, to discriminate 50 different stimuli. The results, shown in Figure 5.2F, exhibit a significantly lower performance for the initial topographic density of (7.5%), from ≈ 1 for ten stimuli (Figure 5.2A) to ≈ 0.3. This drop can be likely attributed to overlapping projections between the sub-networks, since more stimulus-specific pathways naturally lead to more overlap between these regions, causing less discriminable responses. However, this seems to be compensated for by increasing the projection density, with stronger connectivity significantly improving the performance. Thus, our simulations corroborate these previous experiments: increasing the connection density within topographic maps increases the network's computational capacity.

Population activity and state separability

To ensure a perfect linear decoding of the input, population responses elicited by different stimuli must flow along well segregated, stimulus-specific regions in the network's state-space (separation property, see Maass et al. 2002). In this section, we evaluate the quality of these input-state mappings as the representations are transferred progressively through the sub-networks, and identify population activity features that influence the networks' computational capabilities in various scenarios.

When a random network is driven only by background noise, the activity in the first two SSNs is asynchronous and irregular but evolves into a more synchronous regime in SSN_2 (see example activity in Figure 5.3A left, and noise condition in Figure 5.3B). In the last sub-network, the system enters a synchronous regime, which has been previously shown to negatively impact information processing by increasing redundancy in the population activity (Duarte and Morrison, 2014). This excessive synchronization explains the increased firing rates, reaching $\approx 10\,\text{spks/sec}$ in SSN_3 (Figure 5.3D). Previous works have shown that even weak correlations within an input population can induce correlations and fast oscillations in the network (Brunel, 2000). This phenomenon arises in networks with sequentially connected populations and is primarily a consequence of an increase in shared pre-synaptic inputs between successive populations (Shadlen and Newsome, 1998; Tetzlaff et al., 2003; Kumar et al., 2008a). As the feedforward projections gradually increase the convergence of connections between sub-networks, the corresponding magnitude of post-synaptic responses also increases towards the deeper populations. Effectively stronger synapses then shift the network's operating point away

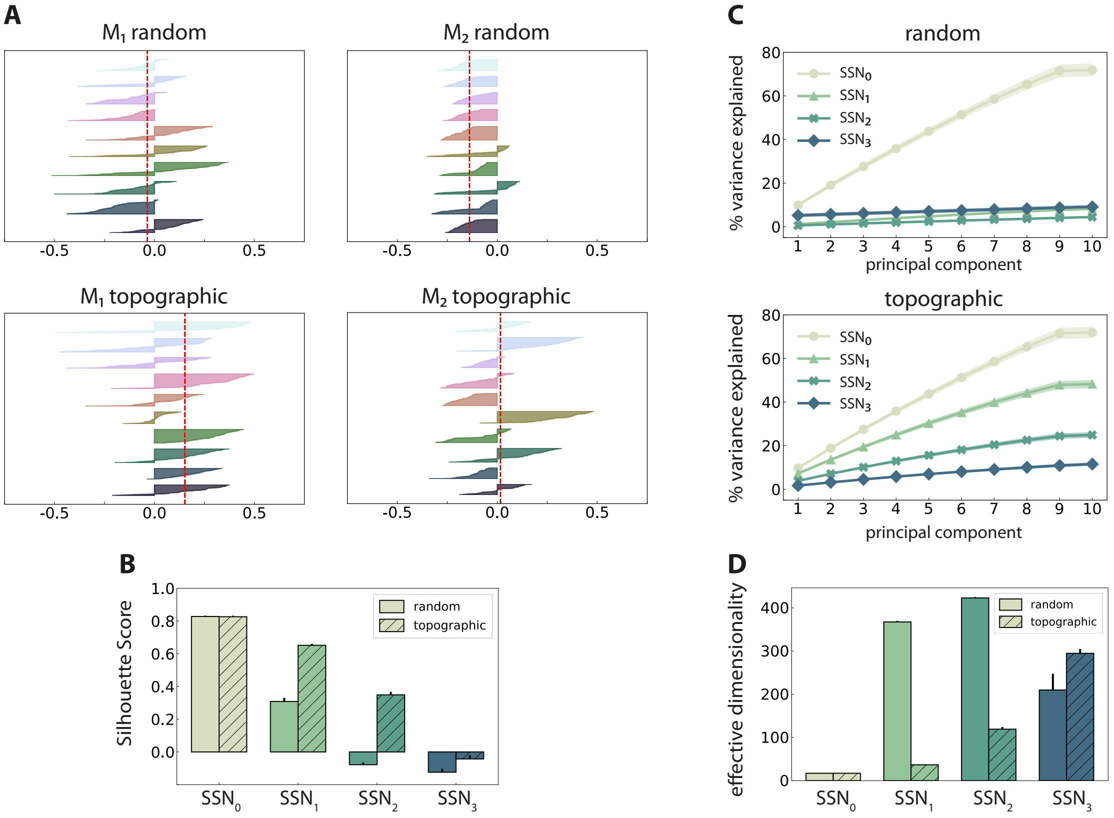


Figure 5.4: **Spatial arrangement and cluster analysis of stimulus-specific state vectors. A**: Distribution of silhouette coefficients for the stimulus-specific clusters in SSN_1 and SSN_2, computed in the space spanned by the first three principal components (PCs) of the state vectors (membrane potentials). Here each stimulus was presented ~50 times, resulting in clusters containing 50 data points and associated coefficients, color-coded and sorted in descending order for each of the ten stimuli used. The vertical lines in red represent the mean over all coefficients (silhouette score) in a single trial. **B**: Trial-averaged silhouette score calculated using the first ten PCs. **C**: Cumulative variance explained by the first ten PCs for random (top) and topographic (bottom) projections. **D**: Effective dimensionality of the state matrix computed on the firing rates (bin size 200 ms). All results are averaged over ten trials, each lasting 100 seconds (500 samples).

from the desired Poissonian statistics. This effect accumulates across the sub-networks and gradually skews the population activity towards states of increased synchrony.

Compared to baseline activity, the presence of a patterned stimulus increases the irregularity in all SSNs except the very first one. This is visualized in the example activity plots in Figure 5.3A (center and right). Furthermore, active input substantially reduces the synchrony in the last two SSNs, allowing the system to globally maintain the asynchronous irregular regime (see random and topographic conditions in Figure 5.3B, C). Such alterations in the population response statistics during active processing have also been confirmed experimentally: *in vivo* recordings show that neuronal activity in awake, behaving animals is characterized by weak correlations and low firing rates in the presence of external stimuli (Ecker et al., 2010; Vaadia et al., 1995).

Despite the beneficial influence of targeted stimulation, it appears that random projections are not sufficient to entirely overcome the effects of shared input and excessive synchronization in the deeper sub-networks (e.g. in SSN_3, CC ≈ 0.12, with a correspondingly high firing rate). The existence of structured connectivity, through conserved topographic maps, on the other hand, allows the system to retain an asynchronous firing profile throughout the network. Whereas the more synchronous activity in random networks, coupled with larger variability in the population responses (Figure 5.3E), contributes to their inability to represent the input in the deeper SSNs, topographic projections lead to more stable and reliable neuronal responses that enable the maintenance of distinguishable stimulus mappings, in line with the performance results observed in Figure 5.2A.

Furthermore, networks with structured connectivity are also more resource-efficient, achieving better performance with lower overall activity (Figure 5.3D). This can be explained by the fact that neurons receiving direct stimulus input in SSN_0, firing at higher rates, project only to a restricted sub-population in the subsequent SSNs, thereby having a smaller impact on the average population activity downstream.

The above observations are also reflected in the geometric arrangement of the population response vectors, as visualized by the silhouette coefficients of a low-dimensional projection of their firing rates in Figure 5.4A (see Section 5.2). As stimulus responses become less distinguishable with network depth, the coefficients decrease, indicating more overlapping representations. This demonstrates a reduction in the compactness of stimulus-dependent state vector clusters, which, although not uniformly reflected for all stimuli, is consistent across sub-networks (only SSN_1 and SSN_2 shown). However, these coefficients are computed using only the first three principal components (PCs) of the firing rate vectors and are trial-specific. We can obtain a more representative result by repeating the analysis over multiple trials and taking into account the first ten PCs (Figure 5.4B). The silhouette scores computed in this way reveal a clear disparity between random and topographic networks for the spatial segregation of the clusters, beginning with SSN_1, in accordance with the classification performances (Figure 5.2A).

We can further assess the effectiveness with which the networks utilize their high-dimensional state-space by evaluating how many PCs are required to capture the majority of the variance in the data (Figure 5.4C). In the input SSN, where the stimulus impact is strongest, the variance captured by each subsequent PC is fairly constant ($\approx 10\%$), reaching around 75% by the ninth PC. This indicates that population activity can represent the input in a very low-dimensional sub-space through narrow, stimulus-specific trajectories. In random networks, however, this trend is not reflected in the subsequent SSNs, where the first ten PCs account for less than 10% of the total variance.

There is thus a significant increase in the effective dimensionality (see Section 5.2) in the deeper sub-networks (Figure 5.4D), a pattern which is also exhibited, to a lesser extent, in the topographic case. As the population activity becomes less entrained by the input, the deeper SSNs explore a larger region of the state-space. Whereas this tendency is consistent and more gradual for topographic networks, it is considerably faster in networks with unstructured projections, suggesting a quicker dispersion of the stimulus representations. Since in these networks, the stimulus does not effectively reach the last SSN (Figure 5.2A), there is no de-correlation of the responses, and the elevated synchrony (Figure 5.3B) leads to a reduced effective dimensionality.

Overall, these results demonstrate that patterned stimuli push the population activity towards an asynchronous-irregular regime across the network, but purely random systems cannot sustain this state in the deeper SNNs. Networks with structured connectivity, on the other hand, display a more stable activity profile throughout the system, allowing the stimuli to propagate more efficiently and more accurately to all sub-networks. Accordingly, the state representations are more compact and distinguishable, and these decay significantly slower with depth than in random networks, in line with the observed classification results (Figure 5.2A).

5.3.2 Memory capacity and stimulus sensitivity

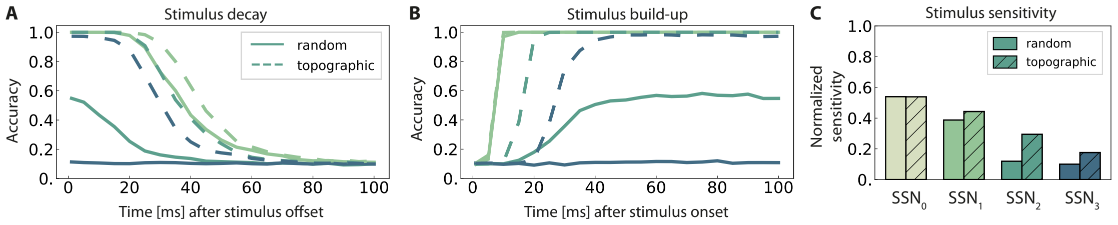


Figure 5.5: **Stimulus sensitivity and temporal evolution of state representations, as indicated by the classification accuracy for 10 stimuli.** **(A)** and **(B)** show the time course of the readout accuracy for the preceding and the current stimulus respectively, with $t = 0$ representing the offset of the previous and onset of the new stimulus. Curves depict the mean accuracy score over 5 trials, with linear interpolation of sampling offsets $t_{\text{samp}} = 1, 5, 10, 15, ..., 100$ ms. Solid and dashed curves represent networks with random connectivity and topography respectively, color-coding according to SSNs (key in **(C)**). The first population SSN_0 was omitted from panels **(A)** and **(B)** to avoid cluttering, as they are identical across the network conditions. **(C)**: The stimulus sensitivity is defined as the area below the intersection of corresponding curves from **(A)** and **(B)**, normalized with respect to maximum performance.

As demonstrated above, both random and topographic networks are able to create unique representations of single stimuli in their internal dynamics and transfer these across multiple recurrent sub-networks. To better understand the nature of distributed processing in these systems, it is critical to investigate how they retain information over time and whether representations of multiple, sequentially presented, stimuli can coexist in a superimposed manner, a property exhibited by cortical circuits as demonstrated by *in vivo* recordings (Nikolic et al., 2009).

To quantify these properties, we use the classification accuracy to evaluate how, for consecutive stimuli, the first stimulus decays and the second stimulus builds up (Figure 5.5A, B). For a given network configuration, the degree of overlap between the two curves indicates how long the system is able to retain useful information about both the previous and the present stimuli (Figure 5.5C). This analysis allows us to measure three important properties of the system: how long stimulus information is retained in each sub-network through reverberations of the current state; how long the network requires to accumulate sufficient evidence to classify the present input; and what are the potential interference effects between multiple stimuli. Note that the procedure used in the following experiments is virtually identical to that in Section 3.1, the only difference being the time at which the network's responses are sampled. In Figure 5.5A, the readout is trained to classify the stimulus identity at increasing time lags after its

offset, whereas in panel B, the classification accuracy is evaluated at various time points after the stimulus onset.

The decay in performance measured at increasing delays after stimulus offset (Figure 5.5A) shows how input representations gradually disappear over time (the *fading memory* property, see Maass et al. 2004). For computational reasons, only the first 100 ms are plotted, but the decreasing trend in the accuracy continues and invariably reaches chance level within the first 150 ms. This demonstrates that the networks have a rather short memory capacity which is unable to span multiple input elements, and that the ability to memorize stimulus information decays with network depth. Adding to the functional benefits of topographic maps, the memory curves reflect the higher overall accuracy achieved by these networks.

We further observe that the networks require exposure time to acquire discernible stimulus representations (Figure 5.5B). The time for classification accuracy to reach its maximum increases with depth, resulting in an unsurprising cumulative delay. Notably, topography enables a faster information build-up beginning with SSN_2.

To determine the *stimulus sensitivity* of a population, we consider the extent of time where useful non-interfering representations are retained in each sub-network. This can be calculated as the area below the intersection of its memory and build-up curves. Following a similar trend to performance and memory, sensitivity to stimulus decreases with network depth and the existence of structured propagation pathways leads to clear benefits, particularly pronounced in the deeper SNNs (Figure 5.5C).

Overall, SNNs located deeper in the network forget faster and take longer (than the inter-SNN delays) to build up stimulus representations. No population can represent two sequential stimuli accurately for a significant amount of time (longer than 100 ms), although topographic maps improve memory capacity and stimulus sensitivity.

5.3.3 Integrating multiple input streams

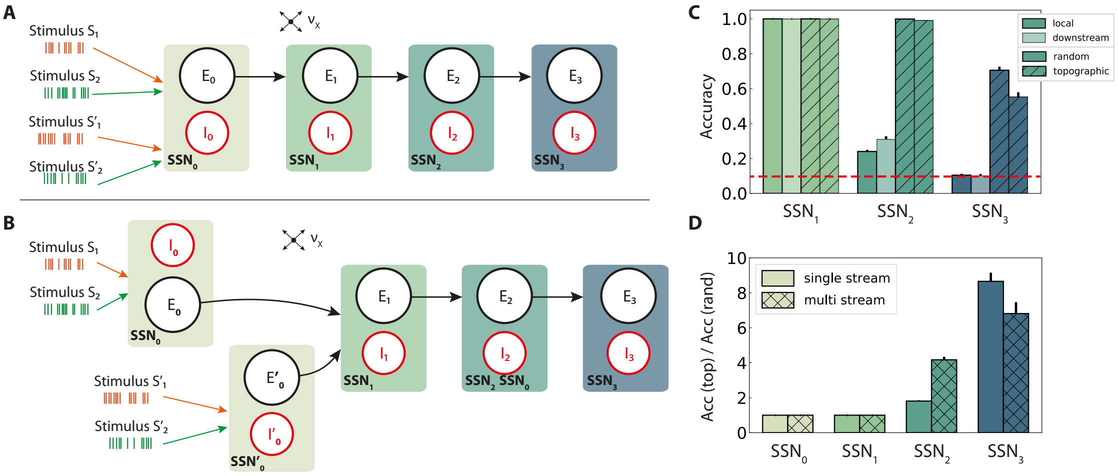


Figure 5.6: **Schematic overview of information integration from two input streams (S and S'), and their performance on the stimulus classification task. A, local integration:** sequential set-up composed of four sub-networks, with two input streams injected into the first population, SSN$_0$, where they are combined and transferred downstream. **B, downstream integration:** as in **A**, but with SSN$_0$ divided into two separate sub-populations SSN$_0$ and SSN$'_0$, each receiving one stream as input and projecting to SSN$_1$. Integration occurs in SSN$_1$. Connection probabilities, weights and other parameters are identical to those in previous scenarios (see Figure 5.2A, B), with the exception of downstream integration (B): to keep the overall excitatory input to SSN$_1$ consistent with local integration, projection densities to SSN$_1$ from the input sub-networks SSN$_0$ and SSN$'_0$ are scaled to $p_{\mathrm{ff}}/2$, while the remaining connections are left unchanged. **(C)**: classification accuracy of ten stimuli from one input stream, in SSN$_1$ − SSN$_3$. **(D)**: Relative performance gain in topographic networks, measured as the ratio of accuracy scores in the single and multiple stream (local integration) scenarios. Results are averaged over ten trials, with dark and light colors coding for local and downstream integration, respectively. The red dashed line represents chance level.

The previous section focuses on a single input stream, injected into a network with sequentially connected sub-networks. Here, we examine the microcircuit's capability to integrate information from two different input streams, in two different scenarios with respect to the location of the integration. The set-up and results are illustrated in Figure 5.6.

In a first step, the set-up from Figure 5.1A is extended with an additional input stream S', without further alterations at population or connectivity level. The two stimulus sets, S and S', are in principle identical, each containing the same number of unique stimuli and connected to specific sub-populations in the networks. Since the inputs are combined locally in the first SSN and the mixed information transferred downstream, we refer to this setup, visualized in Figure 5.6A, as *local integration.* In a second scenario (Figure 5.6B), each input stream is injected into a separate sub-network (SSN_0 and SSN'_0), jointly forming the input module of the system. Here, computation on the combined input happens *downstream* from this first module, with the aim of simulating the integration of information that originated from more distant areas and had already been processed by two independent microcircuits.

Adding a second input stream significantly affects the network activity and the stimulus representations therein, which now must produce distinguishable responses for two stimuli concurrently. Compared to the same setup with a single input source (Figure 5.2A), the performance degrades in both random and topographic networks starting with SSN_2 (Figure 5.6C, D). This suggests that the mixture of two stimuli results in less separable responses as the two representations interfere with each other, with structured connectivity again proving to be markedly beneficial. These benefits become clearer in the deeper sub-networks, as demonstrated in Figure 5.6D where the effects of topography can lead to an 8-fold gain in task accuracy in SSN_3.

As the spatiotemporal structure of the stimuli from both sources are essentially identical, it is to be expected that the mixed responses contain the same amount of information about both inputs. This is indeed the case, as reflected by comparable performance results when decoding from the second input stream (Figure S1).

Interestingly, the location of the integration appears to play no major role in random networks. In networks with topographic maps, however, local integration improves the classification accuracy by around 25% in the last sub-network compared to the downstream case. In the next section, we investigate whether this phenomenon is setup- and task-specific, or reflects a more generic computational principle.

Local integration improves nonlinear computation

In addition to the linear classification task discussed above, we analyze the ability of the circuit to extract and combine information from the two concurrent streams in a more complex, nonlinear fashion. For this, we trained the readouts on the commonly used nonlinear XOR task described in Section 5.2.

We observe that the networks' computational capacity is considerably reduced compared to the simpler classification task, most noticeably in the deeper sub-networks (Figure 5.7A). Although information about multiple stimuli from two input streams could be reasonably represented and transferred across the network, as shown in Figure 5.6C, it is substantially more difficult to perform complex transformations on even a small number of stimuli. This is best illustrated in the last population of topographic

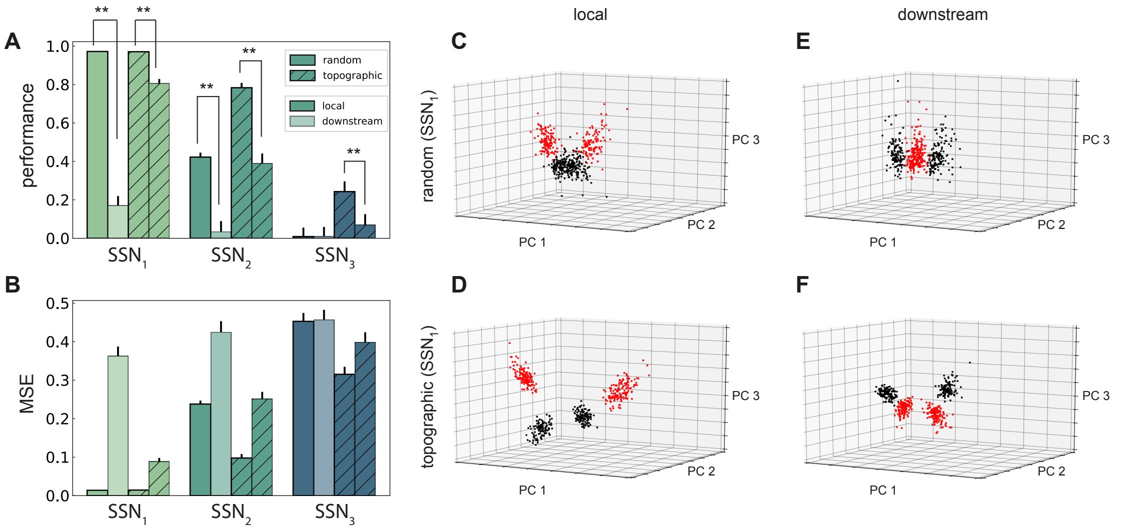


Figure 5.7: **Performance and state-space partitioning in the XOR task.** **(A)**: Task performance on the XOR task measured using the point-biserial correlation coefficient (PBCC) between the XOR on the labels from the two input streams (target) and the raw readout values computed from the membrane potentials. **(B)**: Corresponding mean squared error. Results are averaged over 10 trials, with 2000 training samples and 500 testing samples in each trial. **(C-F)**: PCA projections of 500 data points (low-pass filtered responses) in SSN$_1$, for the four combinations with respect to feed-forward connectivity and location of integration. Same axes in all panels. Colors code the target value, i.e., XOR on the stimulus labels from the two input streams. Random (top) and topographic (bottom) connectivity with local **(C,D)** and downstream **(E,F)** integration.

networks, where the stimulus identity can still be decoded with an accuracy of 70% (Figure 5.6C), but the XOR operation yields performance values close to chance level (PBCC of 0).

In contrast to the identity recognition, for XOR it is clearly more advantageous to fuse the two input streams locally in SSN$_0$, rather than integrating only in SSN$_1$ (Figure 5.7A,B). The differences in performance are statistically significant (two-sided Kolmogorov-Smirnov (KS) test 1.0, p-value < 0.01 for SSN$_1$ and SSN$_2$, and KS-test 0.9 with p-value < 0.01 for SSN$_3$ in topographic networks) and consistent in every scenario and all sub-networks from SSN$_1$ onwards, excepting SSN$_3$ in random networks.

One can gain a more intuitive understanding of the networks' internal dynamics by looking at the state-space partitioning (Figure 5.7C-F), which reveals four discernible clusters corresponding to the four possible label combinations. These low-dimensional projections illustrate two key computational aspects: the narrower spread of the clusters in topographic networks (Figure 5.7D, F) is an indication of their greater representational precision, while the significance of the integration location is reflected in the collapse along the third PC in the downstream scenario (Figure 5.7F). To a lesser extent, these differences are also visible for random networks (Figure 5.7C, E). A more compact representation of the clustering quality using silhouette scores, consistent with these observations, is depicted in Figure S2.

Altogether, these results suggest that it is computationally beneficial to perform nonlinear transformations locally, as close to the input source as possible, and then propagate the result of the computation downstream instead of the other way around. The results were qualitatively similar for both the low-pass filtered spike trains and the membrane potential (see Figure S3). To rule out any possible bias arising from re-scaling the feedforward projections to SSN_1 in the downstream scenario, we also ensured that these results still hold when each of the input sub-populations SSN_0 and SSN_0' projected to SSN_1 with the same unscaled probability p_{ff} as in Figure 5.1B (see Figure S4).

Effective dimensionality depends on the architecture of stimulus integration

Previous studies have suggested that nonlinear integration of multiple input streams is associated with high response dimensionality compared to areas in which little or only linear interactions occur (Rigotti and Fusi, 2016; Barak et al., 2013). To assess if these predictions hold in our model, we consider different stimulus integration schemes and investigate whether the effective response dimensionality correlates with XOR accuracy, which is used to quantify the nonlinear transformations performed by the system.

For simplicity, we focus only on random networks. To allow a better comparison between the integration schemes introduced in Figure 5.6, we explore two approaches to gradually interpolate the downstream scenario towards the local one in an attempt to approximate its properties. First, we distribute each input stream across the two segregated input sub-populations SSN_0 and SSN_0', referred to as *mixed input* (Figure 5.8A). Second, we maintain the input stream separation but progressively merge the two sub-populations into a single larger one by redistributing the recurrent connections (Figure 5.8B). We call this scenario *mixed connectivity*.

Relating these two scenarios is the *mixing factor* (m), which controls the input mapping or the connectivity between the sub-populations, respectively. A factor of 0 represents separated input sources and disconnected sub-populations as in Figure 5.6B; a value of $m = 1$ indicates that the input sub-populations mix contributions from both sources equally (for mixed input), or that the connectivity between and within SSN_0 and SSN_0' are identical (for mixed connectivity). In both cases, care was taken to keep the overall input to the network unchanged, as well as the average in- and out-degree of the neurons.

Combining information from both input streams already in the first sub-populations ($m > 0$), either via mixed input or mixed connectivity, significantly increases the task performance after convergence in the deeper sub-networks.

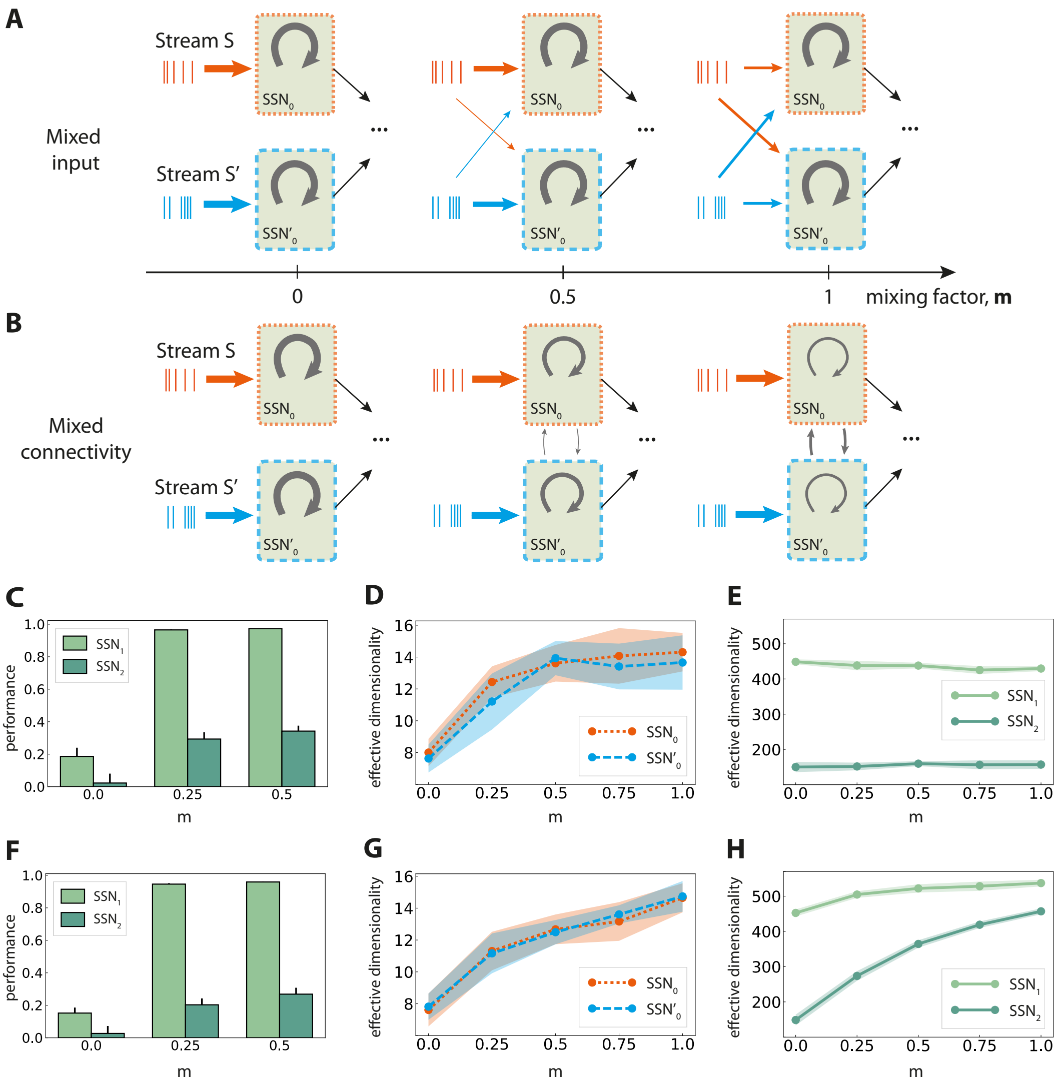


Figure 5.8: **Mixed multi-stream integration and neural dimensionality in random networks.** **(A)**: Downstream integration (only SSN$_0$ shown) with progressive redistribution of the input across SSN$_0$ and SSN$'_0$. The mixing ratio is parameterized by $m \in [0, 1]$, with $m = 0$ representing two information sources mapped exclusively onto the corresponding sub-population, and $m = 1$ indicating equally distributed inputs across them. The overall input to the network is kept constant. Arrow thickness indicates connection density. **(B)**: As in **(A)**, but gradually connecting SSN$_0$ and SSN$'_0$ while keeping the input streams separated. Here, m controls the ratio of within- and between sub-population connection probabilities, with the total number of connections kept constant. **(C)**: XOR performance as a function of m for mixed input; **(D)**: Corresponding effective dimensionality in the SSN$_0$ and SSN$'_0$; **(E)**: Corresponding effective dimensionality in the deeper sub-networks SSN$_1$ and SSN$_2$. **(F-H)**: as in **(C-E)**, but for mixed connectivity. Effective dimensionality is calculated for the membrane potential (mean and standard deviation over 10 trials, calculated using the first 500 PCs).

This is illustrated in Figure 5.8C, F, with $m > 0.5$ yielding similar values. Despite comparable gains in the nonlinear computational performance, the underlying mechanisms appear to differ in the two mixing approaches, as detailed in the following.

In SSN_0 and SSN'_0, the effective dimensionality of the neural responses increases monotonically with the amount of information shared between the two sub-populations (Figure 5.8D, G). This is expected, since the sub-populations are completely independent initially ($m = 0$) and can therefore use more compact state representations for single stimuli. However, diverging patterns emerge after convergence in SSN_1. While the dimensionality does increase with the coefficient m in the mixed connectivity scenario (Figure 5.8H), it remains fairly constant in the mixed input case (Figure 5.8E), despite comparable task performance. Thus, complex nonlinear transformations do not necessarily involve the exploration of larger regions of state-space, but can also be achieved through more efficient representations.

These results also demonstrate the difficulty in defining a clear relation between the ability of the system to perform nonlinear transformations on the input and its response dimensionality. Particularly in the case of larger networks involving transmission across multiple modules, the effective dimensionality can depend on the system's architecture, such as the input mapping and connectivity structure in the initial stages.

5.4 Discussion

This chapter examined the temporal dynamics of the information transferred between sequentially connected modules and explored how different network characteristics enable information integration from two independent sources in a computationally useful manner. Structural differences in the network were proven to greatly influence the dynamics and the downstream computation when combining inputs from two independent sources. In addition to the feedforward connectivity, the ability of the downstream subnetworks to nonlinearly combine the inputs was shown to depend on the location where the input converges, as well as on the extent to which the different input streams are mixed in the initial sub-networks. We therefore anticipate that the degree of mixed selectivity in early sensory stages is predictive of the computational outcome in deeper levels, particularly for nonlinear processing tasks, as we describe in greater detail below.

5.4.1 Representation transfer in sequential hierarchies

The proficiency of randomly coupled spiking networks (see e.g., Maass et al. 2002; Sussillo 2014; Duarte and Morrison 2014) demonstrates that random connectivity can be sufficient for local information processing. Successful signal propagation over multiple modules, however, appears to require some form of structured pathways for accurate and reliable transmission. Our results suggest that these requirements can be achieved by embedding simple topographic projections in the connectivity between the modules.

Such mechanisms might be employed across the brain for fast and robust communication, particularly (but not exclusively) in the early sensory systems, where real-time computation is crucial and where the existence of topographic maps is well supported by anatomical studies (Kaas, 1997; Bednar and Wilson, 2016).

Purely random feedforward connectivity allowed stimulus information to be decoded only up to the third module, whereas incorporating topographic projections ensured almost perfect accuracy in all modules (Figure 5.2A, B). These differences could be attributed to a decrease in the specificity of stimulus tuning with network depth, which is much more prominent for random networks (Figure 5.4). This result suggests that accurate information transmission over longer distances is not possible without topographic precision, thus uncovering an important functional role of this common anatomical feature.

Moreover, topography was shown to counteract the shared-input effect which leads to the development of synchronous regimes in the deeper modules. By doing so, stimulus information is allowed to propagate not only more robustly, but also more resource efficiently, in that the average spike emission is much lower (Figure 5.3D, E). Nevertheless, as the stimulus intensity invariably fades with network depth, the deeper modules capture fewer spatiotemporal features of the input and their response dimensionality increases. This process is clearer in random networks (Figure 5.4C, D), a further indication that topography enforces more stereotypical, lower-dimensional and stimulus-specific response trajectories. The input-state mappings are also retained longer and built up more rapidly in topographic networks (Figure 5.5).

5.4.2 Network architecture and input integration

In biological microcircuits, local connections are complemented by long-range projections which either stem from other cortical regions (cortico-cortical), or different sub-cortical nuclei (e.g., thalamocortical). These different projections carry different information content and thus require the processing circuits to integrate multiple input streams during online processing. The ability of local modules to process information from multiple sources simultaneously and effectively is thus a fundamental building block of cortical processing.

Including a second input source into our sequential networks leads to less discriminable responses, as reflected in a decreased classification performance (Figure 5.6C). Integrating information from the two sources as early as possible in the system (i.e., in the modules closest to the input) was found to be clearly more advantageous for nonlinear computations (Figure 5.6C) and, to a lesser extent, also to linear computations. For both tasks, however, topographic networks achieved better overall performance.

5.4.3 Degree of mixed selectivity predicts computational performance

We have further shown that the effective dimensionality of the neural responses does not correlate with the nonlinear computational capabilities, except in the very first modules (Figure 5.8). These insights are in agreement with previous studies based on fMRI data (Rigotti and Fusi, 2016; Barak et al., 2013), which predicted a high response dimensionality in areas involved in nonlinear multi-stream integration, and lower in areas where inputs from independent sources do not interact at all or solely overlap linearly. These studies considered single circuits driven by input from two independent sources, focusing on the role of mixed selectivity neurons in the convergent population. Mixed selectivity refers to neurons being tuned to mixtures of multiple task-related aspects (Rigotti et al., 2013; Warden and Miller, 2010), which we approximated as a differential driving of the neurons with a variable degree of input from both sources.

Although we did not specifically examine mixed selectivity at a single neuron level, one can consider both the mixed input and mixed connectivity scenarios (Figure 5.8A and B, respectively) to approximate this behavior at a population level. This is particularly the case for the input sub-modules SSN_0 and SSN'_0, where the network's response dimensionality, as expected, increases with the mixing ratio (Figure 5.8D, G). However, the different results we obtained for the deeper modules (Figure 5.8E, H), suggest that the effective dimensionality measured at the neuronal level is not reliable evidence for nonlinear processing in downstream convergence areas (despite similar performance), but instead depends on how information is mixed in the early stages of the system. Further research in this direction, possibly resorting to multimodal imaging data, is needed to determine a clear relation between functional performance, integration schemes and response dimensionality.

In our models, the task performance improved (and plateaued) with increased mixing factors, suggesting no obvious computational disadvantages for large factor values. While this holds for the discrimination capability of the networks, we did not address their ability to generalize. Since the sparsity of mixed selectivity neurons has been previously shown to control the discrimination-generalization trade-off, along with the existence of an optimal sparsity for neural representations (Barak et al., 2013), it would be interesting to analyze the effect of this parameter more thoroughly in the context of hierarchical processing.

Despite the limitations of our models, we have highlighted the importance of biologically plausible structural patterning for information processing in modular spiking networks. Even simple forms of topography were shown to significantly enhance computational performance in the deeper modules. Additionally, architectural constraints have a considerable impact on the effectiveness with which different inputs are integrated, with early mixing being clearly advantageous and highlighting a possibly relevant feature of hierarchical processing. Taken together, these results provide useful constraints for building modular systems composed of spiking balanced networks that enable accurate information transmission.

5.4.4 Limitations and future work

Our analysis consisted of a relatively simple implementation both in terms of the microcircuit composition and the characteristics of topographic maps. Even though abstractions are required in any modeling study, it is important to highlight the inherent limitations and drawbacks.

The network we referred to as *random* in this study (Figure 5.1A) was considered to be the most appropriate to serve as a baseline for the unstructured architecture, due to its simplicity. However, there are many other classes of non-modular networks, such as small-world or scale-free networks, which are likely to display similar or even superior computational characteristics than our baseline. Investigating the behavior and impact of such alternative network structures could be an interesting topic for future research, as they constitute intermediate steps between fully random and modular architectures.

We have employed a simple process to embed topographic maps in unstructured networks (see Section 5.2), whereby the map size (i.e., size of a population involved in a specific pathway) was kept constant in all modules. Cortical maps, however, exhibit more structured and complex spatial organization (Bednar and Wilson, 2016), characterized by a decrease in topographic specificity with hierarchical depth. This, in turn, is likely a consequence of increasingly overlapping projections and increasing map sizes and is considered to have significant functional implications (see e.g., Rigotti et al. 2013), which we did not explore in more detail here. Nevertheless, our results (Figure 5.2F) suggest that, at least for the relatively simple and low-dimensional (considering the network size) tasks employed in this study, overlapping projections have a detrimental effect on the network's discrimination ability. More complex tasks involving high-dimensional mappings would therefore negatively impact the performance of our modular networks. Assuming a one-to-one mapping between input dimensions and stimulus-specific neuronal clusters, a larger task dimensionality would require either fewer neurons per cluster or some compensation mechanism (e.g., stronger or denser projections between the clusters), possibly limiting the task complexity that smaller local circuits can handle. Alternatively, cortical circuits might solve this dimensionality problem by combining multiple modules dynamically, in a task-dependent manner (Yang et al., 2019). We discuss additional neurobiological limitations of our models in Chapter 9.

Chapter 6

Denoising through topographic modularity

6.1 Introduction

In the previous chapter, we established that structured projections can create feature-specific pathways that allow the external inputs to be faithfully represented and propagated through multiple processing modules. Topographic maps were embedded in unstructured networks via a simplified process, by randomly choosing subsets of neurons in the different sub-networks and connecting them with a fixed probability. While this method proved sufficient to determine the functional improvements of structured projections over random networks, it does not allow controlling for the specificity of these maps. In particular, it is difficult to disentangle which connectivity properties (e.g., precision of projections, map size) are critical for improved performance, and the underlying mechanism remains unclear. Moreover, we considered an idealized scenario where the input signal was not corrupted by noise and evaluated the networks' ability to represent and classify the stimulus identity at a single point in time (stimulus offset). These simplifications were useful from a modeling perspective, but they ignored two key aspects of naturalistic stimuli and sensory perception: they are time-continuous and typically noisy (see Section 1.4).

In this chapter, we take into account these aspects and pursue a more systematic approach to investigate the role of topographic projections in processing noisy, dynamic input signals. We manipulate key structural parameters such as modularity, map size and degree of overlap, and evaluate their impact on the network dynamics and computational performance during a continuous signal reconstruction task from noisy inputs. We demonstrate that, by modulating effective connectivity and regional E/I balance, topographic projections additionally serve a *denoising* function, not merely allowing the faithful propagation of input signals, but systematically improving the system's internal representations and increasing signal-to-noise ratio. We identify a critical threshold in the degree of modularity in topographic projections, beyond which the system behaves effectively as a denoising autoencoder[1]. Additionally, we demonstrate that this phe-

[1] Note that the parallel is established here on conceptual, not formal, grounds as the system is capable of retrieving the original, uncorrupted input from a noisy source, but bears no formal similarity to denoising autoencoder algorithms.

nomenon is robust, with the qualitative behavior persisting across very different models. Theoretical considerations and network simulations show that it hinges solely on the modularity of topographic projections and the presence of recurrent inhibition, with the external input and single-neuron properties influencing where/when, but not if, denoising occurs.

Our results suggest that modular structure in feedforward projection pathways can have a significant effect on the system's qualitative behavior, enabling a wide range of behaviorally relevant and empirically supported dynamic regimes. This allows the system to: (i) maintain stable representations of multiple stimulus features (Andersen et al., 2008); (ii) amplify features of interest while suppressing others through winner-takes-all mechanisms (Douglas and Martin, 2004; Carandini and Heeger, 2012); and (iii) dynamically represent different stimulus features as stable and metastable states and stochastically switch among active representations through a winnerless competition effect (McCormick, 2005; Rabinovich et al., 2008; Rost et al., 2018).

6.2 Numerical simulations and theoretical analysis

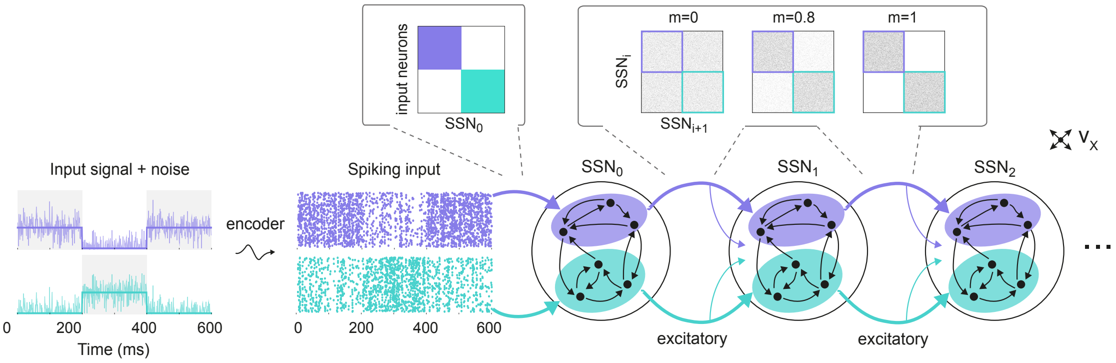


Figure 6.1: **Sequential denoising spiking architecture.** A continuous step signal is used to drive the network. The input is spatially encoded in the first sub-network (SSN$_0$), whereby each input channel is mapped exclusively onto a sub-population of stimulus-specific excitatory and inhibitory neurons (schematically illustrated by the colors; see also inset, top left). This exclusive encoding is retained to variable degrees across the network, through topographically structured feedforward projections (inset, top right) controlled by the modularity parameter m (see Section 6.3). This is illustrated explicitly for both topographic maps (purple and cyan arrows). Projections between SSNs are purely excitatory and target both excitatory and inhibitory neurons.

To investigate the role of structured pathways between processing modules in modulating the fidelity of stimulus representations, we study a network similar to the one described in Chapter 5, Figure 5.1, comprising up to six (not four) sequentially connected sub-networks (SSNs, see Section 6.3 and Figure 6.1). As in the previous chapter, each SSN is a balanced random network composed of leaky integrate-and-fire neurons, but here we simplify to current-based synapses for analytical tractability. In each SSN, neurons are assigned to sub-populations associated with a particular stimulus, which we denote, along with the structured feedforward projections among them, as a *topographic map*. The main difference to model in Chapter 5 consists in the creation of these maps: whereas the stimulus-specific sub-populations were chosen randomly and feedforward projections were restricted within similarly tuned neurons, here these sub-populations are chosen systematically and project to the subsequent SSN with a varying degree of specificity. The specificity of the map is determined by the degree of *modularity* of the corresponding projections matrices (see, e.g. Figure 6.1). Note that in this chapter m denotes modularity, which is independent of the mixing factor represented by the same variable in Section 5.3.3. Modularity is thus defined as the relative density of connections within a stimulus-specific pathway (i.e., connecting sub-populations associated with the *same* stimulus; see Section 6.3). In the following, we study the role of topographic specificity in modulating the system's functional and representational dynamics and its ability to cope with noise-corrupted input signals.

6.2.1 Sequential denoising through structured projections

By systematically varying the degree of modular specialization in the feedforward projections (modularity parameter, m, see Section 6.3 and Figure 6.1), we can control the segregation of stimulus-specific pathways across the network and investigate how it influences the characteristics of neural representations as the signal propagates. If the feedforward projections are unstructured or moderately structured ($m \lesssim 0.8$), information about the input fails to permeate the network, resulting in a chance-level reconstruction accuracy in the last sub-network, SSN_5, even in the absence of noise (see Figure 6.2A-C). However, as m approaches a switching value $m_{\text{switch}} \approx 0.83$, there is a qualitative transition in the system's behavior, leading to a consistently higher reconstruction accuracy across the sub-networks (Figure 6.2B,C), regardless of the amount of noise added to the signal (Figure 6.2E,F).

Beyond this transition point, reconstruction accuracy improves with depth, i.e. the signal is more accurately represented in SSN_5 than in the initial sub-network, SSN_0, with an effective accuracy gain of over 40% (Figure 6.2C, F). While the addition of noise does impair the absolute reconstruction accuracy in all cases (see Supplementary Figure B.1), the denoising effect persists even if the input is severely corrupted ($\sigma_\xi = 3$, see Figure 6.2E, F). This is a counter-intuitive result, suggesting that topographic modularity is not only necessary for reliable communication across multiple populations (see Chapter 5 and Zajzon et al., 2019), but also supports an effective denoising effect, whereby representational precision increases with depth, even if the signal is profoundly distorted by noise.

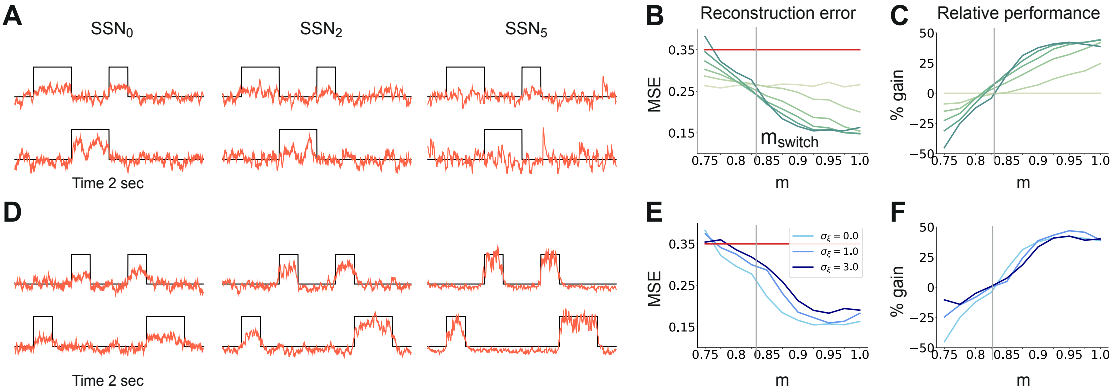


Figure 6.2: **Reconstruction of a continuous step signal.** **(A)**: Signal reconstruction across the network. Single-trial illustration of target signal (black step function) and readout output (red curves) in 3 different SSNs, for $m = 0.75$ and no added noise ($\sigma_\xi = 0$). For simplicity, only two out of ten input channels are shown. **(B)**: Signal reconstruction error in the different SSNs for the no-noise scenario shown in **(A)**. Color shade denotes network depth, from SSN$_0$ (lightest) to SSN$_5$ (darkest). The horizontal red line represents chance level, while the grey vertical line marks the transition (switching) point $m_{\text{switch}} \approx 0.83$ (see main text). Supplementary Figure B.1 shows the task performance for a broader range of parameters. **(C)**: Performance gain across the network, relative to SSN$_0$, for the setup illustrated in **(A)**. **(D)**: As in **(A)** but for $m = 0.9$. **(E)**: Reconstruction error in SSN$_5$ for the different noise intensities. Horizontal and vertical dashed lines as in **(B)**. **(F)**: Performance gain in SSN$_5$, relative to SSN$_0$.

6.2.2 Noise suppression and response amplification

The sequential denoising effect observed beyond the transition point $m_{\text{switch}} \approx 0.83$ results in an increasingly accurate input encoding through progressively more precise internal representations. In general, such a phenomenon could be achieved either through noise suppression, stimulus-specific response amplification or both. In this section, we examine these possibilities by analyzing and comparing the input-driven dynamics of the different sub-networks. The strict segregation of stimulus-specific sub-populations in SSN$_0$ is only fully preserved across the system if $m = 1$, in which case signal encoding and transmission primarily rely on this spatial segregation. Spiking activity across the different SSNs (Figure 6.3A) demonstrates that the system gradually sharpens the segregation of stimulus-specific sub-populations; indeed, in systems with fully modular feedforward projections, activity in the last sub-network is concentrated predominantly in the stimulated sub-populations. This effect can be observed in both excitatory (E) and inhibitory (I) populations, as both are equally targeted by the feedforward excitatory projections. The sharpening effect consists of both *noise suppression* and *response amplification* (Figure 6.3B), measured as the relative firing rates of the non-stimulated $\nu_5^{\text{NS}}/\nu_0^{\text{NS}}$ and stimulated sub-populations $\nu_5^{\text{S}}/\nu_0^{\text{S}}$, respectively. For $m < m_{\text{switch}}$, noise

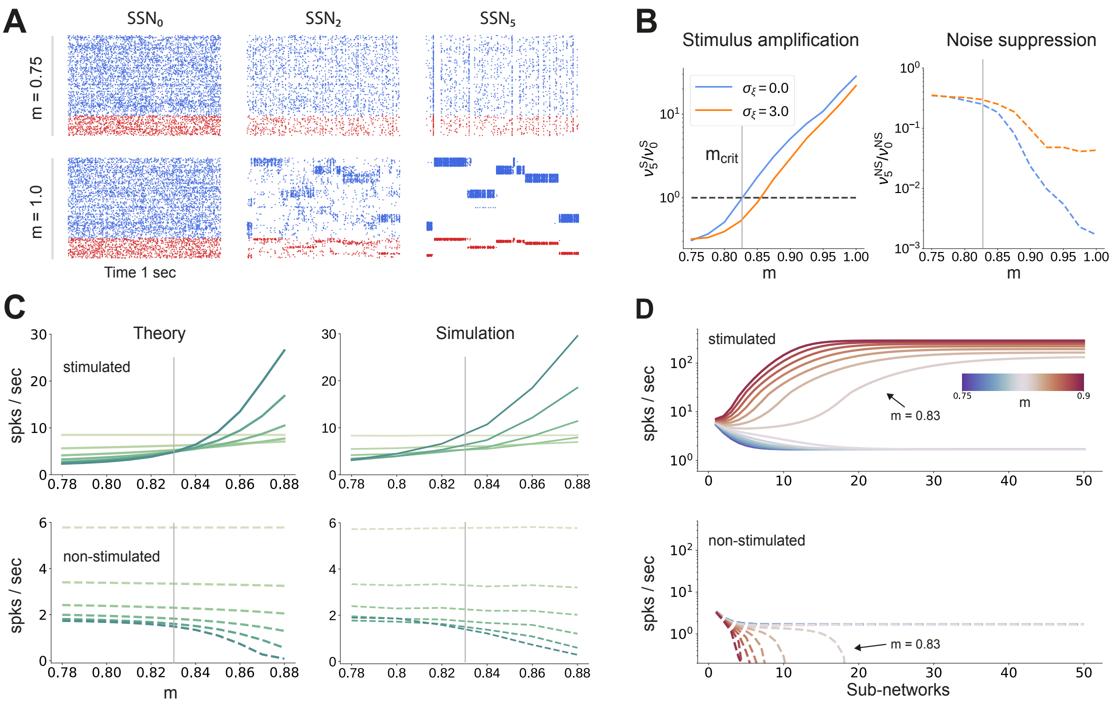


Figure 6.3: **Activity modulation and representational precision. (A)**: 1 second of spiking activity observed across 1000 randomly chosen excitatory (blue) and inhibitory (red) neurons in SSN$_0$, SSN$_2$ and SSN$_5$, for $\sigma_\xi = 3$ and $m = 0.75$ (top) and $m = 1$ (bottom). **(B)**: Mean quotient of firing rates in SSN$_5$ and SSN$_0$ (ν_5/ν_0) for stimulated (S, left) and non-stimulated (NS, right) sub-populations for different input noise levels, describing response amplification and noise suppression, respectively. **(C)**: Mean firing rates of the stimulated (top) and non-stimulated (bottom) excitatory sub-populations in the different SSNs (color shade as in Figure 6.2), for $\sigma_\xi = 0$. For modularity values facilitating an asynchronous irregular regime across the network, the firing rates predicted by mean-field theory (left) closely match the simulation data (right). **(D)**: Mean-field predictions for the stationary firing rates of the stimulated (top) and non-stimulated (bottom) sub-populations, in a system with 50 sub-networks and $\sigma_\xi = 0$. Note that all reported simulation data corresponds to the mean firing rates acquired over a period of 10 seconds and averaged across 5 trials per condition. Supplementary Figure B.2 shows the firing rates as a function of the input intensity λ.

suppression is only marginal and responses within the stimulated pathways are not amplified ($\nu_5^S/\nu_0^S < 1$).

Mean-field analysis of the stationary network activity (see Section 6.3 and Appendix B) predicts that the firing rates of the stimulus-specific sub-populations increase systematically with modularity, whereas the untuned neurons are gradually silenced (Figure 6.3C, left). At the transition point $m_{\text{switch}} \approx 0.83$, mean firing rates across the different sub-networks converge, which translates into a globally uniform signal encoding capacity, corresponding to the zero-gain convergence point in Figure 6.2C, F. As the degree of modularity increases beyond this point, the self-consistent state is lost again as the functional dynamics across the network shifts towards a gradual response sharpening, whereby the activity of stimulus-tuned neurons becomes increasingly dominant (Figure 6.3A-C). The effect is more pronounced for the deeper sub-networks. Note that the analytical results match well with those obtained by numerical simulation (Figure 6.3C, right).

In the limit of very deep networks (up to 50 SSNs, Figure 6.3D) the system becomes bistable, with rates converging to either a high-activity state associated with signal amplification or a low-activity state driven by the background input. The transition point is observed at a modularity value of $m = 0.83$, matching the results reported so far. Below this value, elevated activity in the stimulated sub-populations can be maintained across the initial sub-networks (< 10), but eventually dies out; the rate of all neurons decays and information about the input cannot reach the deeper populations. Importantly, for $m = 0.83$, the transition towards the high-activity state is slower. This allows the input signal to faithfully propagate across a large number of sub-networks (≈ 15), without being driven into implausible activity states.

6.2.3 E/I balance and asymmetric effective couplings

The departure from the balanced activity in the initial sub-networks can be better understood by zooming in at the synaptic level and analyzing how topography influences the synaptic input currents. The segregation of feedforward projections into stimulus-specific pathways breaks the symmetry between excitation and inhibition (see Figure 6.4A) that characterizes the balanced state (Haider et al., 2006; Shadlen and Newsome, 1994), for which the first two sub-networks were tuned (see Section 6.3). E/I balance is thus systematically shifted towards excitation in the stimulated populations and inhibition in the non-stimulated ones. Neurons belonging to sub-populations associated with the active stimulus receive significantly more net overall excitation, whereas the other neurons become gradually more inhibited. This disparity grows not only with modularity but also with network depth. Overall, across the whole system, increasing modularity results in an increasingly inhibition-dominated dynamical regime (inset in Figure 6.4A), whereby stronger effective inhibition silences non-stimulated populations, thus sharpening stimulus / feature representations by concentrating activity in the stimulus-driven sub-populations.

To gain an intuitive understanding of these effects from a dynamical systems perspective, we linearize the network dynamics around the stationary working points of the individual populations (Tetzlaff et al., 2012) in order to obtain the effective connectivity W of the system (see Section 6.3 and Appendix B). The effective impact of a single spike from a presynaptic neuron j on the firing rate of a postsynaptic neuron i (the effective weight $w_{ij} \in W$) is determined not only by the synaptic efficacies J_{ij}, but also by the

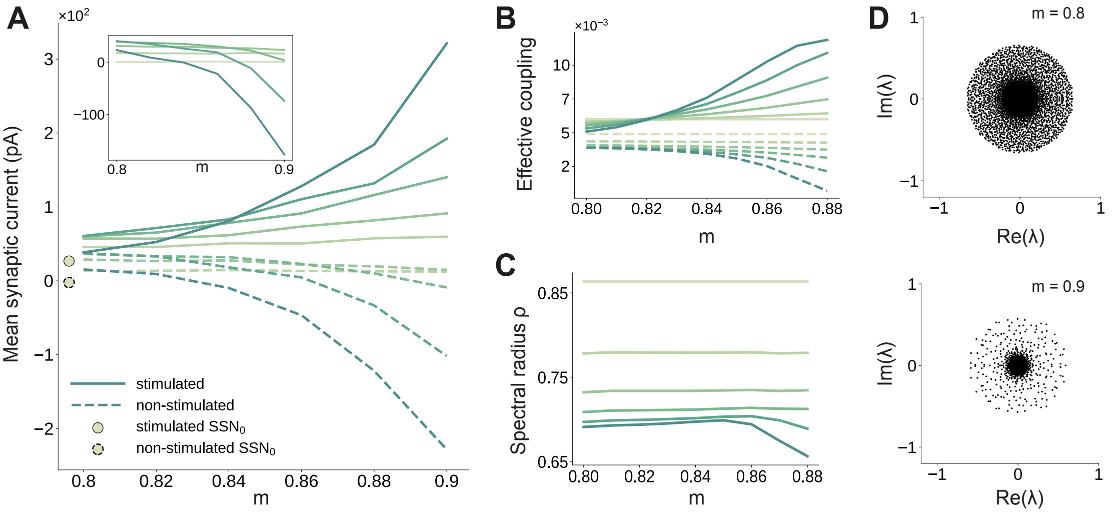


Figure 6.4: **Asymmetric effective couplings modulate the E/I balance and support sequential denoising. (A)**: Mean synaptic input currents for neurons in the stimulated (solid curves) and non-stimulated (dashed curves) excitatory sub-populations in the different SSNs. To avoid clutter, data for SSN_0 is only shown by markers (independent of m). Inset shows the currents (in pA) averaged over all excitatory neurons in the different sub-networks; increasing modularity leads to a dominance of inhibition in the deeper sub-networks. Color shade represents depth, from SSN_1 (light) to SSN_5 (dark). **(B)**: Mean-field approximation of the effective recurrent weights in SSN_5. Curve shade and style as in **(A)**. **(C)**: Spectral radius of the effective connectivity matrices $\rho(W)$ as a function of modularity. **(D)**: Eigenvalue spectra for the effective coupling matrices in SSN_5, for $m = 0.8$ (top) and $m = 0.9$ (bottom). The largest negative eigenvalue (outlier, see Section 6.3), characteristic of inhibition-dominated networks, is omitted for clarity.

statistics of the synaptic input fluctuations to the target cell i that determine its excitability (see Section 6.3, Eq. 6.6). This analysis reveals that there is an increase in the effective synaptic input onto neurons in the stimulated sub-populations as a function of modularity (Figure 6.4B). Conversely, non-stimulated neurons effectively receive weaker excitatory (and stronger inhibitory) drive and become increasingly less responsive (see Figure 6.4A, B). The role of topographic modularity in denoising can thus be understood as a transient, stimulus-specific change in effective connectivity.

For low and moderate topographic precision ($m \lesssim 0.83$), denoising does not occur as the effective weights are sufficiently similar to maintain a stable E/I balance across all populations and sub-networks (Figure 6.4A, B), resulting in a relatively uniform global dynamical state (indicated in Figure 6.4C by a constant spectral radius for $m \lesssim 0.83$, see also Section 6.3) and stable linearized dynamics ($\rho(W) < 1$).

However, as the feedforward projections become more structured, the system undergoes qualitative changes: after a weak transient ($0.83 \lesssim m \lesssim 0.85$) the spectral radius

ρ in the deep SSNs expands due to the increased effective coupling to the stimulated sub-population (Figure 6.4B); the spectral radius eventually ($m \gtrsim 0.85$) contracts with increasing modularity (Figure 6.4C, D). Given that ρ is determined by the variance of W, i.e. heterogeneity across connections (Rajan and Abbott, 2006), this behavior is expected: most weights are in the non-stimulated pathways, which decrease with larger m and network depth (Figure 6.4B). Strong inhibitory currents (Figure 6.4A) suppress the majority of neurons, thereby reducing noise, as demonstrated by the collapse of the bulk of the eigenvalues towards the center for larger m (Figure 6.4D). Indicative of a more constrained state-space, this contractive effect suggests that population activity becomes gradually entrained by the spatially encoded input along the stimulated pathway, whereas the responses of the non-stimulated neurons have a diminishing influence on the overall behavior.

By biasing the effective connectivity of the system, precise topography can thus modulate the balance of excitation and inhibition in the different sub-networks, concentrating the activity along specific pathways. This results in both a systematic amplification of stimulus-specific responses and a systematic suppression of noise (Figure 6.3B). The sharpness/precision of topographic specificity along these pathways thus acts as a critical control parameter that largely determines the qualitative behavior of the system and can dramatically alter its responsiveness to external inputs.

6.2.4 Modulating inhibition

How can the system generate and maintain the elevated inhibition underlying such a noise-suppressing regime? On the one hand, feedforward excitatory input may increase the activity of certain excitatory neurons in $\mathrm{E_i}$ of sub-network $\mathrm{SSN_i}$, which, in turn, can lead to increased mean inhibition through local recurrent connections. On the other hand, denoising could depend strongly on the concerted topographic projections onto $\mathrm{I_i}$. Such structured feedforward inhibition is known to play important functional roles in, e.g., sharpening the spatial contrast of somatosensory stimuli (Mountcastle and Powell, 1959) or enhancing coding precision throughout the ascending auditory pathways (Roberts et al., 2013).

To investigate whether recurrent activity alone can generate sufficiently strong inhibition for signal transmission and denoising, we maintained the modular structure between the excitatory populations and randomized the feedforward projections onto the inhibitory ones ($m = 0$ for $\mathrm{E_i} \rightarrow \mathrm{I_{i+1}}$, compare top panels of Figure 6.5A and B). This leads to unstable firing patterns in the downstream sub-networks, characterized by significant accumulation of synchrony and increased firing rates (see bottom panels of Figure 6.5A and B and Supplementary Figure B.3a, b). These effects, known to result from shared pre-synaptic excitatory inputs (see, e.g. (Shadlen and Newsome, 1998; Tetzlaff et al., 2003; Kumar et al., 2008a)), are more pronounced for larger m and network depth (see Supplementary Figure B.3). Compared with the baseline network, whose

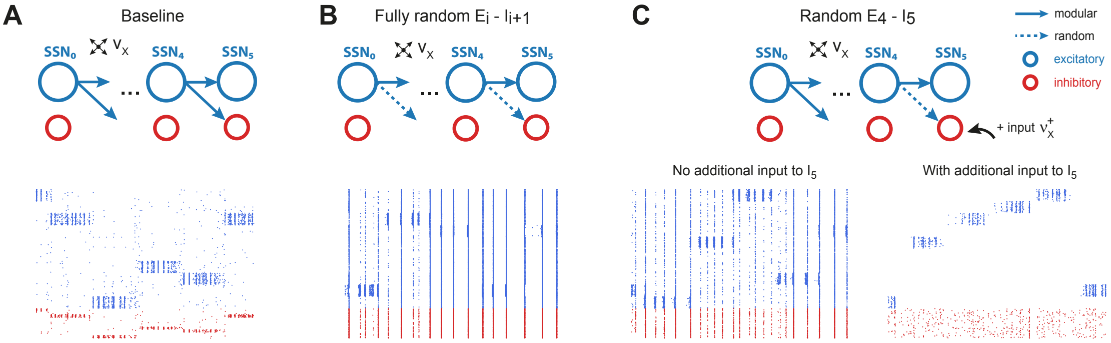


Figure 6.5: **Modular projections to inhibitory populations stabilize network dynamics.** Raster plots show 1 second of spiking activity of 1000 randomly chosen neurons in SSN$_5$, for different network configurations. **(A)**: Baseline network with $m = 0.88$. **(B)**: Unstructured feedforward projections to the inhibitory sub-populations lead to highly synchronized network activity, hindering signal representation. **(C)**: Same as the baseline network in **(A)**, but with random projections for E$_4$ → I$_5$ and additional but unspecific (Poissonian) excitatory input to I$_5$ controlled via ν_X^+. Without such input ($\nu_X^+ = 0$, left), the activity is strongly synchronous, but this is compensated for by the additional excitation, reducing synchrony and restoring the denoising property ($\nu_X^+ = 10$ spks/sec, right). Supplementary Figure B.3 depicts the activity statistics in the last two modules, for the different scenarios.

activity shows clear spatially encoded stimuli (sequential activation of stimulus-specific sub-populations (Figure 6.5A, bottom)), removing structure from the projections onto inhibitory neurons abolishes the effect and prevents accurate signal transmission.

These effects of unstructured inhibitory projections are so marked that they can be observed even if a single set of projections is modified: this can be seen in Figure 6.5C, where only the E$_4$ → I$_5$ connections are randomized. It is worth noting, however, that the excessive synchronization that results from unstructured inhibitory projections (Figure 6.5C bottom left, no additional input condition) can be easily counteracted by driving I$_5$ (the inhibitory population that receives only unstructured projections) with additional uncorrelated external input. If strong enough ($\nu_X^+ \approx 10$ spks/sec), this additional external drive pushes the inhibitory population into an asynchronous regime that restores the sharp, stimulus-specific responses in the excitatory population of the corresponding sub-network (see Figure 6.5C bottom right, and Supplementary Figure B.3c).

These results emphasize the control of inhibitory neurons' responsiveness as the main causal mechanism behind the effects reported. Elevated local inhibition is strictly required, but whether this is achieved by tailored, stimulus-specific activation of inhibitory sub-populations, or by uncorrelated excitatory drive onto all inhibitory neurons appears to be irrelevant and both conditions result in sharp, stimulus-tuned responses in the excitatory populations.

6.2.5 A generalizable structural effect

We have demonstrated that, by controlling the different sub-networks' operating points, the sharpness of feedforward projections allows the architecture to systematically improve the quality of internal representations and retrieve the input structure, even if profoundly corrupted by noise. In this section, we investigate the robustness of the phenomenon in order to determine whether it can be entirely ascribed to the topographic projections (a structural/architectural feature) or if the particular choices of models and model parameters for neuronal and synaptic dynamics contribute to the effect.

To do so, we study two alternative model systems on the signal denoising task. These are structured similarly to the baseline system explored so far, comprising separate sequential sub-networks with modular feedforward projections among them (see Figure 6.1 and Section 6.3), but vary in total size, neuronal and synaptic dynamics. In the first test case, only the models of synaptic transmission and corresponding parameters are altered. To increase biological verisimilitude and following (Zajzon et al., 2019), synaptic transmission is modeled as a conductance-based process, with different kinetics for excitatory and inhibitory transmission, corresponding to the responses of AMPA and GABA_a receptors, respectively. This is the model used in Chapter 5, see corresponding Section 5.2 and supplementary materials for details on the parameters. The results, illustrated in Figure 6.6A, demonstrate that task performance and population activity across the network follow a similar trend to the baseline model (Figure 6.2 and Figure 6.3A, B). Despite severe noise corruption, the system is able to generate a clear, discernible representation of the input as early as SSN_2 and can accurately reconstruct the signal. Importantly, the relative improvement with increasing modularity and network depth is retained. In comparison to the baseline model, the transition occurs for a slightly different topographic configuration, $m \approx 0.85$, at which point the network dynamics converge towards a low-rate, stable asynchronous irregular regime across all populations, facilitating a linear firing rate propagation along the topographic maps (Supplementary Figure B.4).

The second test case is a smaller and simpler network of nonlinear rate neuron models (see Figure 6.6B and Section 6.3) which interact via continuous signals (rates) rather than discontinuities (spikes). Despite these profound differences in the neuronal and synaptic dynamics, the same behavior is observed, demonstrating that sequential denoising is a structural effect, dependent on the population firing rates and thus less sensitive to fluctuations in the precise spike times. Moreover, the robustness with respect to the network size suggests that denoising could also be performed in smaller, localized circuits, possibly operating in parallel on different features of the input stimuli.

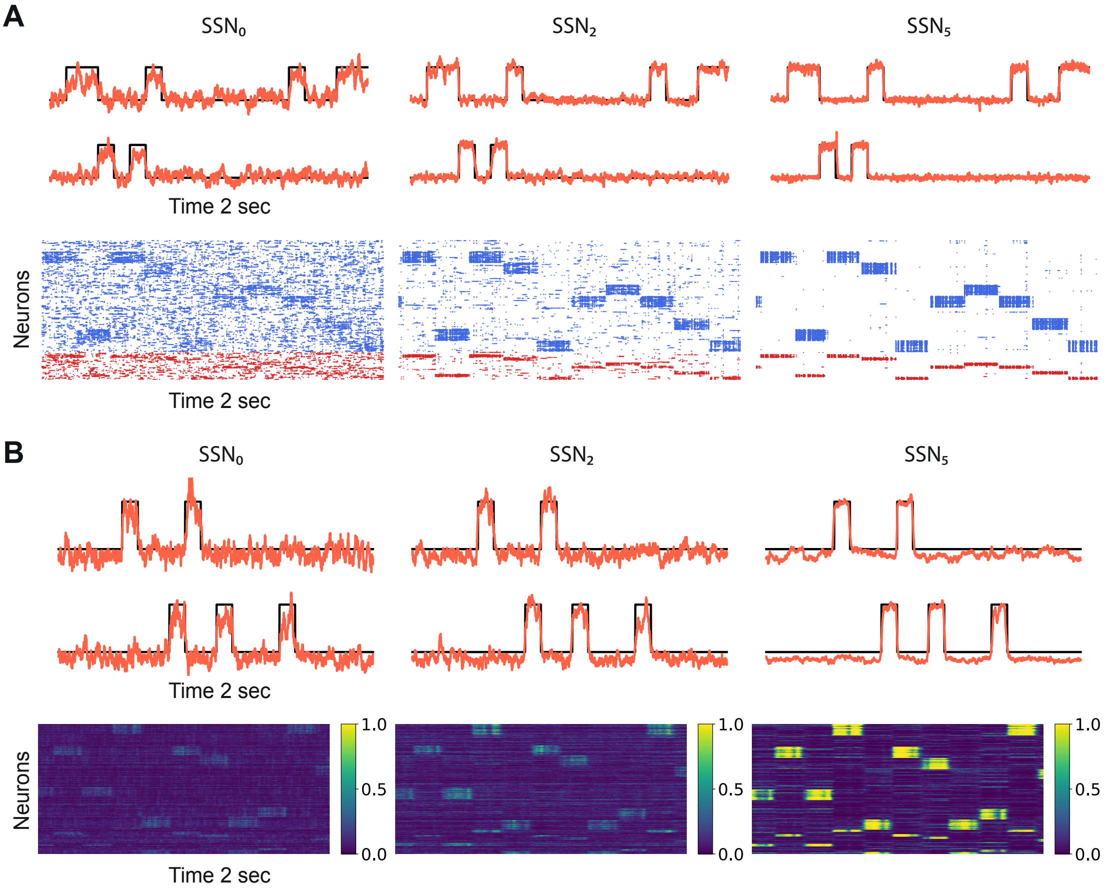


Figure 6.6: **Denoising through modular topography is a robust structural effect. (A)**: Signal reconstruction (top) and corresponding network activity (bottom) for a network with LIF neurons and conductance-based synapses (see Section 6.3). Single-trial illustration of target signal (black step function) and readout output (red curves) in three different SSNs, for $m = 0.9$ and strong noise corruption ($\sigma_\xi = 3$). For simplicity, only two out of ten input channels are shown. Supplementary Figure B.4 shows additional activity statistics. **(B)**: As in **(A)** for a rate-based model with $m = 1$ and $\sigma_\xi = 1$ (see Section 6.3 for details).

6.2.6 Variable map sizes

Despite their ubiquity throughout the neocortex, the characteristics of structured projection pathways are far from uniform (Bednar and Wilson, 2016), exhibiting marked differences in spatial precision and specificity, aligned with macroscopic gradients of cortical organization. This non-uniformity may play an important functional role in supporting feature aggregation (Hagler and Sereno, 2006) and the development of mixed representations (Patel et al., 2014) in higher (more anterior) cortical areas. Here, we

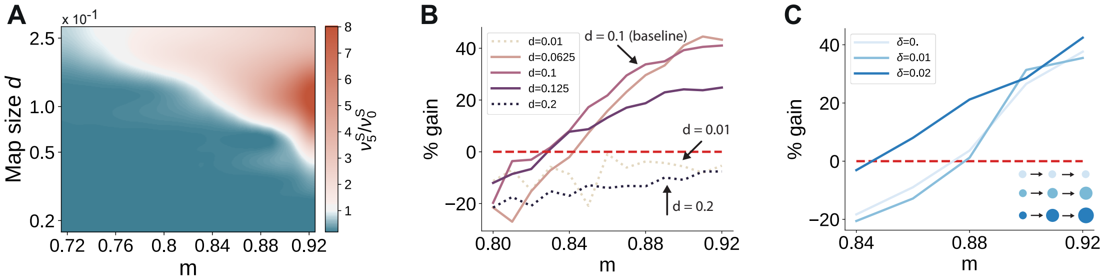


Figure 6.7: **Variation in the map sizes. (A)**: Difference in the firing rates of the stimulated sub-populations in the first and last sub-networks, $\nu_5^{\mathrm{S}} - \nu_0^{\mathrm{S}}$, as a function of modularity and map size (parameterized by d and constant throughout the network, i.e. $\delta = 0$, see Section 6.3). Depicted values correspond to stationary firing rates predicted by mean-field theory, smoothed using a Lanczos filter. Note that, in order to ensure that every neuron was uniquely tuned, i.e. there is no overlap between stimulus-specific sub-populations, the number of sub-populations was chosen to be proportional to the map size ($N_{\mathrm{C}} = 1/d$). **(B-C)**: Performance gain in SSN$_5$ relative to SSN$_0$ (ten stimuli, as in Figure 6.2C, F), for varying properties of structural mappings: **(B)** fixed map size ($\delta = 0$) with color shade denoting map size, and **(C)** linearly increasing map size ($\delta > 0$) and a smaller initial map size $d_0 = 0.04$. The results depict the average performance gains measured across five trials, using the current-based model illustrated in Figure 6.2 (ten stimuli) and no input noise ($\sigma_\xi = 0$). Supplementary Figure B.5 further illustrates how the activity varies across the modules as a function of the map size.

consider two scenarios in the baseline (current-based) model to examine the robustness of our findings to more complex topographic configurations.

First, we varied the size of stimulus-tuned sub-populations (parametrized by d_{i}, see Section 6.3) but kept them fixed across the network. For small sub-populations and intermediate degrees of topographic modularity, the activity along the stimulated pathway decays with network depth, suggesting that input information does not reach the deeper SSNs (see Figure 6.7A and Supplementary Figure B.5). These results place a lower bound on the size of stimulus-tuned sub-populations below which no signal propagation can occur, as reflected by the negative gain in performance for $d = 0.01$ (Figure 6.7B). Whereas denoising is robust to variation around the baseline value of $d = 0.1$ that yielded perfect partitioning of the feedforward projections (see Supplementary Materials), an upper bound may emerge due to increasing overlap between the maps ($d = 0.2$ in Figure 6.7B). In this case, the activity may “spill over” to other pathways than the stimulated one, corrupting the input representations and hindering accurate transmission and decoding. This can be alleviated by reduced or no overlap (as in Figure 6.7A), in which case signal propagation and denoising is successful for larger map sizes ($\nu_5^{\mathrm{S}}/\nu_0^{\mathrm{S}} > 1$ also for $d > 0.1$). We thus observe a trade-off between map size, overlap and the degree of topographic precision that is required to accurately propagate stimulus representations (see Section 6.4).

Second, we took into account the fact that these structural features are known to vary with hierarchical depth resulting in increasingly larger sub-populations and, consequently, increasingly overlapping stimulus selectivity (Smith et al., 2001; Patel et al., 2014; Bednar and Wilson, 2016). To capture this effect, we introduce a linear scaling of map size with depth ($d_{i+1} = \delta + d_i$ for $i \geq 1$, see Section 6.3). The ability of the circuit to gradually clean the signal's representation is fully preserved, as illustrated in Figure 6.7C. In fact, for intermediate modularity ($m < 0.9$) broadening the projections can further sharpen the reconstruction precision (compare curves for $\delta = 0.02$ and $\delta = 0$).

Taken together, these observations demonstrate that a gradual denoising of stimulus inputs can occur entirely as a consequence of the modular wiring between the subsequent processing circuits. Importantly, this effect generalizes well across diverse neuron and synapse models, as well as key system properties, making modular topography a potentially universal circuit feature for handling noisy data streams.

6.2.7 Modularity as a bifurcation parameter

The results so far indicate that the modular topographic projections, more so than the individual characteristics of neurons and synapses, lead to a sequential denoising effect through a joint process of signal amplification and noise suppression. To better understand how the system transitions to such an operating regime, it is helpful to examine its macroscopic dynamics in the limit of many sub-networks (Toyoizumi, 2012; Gajic and Shea-Brown, 2012; Kadmon and Sompolinsky, 2016). We apply standard mean-field techniques (Fourcaud and Brunel, 2002; Helias et al., 2013; Schuecker et al., 2015) to find the asymptotic firing rates (fixed points across sub-networks) of the stimulated and non-stimulated sub-populations as a function of topography (Figure 6.3D). For this, we can approximate the input μ to a group of neurons as a linear function of its firing rate ν with a slope κ that is determined by the coupling within the group and an offset given by inputs from other groups of neurons (orange line in Figure 6.8A). With an approximately sigmoidal rate transfer function, the self-consistent solutions are at the intersections marked in Figure 6.8A.

Formally, all neurons in the deep sub-networks of one topographic map form such a group as they share the same firing rate (asymptotic value). The coupling κ within this group comprises not only recurrent connections of one sub-network but also modular feedforward projections across sub-networks. For small modularity, the group is in an inhibition-dominated regime ($\kappa < 0$) and we obtain only one fixed point at low activity (Figure 6.8A, left). Importantly, the firing rate of this fixed point is the same for stimulated and non-stimulated topographic maps. Any influence of input signals applied to SSN_0 therefore vanishes in the deeper sub-networks and the signal cannot be reconstructed (*fading* regime). As topographic projections become more concentrated (larger m), κ changes sign and gradually leads to two additional fixed points (as conceptually illustrated in Figure 6.8A and quantified in Figure 6.8B by numerically

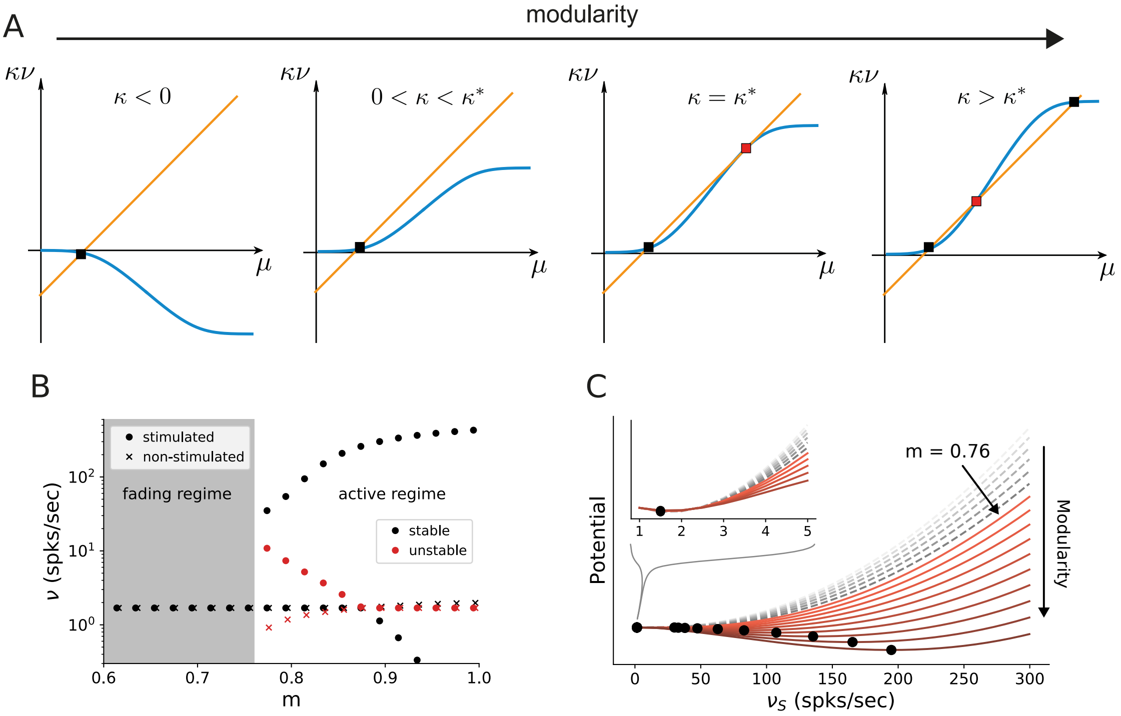


Figure 6.8: **Modularity changes the fixed point structure of the system. (A)**: Sketch for self-consistent solution (for the full derivation see Appendix B.4) for the firing rate of the stimulated sub-population (blue curves) and the linear relation $\kappa\nu = \mu - I$ (orange lines), in the limit of infinitely deep networks. Squares denote stable (black) and unstable (red) fixed points where input and output rates are the same. **(B)**: Bifurcation diagram obtained from the numerical evaluation of the mean-field self-consistency equations Eq. 6.9 and Eq. 6.10 showing a single stable fixed point in the fading regime, and multiple stable (black) and unstable (red) fixed points in the active regime where denoising occurs. **(C)**: Potential energy of the mean activity (see Methods and Eq. B.11 in Appendix B) for increasing topographic modularity. A stable state, corresponding to a local minimum in the potential, exists at a low non-zero rate in every case, including for $m \leq 0.75$ (grey dashed curves, inset). For $m \geq 0.76$ (colored solid curves), a second fixed point appears at progressively larger firing rates. Note that panels **(B-C)** show theoretical predictions obtained from the numerical evaluation of the mean-field self-consistency equations.

solving the self-consistent mean-field equations, see also Appendix B.4): an unstable one (red) that eventually vanishes with increasing m and a stable high-activity fixed point (black). The bistability opens the possibility to distinguish between stimulated and non-stimulated topographic maps and thereby reconstruct the signal in deep sub-networks: in the *active* regime beyond the *critical modularity threshold* (here $m \geq m_{\text{crit}} = 0.76$), a sufficiently strong input signal can drive the activity along the stimulated map to the high-activity fixed point, such that it can permeate the system, while the non-stimulated sub-populations still converge to the low-activity fixed point. Note that this critical modularity represents the minimum modularity value for which bistability emerges. It typically differs from the actual switching point m_{switch}, which additionally depends on the input intensity.

In the potential energy landscape U (see Methods), where stable fixed points correspond to minima, the bistability that emerges for more structured topography $m \geq m_{\text{crit}} = 0.76$ can be understood as a transition from a single minimum at low rates (Figure 6.8C, inset) to a second minimum associated with the high-activity state (Figure 6.8C). Even though the full dynamics of the spiking network away from the fixed point cannot be entirely understood in this simplified potential picture (see Appendix B), qualitatively, more strongly modular networks cause deeper potential wells, corresponding to more attractive dynamical states and higher firing rates (see Supplementary Figure B.10).

Because the intensity of the input signal dictates the rate of different populations in the initial sub-network SSN_0 (Figure 6.9A), it also determines, for any given modularity, whether the rate of the stimulated sub-population is in the basin of attraction of the high-activity (see Figure 6.9B, solid markers and arrows) or low-activity (dashed, blue marker and arrow) fixed point. Denoising, and therefore increasing signal reconstruction, is thus achieved by successively (across sub-networks) pushing the population states down along the different potential gradients.

As reported above, for the baseline network and (standard) input ($\lambda = 0.05$) used in Figure 6.2 and Figure 6.3, the switching point between low and high activity is at $m = 0.83$ (blue markers in Figure 6.9A, C). Stronger input signals move the switching point towards the minimal modularity $m = 0.76$ of the active regime (black markers in Figure 6.9A, C), while weaker inputs only induce a switch at larger modularities (grey markers in Figure 6.9A, C).

Noise in the input simply shifts the transition point to the high-activity state in a similar manner, with more modular connectivity required to compensate for stronger jitter (Figure 6.9D). However, as long as the mean firing rate of the stimulated sub-population in SSN_0 is slightly higher than that of the non-stimulated ones (up to 0.5 spks/sec), it is sufficient to position the system in the attracting basin of the high rate fixed point and the system is able to clean the signal representation. This indicates a remarkably robust denoising mechanism.

6.2.8 Critical modularity for denoising

In addition to properties of the input, the critical modularity marking the onset of the active regime is also influenced by neuronal and connectivity features. To build some intuition, it is helpful to consider the sigmoidal activation function of spiking neurons (Figure 6.10A). The nonlinearity of this function prohibits us from obtaining

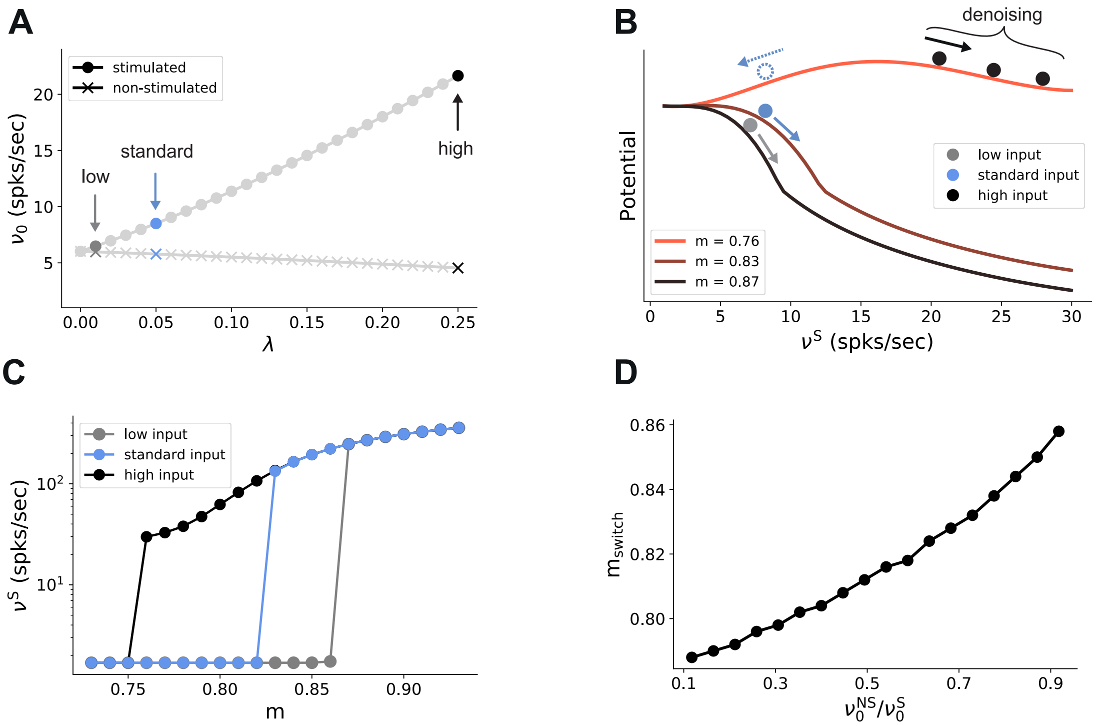


Figure 6.9: **Modularity changes the fixed point structure of the system.** **(A)**: Theoretical predictions for the stationary firing rates of the stimulated and non-stimulated sub-populations in SSN_0, as a function of stimulus intensity (λ, see Section 6.3). Low, standard and high denote λ values of 0.01, 0.05 (baseline value used in Figure 6.2) and 0.25, respectively. **(B)**: Sketch of attractor basins in the potential for different values of m. Markers correspond to the highlighted initial states in **(A)**, with solid and dashed arrows indicating attraction towards the high- and low-activity state, respectively. **(C)**: Firing rates of the stimulated sub-population as a function of modularity in the limit of infinite sub-networks, for the three different λ marked in **(A)**. **(D)**: Modularity threshold for the active regime shifts with increasing noise in the input, modeled as additional input to the non-stimulated sub-populations in SSN_0. Supplementary Figure B.6 shows the dependency of the effective feedforward couplings on different parameters. Note that all panels show theoretical predictions obtained from the numerical evaluation of the mean-field self-consistency equations.

quantitative, closed-form analytical expressions for the critical modularity and requires a numerical solution of the self-consistency equations (Figure 6.8B). However, since the continuous rate model shows qualitatively similar behavior to the spiking baseline model (see Section 6.2.5), we can study a fully analytically tractable model with piecewise linear activation function (Figure 6.10A, B) to expose the dependence of the critical modularity on both neuron and network properties (see detailed derivations in Appendix B).

In this simple model, the output is zero for inputs below $\mu_{\min} = 15$ and at maximum rate $\nu_{\max} = 150$ for inputs above $\mu_{\max} = 400$. In between these two bounds, the output is linearly interpolated $\nu(\mu) = \nu_{\max}(\mu - \mu_{\min})/(\mu_{\max} - \mu_{\min})$. As discussed before, successful denoising is achieved if the non-stimulated sub-populations are silent, $\nu^{\mathrm{NS}} = 0$, and the stimulated sub-populations are active, $\nu^{\mathrm{S}} > 0$. Note that in the following we focus on this ideal scenario representing perfect denoising, but, in principle, intermediate solutions with $\nu^{\mathrm{S}} \gg \nu^{\mathrm{NS}} > 0$ may also occur and could still be considered as successful denoising. Analyzing for which neuron, network and input properties this scenario is achieved, we obtain multiple conditions for the modularity that need to be fulfilled.

The first condition illustrates the dependence of the critical modularity on the neuron model (Figure 6.10C, purple horizontal line)

$$m \geq \frac{(\mu_{\max} - \mu_{\min})N_{\mathrm{C}}}{(1-\alpha)\mathcal{J}\nu_{\max} + (\mu_{\max} - \mu_{\min})(N_{\mathrm{C}} - 1)}, \tag{6.1}$$

where N_{C} is the number of stimulus-specific sub-populations and $\alpha \leq 1$ (typically with a value of 0.25) represents the (reduced) noise ratio in the deeper sub-networks, with α scaling the noise and $1 - \alpha$ scaling the feedforward connections (see Section 6.3). This is necessary to ensure that the total excitatory input to each neuron is consistent across the network. In particular, the critical modularity depends on the dynamic range of input $\mu_{\max} - \mu_{\min}$ and output $\nu_{\max}$. The condition represents a lower bound on the modularity required for denoising. Importantly, while it depends on the effective coupling strength $\mathcal{J}$, the noise ratio α and the number of maps N_{C} (see Section 6.3), it does not depend on the nature of the recurrent interactions (E/I ratio) and the strength of the external background input. In addition, we find two additional critical values of the modularity (cyan and green curves in Figure 6.10C-E), both of which do depend on the strength of the external background input ν_{X} and the recurrent connectivity (E/I ratio γg):

$$m = \frac{N_{\mathrm{C}}}{N_{\mathrm{C}} - 1} - \frac{1}{N_{\mathrm{C}} - 1} \frac{(1-\alpha)\mathcal{J}\nu_{\max}}{\mu_{\max} - \alpha\mathcal{J}\nu_{\mathrm{X}} - \frac{\mathcal{J}}{N_{\mathrm{C}}}(1+\gamma g)\,\nu_{\max}} \tag{6.2}$$

$$m = 1 - \frac{\left(\mu_{\min} - \alpha\mathcal{J}\nu_{\mathrm{X}} - \frac{\mathcal{J}}{N_{\mathrm{C}}}(1+\gamma g)\,\nu_{\max}\right)}{\mathcal{J}(1-\alpha)\nu_{\max} - (N_{\mathrm{C}} - 1)\left(\mu_{\min} - \alpha\mathcal{J}\nu_{\mathrm{X}} - \frac{\mathcal{J}}{N_{\mathrm{C}}}(1+\gamma g)\,\nu_{\max}\right)} \tag{6.3}$$

Depending on the external input strength ν_{X}, these are either upper or lower bounds. In the denominator of these expressions, the total input (recurrent and external) is compared to the limits of the dynamic range of the neuron model. The cancellation between recurrent and external inputs in the inhibition-dominated baseline model typically yields a total input within the dynamic range of the neuron, such that modularity in feedforward connections can decrease the input of the non-stimulated sub-populations to silence them, and increase the input of the stimulated sub-populations to support their activity. The competition between the excitatory and inhibitory contributions ensures that the

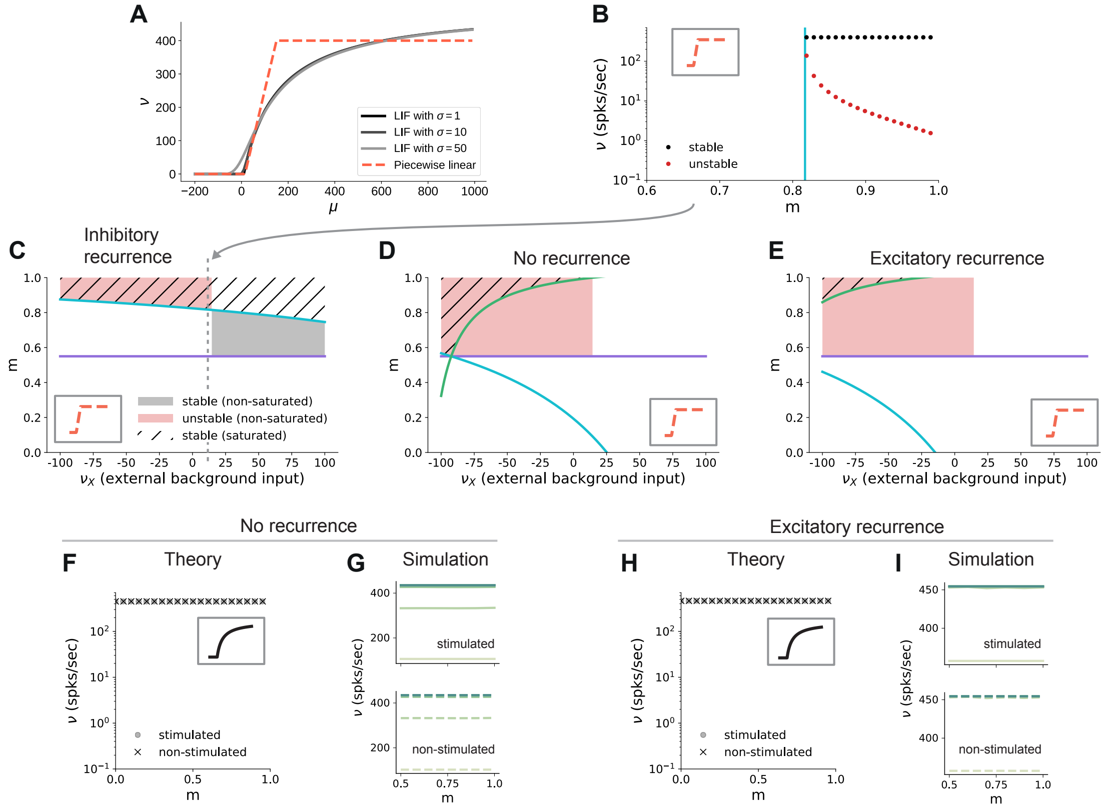


Figure 6.10: **Dependence of critical modularity on neuron and connectivity features.** **(A)**: Activation function $\nu(\mu, \sigma)$ for LIF model as a function of the mean input μ for $\sigma = 1, 10, 50$ (black to gray) and piecewise linear qualitative approximation (red). **(B)**: Bifurcation diagram as in Figure 6.8B, but for piecewise linear activation function. Low-activity fixed points at $\nu = 0$ are not shown, which is always the case for the non-stimulated sub-populations. This panel corresponds to the cross-section marked by the gray dashed lines in **(C)**, at $\nu_X = 12$. Vertical cyan bar corresponds to the lower bound on modularity depicted by the cyan curve in **(C)** for $\nu_X = 12$. **(C)**: Analytically derived bounds on modularity (purple line corresponds to Eq. 6.1, cyan curve to Eq. 6.2) as a function of external input for the baseline model with inhibition-dominated recurrent connectivity ($g = -12$). Shaded regions denote positions of stable (black) and unstable (red) fixed points with $0 < \nu^S < \nu_{\max}$ and $\nu^{NS} = 0$. Hatched area represents region with stable fixed points at saturated rates. Denoising occurs in all areas with stable fixed points (hatched and black shaded regions). $\nu_x < 0$ corresponds to inhibitory external background input with rate $|\nu_x|$. **(D)**: Same as **(C)** for networks with no recurrent connectivity within SSNs (green curve defined by Eq. 6.3). **(E)**: Same as **(C)**, for networks with excitation-dominated connectivity within SSNs ($g = -3$). **(F)**: Same as Figure 6.8B, obtained through numerical evaluation of the mean-field self-consistent equations for the spiking model. All non-zero fixed points are stable, with points representing stimulated (circle) and non-stimulated (cross) populations overlapping. **(G)**: Mean firing rates across SSNs in the current-based model with no recurrent connections (5 seconds of activity, averaged over 5 trials). **(H, I)** Same as **(F, G)**, for networks with excitation-dominated connectivity.

total input does not lead to a saturating output activity. Thus, for inhibitory recurrence, denoising can be achieved at a moderate level of modularity over a large range of external background inputs (shaded black and hatched regions in Figure 6.10C), which demonstrates a robust denoising mechanism even in the presence of changes in the input environment.

In contrast, if recurrent connections are absent, strong inhibitory external background input is required to counteract the excitatory feedforward input and achieve a denoising scenario (Figure 6.10D). Fixed points at non-saturated activity $\nu^{\mathrm{S}} > 0$ are also present for low excitatory external input, but unstable due to the positive recurrent feedback. This is because in networks without recurrence, there is no competition between the recurrent input and the external and feedforward inputs. As a result, the input to both the stimulated and non-stimulated sub-populations is typically high, such that modulation of the feedforward input via topography cannot lead to a strong distinction between the pathways as required for denoising. In these networks, one typically observes high activity in all populations. A similar behavior can be observed in excitation-dominated networks (Figure 6.10E), where the inhibitory external background input must be even stronger to compensate for the excitatory feedforward and recurrent connectivity and reach a stable denoising regime.

Note that inhibitory external input is not in line with the excitatory nature of external inputs to local circuits in the brain and is therefore biologically implausible. One way to achieve denoising in excitation-dominated networks for excitatory background inputs would be to shift the dynamic range of the activation function (see Supplementary Figure B.7), which is, however, not consistent with the biophysical properties of real neurons (distance between threshold and rest as compared to typical strengths of postsynaptic potentials). In summary, we find that recurrent inhibition is crucial to achieve denoising in biologically plausible settings.

These results on the role of recurrence and external input can be transferred to the behavior of the spiking model. While details of the fixed point behavior depend on the specific choice of the activation function, Figure 6.10F and Figure 6.10H show that there is also no denoising regime for the spiking model in the case of no or excitation-dominated recurrence and a biologically plausible level of external input. Instead, one finds high activity in both stimulated and non-stimulated sub-populations, as confirmed by network simulations (Figure 6.10G and Figure 6.10I). Supplementary Figure B.8 further confirms that even reducing the external input to zero does not avoid this high-activity state in both stimulated and non-stimulated sub-populations for $m < 1$.

6.2.9 Input integration and multi-stability

The analysis considered in the sections above is restricted to a system driven with a single external stimulus. However, to adequately understand the system's dynamics, we need to account for the fact that it can be concurrently driven by multiple input streams.

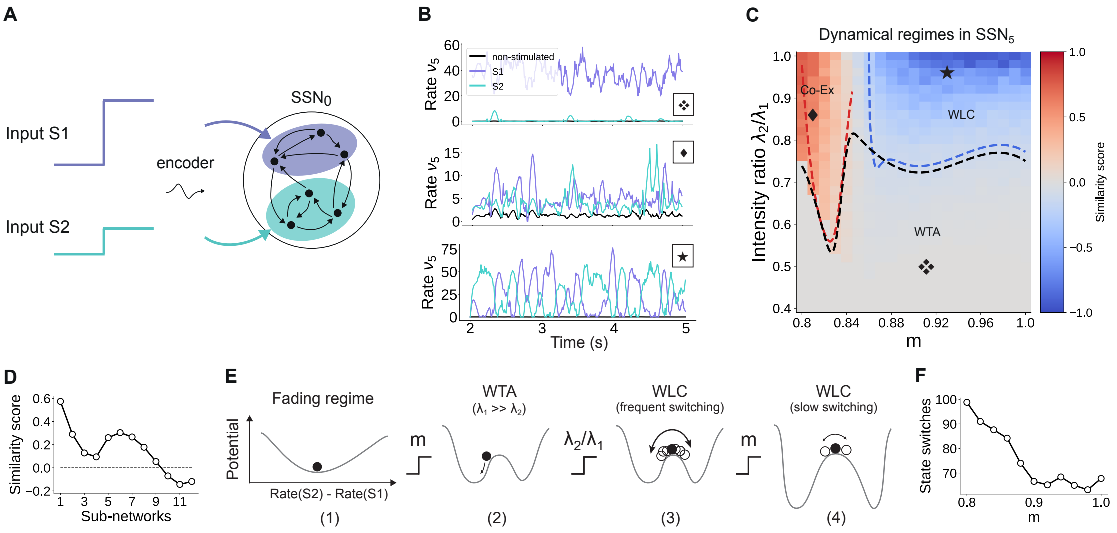


Figure 6.11: **For multiple input streams, topography may elicit a wide range of dynamical regimes.** **(A)**: Two active input channels with corresponding stimulus intensities λ_1 and λ_2, mapped onto non-overlapping sub-populations, drive the network simultaneously. Throughout this section, $\lambda_1 = 0.05$ is fixed to the previous baseline value. **(B)**: Mean firing rates of the two stimulated sub-populations (purple and cyan), as well as the non-stimulated sub-populations (black) for three different combinations of m and ratios λ_2/λ_1 (as marked in **(C)**).**(C)**: Correlation-based similarity score shows three distinct dynamical regimes in SSN_5 when considering the firing rates of two, simultaneously stimulated sub-populations associated with S_1 and S_2, respectively: coexisting (Co-Ex, red area), winner-takes-all (WTA, grey) and winnerless competition (WLC, blue). Curves mark the boundaries between the different regimes (see Section 6.3). Activity for marked parameter combinations shown in **(B)**. **(D)**: Evolution of the similarity score with increasing network depth, for $m = 0.83$ and input ratio of 0.86. For deep networks, the Co-Ex region vanishes and the system converges to either WLC or WTA dynamics. **(E)**: Schematic showing the influence of modularity and input intensity on the system's potential energy landscape (see Section 6.3): (1) in the fading regime there is a single low-activity fixed point (minimum in the potential); (2) increasing modularity creates two high-activity fixed points associated with S1 and S2, with the dynamics always converging to the same minimum due to $\lambda_1 >> \lambda_2$; (3) strengthening S2 balances the initial conditions, resulting in frequent, fluctuation-driven switching between the two states. (4) for larger m values, switching speed decreases as the wells become deeper and the barrier between the wells wider. **(F)**: Switching frequency between the dominating sub-populations in SSN_5 decays with increasing modularity. Data computed over 10 sec, for $\lambda_2/\lambda_1 = 0.9$. Supplementary Figure B.9 and Supplementary Figure B.10 show the evolution of the Co-Ex region over 12 modules and the potential landscape, respectively.

If two simultaneously active stimuli drive the system (see illustration in Figure 6.11A), the qualitative behavior where the responses along the stimulated (non-stimulated) maps are enhanced (silenced) is retained if the strength of the two input channels is sufficiently different (Figure 6.11B, top panel). In this case, the weaker stimulus is not strong enough to drive the sub-population it stimulates towards the basin of attraction of the high-activity fixed point. Consequently, the sub-population driven by this second stimulus behaves as a non-stimulated sub-population and the system remains responsive to only one of the two inputs, acting as a winner-takes-all (WTA) circuit. If, however, the ratio of stimulus intensities varies, two active sub-populations may co-exist (Figure 6.11B, center) and/or compete (bottom panel), depending also on the degree of topographic modularity.

To quantify these variations in macroscopic behavior, we focus on the dynamics of SSN$_5$ and measure the similarity (correlation coefficient) between the firing rates of the two stimulus-specific sub-populations as a function of modularity and ratio of input intensities λ_2/λ_1 (see Section 6.3 and Figure 6.11C). In the case that both inputs have similar intensities but the feedforward projections are not sufficiently modular, both sub-populations are activated simultaneously (Co-Ex, red area in Figure 6.11C). This is the dynamical regime that dominates the earlier sub-networks. However, this is a transient state, and the Co-Ex region gradually shrinks with network depth until it vanishes completely after approximately 9-10 SSNs (see Figure 6.11D).

For low modularity, the system settles in the single stable state associated with near-zero firing rates, as illustrated schematically in the energy landscape in Figure 6.11E (1) (see Section 6.3, Appendix B and Supplementary Materials for derivations and numerical simulations). Above the critical modularity value, the system enters one of two different regimes. For $m > 0.84$ and an input ratio below 0.7 (Figure 6.11C, grey area), one stimulus dominates (WTA) and the responses in the two populations are uncorrelated (Figure 6.11B, top panel). Although the potential landscape contains two minima corresponding to either population being active, the system always settles in the high-activity attractor state corresponding to the dominating input (Figure 6.11E, (2)).

If, however, the two inputs have comparable intensities and the topographic projections are sharp enough ($m > 0.84$), the system transitions into a different dynamical state where neither stimulus-specific sub-population can maintain an elevated firing rate for extended periods. In the extreme case of nearly identical intensities ($\lambda_2/\lambda_1 \geq 0.9$) and high modularity, the responses become anti-correlated (Figure 6.11B, bottom panel), i.e. the activation of the two stimulus-specific sub-populations switches, as they engage in a dynamic behavior reminiscent of winnerless competition (WLC) between multiple neuronal groups (Lagzi and Rotter, 2015; Rost et al., 2018). The switching between the two states is driven by stochastic fluctuations (Figure 6.11E, (3)). The depth of the wells and width of the barrier (distance between fixed points) increase with modularity (see Figure 6.11E, (4) and Supplementary Figure B.10), suggesting a greater difficulty in moving between the two attractors and consequently fewer state changes. Numerical simulations confirm this slowdown in switching (Figure 6.11F).

We wish to emphasize that the different dynamical states arise primarily from the feedforward connectivity profile. Nevertheless, even though the synaptic weights are not directly modified, varying the topographic modularity does translate to a modification of the effective connectivity weights (Figure 6.4B). The ratio of stimulus intensities also plays a role in determining the dynamics, but there is a (narrow) range (approximately between 0.75 and 0.8) for which all 3 regions can be reached through sole modification of the modularity. Together, these results demonstrate that topography can not only lead to spatial denoising but also enable various, functionally important network operating points.

6.2.10 Reconstruction and denoising of dynamical inputs

Until now, we have considered continuous but piecewise constant, step signals, with each step lasting for a relatively long and fixed period of 200 ms. This may give the impression that the denoising effects we report only work for static or slowly changing inputs, whereas naturalistic stimuli are continuously varying. Nevertheless, sensory perception across modalities relies on varying degrees of temporal and spatial discretization (VanRullen and Koch, 2003), with individual (sub-)features of the input encoded by specific (sub-)populations of neurons in the early stages of the sensory hierarchy. In this section, we will demonstrate that denoising is robust to the temporal properties of the input and its encoding, even as we relax many of the assumptions made in previous sections.

We consider a sinusoidal input signal $x(t) = \sin(3t) + \cos(t)$, which we discretize and map onto the network according to the depiction in Figure 6.12A. This approach is similar to previous works, for instance, it can mimic the movement of a light spot across the retina (Klos et al., 2018). By varying the sampling interval dt and the number of channels k, we can change the coarseness of the discretization from step-like signals to more continuous approximations of the input. If we choose a high sampling rate ($dt = 1$ ms) and sufficient channels ($k = 40$), we can accurately encode even fast-changing signals (Figure 6.12B). Given that each SSN is inhibition-dominated and therefore close to the balanced state, the network exhibits a fast-tracking property (van Vreeswijk and Sompolinsky, 1996) and can accurately represent and denoise the underlying continuous signal in the spiking activity (Figure 6.12C, top). This is also captured by the readout, with the tracking precision increasing with network depth (Figure 6.12C, bottom). In this condition, there is a performance gain of up to 50% in the noiseless case (Figure 6.12F, left) and similar values for varying levels of noise (Figure 6.12F, right).

Note that due to the increased number of input channels (40 compared to 10) projecting to the same number of neurons in SSN_0 as before (800), the effective amount of noise each neuron for the same σ_ξ receives is, on average, four times larger than in the baseline network. Moreover, the task was made more difficult by the significant overlap between the maps ($N_\mathrm{C} = 20$) as well as the resulting decrease in neuronal input selectivity. Nevertheless, similar results were obtained for slower and more coarsely sampled signals (Figure 6.12D,E,G).

We found comparable denoising dynamics for a large range of parameter combinations

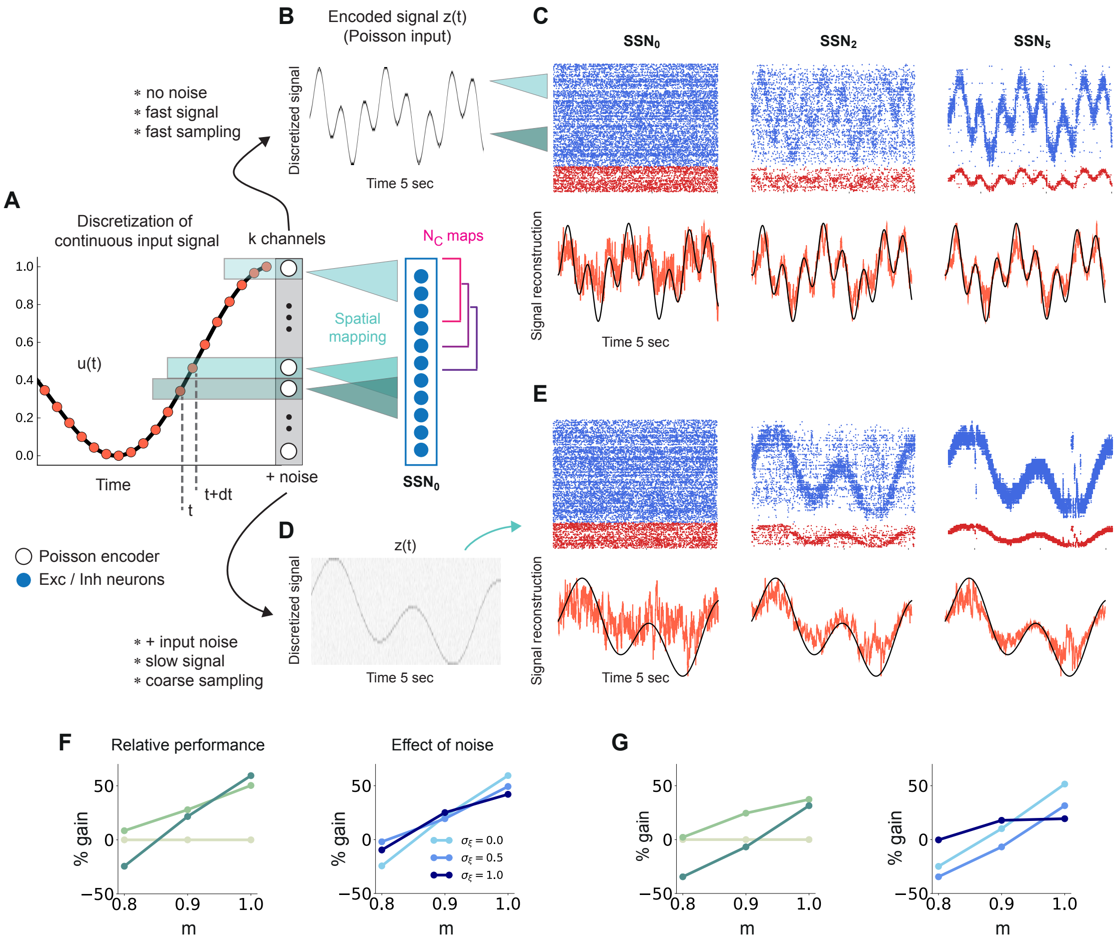


Figure 6.12: **Reconstruction of a dynamic, continuous input signal**. **(A)**: Sketch of the encoding and mapping of a sinusoidal input $x(t)$ onto the current-based network model. The signal is sampled at regular time intervals dt, with each sample binned into one of k channels (active for a duration of dt). This yields a temporally and spatially discretized k-dimensional binary signal $u(t)$ from which the final noisy input $z(t)$ is obtained (see Figure 6.1 and Section 6.3). Unlike the one-to-one mapping in Figure 6.1, here we decouple the number of channels $k = 40$ from that of topographic maps, $N_\text{C} = 20$ (map size is unchanged, $C_\text{i} = 800$). Because $N_\text{C} < k$, the channels project to evenly spaced but overlapping sub-populations in SSN$_0$, while the maps themselves overlap significantly. **(B)**: Discretized signal $z(t)$ and rate encoding for input $x(t) = \sin(10t) + \cos(3t)$, with $dt = 1$ ms and no noise ($\sigma_\xi = 0$). **(C)**: Top panel shows the spiking activity of 500 randomly chosen excitatory (blue) and inhibitory (red) neurons in SSN$_0$, SSN$_2$ and SSN$_5$, for $m = 0.9$. Corresponding target signal $x(t)$ (black) and readout output (red) are shown in the bottom panel. **(D-E)**: Same as **(B-C)**, but for a slowly varying signal (sampled at $dt = 20$ ms), $\sigma_\xi = 0.5\nu_\text{in}$ and $m = 1$. **(F)**: Relative gain in performance in SSN$_2$ and SSN$_5$ compared to SSN$_0$, for $\sigma_\xi = 0$ (left). Color shade denotes network depth. Right panel shows relative gain in SSN$_5$ for different levels of noise $\sigma_\xi \in \{0, 0.5, 1.\}$. **(G)**: Same as **(F)**, but for the slowly changing signal shown in **(D-E)**. Performance results are averaged across five trials. We used 20 seconds of data for training and 10 seconds for testing (activity sampled every 1 ms).

involving the map size, number of maps, number of channels and signal complexity. Although there are limits on the frequencies (and noise intensity) the network can track (see Supplementary Figure B.11), these findings indicate a very robust and flexible phenomenon for denoising spatially encoded sensory stimuli.

6.3 Methods

6.3.1 Network architecture

We consider a feedforward network architecture where each sub-network (SSN) is a balanced random network (Brunel, 2000) composed of $N = 10000$ homogeneous leaky integrate-and-fire neurons, grouped into a population of $N^{\mathrm{E}} = 0.8N$ excitatory and $N^{\mathrm{I}} = 0.2N$ inhibitory units. Within each sub-network, neurons are connected randomly and sparsely, with a fixed number of $K_{\mathrm{E}} = \epsilon N^{\mathrm{E}}$ local excitatory and $K_{\mathrm{I}} = \epsilon N^{\mathrm{I}}$ local inhibitory inputs per neuron. The sub-networks are arranged sequentially, i.e. the excitatory neurons $\mathrm{E_i}$ in $\mathrm{SSN_i}$ project to both $\mathrm{E_{i+1}}$ and $\mathrm{I_{i+1}}$ populations in the subsequent sub-network $\mathrm{SSN_{i+1}}$ (for an illustrative example, see Figure 6.1a). There are no inhibitory feedforward projections. Although projections between sub-networks have a specific, non-uniform structure (see next section), each neuron in $\mathrm{SSN_{i+1}}$ receives the same total number of synapses from the previous SSN, K_{FF}.

In addition, all neurons receive K_{X} inputs from an external source representing stochastic background noise. For the first sub-network, we set $K_{\mathrm{X}} = K_{\mathrm{E}}$, as it is commonly assumed that the number of background input synapses modeling local and distant cortical input is in the same range as the number of recurrent excitatory connections (see e.g., Brunel, 2000; Kumar et al., 2008b; Duarte and Morrison, 2014). To ensure that the total excitatory input to each neuron is consistent across the network, we scale K_{X} by a factor of $\alpha = 0.25$ for the deeper SSNs and set $K_{\mathrm{FF}} = (1 - \alpha)K_{\mathrm{E}}$, resulting in a ratio of 3:1 between the number of feedforward and background synapses.

6.3.2 Modular feedforward projections

Within each SSN, each neuron is assigned to one or more of N_{C} sub-populations *SP* associated with a specific stimulus ($N_{\mathrm{C}} = 10$ unless otherwise stated). This is illustrated in Figure 6.1a for $N_{\mathrm{C}} = 2$. We choose these sub-populations to minimize their overlap within each $\mathrm{SSN_i}$, and control their effective size $C_{\mathrm{i}}^{\beta} = d_{\mathrm{i}} N^{\beta}, \beta \in [\mathrm{E}, \mathrm{I}]$, through the scaling parameter $d_{\mathrm{i}} \in [0, 1]$. Depending on the size and number of sub-populations, it is possible that some neurons are not part of any or that some neurons belong to multiple such sub-populations (overlap).

Map size. In what follows, a topographic map refers to the sequence of sub-populations in the different sub-networks associated with the same stimulus. To enable a flexible manipulation of the map sizes, we constrain the scaling factor d_{i} by introducing a step-wise linear increment δ, such that $d_{\mathrm{i}} = d_0 + i\delta, i \geq 1$. Unless otherwise stated, we set $d_0 = 0.1$ and $\delta = 0$. Note that all SPs within a given SSN have the same size. In this study, we will only explore values in the range $0 \leq \delta \leq 0.02$ to ensure consistent map sizes across the system, i.e., $0 \leq d_{\mathrm{i}} \leq 1$ for all $\mathrm{SSN_i}$ (see constraints in Appendix B.3).

Modularity. To systematically modify the degree of modular segregation in the topographic projections, we define a modularity parameter that determines the relative probability for feedforward connections from a given SP in $\mathrm{SSN_i}$ to target the corresponding SP in $\mathrm{SSN_{i+1}}$. Specifically, we follow (Newman, 2009; Pradhan et al., 2011) and define $m = 1 - \frac{p_0}{p_{\mathrm{c}}} \in [0, 1]$ as the ratio of the feedforward projection probabilities between neurons belonging to different SPs (p_0) and between neurons on the same topographic map (p_{c}). According to the above definition, the feedforward connectivity matrix is random and homogeneous (Erdős-Rényi graph) if $m = 0$ or $d_{\mathrm{i}} = 1$ (see Figure 6.1a). For $m = 1$ it is a block-diagonal matrix, where the individual SPs overlap only when $d_{\mathrm{i}} > 1/N_{\mathrm{C}}$. In order to isolate the effects on the network dynamics and computational performance attributable exclusively to the topographic structure, the overall density of the feedforward connectivity matrix is kept constant at $(1 - \alpha) * \epsilon = 0.075$ (see also previous section). We note that, while providing the flexibility to implement the variations studied in this manuscript, this formalism has limitations (see Appendix B.3).

6.3.3 Neuron and synapse model

We study networks composed of leaky integrate-and-fire (LIF) neurons with fixed voltage threshold and static synapses with exponentially decaying postsynaptic currents or conductances. The sub-threshold membrane potential dynamics of such a neuron evolves according to:

$$\tau_{\mathrm{m}} \frac{dV(t)}{dt} = (V_{\mathrm{rest}} - V(t)) + R\left(I^{\mathrm{E}}(t) + I^{\mathrm{I}}(t) + I^{\mathrm{X}}(t)\right) \tag{6.4}$$

where τ_{m} is the membrane time constant, and RI^{β} is the total synaptic input from population $\beta \in [E, I]$. The background input I^{X} is assumed to be excitatory and stochastic, modeled as a homogeneous Poisson process with constant rate ν_{X}. Synaptic weights J_{ij}, representing the efficacy of interaction from presynaptic neuron j to postsynaptic neuron i, are equal for all realized connections of a given type, i.e., $J_{\mathrm{EE}} = J_{\mathrm{IE}} = J$ for excitatory and $J_{\mathrm{EI}} = J_{\mathrm{II}} = gJ$ for inhibitory synapses. All synaptic delays and time constants are equal in this setup. For a complete, tabular description of the models and model parameters used throughout this study, see Appendix B.2.

Following previous works (Zajzon et al., 2019; Duarte and Morrison, 2014), we choose the intensity of the stochastic input ν_{X} and the E-I ratio g such that the first two sub-networks operate in a balanced, asynchronous irregular regime when driven solely by background input. This is achieved with $\nu_{\mathrm{X}} = 12$ spks/sec and $g = -12$, resulting in average firing rates of ~ 3 spks/sec, coefficient of variation (CV_{ISI}) in the interval $[1.0, 1.5]$ and Pearson cross-correlation (CC) ≤ 0.01 in SSN_0 and SSN_1.

In Section 6.2.5 we consider two additional systems, a network of LIF neurons with conductance-based synapses and a continuous firing rate model. The LIF network is described in detail in Section 5.2 of the previous chapter. Spike-triggered synaptic conductances are modeled as exponential functions, with fixed and equal conduction delays for all synapses. Key differences to the current-based model include, in addition to the biologically more plausible synapse model, longer synaptic time constants and stronger input (see also Zajzon et al. (2019) and Supplementary Table B.3 for the numerical values of all parameters).

The continuous rate model contains $N = 3000$ nonlinear units, the dynamics of which are governed by:

$$\begin{aligned} \tau_x \frac{\mathrm{d}\boldsymbol{x}}{\mathrm{dt}} &= -\boldsymbol{x} + \boldsymbol{J}\boldsymbol{r} + J^{\mathrm{in}}\boldsymbol{u} - \boldsymbol{b}^{\mathrm{rec}} + \sqrt{2\tau_{\boldsymbol{x}}}\sigma_{\mathrm{X}}\boldsymbol{\xi} \\ \boldsymbol{r} &= 0.5(1 + \tanh(\boldsymbol{x})) \end{aligned} \tag{6.5}$$

where $\boldsymbol{x}$ represents the activation and $\boldsymbol{r}$ the output of all units, commonly interpreted as the synaptic current variable and the firing rate estimate, respectively. The rates r_{i} are obtained by applying the nonlinear transfer function $\tanh(x_{\mathrm{i}})$, modified here to constrain the rates to the interval $[0, 1]$. $\tau_x = 10$ ms is the neuronal time constant, $\boldsymbol{b}^{\mathrm{rec}}$ is a vector of individual neuronal bias terms (i.e. a baseline activation), and J and J^{in} are the recurrent (including feedforward) and input weight matrices, respectively. These are constructed in the same manner as for the spiking networks, such that the overall connectivity, including the input mapping onto SSN_0, is identical for all three models. Input weights are drawn from a uniform distribution, while the rest follow a normal distribution. Finally, $\boldsymbol{\xi}$ is a vector of N independent realizations of Gaussian white noise with zero mean and variance scaled by σ_{X}. The differential equations are integrated numerically, using the Euler–Maruyama method with step $\delta t = 1$ ms, with specific parameter values given in Supplementary Table B.5.

6.3.4 Signal reconstruction task

We evaluate the system's ability to recover a simple, continuous step signal from a noisy variation using linear combinations of the population responses in the different SSNs (Maass et al., 2002). This is equivalent to probing the network's ability to function as a denoising autoencoder (Bengio et al., 2013).

To generate the N_{C}-dimensional input signal $u(t)$, we randomly draw stimuli from a predefined set $S = \{S_1, S_2, ..., S_{N_{\mathrm{C}}}\}$ and set the corresponding channel to active for a fixed duration of 200 ms (Figure 6.1, left). This binary step signal $u(t)$ is also the target signal to be reconstructed. The effective input is obtained by scaling $u(t)$ with the input rate ν_{in}, and adding a Gaussian white noise process with zero mean and variance σ_ξ^2. Rectifying the resulting signal leads to the final form of the continuous input signal $z(t) = (\nu_{\text{in}} u(t) + \xi(t))_+$. This allows us to control the amount of noise in the input, and thus the task difficulty, through a single parameter σ_ξ.

To deliver the input to the circuit, the analog signal $z(t)$ is converted into spike trains, with its amplitude serving as the rate of an inhomogeneous Poisson process generating independent spike trains. We set the scaling amplitude to $\nu_{\text{in}} = K_E \lambda \nu_{\text{X}}$, modelling stochastic input with fixed rate $\lambda \nu_{\text{X}}$ from $K_E = 800$ neurons. If not otherwise specified, $\lambda = 0.05$ holds, resulting in a mean firing rate below 8 spks/sec in SSN_0 (see Figure 6.3C).

Each input channel k is mapped onto one of the N_{C} stimulus-specific sub-populations of excitatory and inhibitory neurons in the first (input) sub-network SSN_0, chosen according to the procedure described above (see also Figure 6.1). This way, each stimulus S_{k} is mapped onto a specific set of sub-populations in the different sub-networks, i.e., the topographic map associated with S_{k}.

For each stimulus in the sequence, we sample the responses of the excitatory population in each SSN_{i} at fixed time points (once every ms) relative to stimulus onset. We record from the membrane potentials V_m as they represent a parameter-free and direct measure of the population state (Duarte et al., 2018; van den Broek et al., 2017). The activity vectors are then gathered in a state matrix $X_{\text{SSN}_i} \in \mathbb{R}^{N^{\text{E}} \times T}$, which is then used to train a linear readout to approximate the target output of the task (Lukoševičius and Jaeger, 2009). We divide the input data, containing a total of 100 stimulus presentations (yielding $T = 20000$ samples), into a training and a testing set (80/20 %), and perform the training using ridge regression (L2 regularization), with the regularization parameter chosen by leave-one-out cross-validation on the training dataset.

Reconstruction performance is measured using the normalized root mean squared error (NRMSE). For this particular task, the effective delay in the build-up of optimal stimulus representations varies greatly across the sub-networks. To close in on the optimal delay for each SSN_{i}, we train the state matrix $X_{\text{SSN}_{\text{i}}}$ on a larger interval of delays and choose the one that minimizes the error, averaged across multiple trials.

6.3.5 Effective connectivity and stability analysis

To better understand the role of structural variations on the network's dynamics, we determine the network's effective connectivity matrix W analytically by linear stability analysis around the system's stationary working points (see Appendix B for the complete derivations). The elements $w_{ij} \in W$ represent the integrated linear response of a target neuron i, with stationary rate ν_{i}, to a small perturbation in the input rate ν_{j} caused by a spike from presynaptic neuron j. In other words, w_{ij} measures the average number of additional spikes emitted by a target neuron i in response to a spike from the presynaptic neuron j, and its relation to the synaptic weights is defined by (Tetzlaff et al., 2012; Helias et al., 2013):

$$
\begin{aligned}
w_{ij} = \frac{\partial \nu_i}{\partial \nu_j} &= \widetilde{\alpha} J_{ij} + \widetilde{\beta} J_{ij}^2 \\
\text{with} \quad \widetilde{\alpha} &= \sqrt{\pi}\,(\tau_\mathrm{m}\nu_i)^2\,\frac{1}{\sigma_i}\,(f(y_\theta) - f(y_\mathrm{r})) \\
\text{and } \tilde{\beta} &= \sqrt{\pi}\,(\tau_\mathrm{m}\nu_i)^2\,\frac{1}{2\sigma_i^2}\,(f(y_\theta)y_\theta - f(y_\mathrm{r})y_\mathrm{r})\,.
\end{aligned}
\tag{6.6}
$$

Note that in Figure 6.3 we ignore the contribution $\widetilde{\beta}$ resulting from the modulation in the input variance σ_j^2 which is significantly smaller due to the additional factor $1/\sigma_i \sim \mathcal{O}(1/\sqrt{N})$. Importantly, the effective connectivity matrix W allows us to gain insights into the stability of the system by eigenvalue decomposition. For large random coupling matrices, the effective weight matrix has a spectral radius $\rho = \max_k(\mathrm{Re}\{\lambda_\mathrm{k}\})$ which is determined by the variances of W (Rajan and Abbott, 2006). For inhibition-dominated systems, such as those we consider, there is a single negative outlier representing the mean effective weight, given the eigenvalue λ_k^* associated with the unit vector. The stability of the system is thus uniquely determined by the spectral radius ρ: values smaller than unity indicate stable dynamics, whereas $\rho > 1$ leads to unstable linearized dynamics.

6.3.6 Fixed point analysis

For the mean-field analysis, the N_C sub-populations in each sub-network can be reduced to only two groups of neurons, the first one comprising all neurons of the stimulated SPs and the second one comprising all neurons in all non-stimulated SPs. This is possible because 1) the firing rates of the excitatory and inhibitory neurons within one SP are identical, owing to homogeneous neuron parameters and matching incoming connection statistics, and 2) all neurons in non-stimulated SPs have the same rate ν^NS that is in general different from the rate of the stimulated SP ν^S. Here we only sketch the main steps, with a detailed derivation given in Appendix B.4.

The mean inputs to the first sub-network can be obtained via

$$
\begin{aligned}
\mu^\mathrm{S} &= (1+\lambda)\mathcal{J}\nu_\mathrm{x} + \frac{1}{N_\mathrm{C}}\mathcal{J}\,(1+\gamma g)\,\nu^\mathrm{S} + \frac{N_\mathrm{C}-1}{N_\mathrm{C}}\mathcal{J}\,(1+\gamma g)\,\nu^\mathrm{NS}\,, \\
\mu^\mathrm{NS} &= \qquad\quad \mathcal{J}\nu_\mathrm{x} + \frac{1}{N_\mathrm{C}}\mathcal{J}\,(1+\gamma g)\,\nu^\mathrm{S} + \frac{N_\mathrm{C}-1}{N_\mathrm{C}}\mathcal{J}\,(1+\gamma g)\,\nu^\mathrm{NS}
\end{aligned}
\tag{6.7}
$$

where $\gamma = K_\mathrm{I}/K_\mathrm{E}$ and $\mathcal{J} = \tau_\mathrm{m} K_\mathrm{E} J$. Both equations are of the form

$$\kappa\nu = \mu - I \tag{6.8}$$

where κ is the effective self-coupling of a group of neurons with rate ν and input μ, and I denotes the external inputs from other groups. Equation (6.8) describes a linear relationship between the rate ν and the input μ. To find a self-consistent solution for the rates ν^{S} and ν^{NS}, the above equations need to be solved numerically, taking into account in addition the f-I curve $\nu(\mu)$ of the neurons that in the case of leaky integrate-and-fire model neurons also depends on the variance σ^2 of inputs. The latter can be obtained analogous to the mean input μ (see Appendix B). Note that for general nonlinearity $\nu(\mu)$ there is no analytical closed-form solution for the fixed points.

Starting from SSN_1, networks are connected in a fixed pattern such that the rate ν_i in SSN_i also depends on the excitatory input from the previous sub-network SSN_{i-1} with rate ν_{i-1}. For a fixed point, we have $\nu_i = \nu_{i-1}$ (Toyoizumi, 2012). In this case, we can effectively group together stimulated/non-stimulated neurons in successive sub-networks and re-group equations for the mean input in the limit of many sub-networks, obtaining the simplified description (details see Appendix B.4)

$$\mu^{\mathrm{S}} = \alpha \mathcal{J} \nu_{\mathrm{x}} + \kappa_{\mathrm{S,S}}\ \nu^{\mathrm{S}} + \kappa_{\mathrm{S,NS}}\ \nu^{\mathrm{NS}} \tag{6.9}$$

$$\mu^{\mathrm{NS}} = \alpha \mathcal{J} \nu_{\mathrm{x}} + \kappa_{\mathrm{NS,S}}\ \nu^{\mathrm{S}} + \kappa_{\mathrm{NS,NS}}\ \nu^{\mathrm{NS}} \tag{6.10}$$

The scaling terms of the firing rates incorporate the recurrent and feedforward contributions from the stimulated and non-stimulated groups of neurons. They depend solely on some fixed parameters of the system, including modularity m (see Appendix B). Importantly, Eq. 6.9 and Eq. 6.10 have the same linear form as Eq. 6.8 and can be solved numerically as described above. Again, for general nonlinear $\nu(\mu)$ there is no closed-form analytical solution, but see below for a piecewise linear activation function $\nu(\mu)$. The numerical solutions for fixed points are obtained using the root finding algorithm `root` of the `scipy.optimize` package (Virtanen et al., 2020). The stability of the fixed points is obtained by inserting the corresponding firing rates into the effective connectivity Eq. 6.6 on the level of stimulated and non-stimulated sub-populations

$$\begin{pmatrix} \kappa_{\mathrm{S,S}}(m)\tilde{\alpha}(\nu^{\mathrm{S}}) & \kappa_{\mathrm{S,NS}}(m)\tilde{\alpha}(\nu^{\mathrm{NS}}) \\ \kappa_{\mathrm{NS,S}}(m)\tilde{\alpha}(\nu^{\mathrm{S}}) & \kappa_{\mathrm{NS,NS}}(m)\tilde{\alpha}(\nu^{\mathrm{NS}}) \end{pmatrix} \tag{6.11}$$

and evaluating its eigenvalues.

The structure of fixed points for the stimulated sub-population (see discussion in Section 6.2.7) can furthermore be intuitively understood by studying the potential landscape of the system. The potential U is thereby defined via the conservative force $F = -\frac{dU}{d\nu^{\mathrm{S}}} = -\nu^{\mathrm{S}} + \nu(\mu, \sigma^2)$ that drives the system towards its fixed points via the equation of motion $\frac{d\nu^{\mathrm{S}}}{dt} = F$ (Wong, 2006; Litwin-Kumar and Doiron, 2012; Schuecker et al., 2017). Note that μ and σ^2 are again functions of ν^{S} and ν^{NS}, where the latter is the self-consistent rate of the non-stimulated sub-populations for given rate ν^S of the stimulated sub-population, $\nu^{\mathrm{NS}} = \nu^{\mathrm{NS}}(\nu^{\mathrm{S}})$ (details see Appendix B.4).

6.3.7 Critical modularity and relation to recurrent connectivity, background input and single-neuron activation function

While the self-consistent equations for the fixed points cannot be solved analytically for the nonlinear activation function of spiking models, we here turn to an analytically simpler but qualitatively similar piecewise linear activation function (Figure 6.10A). The latter yields linear self-consistency equations that can be solved analytically to obtain conditions on the modularity for denoising (details see Appendix B.4.3). Successful denoising thereby requires the non-stimulated sub-populations to be silent, $\nu^{\mathrm{NS}} = 0$, and the stimulated sub-populations to be active $\nu^{\mathrm{S}} > 0$. The analysis yields multiple conditions for denoising. The first condition

$$m \geq \frac{(\mu_{\max} - \mu_{\min})N_{\mathrm{C}}}{(1-\alpha)\mathcal{J}\nu_{\max} + (\mu_{\max} - \mu_{\min})(N_{\mathrm{C}} - 1)} \tag{6.12}$$

mainly depends on parameters of the activation function, namely the dynamic range of input $\mu_{max} - \mu_{min}$ and output ν_{max}. In particular, it does not depend on the nature of the recurrent interactions and the external background input. In addition, we find further critical values of the modularity for denoising (Figure 6.10C)

$$m = \frac{N_{\mathrm{C}}}{N_{\mathrm{C}} - 1} - \frac{1}{N_{\mathrm{C}} - 1}\frac{(1-\alpha)\mathcal{J}\nu_{\max}}{\mu_{\max} - \alpha\mathcal{J}\nu_x - \frac{\mathcal{J}}{N_{\mathrm{C}}}\left(1+\gamma g\right)\nu_{\max}} \tag{6.13}$$

$$m = 1 - \frac{\left(\mu_{\min} - \alpha\mathcal{J}\nu_x - \frac{\mathcal{J}}{N_{\mathrm{C}}}\left(1+\gamma g\right)\nu_{\max}\right)}{\mathcal{J}(1-\alpha)\nu_{\max} - (N_{\mathrm{C}} - 1)\left(\mu_{\min} - \alpha\mathcal{J}\nu_x - \frac{\mathcal{J}}{N_{\mathrm{C}}}\left(1+\gamma g\right)\nu_{\max}\right)} \tag{6.14}$$

that are either upper or lower bounds depending on the strength of the external input. In the denominator of these expressions the total input, recurrent and external, is compared against the limits of the dynamic range of the neuron model. The cancellation between recurrent and external inputs in the inhibition-dominated baseline model typically yields a total input within the dynamic range of the neuron such that modularity in feedforward connections can decrease the input of the non-stimulated sub-populations to silence them and increase the input of the stimulated sub-populations to support their activity. In networks without recurrent connections or excitation-dominated connectivity, the total input is typically large as the external input is excitatory in biologically plausible settings. This leads to strong activity in both stimulated and non-stimulated populations that cannot be modulated sufficiently by topographic feedforward connections. While the analysis yields fixed points for excitatory external input also in this scenario, these fixed points are unstable due to the positive recurrent feedback in the system (see Appendix B.4.3)

6.3.8 Multiple inputs and correlation-based similarity score

In Figure 6.11 we consider two stimuli S_1 and S_2 to be active simultaneously for 10 s. Let SP_1 and SP_2 be the two corresponding SPs in each sub-network. The firing rate of each SP is estimated from spike counts in time bins of 10 ms and smoothed with a Savitzky-Golay filter (length 21 and polynomial order 4). We compute a similarity score based on the correlation between these rates, scaled by the ratio of the input intensities λ_2/λ_1 (with λ_1 fixed). This scaling is meant to introduce a gradient in the similarity score based on the firing rate differences, ensuring that high (absolute) scores require comparable activity levels in addition to strong correlations. To ensure that both stimuli are decodable where appropriate, we set the score to 0 when the difference between the rate of SP_2 and the non-stimulated SPs was $< 1\,\text{spks/sec}$ (SP_1 had significantly higher rates). The curves in Figure 6.11C mark the regime boundaries: coexisting (Co-Ex) where score is > 0.1 (red curve); winnerless competition (WLC) where score is < -0.1 (blue); winner-takes-all (WTA, grey) and where the score is in the interval $(-0.1, 0.1)$, and either $\lambda_2/\lambda_1 < 0.5$ holds or the score is 0. While the Co-Ex region is a dynamical regime that only occurs in the initial sub-networks (Figure 6.11D), the WTA and WLC regimes persist and can be understood again with the help of a potential U, which in this case is a function of the rates of the two SPs (details see Appendix B).

6.3.9 Numerical simulations and analysis

The numerical simulations for the spiking networks were conducted using NMSAT v0.2 (Duarte et al., 2017b) and NEST 2.18.0 (Jordan et al., 2019), while FNA was used for the continuous rate network (see Chapter 4). The code package to reproduce all figures is available as supplementary data of the publication (Zajzon et al., 2023), with the full code package to reproduce the simulation results available on Zenodo Zajzon et al. (2022). The complete set of parameters can be found in Appendix B.

6.4 Discussion

The presence of stimulus- or feature-tuned sub-populations of neurons in primary sensory cortices (as well as in downstream areas) provides an efficient spatial encoding strategy (Pouget et al., 1999; Seriès et al., 2004; Tkačik et al., 2010) that ensures the relevant computable features are accurately represented. Here, we propose that beyond primary sensory areas, modular topographic projections play a key role in preserving accurate representations of sensory inputs across many processing modules. Acting as a structural scaffold for a sequential denoising mechanism, we show how they simultaneously enhance relevant stimulus features and remove noisy interference. We demonstrate this phenomenon in a variety of network models and provide a theoretical analysis that indicates its robustness and generality.

When reconstructing a spatially encoded input signal corrupted by noise in a network of sequentially connected populations, we find that a convergent structure in the feedforward projections is not only critical for successfully solving the task, but that the performance increases significantly with network depth beyond a certain modularity (Figure 6.2). Through this mechanism, the response selectivity of the stimulated sub-populations is sharpened within each subsequent sub-network, while others are silenced (Figure 6.3). Such wiring may support efficient and robust information transmission from the thalamus to deeper cortical centers, retaining faithful representations even in the presence of strong noise. We demonstrate that this holds for a variety of signals, from approximately static (stepwise) to smoothly and rapidly changing dynamic inputs (Figure 6.12). Thanks to the balance of excitation and inhibition, the network can track spatially encoded signals on very short timescales and is flexible with respect to the level of spatial and temporal discretization.

More generally, topographic modularity, in conjunction with other top-down processes (Kok et al., 2012), could provide the anatomical substrate for the implementation of a number of behaviorally relevant processes. For example, feedforward topographic projections on the visual pathway could contribute, together with various attentional control processes, to the widely observed *pop-out effect* in the later stages of the visual hierarchy (Brefczynski-Lewis et al., 2009; Itti et al., 1998). The pop-out effect, at its core, assumes that in a given context some neurons exhibit sharper selectivity to their preferred stimulus feature than the neighboring regions, which can be achieved through a winner-take-all (WTA) mechanism (see Figure 6.11 and (Himberger et al., 2018)).

The WTA behavior underlying the denoising is caused by a re-shaping of the E/I balance across the network (see Figure 6.4). As the excitatory feedforward projections become more focused, they modulate the system's effective connectivity and thereby the gain on the stimulus-specific pathways, gating or allowing (and even enhancing) signal propagation. This change renders the stimulated pathway excitatory in the active regime (see Figure 6.8), leading to multiple fixed points such as those observed in networks with local recurrent excitation (Renart et al., 2007; Litwin-Kumar and Doiron, 2012). While the high-activity fixed point of such clustered networks is reached over time, in our model it unfolds progressively in space, across multiple populations. Importantly, in the range of biologically plausible numbers of cortical areas relevant for signal transmission (up to ten for some visual stimuli, see Felleman and Van Essen, 1991; Hegdé and Felleman, 2007) and intermediate modularity, the firing rates remain within experimentally observed limits and do not saturate. The basic principle is similar to other approaches that alter the gain on specific pathways to facilitate stimulus propagation, for example through stronger synaptic weights (Vogels and Abbott, 2005), stronger nonlinearity (Toyoizumi, 2012), tuning of connectivity strength and neuronal thresholds (Gajic and Shea-Brown, 2012), via detailed balance of local excitation and inhibition (amplitude gating (Vogels and Abbott, 2009)) or with additional sub-cortical structures (Cortes and van Vreeswijk, 2015). Additionally, our model also displays some activity characteristics reported previously, such as the response sharpening observed for synfire chains (Diesmann et al., 1999) or (almost) linear firing rate propagation (Kumar et al., 2010a) (for intermediate modularity).

While our findings build on the above results, we here show that the experimentally observed topographic maps may serve as a structural denoising mechanism for sensory stimuli. In contrast to most works on signal propagation where noise mainly serves to stabilize the dynamics and is typically avoided in the input, here the system is driven by a continuous signal severely corrupted by noise. Taking a more functional approach, this input is reconstructed using linear combinations of the full network responses, rather than evaluating the correlation structure of the activity or relying on precise firing rates. Focusing on the modularity of such maps in recurrent spiking networks, our model also differs from previous studies exploring optimal connectivity profiles for minimizing information loss in purely feedforward networks (Renart and van Rossum, 2012; Zylberberg, 2017), also in the context of sequential denoising autoencoders (Kadmon and Sompolinsky, 2016) and stimulus classification (Babadi and Sompolinsky, 2014), which used simplified neuron models or shallow networks, made no distinction between excitatory and inhibitory connections, or relied on specific, trained connection patterns (e.g., chosen by the pseudo-inverse model). Although the bistability underlying denoising can, in principle, also be achieved in such feedforward or networks without inhibition, our theoretical predictions and network simulations indicate that for biologically constrained circuits (i.e., where the background and feedforward / long-range input is excitatory), inhibitory recurrence is indispensable for the spatial denoising studied here (see Section 6.2.8). Recurrent inhibition compensates for the feedforward and external excitation, generating competition between the topographic pathways and allowing the populations to track their input rapidly.

Moreover, our findings provide an explanation for how low-intensity stimuli ($1-2$spks/sec above background activity, see Figure 6.3 and Supplementary Materials) could be amplified across the cortex despite significant noise corruption, and relies on a generic principle that persists across different network models (Figure 6.6) while also being robust to variations in the map size (Figure 6.7). We demonstrated both the existence of a lower and upper (due to increased overlap) bound on their spatial extent for signal transmission, as well as an optimal region for which denoising was most pronounced. These results indicate a trade-off between modularity and map size, with larger maps sustaining stimulus propagation at lower modularity values, whereas smaller maps must compensate through increased topographic density (see Figure 6.7A and Supplementary Materials). In the case of smaller maps, progressively enlarging the receptive fields enhanced the denoising effect and improved task performance (Figure 6.7C), suggesting a functional benefit for the anatomically observed decrease in topographic specificity with hierarchical depth (Bednar and Wilson, 2016; Smith et al., 2001). One advantage of such a wiring could be spatial efficiency in the initial stages of the sensory hierarchy due to anatomical constraints, for instance the retina or the lateral geniculate nucleus. While we get a good qualitative description of how the spatial variation of topographic maps influences the system's computational properties, the numerical values in general are not necessarily representative. Cortical maps are highly dynamic and exhibit more complex patterning, making (currently scarce) precise anatomical data a prerequisite for more detailed investigations.

Finally, our model relates topographic connectivity to competition-based network dynamics. For two input signals of comparable intensities, moderately structured projections allow both representations to coexist in a decodable manner up to a certain network depth, whereas strongly modular connections elicit winnerless competition (WLC) like behavior characterized by stochastic switching between the two stimuli (see Figure 6.11).

Importantly, all these different dynamical regimes emerge progressively through the hierarchy and are not discernable in the initial modules. Previous studies reporting on similar dynamical states have usually considered either the synaptic weights as the main control parameter (Lagzi and Rotter, 2015; Lagzi et al., 2019; Vogels and Abbott, 2005) or studied specific architectures with clustered connectivity (Schaub* et al., 2015; Litwin-Kumar and Doiron, 2012; Rost et al., 2018). Our findings suggest that in a hierarchical circuit a similar palette of behaviors can be also obtained given appropriate effective connectivity patterns modulated exclusively through modular topography. Although we used fixed projections throughout this study, these could also be learned and shaped continuously through various forms of synaptic plasticity (see e.g., Tomasello et al., 2018). To achieve such a variety of dynamics, cortical circuits most likely rely on a combination of all these mechanisms, i.e., pre-wired modular connections (within and between distant modules) and heterogeneous gain adaptation through plasticity, along with more complex processes such as targeted inhibitory gating.

Overall, our results highlight a novel functional role for topographically structured projection pathways in constructing reliable representations from noisy sensory signals, and accurately routing them across the cortical circuitry despite the plethora of noise sources along each processing stage.

Chapter 7

Overcoming temporal compression during sequence learning

7.1 Introduction

The previous chapters investigated how a modular connectivity can support accurate representations of discrete stimuli and continuous signals and their transmission across a series of simple cortical circuits, without accounting for temporal structure in the input. Such representations are a prerequisite for any further processing, but navigation in a dynamic environment requires actions and decisions that are precisely coordinated in time and space, matching the spatio-temporally structured stimuli upon which they are based. As discussed in Section 1.5, the ability to learn, process and predict sequential patterns is therefore a critical component of cognition, with recent experimental findings showing a multitude of brain regions to be involved in sequence processing (Dehaene et al., 2015; Wilson et al., 2018; Henin et al., 2021). Recordings in the primary visual cortex indicate that even early sensory areas are capable of learning and recalling not just the order of a series of stimulus patterns, but also the duration of the individual elements (Xu et al., 2012; Gavornik and Bear, 2014). In fact, the ability to represent both the ordinal and temporal components of a sequence are two of the most fundamental requirements for any system processing sequential information.

However, as reviewed in Section 3.3.3, most existing models of unsupervised biological sequence learning address only the first of these two criteria, focusing on acquiring the order of elements and typically failing to account for their duration. They either cannot intrinsically represent the time intervals (Klos et al., 2018; Bouhadjar et al., 2022), or they assume a fixed and identical duration for each element that is limited by the architecture (Maes et al., 2020), or they produce longer sequences that arise spontaneously even in the absence of structured input (and hence are not related to it, Fiete et al., 2010).

Seeking to unify these computational features, Cone and Shouval (2021) recently proposed a novel, biophysically realistic spiking network model that avoids the problem of temporal compression while maintaining the precise order of elements during sequence replay. Relying on a laminar structure, as well as experimentally observed cell properties, the system uses a local, eligibility-based plasticity rule (3-factor learning rule, see Section 2.3) to learn the order of elements by mapping out a physical path between stimulus-tuned columns, with the duration of each item being encoded in the recurrent activations within the corresponding column. The learning rule, based on the competition between two eligibility traces and a globally available reward signal, is grounded in recent experimental findings (He et al., 2015; Huertas et al., 2016). This modular architecture allows the network to flexibly learn and recall sequences of up to eight elements with variable length, but only with simple transitions between items (first-order Markovian). More intricate sequences with history dependence (i.e., higher-order Markovian) can be learned, but require additional structures for memory. Given the increased complexity, this ability is only demonstrated in a continuous rate-based model.

The code for the model is available in *MATLAB*. As this is a proprietary, closed-source software, models expressed in this manner have accessibility issues (not every scientist can afford a license) and bear a greater risk of becoming non-executable legacy code, if the code is not regularly maintained (for an example, see to Brinke et al., 2022). Additionally, as *MATLAB* is a general purpose numeric computing platform, the researcher must develop all neuroscientific models and simulation algorithms *de novo*, which presents a higher risk for implementation errors and poorly-suited numerics (Pauli et al., 2018).

In this chapter we therefore present a replication of the original study, which serves the twin purpose of testing the original findings and providing a more accessible version of the model to the computational neuroscience community. Specifically, we re-implement their model in NEST using FNA (see Chapter 4), thus ensuring an open source, reusable and maintainable code base.

Here, we use the term *replication* in the R^5 sense described by Benureau and Rougier (2018), i.e. striving to obtain the same results using an independent code base, whereas a *reproduction* (R^3) of the model would have been achieved if we had obtained the results of the original study using the original code. However, others have argued these terms should be used the other way around: see Plesser (2018) for an overview and analysis.

7.2 Methods

The model analyzed in this chapter is described in full detail in the original work of Cone and Shouval (2021). Nevertheless, given that we found numerous discrepancies between the model description and implementation, we present all the key properties and parameters that are necessary for a successful replication of the results.

7.2.1 Network architecture

The central characteristic of the network architecture is the modular columnar structure (see Figure 7.1A, B), where each of the N_C columns is associated with a unique sequence

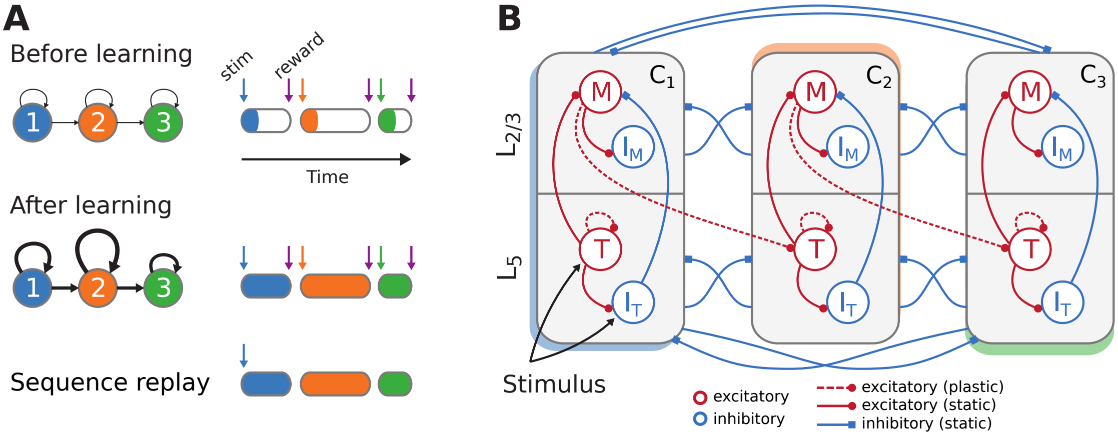


Figure 7.1: **Sequence learning task and network architecture. (A)** A sequence of three intervals (elements) is learned by a network with as many dedicated populations (columns). The individual populations are stimulated sequentially, with a global reward signal given at the beginning and the end of each element. After training, the recurrent and feedforward weights are strengthened, and the sequence is successfully recalled following a cue. The fullness of the colored sections on the right illustrates the duration of the activity (firing rates) above a certain threshold. **(B)** Each stimulus-specific column is composed of two excitatory, *Timers* (T) and *Messengers* (M), and two corresponding inhibitory populations, I_{T} and I_{M}. Solid (dashed) arrows represent fixed static (plastic) connections. Cross-columnar inhibition always targets the excitatory population in the corresponding layer (L_5 or $L_{2/3}$).

element (stimulus). Each column contains two excitatory (Timer and Messenger) and two associated inhibitory populations $I_{\mathrm{T}}$and I_{M}, roughly corresponding to L_5 and $L_{2/3}$ in the cortex. In the following, we will refer to these cell populations as T^i, M^i, I^i_{T} and I^i_{M}, respectively, where the superscript i denotes the column C_i.

Each of the above populations is composed of $N = 100$ leaky integrate-and-fire neurons, with the exception of the network simulated in Section 7.3.4, where $N = 400$. The wiring diagram of the baseline network used in Cone and Shouval (2021) is schematically illustrated in Figure 7.1B. Within a column C_i, T^i cells connect to I^i_{T} and M^i, in addition to recurrent connections to other T^i cells. M^i neurons excite the local inhibitory population I^i_{M}, and are inhibited by I^i_{T}. Inhibition onto the excitatory cells also exists between the columns in a layer-specific manner, i.e., $I^i_{\mathrm{T}} \rightarrow T^j$ and $I^i_{\mathrm{M}} \rightarrow M^j$, with $i \neq j$. Lastly, M^i cells in C_i connect in a feedforward manner to T^{i+1} cells in the subsequent column C_{i+1}. All connections within the same and between different populations have a density of $\varphi = 0.26$. Note that only the feedforward projections $M^i \rightarrow T^{i+1}$ and the recurrent $T^i \rightarrow T^i$ connections are subject to plasticity (see below); all other connections are static. The plastic weights are initialized close to 0 and the static weights are normally distributed around their mean values with a standard deviation of 1.

The complete set of parameters for the architecture proposed in the original study, as well as the variants described below are specified in the Appendix C.

Scaled model

For the scaled network model described in Section 7.3.4, the number of neurons in each populations was increased to $N' = 400$ from $N = 100$. To keep the input variance constant, in the standard scaling scenario (Figure 7.6A) we followed the common approach for balanced random networks (van Vreeswijk and Sompolinsky, 1998; Litwin-Kumar and Doiron, 2012) and reduced all non-plastic synaptic weights by multiplying them with $1/\sqrt{N'/N}$. In addition, we halved the standard deviation σ_ξ of the background noise such that the firing rates were in the same range as for the baseline network. To restore the functional aspects of the network, additional tuning was required for most of the projections, see Table C.4.

All-to-all cross-columnar connectivity

In Section 7.3.5, the baseline network is modified by instantiating plastic excitatory connections between all columns $M^i \rightarrow T^j$, $(i \neq j)$ rather than solely between the columns representing consecutive elements of the stimuli (see Figure 7.7A). All other parameters are unchanged.

Alternative wiring with local inhibition

The functionally equivalent network analyzed in Section 7.3.6 required multiple modifications (see Figure 7.8A). Inhibitory connections are local to the corresponding layer, with connections $I^i_\mathrm{T} \rightarrow T^i$ and $I^i_\mathrm{M} \rightarrow M^i$. Timer cells T^i project to both M^i and I^i_M, as well as to I^j_T in other columns C_j. In layer $L_{2/3}$, M^i cells project to T^{i+1} and I^j_M, $i \neq j$.

7.2.2 Neuron model

The networks are composed of leaky integrate-and-fire (LIF) neurons, with fixed voltage threshold and conductance-based synapses. The dynamics of the membrane potential V_i for neuron i follows:

$$C_\mathrm{m} \frac{dV_i}{dt} = g_\mathrm{L} \left(V_\mathrm{rest} - V_i(t) \right) + I^\mathrm{E}_i(t) + I^\mathrm{I}_i(t) + \xi(t) \tag{7.1}$$

where the leak-conductance is given by g_L, I^E_i and I^I_i represent the total excitatory and inhibitory synaptic input currents, and ξ is a noise term modeled as Gaussian white noise with standard deviation $\sigma_\xi = 100$, unless otherwise stated. This noise term is sufficient to cause a low baseline activity of around $1 - 2$spks/sec. Upon reaching a threshold $V_\mathrm{th} = -55\,\mathrm{mV}$ ($-50\,\mathrm{mV}$ for inhibitory neurons), the voltage is reset to V_reset

for a refractory period of $t_{\text{ref}} = 3$ ms . Note that the higher threshold for inhibitory neurons is critical for the faster decay of their activity compared to Timer cells.

The dynamics of the synaptic conductances are modeled as exponential functions with an adaptation term, with fixed and equal conduction delays for all synapse types. The equations of the model dynamics, along with the numerical values for all parameters are summarized in Appendix C.

In all figures depicting firing rates, these are estimated from the spike trains using an exponential filter with time constant $\tau_{\text{r}} = 40\,\text{ms}$.

7.2.3 Eligibility-based learning rule

The main assumption of the learning rule is the availability of two synaptic eligibility traces at every synapse T^p_{ij} and T^d_{ij}, representing long-term potentiation (LTP) and depression (LTD), which can be simultaneously activated through the Hebbian firing patterns.

For $a \in \{p, d\}$, the dynamics of the traces follows:

$$\tau^a \frac{dT^a_{ij}(t)}{dt} = -T^a_{ij}(t) + \eta^a H_{ij}(t) \left(T^a_{\max} - T^a_{ij}(t)\right), \tag{7.2}$$

where τ^a is the time constant, η^a is a scaling factor, and $T^a_{\max}$ is the saturation level of the trace. $H_{ij}(t)$ is the Hebbian term defined as the product of firing rates of the pre- and postsynaptic neurons:

$$H_{ij}(t) = \begin{cases} r_i(t) r_j(t) & \text{if } r_i(t) r_j(t) > r_{\text{th}} \\ 0 & \text{otherwise} \end{cases}, \tag{7.3}$$

with r_{th} ($r^{\text{ff}}_{\text{th}}$) representing different threshold values for recurrent T to T (feedforward M to T) connections. Note that while this equation is used in both the original MATLAB implementation and in our re-implementation in NEST, the Hebbian terms in the equations in Cone and Shouval (2021) are further normalized by $T^a_{\max}$. For a detailed analysis of the learning convergence, see the original study.

These activity-generated eligibility traces are silent and transient synaptic tags that can be converted into long-term changes in synaptic strength by a third factor, $R(t)$ which is modeled here as a global signal using a delta function, $R(t) = \delta(t - t_{\text{reward}} - d_{\text{reward}})$, and is assumed to be released at each stimulus onset/offset. Although typically signals of this sort are used to encode a *reward*, they can also, as is the case here, be framed as a *novelty* signal indicating a new stimulus. Hence, the synaptic weights w_{ij} are updated through

$$\frac{dw_{ij}}{dt} = \eta R(t) \left(T^p_{ij} - T^d_{ij}\right) \tag{7.4}$$

where η (η_{ff} for feedforward) is the learning rate. Following the reward signal, which has a duration of 25 ms, the eligibility traces are "consumed" and reset to zero, and their activation is set into a short refractory period of 25 ms. In practice, although the weight updates are tracked and evolve during each reward period according to Eq. 7.4, they are only updated at the end of the trial. However, this does not affect the results in any significant manner (data not shown).

7.2.4 Stimulation protocol

Stimulus input is modeled as a 50 ms step signal, encoded as Poisson spike trains with a rate $\nu_{\text{in}} = 30\,\text{spks/sec}$. In the baseline and the extended network discussed in Section 7.3.5, this input is injected into both T^i and I_{T}^i cells, with synaptic weights w_{in}. In the network discussed in Section 7.3.6, the input is restricted to T^i.

The training process of a network instance consists of 100 trials (unless otherwise stated), and in each trial the corresponding columns are stimulated at certain time points according to the input sequence, with the interval between elements representing the duration of the stimulus. At the beginning of each trial, the state of the neurons (membrane potential) and the eligibility traces are reset to their initial values. The test phase consists of multiple trials (usually 50), where the sequence is replayed upon a cued stimulation of the first column.

7.2.5 Numerical simulations and analysis

All numerical simulations were conducted using the Functional Neural Architectures (FNA) toolkit v0.2.1 (Duarte et al., 2021), described in Chapter 4. To ensure the reproduction of all the numerical experiments and figures presented in this study, and abide by the recommendations proposed in Pauli et al. (2018), we provide a complete code package that implements project-specific functionality within FNA (see Appendix C) using NEST 2.20.0 (Fardet et al., 2020). For consistency checks with the reference implementation, we used *MATLAB* version R2020b.

7.3 Simulation results and reproducibility analysis

To investigate how temporal sequences of variable durations can be acquired by cortical circuits, Cone and Shouval (2021) propose a chain-like modular architecture where each population (module) is tuned to a specific element in the sequence, and learning translates to modifications of the synaptic weights within and between modules, based on reward signals. The model is schematically illustrated in Figure 7.1A. Following a training period where the modules are stimulated in a particular order over multiple trials, the network should be able to recall/replay the complete sequence from a single cue.

If learning was successful, both the order and duration of the elements can be recalled faithfully.

Initially, each module exhibits only a transient activity in response to a brief stimulus (50ms, see Section 7.2), as the connections are relatively weak. The duration of each sequence element is marked by a globally available reward signal, forming the central component of a local reinforcement learning rule based on two competing, Hebbian-modulated eligibility traces (Huertas et al., 2016). This synapse-specific rule is used to update the weights of both recurrent and feedforward connections, responsible for the duration of and transition between elements, respectively. After learning, these weights are differentially strengthened, such that during a cued recall the recurrent activity encodes the current element's extent, while the feedforward projections stimulate the module associated with the next sequence element.

The modules correspond to a simplified columnar structure roughly mapping to L2/3 and L5 in the cortex. The columns are composed of two excitatory populations, *Timer* (T) and *Messenger* (M), and two associated inhibitory populations I_{T} and I_{M} (Figure 7.1B). Timer cells learn to represent the duration through plastic recurrent connections, while Messenger cells learn the transitions to the column associated with the next sequence element. Note that, unless otherwise mentioned, feedforward projections exist only between columns corresponding to consecutive items in the input sequence. In other words, the sequence transitions are physically traced out from the onset, only the weights are learned (see also Section 7.4). Cross-inhibition between the columns gives rise to a soft winner-take-all (WTA) behavior, ensuring that only one column dominates the activity.

7.3.1 Sequence learning and recall

This modular architecture allows the system to robustly learn and recall input sequences with variable temporal spans. Figure 7.2A depicts the population responses before and after the network has learned four time intervals, $500, 1000, 700$ and 1800ms (see also Figure 3 in Cone and Shouval, 2021). At first, stimulation of one column produces a brief response, with initial transients in the stimulated Timer and L_5 inhibitory cells I_{T} (see Figure 7.2A, top panel and inset). With the inhibitory firing rate decaying faster than the Timers' due to a higher threshold and lack of recurrence (see Section 7.2), there is a short window when the net excitation from the Timer cells elicit stronger responses from the Messenger cells.

During training, when each column is stimulated sequentially, the recurrent Timer projections are strengthened such that their responses extend up to the respective reward signal (green vertical bars). At the same time, the feedforward projections from the Messenger cells on to the next column are also enhanced, such that upon recall (stimulation of first column), they are sufficient to trigger a strong response in the corresponding Timer cells. This chain reaction allows a complete replay of the original sequence, pre-

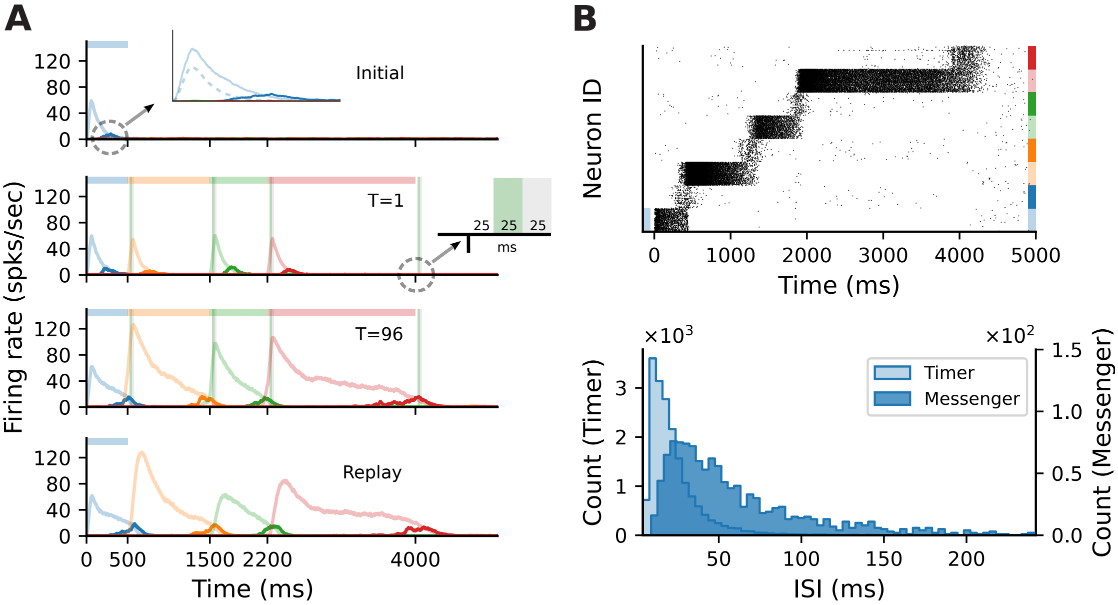


Figure 7.2: **Learning to replay a simple sequence without temporal compression. (A)**: Firing rates of the excitatory populations during learning (top three plots) and recall (bottom plot) of four time intervals $(500, 1000, 700, 1800\text{ms})$. Light (dark) colors represent T (M) cells. Dashed light blue curve in top panel inset shows the inhibitory population I_{T} in L_5. Green (grey) vertical bars show the 25ms reward (trace refractory) period, 25ms after stimulus offset (see inset). **(B)**: Spiking activity of excitatory cells (top) and corresponding ISI distributions (bottom), during recall, for the network in **(C)**. In the raster plot, neurons are sorted by population (T, M) and sequentially by column (see color coding on the right).

serving both the order and intervals. The activity propagation during recall is illustrated in Figure 7.2B (see Figure 3S4 in Cone and Shouval, 2021). The network displays realistic spiking statistics (coefficient of variation of 1.35 and 0.95 for Timer and Messenger cells), with Messenger cells having lower firing rates than Timer cells, roughly consistent with the experimentally observed values (Liu et al., 2015).

7.3.2 Learning and recall precision

The model exhibits fluctuations in the learning process and recall accuracy of sequences as a consequence of noise and the stochastic nature of spiking networks. For sequences of intermediate length, the recall times typically vary within ±10-15% of the target duration (see Figure 7.3A, left). However, this range depends on several parameters and generally increases with duration or sequence length (see Figure C.1). Nevertheless, averaged over multiple network instances, these effects are attenuated and learning becomes more precise (Figure 7.3A, right).

These fluctuations can also be observed at the level of synaptic weights. Whereas

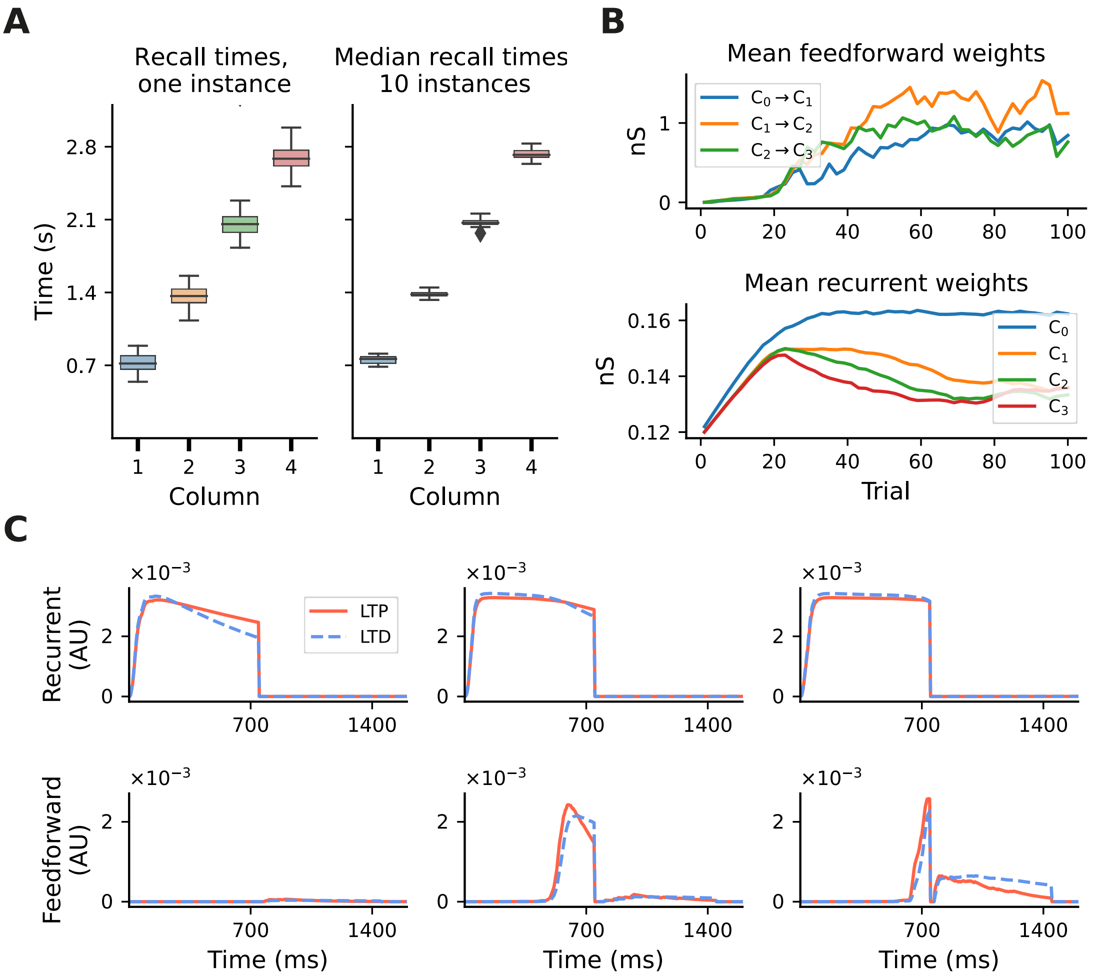


Figure 7.3: **Accuracy of recall and evolution of learning.** Results shown for a sequence of four intervals of 700ms. **(A)**: Fluctuations in learning and sequence recall. We define *recall time* as the time at which the rate of the Timer population drops below 10 spks/s. Left: recall times for 30 trials after learning, for one network instance. Right: distribution of the median recall times over 10 network instances, with the median in each network calculated over 30 replay trials. **(B)**: Mean synaptic weights for feedforward (Messenger to Timer in subsequent columns, top) and recurrent (Timer to Timer in the same column, bottom) connections for one network instance. **(C)**: Mean LTP and LTD traces for the recurrent (top) and feedforward (bottom) connections, for learning trials T= 3, T= 15 and T= 35 and one network instance.

the recurrent weights in the Timer populations converge to a relatively stable value after about 70 trials (see Figure 7.3B, bottom panel, and Figure 3S2 in Cone and Shouval, 2021), the feedforward weights display a larger variability throughout training (top panel). For the recurrent connections, convergence to a fixed point in learning can be formally demonstrated (see proof in Cone and Shouval, 2021). As a Hebbian learning rule (see Section 7.2), the two competing LTP and LTD eligibility traces are activated upon recurrent activity in the Timer population. Assuming that both traces saturate quickly, with a slightly higher LTD peak, and given a larger time constant for the LTP trace, the LTD trace will decay sooner, resulting in the facilitation of recurrent synapses during the reward period (Figure 7.3C, top panel). Learning converges when the net difference between the two traces is zero at the time of reward.

For the feedforward weights, an analytical solution is more difficult to derive. Due to Hebbian co-activation of Messenger cells and Timer cells in the subsequent module, the traces are activated (non-zero) shortly before the reward period, temporarily reset following reward, and reactivated during the next trial (Figure 7.3C, bottom panel). The net weight change is thus the sum of trace differences over two subsequent reward periods. Empirically, learning nevertheless tends to converge to some relatively stable value if feedforward projections only exist between columns coding for subsequent input elements. However, because the reward signal is globally available at each synapse, all projections from a Messenger population to any other module could, in theory, be facilitated, as long as there is some temporal co-activation. We elaborate on this aspect in Section 7.3.5.

7.3.3 Model robustness

Although formally learning convergence is only guaranteed for the recurrent Timer connections, Cone and Shouval (2021) report that in practice the model behaves robustly to variation of some connectivity and learning parameters. However, the range of parameter values and sequence lengths they analyzed (see their Figure 5 and supplements) does not give a complete account of the parameters' influence and the model's limits. To test model robustness more thoroughly, we varied a number of the synaptic weights and learning parameters beyond those considered in the original work and measured the consistency in the recall times of a sequence composed of four 700ms intervals.

First, we varied the excitatory and inhibitory projections onto Messenger cells within a column, in an interval of ±20% of their baseline value. This is the range explored in Cone and Shouval (2021, see their Figure 5), but only qualitative results of the population activities were reported and only for a subset of all possible combinations. In the baseline network, on average 17 out of 50 reported recall times were off by ±140 ms (or 20% of correct interval) when measured relative to their expected onset time, whereas these values varied between 15 and 22 for the tested parameter configurations (see Figure 7.4A, top left). Averaged across all four columns, the outliers decreased to a range between 11-15 (Figure 7.4A, bottom left). Next, we used a modified z-score based on the median absolute deviation (Iglewicz and Hoaglin, 1993) to evaluate the distribution of the absolute recall times (not relative to their expected onset).

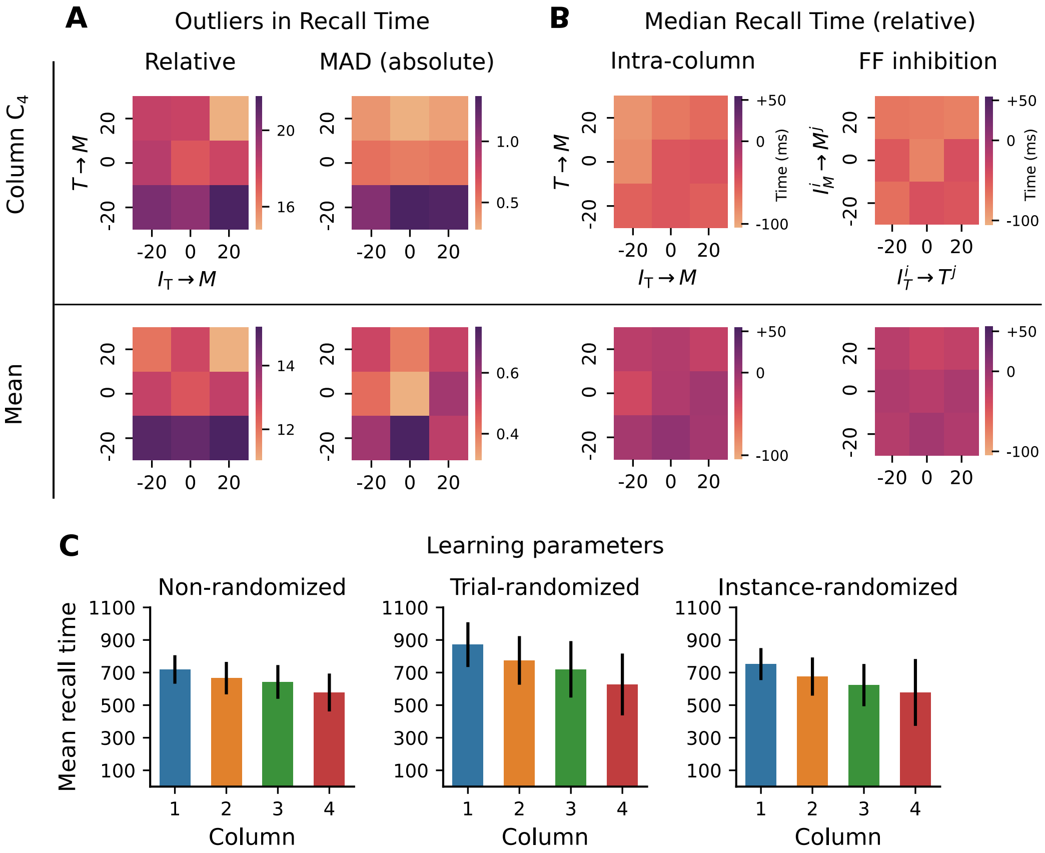


Figure 7.4: **Robustness to variation in synaptic weights and learning parameters.** Model trained on a sequence of four elements, each with a duration of 700ms. For the Timer cells, we define *relative recall time* as the recall time relative to stimulation onset, i.e., the time from the expected onset time $(0, 700, 1400, 2100)$ in the sequence until the rate drops below a threshold of 10spks/s. Conversely, *absolute recall time* is simply the time when the rate drops below threshold (relative to 0). **(A)**: Number of outlier intervals reported during 50 recall trials, as a function of the percentage change of two synaptic weights within a column: excitatory Timer to Messenger, and inhibitory $I_\mathrm{T}$to Messenger. Top row shows the number of outliers, defined as a deviation of ±140 ms from the correct interval relative to expected onset (left), and the number of outliers detected using a modified z-score (threshold > 3, right panel) based on the median absolute deviation in column C_4 (see main text). Bottom row shows the respective outliers averaged over all four columns. **(B)**: Deviation of the median recall time from the expected 700ms, as a function of the excitatory and inhibitory synaptic weights onto the Messenger cells in a column (left), and as a function of the cross-columnar $(C_i \neq C_j)$ inhibitory synaptic weights within the same layers (right). Top and bottom row as in **(A)**. All data in **(A)** and **(B)** is averaged over 20 network instances. **(C)**: Mean recall time of a four-element sequence of 700ms intervals, over 50 recall trials of a single network instance. Left: baseline network. Center: during each training trial, the learning parameters (see main text) are drawn randomly and independently from a distribution of ±20% around their baseline value. Error bars represent the standard deviation. Right: the set of learning parameters is drawn randomly once for each network instance, with data shown averaged over 10 instances.

These were centered closely around the mean recall time in each column, with the number of outliers decreasing significantly to below 1.5 (3% of recall trials, Figure 7.4A, right). These results suggest that the recall times are relatively consistent for each column (narrowly distributed), but the absolute deviations from the expected values increase with the element's position in the sequence.

In other words, the errors and variability accumulate with sequence length, with the network being particularly sensitive to the weaker excitatory connections from Timer onto Messenger cells (see $\Delta w = -20\%$ for $T \rightarrow M$). In fact, these errors manifest in recalling increasingly shorter intervals (Figure 7.4B, left), with the last column reporting on average close to 600 ms instead of 700 ms. Averaged across all columns, the median recall time is more accurate. Similar results are obtained for variations in the inhibitory projections between columns (Figure 7.4B, right).

The model displays similar robustness to variations in the eligibility trace time constants (τ^p, τ^d, τ^p_{ff}, τ^d_{ff}) and the variables scaling the Hebbian contribution to the trace dynamics (η^p, η^d, η^p_{ff}, η^d_{ff}, see Section 7.2). Whereas in the original work, this analysis was performed with a sequence of two elements of 500 ms each (see Figure 5 - supplement 1 in Cone and Shouval, 2021), here we use a sequence of four 700 ms elements. Compared to the baseline network (Figure 7.4C, left), where the median recall time decays only slightly with sequence length, randomizing the learning parameters in each learning trial not only increases the median recall time across all columns, but it also leads to greater variability in the replayed sequences (Figure 7.4C, center). Randomizing the learning parameters once per network instance does, on average, lead to results closer to the baseline case, but further increases the recall variability in the last column (Figure 7.4C, right - analysis not performed in Cone and Shouval, 2021).

These results demonstrate that the system copes well with intermediate perturbations to the baseline parameters with respect to the afferent weights for the Messenger population, the cross-columnar inhibition and the learning rule variables.

While the Timer and Messenger cells are responsible for maintaining a sequence element in the activity and signaling the onset of subsequent ones, the dynamics of the inhibitory populations orchestrate the timing of the individual components. For example, through their characteristic activity curve, the inhibitory cells in L_5 simultaneously control the activity of the Messenger cells in their own column and the onset of the Timer populations in the next column. By modifying the synaptic weight from the Timer cells to the inhibitory population in their column ($w_{T \rightarrow I_{\text{T}}}$), and thus controlling direct excitation, we sought to understand how these inhibitory cells impact learning.

For values significantly lower than baseline ($< -25\%$, grey area in Figure 7.5A), the network fails to recall sequences in a reliable manner (Figure 7.5B), in particular sequences containing more than two elements. In addition, the recall times vary significantly across the columns in the case of reduced weights. As the weights increase, the stronger net excitation causes longer-lasting inhibition by I_{L_5}, delaying the activation of the Messenger cells (Figure 7.5C). This leads to an over-estimation of the elements'

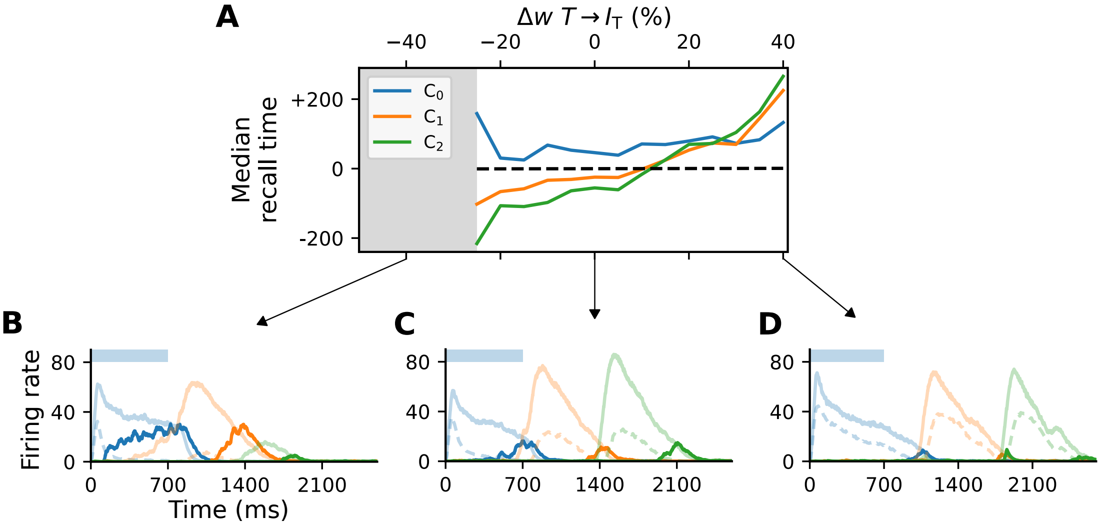


Figure 7.5: **Activity of L$_5$ inhibitory population is critical for accurate learning. (A)**: Deviation of the median recall time of three intervals of 700ms, as a function of the change in synaptic weights $T \rightarrow I_{\mathrm{T}}$ relative to baseline ($\Delta w = 0$). Grey area ($< -25\%$) marks the region where learning is unstable (not all elements can be recalled robustly). Data is averaged over 5 network instances. **(B-D)**: Characteristic firing rates during recall for values deviations of -25, 0 and 40% relative to baseline. Solid curves represent the excitatory populations as in Figure 7.2, while dashed curves indicate the respective inhibitory populations I_{T}in C_i.

duration, which increases with the element's position in the sequence (up to +200ms for $\Delta w_{T \rightarrow I_{\mathrm{T}}} = 40\%$, Figure 7.5D).

Although these observations suggest a robust learning mechanism, they also indicate an intrinsic and consistent bias of the model for reporting increasingly shorter intervals and larger variability in the recall times of longer sequences.

7.3.4 Model scaling

In the previous section, we investigated the sensitivity of the model to the choice of synaptic weights, but a broader definition of robustness also encompasses invariance to the size of the different populations. Ideally, the model should retain its dynamical and learning properties also for larger network sizes, without the need for manual recalibration of the system parameters. In balanced random networks, increasing the network size by a factor of m and decreasing the synaptic weights by a factor of $\sqrt{m}$ should maintain the activity characteristics (van Vreeswijk and Sompolinsky, 1998; Litwin-Kumar and Doiron, 2012; van Albada et al., 2015). The model studied here differs significantly from these systems with respect to features such as the ratio of excitation and inhibition (1:1, not 4:1), or strong recurrent connectivity in the small N regime, which results in

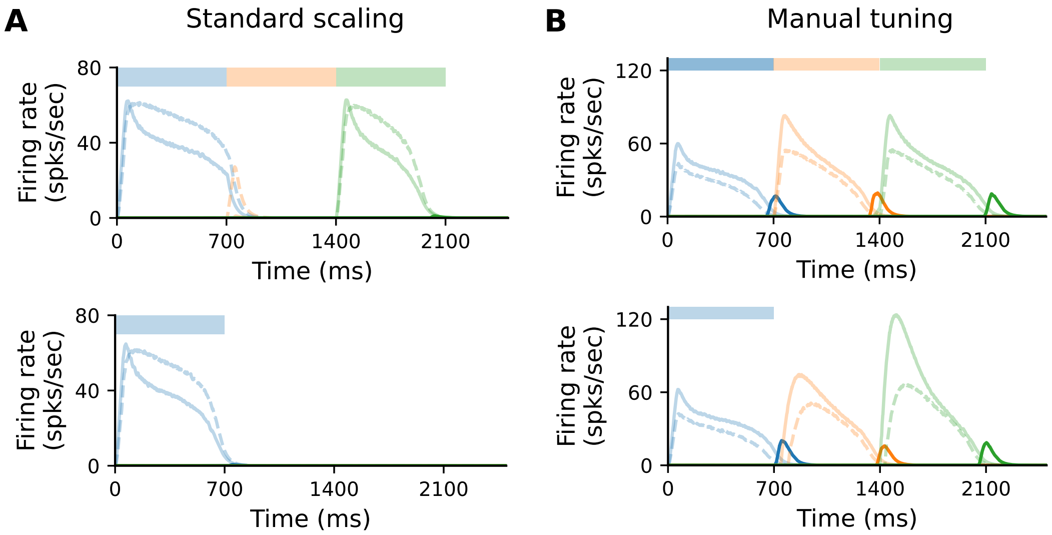


Figure 7.6: **Scaling the model requires manual retuning of parameters. (A)**: Characteristic firing rates during training (top) and recall (bottom) of a sequence composed of three 700ms intervals, in a larger network where each population is composed of $N' = 400$ cells. All static weights have been scaled down by $1/\sqrt{N'/N}$ (see Section 7.2). Solid curves show Timer (light) and Messenger (dark) cells, dashed curves I_{T}cells. **(B)**: As in (A), with further manual tuning of specific weights. For details, see Section 7.2 and Table C.4.

significant fluctuations driven by noise. Furthermore, the stereotypical activation patterns underlying sequence learning and replay are significantly more complex. These considerations suggest that successful scaling may require additional modifications of the connectivity.

In the original formulation of the model, each population (Messenger, Timer, inhibitory) consists of 100 neurons. To study how well the model scales for $N' = 400$, we kept all parameters unchanged and scaled all non-plastic weights by $1/\sqrt{N'/N}$ (see Table C.4). Under such standard scaling, the system fails to learn and recall sequences (Figure 7.6A), primarily due to the high firing rates of I_{T} cells. These decay slower than the corresponding Timer cells, inhibiting the Timer population in the subsequent column and thus prohibiting a correct sequential activation during training.

Nevertheless, it is possible to find a set of parameters (see Section 7.2 and Table C.4) for which learning unfolds as expected; this is illustrated in Figure 7.6B. The critical component here is the activity of I_{T} (see also Figure 7.5). This must fulfill three criteria: first, it must decay slightly faster than the rate of the Timer population in the same column; second, it must sufficiently inhibit the Timer populations in all other columns to enable a WTA dynamics; third, the WTA inhibition of the Timer populations must be weak enough that they can still be activated upon stimulation. One way to achieve this is by further decreasing the local weights $w_{T \to I_{\mathrm{T}}}$within a column and the cross-columnar inhibition $w_{I^i_{\mathrm{T}} \to T^j}$. This indicates that, given the right set of parameters, the dynamics underlying the learning process are independent of the network size. Although it is outside the scope of this work, scaling can be likely achieved for a wider range of model sizes, as long as the core properties described above are retained.

7.3.5 Projections between all columns

In the original implementation of Cone and Shouval (2021), and in contrast to the description in the paper, excitatory projections between columns were only allowed in a feedforward manner, thus hard-wiring the order of the sequence elements. Since such a predetermined and stimulus-dependent connection pattern weakens the model's claims of biological plausibility, we probed the model's ability to learn when this constraint was relaxed.

To this end, we extended the baseline network with additional projections from Messenger cells in column C_i to Timer cells in all other columns C_j, $(i \neq j)$ as depicted in Figure 7.7A. As the weights of these projections are initialized close to 0, no further measures were necessary to maintain the same activity level as the baseline network. Although learning initially proceeded as before, the activity soon lost its stereotypical temporal structure and the learning process is corrupted (Figure 7.7B). After only a few dozen trials, the activation order of the columns did not match the stimulation, with multiple populations responding simultaneously. Such random, competitive population responses also continued throughout the recall trials.

This behavior arises because projections from the Messenger cells to all columns are incorrectly strengthened, not just between subsequent ones reflecting the order of the input sequence. Figure 7.7C illustrates such an example, with synaptic weights from Messenger cells in C_2 to all other columns C_j being equally strengthened, instead of only to C_3. Naturally, this effect is detrimental because Messenger cells can activate multiple Timer populations at once, introducing a stochasticity in the network that abolishes the unique sequential activation required for accurate learning and recall. In other words, the physical pathway encoding the transitions between sequence elements can not be uniquely traced out as in the baseline network.

According to the Hebbian-based plasticity rule (see Section 7.2), synaptic weights are modified during the reward period only if there is a co-activation of the pre- and post-synaptic neurons. This means that connections from M cells in a column C_i to T cells in any C_j may be strengthened if there is temporal co-activation of the two populations. While this is the intended behavior for subsequent columns C_i and C_{i+1}, Timer cells in other columns may also spike due to the background noise, thereby enhancing the corresponding connections. Obviously, in the pre-wired (baseline) network this is not an issue, as only subsequent columns are connected.

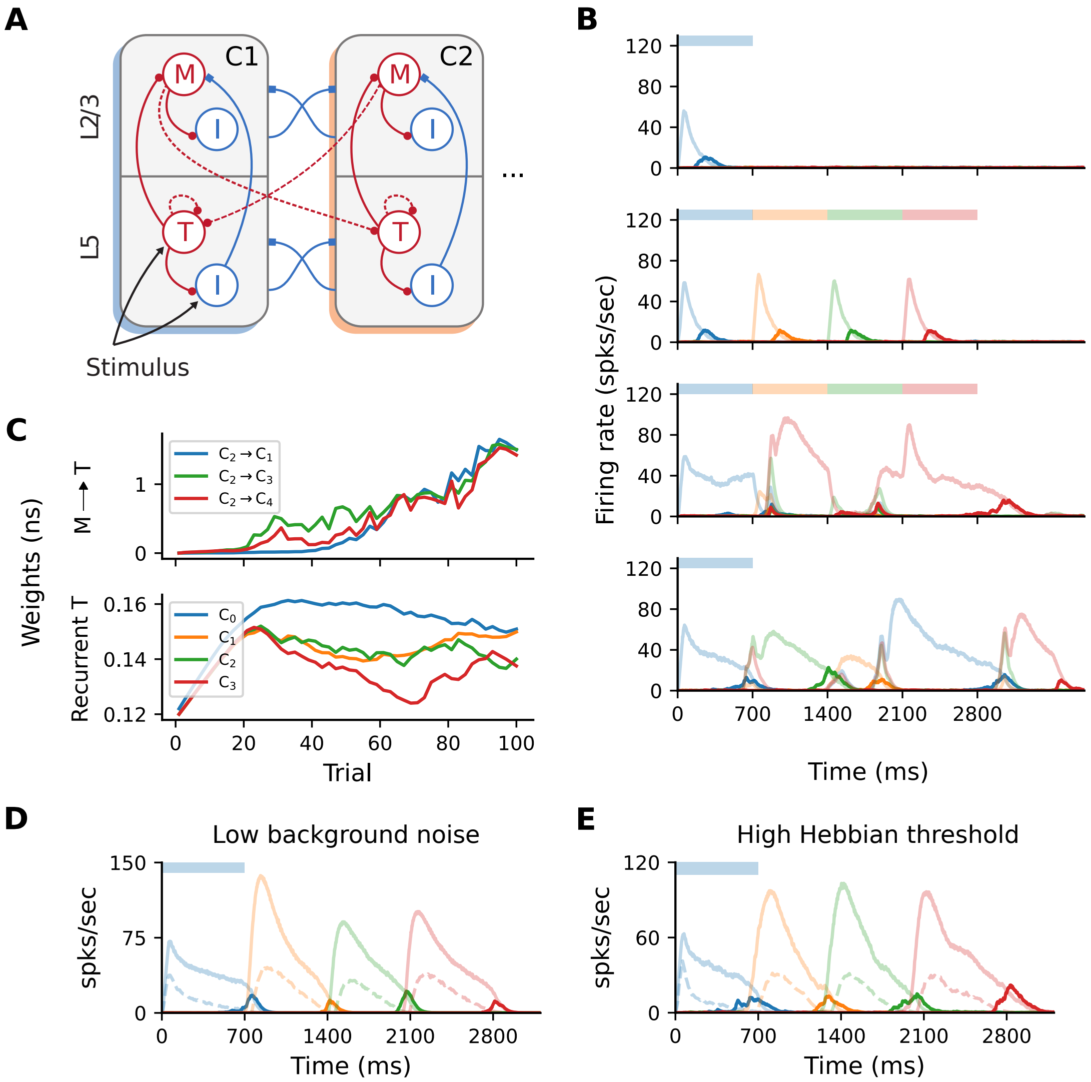


Figure 7.7: **All-to-all cross-columnar excitation prohibits learning. (A)**: Extending the original architecture described in Figure 7.1B, $M \rightarrow T$ connections exist between all columns $C_i \rightarrow C_j$ $(i \neq j)$ and are subject to the same plasticity. **(B)**: Firing rates of the excitatory populations during learning and recall of four time intervals (each 700 ms). Initially, learning evolves as in Figure 7.2A, but the activity becomes degenerated and the sequence can not be recalled correctly (lower panels). **(C)**: Evolution of the cross-columnar (from C_2, top panel) and recurrent Timer synaptic weights (bottom panel). The transition to the next sequence cannot be uniquely encoded as the weights to all columns are strengthened. **(D)**: Sequence recall after 100 training trials in a network with a low background noise (50% of the baseline value, $1/2\sigma_\xi$). **(E)**: Sequence recall after 100 training trials in a network with a higher Hebbian activation threshold for the cross-columnar projections $r_{\mathrm{th}}^{\mathrm{ff}} = 30$ spks/sec (instead of the baseline 20 spks/sec).

One straightforward solution to overcome this problem is to reduce the background noise below the spiking threshold, thereby ensuring that only the stimulated populations are active and no “cross-talk” occurs through spurious spiking. Doing so allows the network to regain its functional properties (Figure 7.7D), pending some minor additional parameter tuning (see Section 7.2). However, from the point of view of biological plausibility, this has the disadvantage that neurons spike exclusively during their preferred stimulus.

Alternatively, it is possible to compensate for the low-rate spontaneous spiking by raising the activation threshold for the Hebbian term, $r_{\rm th}^{\rm ff}$ (see Section 7.2). For instance, increasing from the baseline value of 20 to 30 spks/sec is sufficient to ensure that only the stimulated populations reach these rates. Thus, only synapses between stimulated populations are modified, and the learning process is not affected (Figure 7.7E). The role and plausibility of such thresholds is detailed in the Discussion.

7.3.6 Alternative wiring with local inhibition

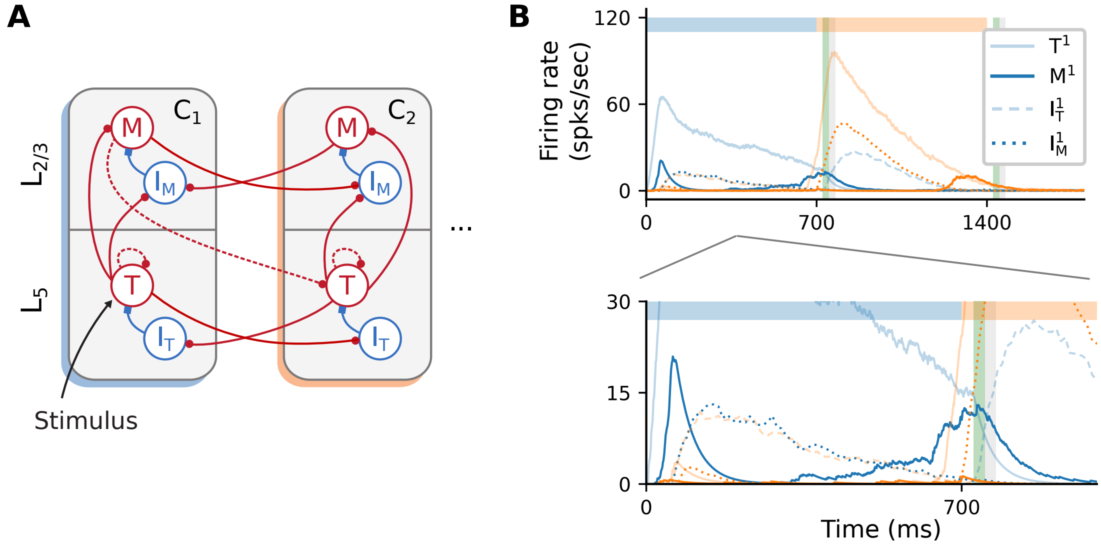


Figure 7.8: **Alternative wiring with local inhibition and only excitatory cross-columnar projections.** **(A)**: Architecture with local inhibition functionally equivalent to Figure 7.1B. Inhibitory projections are now local to the column, and feedforward inhibition is achieved via cross-columnar excitatory projections onto the I populations. **(B)**: Recall of a sequence composed of two 700ms intervals. Inset (bottom panel) zooms in on the activity at lower rates. As before, color codes for columns. Color shade represents populations in L_5 (light) and $L_{2/3}$ (dark), with solid curves denoting excitatory populations. Dashed (dotted) curves represent the inhibitory cells $I_{\rm T}$ ($I_{\rm M}$).

Unlike cortical circuits, where inhibition is assumed to be local (Douglas and Martin, 2004; Fino and Yuste, 2011; Tremblay et al., 2016), the original architecture described in Figure 7.1B relies on (long-range) inhibitory projections between columns to ensure a soft WTA mechanism in the presence of background activity. This aspect is briefly discussed in Cone and Shouval (2021), and the authors also propose an alternative, biologically more plausible and functionally equivalent network architecture (see their Figure 9). As schematically illustrated in Figure 7.8A, cross-columnar inhibition can be replaced by local inhibition and corresponding excitatory projections onto these circuits. In contrast to the baseline network, where both Timer and inhibitory cells in L_5 were stimulated, here only Timer cells received input. Otherwise, excitation onto I_T would soon silence the Timer cells, prohibiting the longer timescales required for encoding the input duration.

As a proof-of-concept, we empirically derived a set of parameters (see Table C.5) for such a circuit and found that the core network dynamics and learning process can, in principle, be retained (Figure 7.8B). However, a significant discrepancy from the baseline behavior concerns the initial transient of the Messenger cells in the first column C_1 (solid, dark blue curve in Figure 7.8B, bottom panel). This occurs because inhibition onto the Messenger cells from I_M (dotted, dark blue curves) is slower (due to higher firing threshold) than the excitation from the Timer cells. This results in a brief period of higher Messenger activity before inhibition takes over and silences it. Although this behavior is different from the baseline model, it does not appear to impact learning, and it is in fact consistent with the experimental data from the primary visual cortex (Liu et al., 2015).

7.4 Discussion

Given that the ability to learn and recall temporal sequences may be a universal functional building block of cortical circuits, it is paramount that we understand how such computational capacities can be implemented in the neural substrate. While there have been numerous approaches to model sequence processing in spiking networks, many of these are either unable to capture important functional aspects (e.g., order and duration of sequences), or rely on biophysically unrealistic assumptions in their structure or learning rules. In this work, we investigated a recent model proposed by Cone and Shouval (2021), which attempts to overcome these weaknesses.

Since here we focused particularly on the reproducibility and replicability aspects, our work provides only limited improvements over the original model. Thus, major modifications such as changes to the learning rule or the evaluation of more complex sequence learning tasks are beyond the scope of our study. However, by re-implementing the model in the NEST simulator, we were able to qualitatively replicate the main findings of the original work, find some of the critical components and assumptions of the model, and highlight its strengths and limitations. More importantly, we provide a complete set of parameters and implementation details for a full replication of the model. As computational studies are becoming increasingly significant across many scientific disciplines, ease of reproduction and replication becomes an ever more important factor, not just to allow efficient scientific progress, but also to ensure a high quality of the work. These points are well illustrated by a notable outcome of this study: as a result of our findings, the authors of the original study have corrected their article (Cone and Shouval, 2023) and modified their published code to enable full replication and correct the inconsistencies and errors discovered in their work (see updated repository on ModelDB), as listed below.

Critical parameters		
Name	**Value**	**Description**
$V_{\mathrm{th}}^{\mathrm{I}}$	-50 mV	Spiking threshold for inhibitory neurons ⊘
r_{th}	10 Hz	Hebbian activation threshold (recurrent connections) ⊘
$r_{\mathrm{th}}^{\mathrm{ff}}$	20 Hz	Hebbian activation threshold (feedforward connections) ⊘
T_{max}^{p}	0.0033	Saturation level of LTP trace (recurrent connections) ⊛
T_{max}^{d}	0.00345	Saturation level of LTD trace (recurrent connections) ⊛
$T_{\mathrm{max}}^{p,\mathrm{ff}}$	0.0034	Saturation level of LTP trace (feedforward connections) ⊛
$T_{\mathrm{max}}^{d,\mathrm{ff}}$	0.00345	Saturation level of LTD trace (feedforward connections) ⊛
η^{p}	45×3500 ms^{-1}	Activation rate of LTP trace (recurrent connections) ⊛
η^{d}	25×3500 ms^{-1}	Activation rate of LTD trace (recurrent connections) ⊛
η_{ff}^{p}	20×3500 ms^{-1}	Activation rate of LTP trace (feedforward connections) ⊛
η_{ff}^{d}	15×3500 ms^{-1}	Activation rate of LTD trace (feedforward connections) ⊛
$\tau_{\mathrm{syn}}^{\mathrm{exc,inp}}$	10 ms	Excitatory synaptic time constant of the input connections ⊘

Table 7.1: **Critical parameters necessary for accurate learning.** Symbols denote different discrepancy types: ⊘ represents parameters not mentioned in the study, and ⊛ parameters with only relative but no exact values given.

7.4.1 Reproducibility

The original model is described in Cone and Shouval (2021), with most parameters provided as Supplementary Information, along with a publicly available MATLAB implementation on ModelDB[1]. However, while the results are reproducible using the provided implementation in the R^3 sense described by Benureau and Rougier (2018), a successful

[1] http://modeldb.yale.edu/266774

Parameter values required for numerical reproducibility		
w_{in}	100 nS	Weights of input connections $\odot$
σ_ξ	$\mathcal{N}(0, 100)$	Gaussian white noise in the neuron model $\odot$
d_{reward}	25 ms	Delay of reward signal relative to the onset of the next sequence element $\oslash$
$\tau_{\text{syn}}^{\text{exc}}$	80 ms	Excitatory synaptic time constant (EE and IE) within the network $\diamond$
$\tau_{\text{syn}}^{\text{inh}}$	10 ms	Inhibitory synaptic time constant (EI) $\diamond$
τ_{ref}	3 ms	Refractory period $\diamond$
φ	0.26	Connection density for all connections (including recurrent) $\diamond$
ν_{in}	30 Hz	Rate of the Poisson input $\diamond$
η	0.16	Learning rate for recurrent connections $\diamond$
η_{ff}	20	Learning rate for feedforward connections $\diamond$

Table 7.2: **Parameter values needed for obtaining numerically similar results to those reported in Cone and Shouval (2021).** Symbols $\oslash$ and $\circledast$ as in Table 7.1. Additionally, $\odot$ denotes parameters with no specific values given, while $\diamond$ denotes a mismatch between the values reported in the paper and the ones used in the reference implementation.

replication in the R^5 sense would not have been possible based solely on the information in the manuscript and Supplementary Tables, given that a number of parameters are either under-specified or omitted entirely. Table 7.1 and Table 7.2 give an overview of the more important discrepancies between the description and original implementation, categorized by their relevance and type of mismatch.

Table 1 lists omitted (or inaccurately stated) critical parameters, i.e. those that are necessary for the model to carry out the computational tasks that are central to the original study. Such oversights are particularly problematic, as they not only make replication more challenging but also make implicit model assumptions opaque. An illustrative example of an omitted critical parameter is the spiking threshold for the inhibitory neurons, V_{th}, which is 5 mV higher than the threshold for the excitatory neurons. This is important, as it results in the inhibitory rates decaying slightly faster than the Timer cells, thus activating the Messenger cells at the appropriate time. In the absence of this dynamical feature, learning fails (see for example Figure 7.6A). While there is some experimental evidence for such a difference in the spiking threshold, it varies significantly across different cell types and recording locations (Tripathy et al., 2015). Similarly, the activation thresholds for the Hebbian learning, r_{th} , are necessary to ensure that spontaneous spiking resulting from the neuronal noise does not lead to potentiation of unwanted synapses, in particular if connections between all columns are allowed (see Figure 7.7). Without such thresholds, learning still converges in the baseline network, but the fixed point of the feedforward weights is shifted, stabilizing at a lower value than in the baseline system (see Figure C.2). Therefore, the role and optimal value for the thresholds likely depend on the amount of noise and spontaneous activity in the network.

A further example is the parameterization of the eligibility traces. Whereas the time constants of the eligibility traces determine their rise and decay behavior, the saturation levels $T^{\alpha}_{\mathrm{max}}$ can profoundly impact learning. For the Timer cells, although their exact values (not provided in the original work) are not essential, the order of magnitude is still critical; they must be carefully chosen to ensure that the traces saturate soon after stimulus onset, and the falling phase begins before the next reward period (see also Huertas et al., 2015). In other words, even though the parameter space is underconstrained and multiple values can lead to accurate learning (Huertas et al., 2016), these nevertheless lie within a restricted interval which is difficult to determine given only the relative values as in the original work: for instance, a value of $T^{\mathrm{d}}_{\mathrm{max}} = 1$ and $T^{\mathrm{d}}_{\mathrm{max}} = 0.95$ will lead to an abrupt increase in the recurrent Timer weights and learning fails. If the traces do not saturate, learning becomes more sensitive to the trace time constants and the range of time intervals that can be learned with one set of parameters shrinks significantly. Moreover, the excitatory input synapses have a shorter time constant of 10 ms than in the rest of the network, which is required for the fast initial ramp-up phase of the Timer cell activity.

Table 7.2 summarizes other, less critical parameters, which are nonetheless necessary to achieve qualitatively similar activity levels to those presented in the original work. These include input related parameters (input weights, input rate), as well as the neuronal noise. Whereas some of these discrepancies are due to omission (e.g., noise) or mismatch between the reported and used values (e.g., learning rate), others arise from tool- and implementation particularities. For instance, for $N = 100$ the random number generation in MATLAB results in an effective connectivity $\varphi =\sim 0.26$ instead of the 0.3 reported in Cone and Shouval (2021), while the effective refractory period is 3 instead of 2 ms, as threshold crossings are registered with a delay of one simulation step. Although these parameters influence the activity level in the network, they do not directly impact the learning process; the key computational features claimed for the model are maintained.

7.4.2 Learning cross-columnar projections

One of the key properties of the model is the ability to learn the order of temporal sequences, achieved by learning the transitions between stimulus-specific populations encoding the sequence elements. However, Cone and Shouval (2021) state that "Messenger cells can only learn to connect to (any) Timer cells outside of their column", which we interpret as an assertion that Timer cells make connections to Messenger cells in all other columns. In practice, the authors' reference implementation restricts these to subsequent columns only. This means that the order of the sequence is hardwired into the connectivity, and the system is only learning the duration of the elements. As we demonstrated in Section 7.3.5, with the baseline parameters the network fails to learn if this restriction is relaxed and feedforward projections are indeed allowed between any columns.

A simple way to circumvent this problem is to ensure that neurons outside the populations coding for the current stimulus remain completely (or sufficiently) silent, to avoid the co-activation necessary for Hebbian synaptic potentiation (see Figure 7.7D). Although such an idealized behavior may be an appropriate solution from a modeling perspective, neurons in the cortex are rarely tuned exclusively to particular stimuli. Instead, most cells spike irregularly (typically at a low rate) even in the absence of input (ongoing activity, see e.g., Arieli et al., 1996), and many respond to multiple different inputs (Walker et al., 2011; Rigotti et al., 2013; de Vries et al., 2020).

A biologically more plausible alternative is to increase the Hebbian activation threshold r_{th} , such that noise-induced spontaneous activity does not lead to a modification of the synaptic strength. However, this introduces an additional, critical parameter in the model. Furthermore, such hard thresholds are coupled to the intensity of background activity and spontaneous spiking, with occasional higher rates possibly destabilizing the learning process.

7.4.3 Functional and neurophysiological considerations

As argued in Section 1.5 and Chapter 3, a generic model of sequence processing should not only replay simple sequences, but also be able to perform chunking and handle non-adjacent dependencies in the input (Fitch and Martins, 2014; Wilson et al., 2018; Hupkes et al., 2019). Although Cone and Shouval (2021) discuss and provide an extension of the baseline network for higher-order Markovian sequences, the computational capacity of the model is fundamentally limited by the requirement of a unique stimulus-column (or stimulus-population) mapping. This characteristic means that for certain tasks, such as learning (hierarchical) compositional sequences (i.e., sequences of sequences), the model size would increase prohibitively with the number of sequences, as one would require a dedicated column associated with each possible sequence combination. In addition, it would be interesting to evaluate the model's ability to recognize and distinguish statistical regularities in the input in tasks such as chunking, which involve one or more sequences interleaved with random elements.

In their study, Cone and Shouval (2021) demonstrate that the extended, rate-based network can learn multiple, higher-order Markovian sequences when these are presented successively. For first-order Markovian sequences, this should also hold for the baseline spiking network model, contingent on preserving the unique stimulus-to-column mapping. However, it is also important to understand how the model behaves when two sequences are presented *simultaneously.* This depends on the interpretation and expected behavior, and to the best of our knowledge there is little experimental and modeling work on this (but see, e.g., Murray and Escola, 2017). Nevertheless, if the two sequences are considered to be *independent*, we speculate that the networks will not be able to learn and treat them as such for multiple reasons. Assuming that projections between all columns are allowed (with the appropriate measures, see Section 7.3.5), in the spiking model the connections between the columns associated with the different sequences would also be strengthened upon temporal co-activation: for two simultaneously initiated sequences S1 and S2, the cross-columnar projections between a column $C_{\mathrm{i}}^{\mathrm{S1}}$ associated with S1 and another column $C_{\mathrm{i+1}}^{\mathrm{S2}}$ coding for an element at position $i+1$ in S2 would be (incorrectly) strengthened. In the case of the extended rate network, the context representations may mix and interfere in the external reservoir, and the issue of temporal co-activation discussed above is also likely to occur.

Moreover, convergence of learning in the cross-columnar synapses depends on the existence of two consecutive reward periods. As described in Section 7.3.2 and illustrated in Figure 7.3C, during the first reward (associated with the current sequence element) the weights are potentiated, even after the weights have reached a fixed point. However, a second reward, during which the weights are depressed, is necessary to achieve a net zero difference in the LTP and LTD traces at lower weight values. Although learning would converge even without a second reward, the fixed point will be different (higher), and thus convergence would occur for larger weights (possibly too large for stable firing rates). Given that the reward (novelty) signal is globally released both before and after each sequence element in the interpretation of Cone and Shouval (2021), the existence of a reward after the final element is guaranteed and therefore this is not an issue for the stimulation protocol used in the original and our study. If, on the other hand, we interpret the reward as a novelty signal indicating the next stimulus, we would not expect it to be present in this form after the last element of the sequence. In this case, the cross-columnar projections marking the transition from the penultimate to the ultimate element may not be learned accurately (weights would still converge, but likely to larger values than appropriate).

While a solution to the above issues is beyond the scope of this work, we speculate that a more granular architecture, in which multiple stimulus-specific sub-populations could form different cell assemblies within a single column, would be more in line with experimental evidence from the neocortex. Some functional specialization of single cortical columns has been hypothesized (Mountcastle, 1997; Harris and Shepherd, 2015), but such columns are typically composed of a number of cell groups responsive to a wider range of stimuli. We assume that mapping the model to such an extended columnar architecture would require a more complex, spatially-dependent connectivity to ensure similar WTA dynamics.

As we demonstrated in Section 7.3.6, the model is relatively flexible with respect to the precise wiring patterns, as long as certain core, inhibition-related properties are preserved. Given that long-range projections in the neocortex are typically excitatory

(Brown and Hestrin, 2009; Douglas and Martin, 2004), the original architecture (see Figure 7.1B) was implausible due to its reliance on cross-columnar inhibition. The relative ease in adapting the wiring to have only local inhibition is indicative of simple yet powerful and modular computational mechanisms, suggesting that these may be used as building blocks in more complex sequence learning architectures.

Despite these limitations and sensitivity to some parameters, the model presented by Cone and Shouval (2021) is an important step towards a better understanding of how cortical circuits process temporal information. While its modular structure enabling spatially segregated representations may be more characteristic for earlier sensory regions, the proposed local learning rule based on rewards, partially solving the credit assignment problem, is a more universal mechanism likely to occur across the cortex.

Chapter 8

Benchmarking and critical evaluation of biologically-plausible models of sequence learning

8.1 Introduction

In the original study by Cone and Shouval (2021) that we replicated in the previous chapter, the authors investigated only one sequence with the spiking version, as well as two more complex, context-dependent inputs using the simplified rate-based model. While this may be sufficient to illustrate the ability of a model to learn specific sequences or individual rules, it does not say much about its limitations and capacity to learn multiple interlinked rules that are part of a generative process. As argued in Chapter 3, models of sequence processing should be evaluated and benchmarked within a formal framework that bridges theoretical and experimental findings, such as AGL.

This chapter, focusing on the acquisition, representation and use of sequential dependencies in biologically-inspired models, aims to provide the initial steps towards such a benchmarking framework. From a computational perspective, it is imperative that the ability of models to represent structural regularities in complex symbolic sequences is assessed in a systematic and quantifiable manner. More importantly, such a framework can facilitate the comparison of existing and future models, with the potential to shed light on the underlying mechanisms (architecture and learning algorithms) that can then be corroborated with experimental findings.

Building on Chapter 3, here we introduce the theoretical concepts and tools constituting the framework, as well as some of the computational aspects and tasks that should be considered by any general sequence processor. As a proof-of-concept, we then use the framework to compare some of the more prominent biologically-compatible models of sequence learning proposed in the last decade. The tools and results presented in this chapter will form an integral part of a larger meta-modeling study where we re-implement and re-test a variety of existing models.

8.2 Model selection

We consider sequence learning models that are not only derived from clear neurobiological inspiration but also preserve a certain degree of biophysical faithfulness. From the zoo of spiking models reviewed in Section 3.3, we select three for which an implementation in NEST exists or has been completed as part of this work: the (CS) model be Cone and Shouval (2021) analyzed in Chapter 7; a model (KM) based on dynamic assemblies and WTA circuits proposed by Klampfl and Maass (2013); and the spiking Hierarchical Temporal Memory (spkTM) with dendritic processing introduced by Bouhadjar et al. (2022).

Although a detailed description of these models is outside the scope of this chapter, Table 8.1 and Table 8.2 succinctly summarize their key architectural and learning properties. These cover a wide range of biological features, including columnar and WTA networks, dendritic processing and sequentially activated cell assemblies, as well as unsupervised and reward-modulated plasticity rules. As such, they employ different stimulus encoding mechanisms and are designed to perform only partially overlapping sets of tasks. Common to all three models are learning the order of varyingly complex sequences and replaying them upon a cued signal. Although we will only focus on the former one here, these represent only a small subset of tasks that more general sequence processing systems should, ideally, be able to perform (see Chapter 9 for a more detailed discussion).

Note that the spiking version of the CS model was designed for deterministic sequences obeying the Markov property, as explicitly stated in the original study. To handle context dependence, the model requires additional structures for memory which are not straightforward to achieve in spiking networks. Nevertheless, we include the spiking model here because of its numerous biophysically plausible features while taking into account this limitation in our discussions.

8.2.1 Implementation considerations

An important technical aspect of the larger meta-modeling study is the availability of all models in a common, established neural simulator such as NEST. While spkTM was originally developed and implemented in NEST, the other models had to be re-implemented. Although this is often a painful and time-consuming process, there are several advantages to this approach. It enables a shared code-base for running and evaluating all models, efficient simulations for complex and longer tasks, and, as demonstrated in Chapter 7, it may help uncover any hidden assumptions or implementation peculiarities in the original codebase.

The re-implementation of CS has been thoroughly tested and the results published (see Chapter 7). Given that no publicly available implementation of the KM model exists,

Model	Architecture	Neuron types	Synaptic transmission
CS	• Modular network comprising a number of stimulus-selective, laminar microcircuits ("core neural architecture", CNA) • Sparse, specific within- and between columnar connectivity	• Noisy LIF • Excitatory and inhibitory "Timer" and "Messenger" cell sub-populations in each CNA	• Conductance-based with exponential decay • Depressing (self-limiting) synapses
KM	• Reservoir of WTA circuits, composed of unconnected excitatory neurons • Sparse, distance-dependent connectivity between circuits	• Excitatory LIF with stochastic (Poissonian) firing • Symbolic (divisive) inhibition	• Alpha-shaped postsynaptic potentials (EPSP)
spkTM	• Sparse random connectivity between excitatory neurons (plastic) • Local recurrent connectivity between excitatory and inhibitory neurons (static)	• Excitatory neurons: LIF with nonlinear input integration (dendritic action potentials) • Inhibitory neurons: LIF	• Exponential or alpha-shaped postsynaptic currents (PSCs)

Table 8.1: Summary of architectural, neuronal and synaptic features of the selected models.

we achieved only a qualitative replication of the original results based on the source code provided in private correspondence with the authors. While this was crucial even for the partial replication, it has also uncovered some inconsistencies between the model parameters and equations provided in the publication and the actual implementation. These include higher firing rates in the spike patterns (5 instead of 3 Hz), different STDP equations, and spatially-dependent connectivity at the level of WTA circuits and not single neurons (see Michau, 2022). Due to the higher number of patterns used here, we also increased the upper limit of the WTA size from 10 to 25, a value that appeared to be used to the multi-stimuli tasks in the provided implementation. Additional inconsistency in the model parameters for the different tasks, together with the absence of plotting routines for the figures, meant that while we could qualitatively reproduce the stimulus-specific sequential activation patterns for individual patterns, the results for context-dependent representations (e.g., Figure 9 in Klampfl and Maass, 2013) are inconclusive. Because of this, our findings with respect to this model should be interpreted with the appropriate caution.

Furthermore, the homeostatic plasticity in spkTM is disabled in all tasks. This mechanism relies on a time constant that depends on the sequence length, which was fixed in the original study (Bouhadjar et al., 2022). Choosing an optimal value for the sequences

Model	Learning rules	Input encoding	Functional Goal
CS	• Reward-gated, multiplicative Hebbian learning active on recurrent and feedforward connections • Same rule but different parameters for timing (recurrent) and transition (feedforward)	• 50 ms pulsed Poissonian spike train with fixed rate • Spatial mapping on unique, stimulus-specific sub-populations	• Learning and cued replay of first-order deterministic Markovian sequences (higher-order sequences with the rate-based version) • Token duration, serial order and cued replay
KM	• Short-term plasticity on recurrent synapses • Truncated STDP on all synapses	• Spatio-temporal spike patterns overlaid on top of background noise • Patterns are presented to all neurons (all-to-all)	• Emergence of stimulus-specific, dynamic / sequential activation patterns • Context-dependent interlinking of activation patterns • Spontaneous and cued replay
spkTM	• Homeostatic spike-timing dependent structural plasticity in excitatory-to-excitatory connections	• Presentation of sequence element modeled by single spike • Spatial mapping on unique, stimulus-specific sub-populations (columns)	• Sequence prediction and replay in a context dependent manner

Table 8.2: Summary of learning rules, input encoding and functional goal of the tested models.

of varying lengths considered here requires careful consideration and is beyond the scope of this work. We therefore bypass the problem with the remark that future iterations should devote more time to it, given its possible role for the stability of learning.

8.3 Model evaluation and benchmarking

As described in Chapter 3, sequence learning, and in particular symbolic sequence processing, has a rich history in the fields of cognitive science, psycholinguistics, and theoretical computer science due to its significant role in cognition.

We adopt the same convention as throughout this thesis and focus on finite symbolic sequences of length T denoted as $S_T = \sigma_1, \sigma_2, \ldots, \sigma_T$, where the sequences consist of $|\Sigma|$ distinct symbols (tokens) selected from a finite alphabet Σ. To gauge the effectiveness of different model systems as general and versatile sequence processors, we examine several relevant properties.

8.3.1 Pattern perception

The first requirement that a sequence learner must possess is the ability to perceive, represent and recognize the (often variable and noisy) spatiotemporal regularities and patterns that constitute the tokens or elements of a sequence. This needs to be done robustly and flexibly, as the system ought to generalize beyond the token identity.

The initial set of tasks aims to quantify the ability of the different models to perceive patterns in dynamic input signals with different spatiotemporal properties. Given that the different models are equipped with different signal encoding mechanisms, the quality and robustness of internal representations can drastically vary among different systems.

Note that, since these measurements refer to token-level properties, it is indifferent if there is an internal structure to the sequence, or if we are just looking at a completely random sequence of elements. However, because the structure of the sequence (or lack of it) may affect the learning process and implicitly the internal token representations, for these tasks we compare the results for random and ordered sequences.

Stimulus Encoding: Individual tokens σ_i are abstract entities, which can represent any static or time-varying input (see also Section 4.2.1 and Section 5.2.4). Ideally, the manner in which individual stimuli are encoded in the system's dynamics (i.e. the accuracy of internal representations) ought to be robust to variations in the encoding properties. However, the characteristics of this initial mapping largely determine how accurately can the system *perceive* the stimuli as unique, identifiable tokens. Generically, a discrete symbolic sequence can be *unfolded* in time as:

$$u(t) = \rho_u^k \left(\hat{\mathbf{u}}_n \times \delta(t - n\Delta) \right) * g \tag{8.1}$$

where ρ_u^k determines the amplitude or intensity of stimulus k, $\hat{\mathbf{u}}_n$ is a stimulus feature vector (most frequently a one-hot encoded representation), Δ is the token period and g is a fixed kernel.

Stimulus timing: Δ specifies the temporal regularity of the sequence unfolding and comprises the stimulus duration Δ_{stim} and the inter-stimulus-interval Δ_{ISI} defined in ms. In periodic sequences, these are fixed parameters, whereas aperiodicity can be introduced by varying Δ_{stim} and/or Δ_{ISI}. Our first set of tasks sets out to evaluate the ability of the systems to recognize sequence elements (classification accuracy) depending on their duration and intervals. Unless otherwise specified, we set Δ_{stim} and Δ_{ISI} to their default (original) values, which are 50 and 450 ms for CS , 300 and 0 ms for KM , and 1 (amounting to a single spike) and 40 ms for spkTM . Note that real-world sequences have a lot of variability in the distribution of element durations.

Tasks: We consider the following token-level tasks:

Token representation The first and simplest task is to classify the identity of each token at the end of its presentation. Input sequences are drawn from the alphabet $\Sigma = \{A,B,C,D\}$, and are presented in batches of four either in a random order (e.g., ACBC or BDCA) or as one ordered sequence ABCD. To quantify the influence of stimulus timing on the representations, we vary Δ_{stim} and Δ_{ISI} during both learning (training) and testing phases. To evaluate spkTM on this task, we modify its input encoding from a single spike to Poisson spike trains with constant rate $\nu_{\text{spkTM}} = 50$ Hz and duration Δ_{stim} .

Passive memorization To probe how long a token can be recognized after it was presented, the same alphabet Σ is used with the input sequences consisting of four randomly chosen tokens. Recognition performance is then evaluated at increasing time delays t_{samp} after the stimulus offset Δ_{stim} , which is selected fairly for each model. Δ_{ISI} is varied between $0 - 500$ ms to additionally quantify any possible effects it may have on the model behavior.

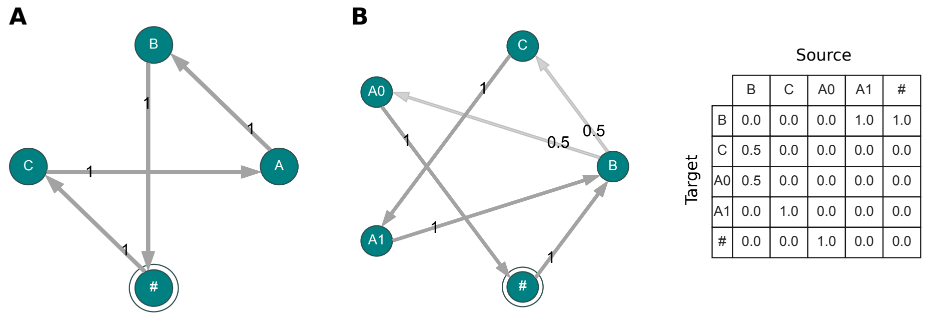
Time-warp invariance If a model learns to represent the elements of a sequence, ideally it should recognize these even if their temporal properties Δ are slightly altered. We measure such time-warp invariance by multiplying the respective Δ_{stim} or Δ_{ISI} with a fixed factor during the test phase. For CS and spkTM (note the rate-based encoding here), this involves modifying the extent of the Poisson input (but not its amplitude), whereas for the KM each spike in the input pattern is simply shifted by the warping factor. Similarly to the token representation task, both random and ordered inputs are used.

Token capacity The last token level task aims to measure how many different tokens can the model accurately distinguish. Given that this is strictly determined by the architecture for CS and spkTM , the task is only evaluated for KM on both random and ordered sequences for an increasing number of distinct tokens.

8.3.2 Rule learning

In addition to simple representation of tokens, a sequence learner must acquire the rules according to which the observed sequences are generated. To quantify this, we probe each model's ability to memorize, distinguish and predict the elements of a large number of sequences that have varying but systematic complexity.

We consider a string (sequence) S_i as an element of (a potentially infinite) formal language, generated by a (finite-state) grammar $\mathcal{G}$ as introduced in Section 3.1.



Target \ Source	B	C	A0	A1	#
B	0.0	0.0	0.0	1.0	1.0
C	0.5	0.0	0.0	0.0	0.0
A0	0.5	0.0	0.0	0.0	0.0
A1	0.0	1.0	0.0	0.0	0.0
#	0.0	0.0	1.0	0.0	0.0

Figure 8.1: **Generating complex grammars in a controlled manner.** Grammar alphabet contains three non-terminal symbols (letters) and one terminal symbol (#). **(A):** Transition diagram of a grammar with no ambiguities and one initial state (C). **(B):** Transition diagram (left) and probability transition matrix (right) of a grammar with one initial state (B), one ambiguous symbol (A, $a = 1$) and an ambiguity depth of two ($d = 2$). Ambiguous states A0 and A1 have an additional index that is not part of the symbol. All outgoing transitions from a state have equal probability. Note that the state corresponding to the terminal is an absorbing one, and all grammatical strings are required to end there. The transitions from this state only denote the initial states, but are required in the transition matrix for computing the grammar complexity (Warren and Schroeder, 2015).

To create grammars with a desired complexity in a controlled manner, we introduce two additional ambiguity variables that regulate the memory (of previous states) required for valid transitions (to subsequent states): the number of ambiguous states a and the ambiguity depth d. The first specifies how many states can occur in different contexts, or equivalently, how many symbols from the alphabet are repeatable. The ambiguity depth represents the number of distinct instances of an ambiguous state, each of which can appear in its own context.

Figure 8.1 illustrates two example grammars without and with ambiguity. Following the Markovian-style notational convention used in Warren and Schroeder (2015), the symbols are depicted in the nodes (states). The ambiguous states have an additional subscript denoting their unique index (ambiguity depth), and transition probabilites are written on the edges. Note that the subscripts are necessary to disambiguate the grammar, but they are not part of the symbol. The language defined by the grammar with no ambiguity contains only one word CAB, whereas the one with ambiguity has an infinite cardinality and can be defined as $\mathcal{L} = \{(\mathrm{BCA})^n\mathrm{BA} \mid n \geq 0\}$, and includes words like BA and BCABA.

Topological entropy: As a measure for grammar complexity, we use the information-theoretic Topological Entropy (TE) introduced by Robinson (1998) and applied to AGs by Bollt and Jones (2000). In these works, an AG is treated as a dynamical system that can generate an infinite number of sequences from a finite set of symbols, with the TE metric calculated using Markov chain analysis methods. For a grammar $\mathcal{G}$ that has s_n unique strings of length n, the TE is defined as

$$h(\mathcal{G}) = \lim_{n\to\infty} \frac{\log_e s_n}{n}. \tag{8.2}$$

Because s_n tends to increase exponentially as a function of the string length n, taking the logarithm yields an approximately linear dependence of n (Warren and Schroeder, 2015). As such, TE measures the "[exponential] growth rate of the number of [unique] words of length n as n goes to infinity" (Robinson, 1998), quantifying the intuitive premise that the complexity of an AG increases with the number of unique grammatical strings it can produce.

The TE of a grammar $\mathcal{G}$ can be computed from its topological (boolean) transition matrix M, obtainable through binarization of the probability transition matrix (see e.g. Figure 8.1B). If N is the number of unique (indexed) states in $\mathcal{G}$, then M is an $N \times N$ matrix satisfying the Markov property, with its TE defined as

$$h(\mathcal{G}) = \log_e(\lambda_1), \tag{8.3}$$

where λ_1 is the largest real eigenvalue of M (always positive). In this study we use the subscript approach for handling ambiguities (Warren and Schroeder, 2015), but in principle one can avoid indexed states and build the transition matrix using a "lifting technique" as proposed by Bollt and Jones (2000). This method involves starting with the single-symbol transition matrix, and finding the shortest n-gram length l (the lift) for which the transition matrix between all unique l-grams captures all possible transitions in $\mathcal{G}$. Not only does this method yield a much larger matrix $N^l \times N^l$ for a l-lift, but it is also very difficult to accurately calculate it in practice (Warren and Schroeder, 2015).

In order to accurately calculate the TE, the topological matrix should satisfy a number of conditions. Formulated concisely, all states must be reachable from at least one initial state $s \in S_{\text{in}}$, and there should be a path of finite length from each state to (one of) the terminal state(s). Adding a loop from the terminal to all initial states ensures that these constraints are obeyed, and additionally guarantees that the loop cycles indefinitely generating infinitely many sequences. Note that while we use # as the terminal state (symbol) in this study and add these loops for the complexity computation, this serves only as a string separator when generating and processing input sequences for the models.

Compared to other complexity measures such as TE computed using the lift method, Shannon entropy or compressibility of strings generated by the grammar, the TE method with subscripts used here is the most sensitive and consistent with the grammar complexity as a function of ambiguity (Figure 8.2).

Of the factors influencing the value of TE, the density and size of the topological matrix are perhaps the most important. As TE typically grows with the number of

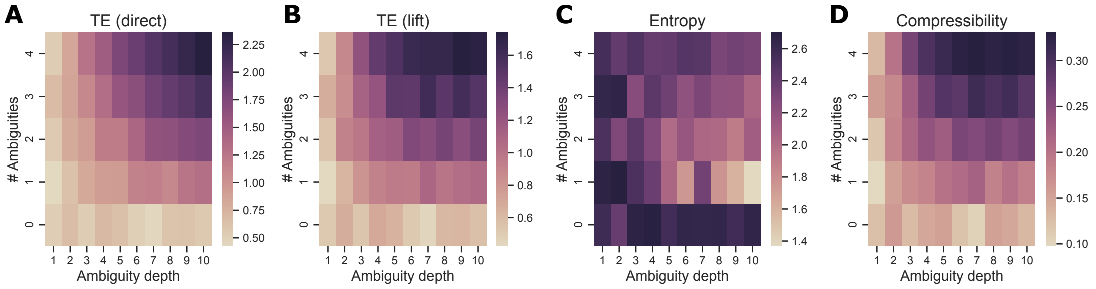


Figure 8.2: **Comparison of grammar complexity measures as a function of ambiguity.** **(A):** Topological entropy computed using the subscript method (Warren and Schroeder, 2015). **(B):** Topological entropy computed using the lift method (Bollt and Jones, 2000). **(C):** Shannon entropy of a sequence of grammatical strings. **(D):** Compressibility ratio of grammatical strings, computed using the DEFLATE algorithm (Deutsch, 1996). Higher value means less compressible. The metrics in **(B)-(D)** were computed on 10000 strings drawn randomly from the respective grammars. Note that the depth parameter is ignored for no ambiguities.

states and transitions in the matrix, increasing the alphabet size, the number of initial states, and the ambiguities (determining the unique states) all lead to higher TE values.

Ambiguous adjacent dependencies: Using the grammar and complexity definitions above, here we restrict the analysis to a single task of learning sequences with ambiguous adjacent dependencies. To evaluate this, we randomly generate grammars of increasing complexity by controlling the level of ambiguity as described above. We tested for up to three ambiguous states and ambiguity depth of four, and either one or two initial states. For each of these parameter combinations, we randomly generate 5 transition matrices (corresponding to 5 grammars) with density ≈ 0.25 and equal outgoing transition probabilites. If any of the constraints are violated, the grammar is discarded and the process is resumed until a valid one is found. We tested alphabet sizes of four and six tokens, with the length of each string truncated to $2|\Sigma|$ to reduce the computational load. Combinations with $a > 0$ and $d = 0$ are considered to be invalid and ignored.

8.3.3 Task evaluation

Task performance is measured using the reservoir computing (RC) approach (see Chapter 2) and the FNA framework described in Chapter 4 and used in the studies presented in Chapter 5 and Chapter 6. Given the vastly different characteristics of the studied models, the method used for constructing the appropriate state matrices was tailored for each model individually in order to ensure a fair comparison and best performance.

Sampling the population responses: For CS , the state matrix X is typically (except for passive memorization) built by sampling the instantaneous firing rates of the Timer populations in all columns at the offset of each token (Δ_{stim}). One could in principle consider the full network instead of only Timer cells, but in practice these are sufficient to decode any stimulus-related information. The rates are estimated using an exponential kernel with time constant $\tau_w = 40\,\text{ms}$, unless otherwise stated.

For the KM model, the state variables are the low-pass filtered spike trains of all the neurons, using a time constant of 20 ms as in the original study.

In the case of spkTM , we consider the somatic spikes of a neuron j, emmitted within a fixed window w relative to the onset t^{σ_i} of a stimulus σ_i, as well as the dendritic action potentials (dAPs) sampled at a fixed delay relative to t^{σ_i}:

$$\begin{aligned} p^{\sigma_i}_{\text{som},j} &= \sum_{t=t^{\sigma_i}}^{t^{\sigma_i}+w} \sum_k \delta(t + t_{\text{samp}} - t^k_{\text{som},j}) \\ p^{\sigma_i}_{\text{dAP},j} &= \sum_k \delta(t^{\sigma_i} + t_{\text{samp}} + w - t^k_{\text{dAP},j}) \end{aligned} \tag{8.4}$$

where $t^k_{\text{som},j}$ and $t^k_{\text{dAP},j}$ are the somatic and dendritic spikes, $w = \Delta_{\text{stim}} + min(d_{\text{dAP}}, \Delta_{\text{ISI}})$ is the window size, and $d_{\text{dAP}} = 14$ a small delay aimed ensuring that any dAP occurring after stimulus offset is captured. t_{samp} is a sampling offset that is non-zero only for the passive memory task, in which case it shifts the sampling window to the right. The sampled state variable for neuron j is then the binarized variable $x^{\sigma_\text{i}}_j$:

$$x^{\sigma_\text{i}}_j = H(p^{\sigma_i}_{\text{som},j} + p^{\sigma_i}_{\text{dAP},j} - 1), \tag{8.5}$$

where H is the Heaviside step function, leading to $x^{\sigma_\text{i}}_j = 1$ if at least one somatic spike occurs within the considered window, or if there is a dAP soon afterwards.

This approach takes into account both the sparse, context-dependent stimulus representations through somatic spikes, as well as the prediction of next tokens encoded in the dendritic activity.

Training and testing procedure: All token-level tasks involve training the readout to classify the currently active token using the matrix X. The training phase consists of 200 strings whereby each token is presented 200 times on average. Generally all models are able to learn such a simple sequence after around 100 presentations, so we discard the first half of the training data and only use the second half to train the linear readouts. This ensures that the readouts are trained on valid activity, i.e., only after the model has completed learning. Test data consists of 25 input sequences, yielding an 80/20 ratio for the effective data.

Rule learning in grammars with ambiguous adjacent dependencies comprises three (sub-)tasks: token classification, n-step memory and n-step prediction. In addition to recognizing the current identity, for each token we ask which element was presented n tokens before (memory) and which one would occur n steps ahead (prediction). Note that because n is limited by the length of each string and thus varies, we use # EOS symbol to demarcate the strings and ensure that memory and prediction does not involve distinct strings.

For these tasks we draw up to 100 strings for each grammar with $|\Sigma| = 4$ ($|\Sigma| = 6$ in Figure 8.12), which are then randomly concatenated (repetitions are allowed) in each batch until a fixed but model-dependent batch size is reached. This is chosen such that the concatenated sequence length approximately matches the number of model parameters (sampled neurons). On average, this gives a batch size of 400 for CS , 675 for KM and 750 for spkTM . We use 10 batches for both training and testing. Such large datasets are meant to ensure sufficient training data and test accuracy even for very complex grammars. Note that this procedure differs from typical machine learning studies, where the model is evaluated on previously unseen data during the test phase. However, given the limited computational power of the models considered here, measuring recognition instead of generalization will still yield valuable insights into their capabilities.

As a performance measure we typically use the standard accuracy metric, which works well for the token-level tasks because each token occurs with the same probability. However, this assumption often does not hold for complex grammars. The imbalance in the token frequency can bias the accuracy and make it difficult to estimate a random baseline. In such cases, we instead use the Cohen's kappa statistic (Cohen, 1960). It is defined as

$$\kappa = \frac{p_0 - p_e}{1 - p_e}, \tag{8.6}$$

where, in the current classification context, p_0 is the empirical accuracy of the model and p_e is the expected agreement between the model predictions and the actual labels as if happening by chance. κ takes values between -1 and 1, 0 being chance level and values towards 1 representing good performance.

In all cases, readouts are trained with ridge regression (see previous chapters).

8.4 Results

We begin with a series of token-level tasks that are intended to shed light on the internal representations and robustness of models to variations in the input. In the second part, we focus on the models' ability to acquire the rules underlying temporally-structured sequences, as well as autonomously replaying learned sequences. In cases where the task is simply unsuitable for a given model or its behavior is straightforward, we exclude it from the corresponding experiment. The arguments for such exclusions and the implications for the model's capabilities are then discussed in-depth.

8.4.1 Pattern perception

Irrespective of whether the input sequence is structured or simply random, perception requires that the individual elements (tokens) of the sequence are represented in a distinguishable manner. The ability to handle and recognize spatiotemporal patterns in a robust manner is therefore a fundamental requirement of any sequence learning model. In a first set of tasks, we evaluate this ability for both ordered and random sequences, and discuss the strengths and weaknesses of the models' properties, including the encoding mechanisms and the characteristics of the patterns they can process.

Token representation

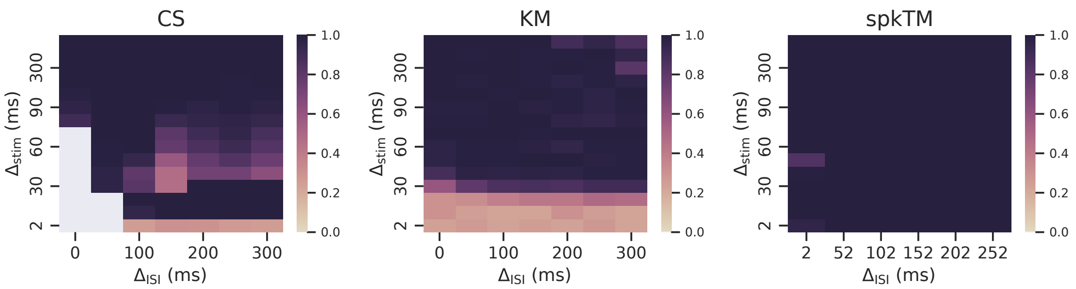


Figure 8.3: **Classification accuracy as a function of stimulus duration and inter-stimulus interval.** Performance is shown for randomly ordered sequences (results are qualitatively similar to ordered sequences). For CS, we consider entries with $\Delta_{\mathrm{stim}} + \Delta_{\mathrm{ISI}} < 75$ ms to be undefined, which corresponds to the sum of reward delay, reward and trace refractory windows that follow each stimulus. For each model, the population responses were sampled at the stimulus offset (not including Δ_{ISI}).

To measure the accuracy of token representations, we used randomly ordered sequences of four elements and trained a linear classifier to predict the token identity from the population responses obtained at stimulus offset (see details on sampling in Section 8.3). By varying the stimulus duration Δ_{stim} as well as the interval between tokens Δ_{ISI} , we probed the range of temporal properties that enables accurate representations (Figure 8.3). For good performance, CS and KM require stimuli that are longer than 10 ms and $20 - 30$ ms, respectively. Although Δ_{ISI} does not have a significant impact, for CS we observe a range of values with reduced performance, which we discuss in a moment.

For such unstructured sequences, the CS results are somewhat misleading because the model learns an all-to-all connectivity between the token-specific columns when an element can be followed by any other one with equal probability. The current token is only decodable due to the active column inhibiting the other populations, so that when we sample at stimulus offset there is a single token being represented. As soon as the rate of the stimulated Timer population decays, any other element can become activated. This may also explain the low performance in certain cases, which appears to depend jointly on Δ_{stim} and Δ_{ISI} . For some parameter combinations, multiple overlapping representations may arise as a consequence of the Timer population responses, weak cross-columnar inhibition towards the end of each token, as well as the short Δ_{ISI} . This effect is only observable in a small parameter range because shorter Δ_{stim} avoid "spillover" Timer responses between tokens while longer ones ensure a sustained and effective cross-columnar inhibition.

The spkTM , on the other hand, is insensitive to any temporal variation in the input. This is expected given that we are interested only in the stimulus identity, which should be distinguishable towards the offset if there is at least one input spike to the corresponding population causing sufficient somatic spiking (condition satisfied here). At first glance, these results contradict the original study (Bouhadjar et al., 2022), which found a working range for Δ_{ISI} between 10 and 75 ms and expected degraded performance for extended somatic activity. While we do not observe these detrimental effects here, they would likely occur for more complex sequence prediction tasks.

Passive memorization

A prerequisite for any complex sequence processing, which invariably involves context-dependent computations, is the ability to maintain information about past elements. While this is evaluated for structured sequences later on, here we use random input to measure how long the representation of a token is preserved after stimulus offset (see Section 8.3.1). Randomness enables investigating the system's passive memory at the token level, eliminating the possibility that an ordered sequence is learned (and memorized) as a whole.

Despite learning an all-to-all connectivity as in the previous task, the CS model exhibits a passive memory that approximately matches $\Delta_{\text{stim}} + \Delta_{\text{ISI}}$ (Figure 8.4A). Because the active stimulus inhibits the populations associated with the other tokens, its representation is distinguishable as long as the rates are sufficiently elevated to maintain the cross-columnar suppression. Soon after the presentation of the next token, this representation is overwritten entirely and performance drops towards chance level, also illustrated by the lack of memory beyond a single token (Figure 8.4B). The slightly better performance for $\Delta_{\text{ISI}} = 100$ ms and sampling delays of around 100 ms are likely caused by the interplay of these parameters and the stimulus duration that lead to low cross-columnar inhibition and competition between multiple Timer populations during a short time window. The passive memory of the model thus cannot be significantly longer than Δ_{ISI} when the tokens are presented sequentially, which is in line with the model's design to learn the time interval until the next token or reward period.

On the other hand, the passive memory of the KM model is between 80 and 300 ms for a fixed $\Delta_{\text{stim}} = 300$ ms. It increased to more than 200 ms when the interval between

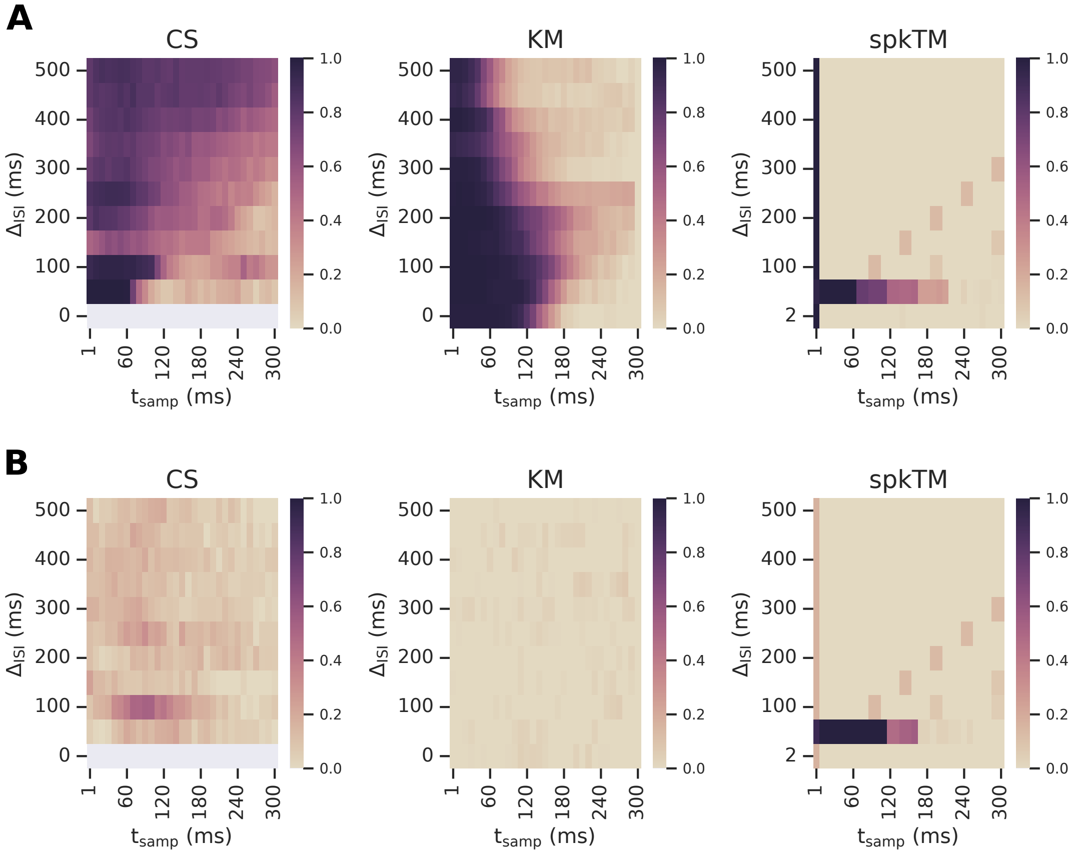


Figure 8.4: **Passive memory of randomly ordered tokens.** Performance is measured using the κ-score, as a function of the Δ_{ISI} and the sampling delay t_{samp} relative to stimulus offset. **(A)** Delayed classification of a stimulus at increasing time lag after its offset. **(B)** Same as **(A)**, but for delayed 1-step memory. Results are averaged over 5 network instances for each Δ_{ISI} .

tokens was approximately in the range of $50 - 250$ ms, with the peak memory found for $\Delta_{\text{ISI}} = 200 - 250$ ms. Given that this optimal range decreases with Δ_{ISI} beyond these values, the timely presentation of the subsequent token appears to extend the memory of the system. An alternative explanation may be that the token representation is longer or simply shifted in time by approximately 250ms. If this was the case, one would expect similar performance for larger Δ_{ISI} as well. One reason why this does not occur could be that longer noisy intervals are detrimental to the learning process.

The spkTM model exhibits more complicated dynamics. In this scenario, the input consists of a single spike ($\Delta_{\text{stim}} = 1$, see Section 8.3.3) and the potentiation rate λ_+ was scaled by a factor of 10 to allow the characteristic behavior to emerge despite the small training set. The stimulus can always be decoded accurately when sampling directly after the stimulus, but the information fades soon thereafter (Figure 8.4A). There is a notable exception for $\Delta_{\text{stim}} \approx= 40 - 50$ ms, which corresponds roughly to the $\Delta_{\text{ISI}} = 40$ ms. This compact range for the intervals allows the emergence of context-encoding dendritic APs, which prolongs the system's memory across multiple stimuli (Figure 8.4B) and highlights its dependence on the decay time constant of the dAPs (set to 40 ms for this task). Outside this range, the interval is either too short or too long for such dAPs to develop accurately. In addition, there is a slightly above-chance performance when sampling at offsets multiple of the ISI. The reason for this is not straightforward and requires a more in-depth analysis.

Time-warp invariance

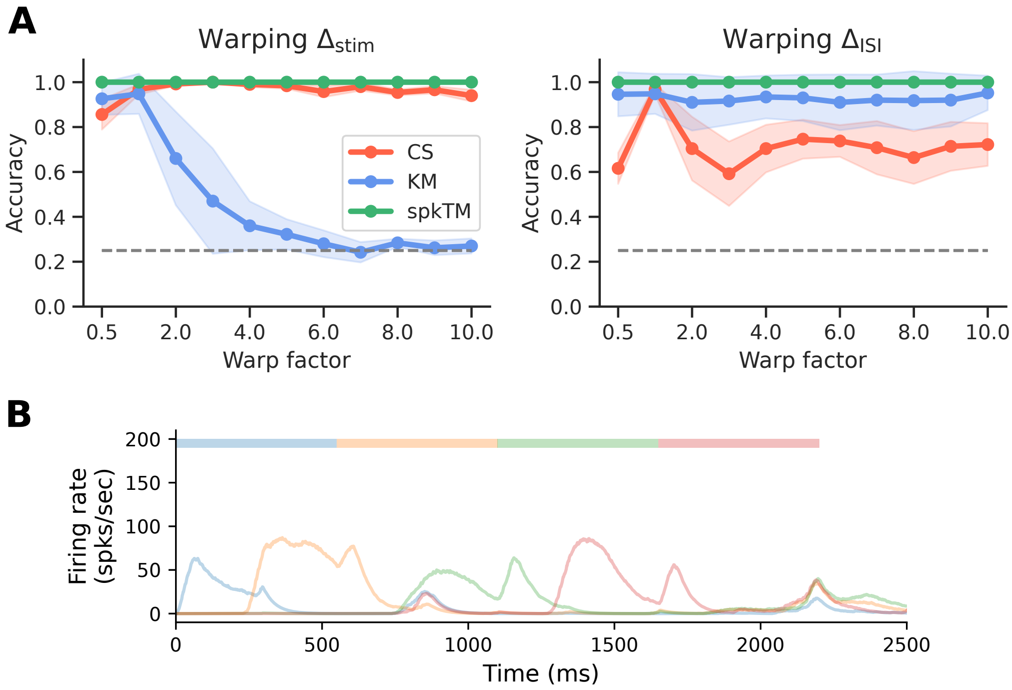


Figure 8.5: **Robustness to time-warped inputs. (A)**: Classification accuracy for a single ordered sequence composed of 4 tokens, as a function of the warping factor applied to the stimulus duration (left) and inter-stimulus intervals (right). During training, the Δ_{stim} was fixed to 50 ms for CS , 300 ms for KM , and 1 ms (single spike) for spkTM . Δ_{ISI} was set to 250 ms for all models. Results are averaged across 5 network instances and are qualitatively similar for different baseline Δ_{ISI} values. **(B)**: Firing rates of Timer populations in the CS model during the test phase, for a Markovian sequence and warp factor of 2 applied to the Δ_{ISI} .

Once a sequence is learned, humans and other animals can recognize it even if there are slight variations in the durations of the individual elements or the time interval between their presentations. To measure whether the models are also invariant to time-warped sequences, we trained them using the regular procedure and warped the stimulus durations or the inter-stimulus intervals during the test phase. Although we only show results for ordered sequences, they are qualitatively similar to random ones.

When considering the responses at stimulus offset, CS is mostly invariant to warping of stimulus durations, with a slight drop in performance for shorter inputs (see Figure 8.5A). This is due to the stimulated Timer cells being the only active population for the duration of the input thanks to cross-columnar inhibition. Furthermore, time-warping does not destroy the order learned Markovian sequence during recall but it may alter the durations of replayed tokens. When the Δ_{ISI} is warped, the performance degrades because the model tries to replay the sequence at the learned pace and warping induces a shift in the stimulation that leads to inconsistent activity (typically the third token is misclassified, see Figure 8.5B).

The KM model behaves in a somewhat opposite manner. Although it is invariant to changes in the ISI, it is more sensitive to warping of stimulus duration and is largely unable to encode the input to beyond a warp factor of 4 (Figure 8.5A). Considering that the number of spikes in the input pattern is unchanged but their timing simply shifted, the high accuracy up to a shift of 900 ms (factor or 3) is nevertheless suggestive of strong robustness.

In contrast, the spkTM model achieves perfect accuracy across the board. As demonstrated previously in Figure 8.3, the model is insensitive to longer stimuli and intervals when evaluated on a simple token classification task as the spatially-encoded input induces sufficiently distinct representations. However, this may not be the case when processing longer sequences.

Token capacity

Another relevant characteristic is the capacity of the model, i.e., the number of different tokens it can encode and process. For the CS and spkTM models, this is fixed and predetermined, and corresponds to the number of columns. As such, scaling the model requires adapting the network for every scenario. The capacity of the KM , on the other hand, is not predefined. As illustrated in Figure 8.6, the model can encode up to 150 tokens if they are presented randomly, and more than 200 tokens if they compose an ordered sequence. Although the capacity decreases with the alphabet size in both cases, the slower decay for ordered sequences indicates that temporal structure in the input enables more robust and distinguishable representations. The actual number likely depends on the network size, but it is also possible that the stimulus representations are dynamic, with a higher number of tokens leading to increasingly fewer neurons coding each one.

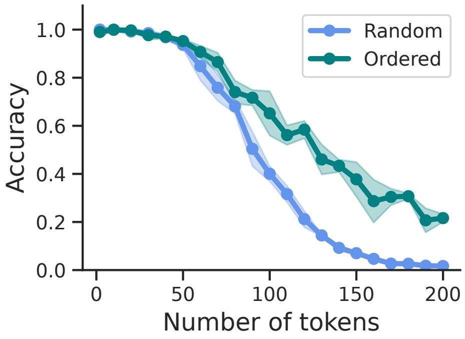


Figure 8.6: **KM performance as a function of the alphabet size.** Classification accuracy is shown for sequences with random elements and a single, ordered sequence. In both cases, sequence length matched the alphabet size (number of tokens). Mean results across 5 network instances are shown for $\Delta_{\text{stim}} = 300$ ms and $\Delta_{\text{ISI}} = 0$ (as in the original model), with shaded area representing the standard deviation. On average, each token was presented 150 times during training and 30 times during testing.

8.4.2 Rule learning

Real-life sequences deviate significantly from randomness, and messages that convey meaningful information consist of strings composed of a limited set of symbols. The patterns we observe in these sequences are a result of intricate generative models, such as environmental dynamics or the thought processes of a conversational partner. To be regarded as a true sequence learner, an entity or a model must possess the capability to comprehend the underlying rules governing the generation of the observed sequences.

Acquisition and internalisation of generative rule systems

Adopting conventions from symbolic processing and AGL studies, we consider sequences generated by artificial grammars that encode transition rules and probabilities that ought to be learned by the models. Typically visualized using a transition diagram (see Figure 8.7A), the complexity of the grammar and the sequences it can generate can be estimated using various measures, including topological entropy (TE) that we employ here (see Section 8.3). We focus on three aspects of sequence processing: memory, classification and prediction.

Figure 8.7C illustrates the performance of the three models for the non-deterministic grammar shown on the left, which can generate relatively complex sequences with only one ambiguous state (TE = 0.46). For this particular grammar, spkTM outperforms CS and KM on some memory tasks, particularly for one and two steps, while the memory and prediction accuracy decreases with larger n at a similar rate for both models. Although the magnitude of the κ coefficient is difficult to interpret (McHugh, 2012), all

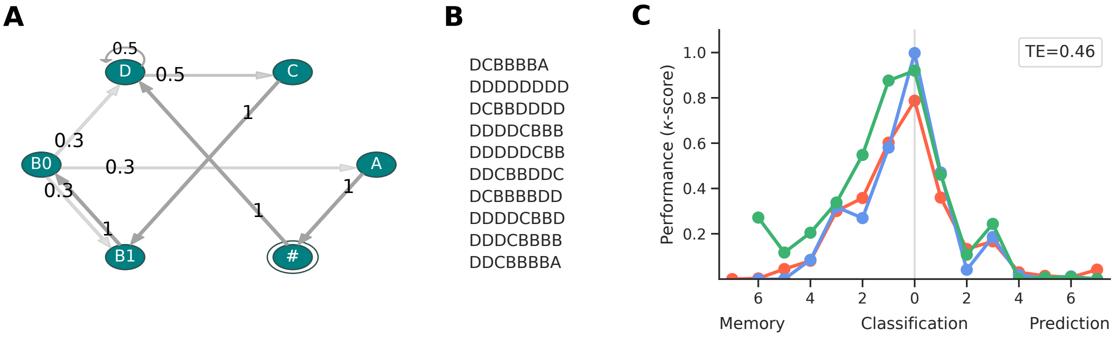


Figure 8.7: **Example grammar and model performance for the rule learning task. (A)**: Transition diagram depicting a grammar with 4 tokens, one ambiguous state (B) and ambiguity depth of two (B_0, B_1). The grammar has a complexity of TE $= 0.46$, with all strings starting with D (initial state). **(B)**: Example strings drawn from the grammar. These are truncated at length $2|\Sigma|$ (see Section 8.4.2, terminal symbol # not shown). **(C)**: Model performance for the grammar in **(A)** for classification, n-step memory and n-step prediction tasks ($n = 7$).

models appear to remember the previous 2-3 and the upcoming token reasonably well.

For these and following results, it is important to note that the κ coefficient is undefined for a single true and predicted label (single class). Assuming perfect accuracy, this means that the maximum memory (and prediction) performance is bounded from above, in addition to the length l of the longest string(s) of the grammar, also by the largest n for which there are at least two substrings of length $n+1$ that differ in their first (last, for prediction) token. For example, 3-step memory and prediction are undefined for the grammar G_1 with $\mathcal{L}_{G_1} = \{\text{AACD}, \text{AABD}\}$, but 3-step prediction is defined for G_2 with $\mathcal{L}_{G_2} = \{\text{AACD}, \text{AABB}\}$. In the next section, we analyze representative grammars and discuss to what extent are these results biased by the readout mechanism.

Readout and model bias

Task performance is often biased by the choice of state variables used to extract information from the system (van den Broek et al., 2017). For instance, the common approach of using the filtered spike trains may artificially increase the system's memory if a kernel with a long time constant is used. This appears to be the case for the CS model, where the performance of a readout trained on the low-pass filtered input spikes was higher than that of a readout trained directly on the (embedded) input labels (compare dashed red and yellow curves in Figure 8.8B). Reducing the filtering time constant from 40 ms to 4 ms removes this bias (dotted curve in Figure 8.8B), and the result on the encoded input matches the expected, memoryless baseline. The actual performance of the CS model indicates that it is able to maintain a representation of the previous stimulus (in line with Figure 8.4A), but not beyond that. In contrast, the KM model does not possess

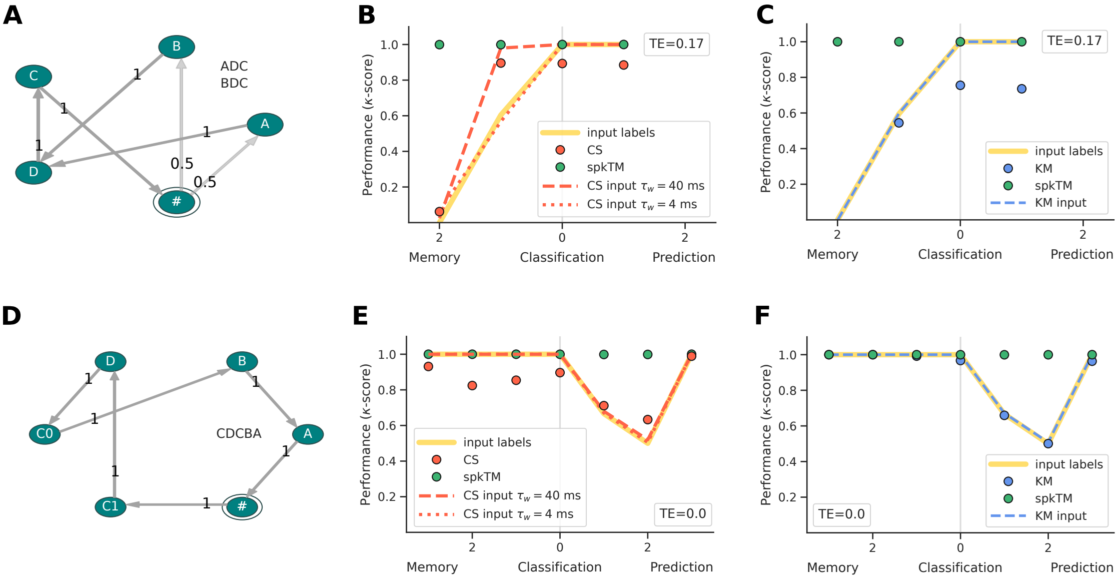


Figure 8.8: **Readout and model bias.** **(A)**: Grammar with no ambiguities and $|\mathcal{L}_G| = 2$. **(B)**: Performance of the spkTM and CS models for the grammar in **(A)** (circles). Yellow curve denotes the baseline performance of a readout trained on the (one-hot) embedded input labels, while the dashed (dotted) red curves represent a readout trained on the instantaneous firing rates estimated from the input (Poisson) spike trains, using a filtering time constant of $\tau_\text{w} = 40$ ms (4 ms). This estimation (with $\tau_\text{w} = 40$ ms) corresponds to the population rates used as the state variable for the model results (circles, see Section 8.3.3 for details). **(C)**: Performance of the spkTM and KM models (circles). Dashed blue curve denotes the performance of a readout trained on the input spike trains (pattern and noise) of the KM model, low-pass filtered using $\tau = 20$ ms as done for the population state variables (blue circles). **(D)**: Grammar with two ambiguous states C0 and C1, which can generate only one string. **(E-F)**: Same as **(B-C)**, but for the grammar in **(C)**.

any memory and also degrades the input representations (Figure 8.8C).

Importantly, estimating the performance directly on the input labels allows us to establish an unbiased, model-agnostic baseline that captures the deterministic (static) parts of the input sequences. This becomes more relevant when considering the ability to predict upcoming elements, as illustrated through a grammar with one ambiguity (Figure 8.8C). With the exception of spkTM , none of the models performs significantly better than baseline (Figure 8.8E,F). These results on relatively simple tasks suggest that CS and KM are not equipped with context-dependent processing capabilities, whereas spkTM achieves close to perfect score. In the following sections, we extend this analysis to grammars of various complexities.

Rule complexity and task performance

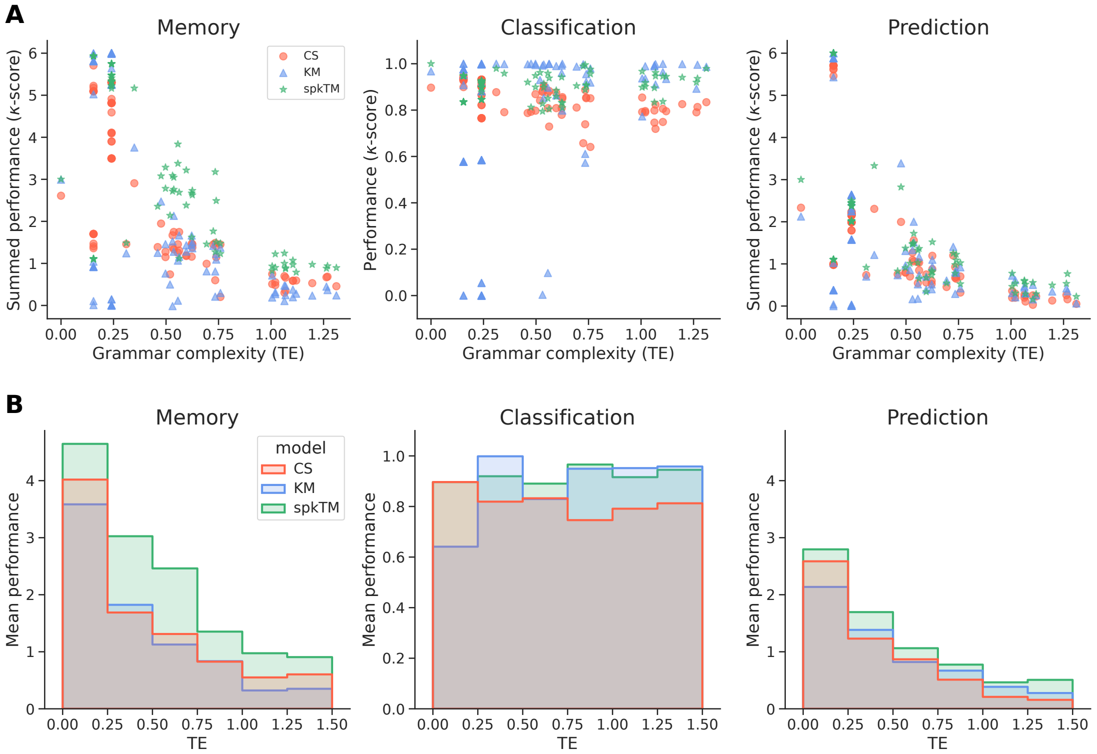


Figure 8.9: **Model performance on a variety of complex sequences.** **(A)**: Grammar complexity and corresponding performance (κ-score) for n-step memory, classification, and n-step prediction tasks. For the memory and prediction tasks, each data point represents the sum of performances over all valid n-steps, which may differ for individual grammars but cannot be greater than $2|\Sigma| - 1 = 7$. Data contains 85 grammars with $|\Sigma| = 4$ and a single initial state (see Section 8.3.3 for more details). Note that due to the randomized generation process, there can be multiple instances of the same grammar or with the same complexity. **(B)**: Histogram of the mean performances from **(A)**, with bin size of 0.25. Each bin represents the summed κ-scores, averaged by the bin count (number of grammars in that complexity range).

The ability to memorize and predict elements also drops with increasing grammar complexity (Figure 8.9). Despite some variability in the performance distribution of CS and KM for similarly complex grammars (Figure 8.9A), the trend holds for all models. As explained above, even in ideal conditions, the maximum performance on the memory and prediction task is constrained by the length, structure and number of the generated strings. These may be smaller for low-complexity grammars containing fewer and shorter strings, as illustrated by the grammar with TE = 0 in Figure 8.9A, which generated only the string CDCBA·

In general, spkTM performs significantly better than the other two models on all tasks. For low-complexity grammars (TE $<\approx 0.3$), spkTM can accurately remember up to 5 tokens on average. This number decreases to around 3 for intermediate complexity (TE $<\approx 0.75$) and to 1 for more complex ones. Prediction follows a similar trend but is somewhat lower, which for the low-complexity grammars can be partially attributed to the sequence structures and the upper bound of the κ-score as discussed above. Note that these results represent memory and prediction performance summed over all n-steps, so a value of 1 is not equivalent to perfect decoding for one single n-step.

The other two models perform similarly across all tasks, with KM achieving somewhat better classification performance for larger TEs. At the same time, in some instances KM fails completely (classification close to 0), indicating that for such complex sequences and long learning phases, the model is unstable with the chosen parameters. For TE $>\approx 0.7$, neither CS nor KM achieves a significantly above-chance accuracy on the memory and prediction tasks. The mean performances reinforce the observation (see Figure 8.8) that, in most cases, both models struggle with contextual information beyond a single step. However, these results are somewhat difficult to interpret because the data points do not capture how the performance changes with n.

To get a better sense for the performance distribution across the different n-steps, we pool the data in Figure 8.9 over all low-, intermediate and high-complexity grammars, and plot the performance as a function of n in Figure 8.10. Although the thresholds are somewhat arbitrary, this categorization is useful to clearly illustrate how performance varies with complexity. Except for 1-step memory where CS and spkTM are comparable for simple grammars (Figure 8.10A), spkTM performs statistically the best on all tasks while KM is typically the worst. For TE < 0.5, the relatively high performances on the memory and prediction tasks even for $n = 6$ is suggestive of little variation in the structure of the sequences, and possibly reflect the bias of our evaluation method for such simple grammars (see also Figure 8.8).

However, any bias should become negligible with increasing complexity, as illustrated in Figure 8.10B,C. In these cases, we observe the expected, more stereotypical performance decay for larger n. Beyond $n = 2$ for intermediate and $n = 1$ for highly complex grammars, the performance of CS and KM is hardly above chance level. spkTM fares better on the memory tasks for intermediate difficulty, with good performance even for 6-steps, but prediction also degrades significantly beyond 3-steps. For very complex rules, even spkTM is limited to 2-step memory and 1-step prediction. To further disentangle these results, we next take a closer look at examples of characteristic grammars.

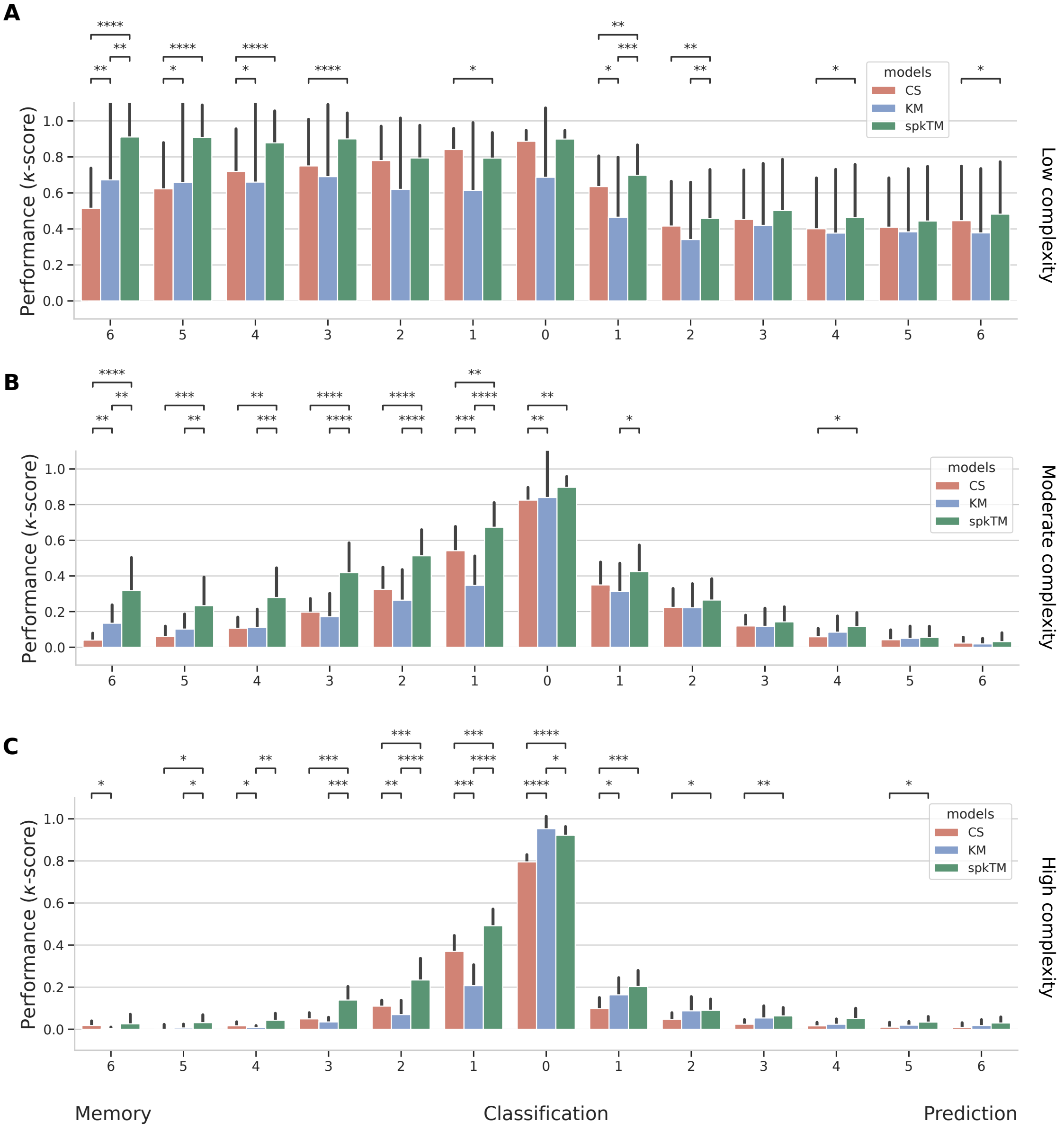


Figure 8.10: **Average model performance for the memory, classification and prediction tasks.** Results are pooled over all grammars within a specified complexity range from the full distribution illustrated in Figure 8.9. These are $[0, 0.5)$ in **(A)**, $[0.5, 1.0)$ in **(B)** and $[1.0, 1.5)$ in **(C)**. Significance was estimated with the Mann-Whitney double-sided test using the `statannotations` package (Charlier et al., 2022).

Representative examples

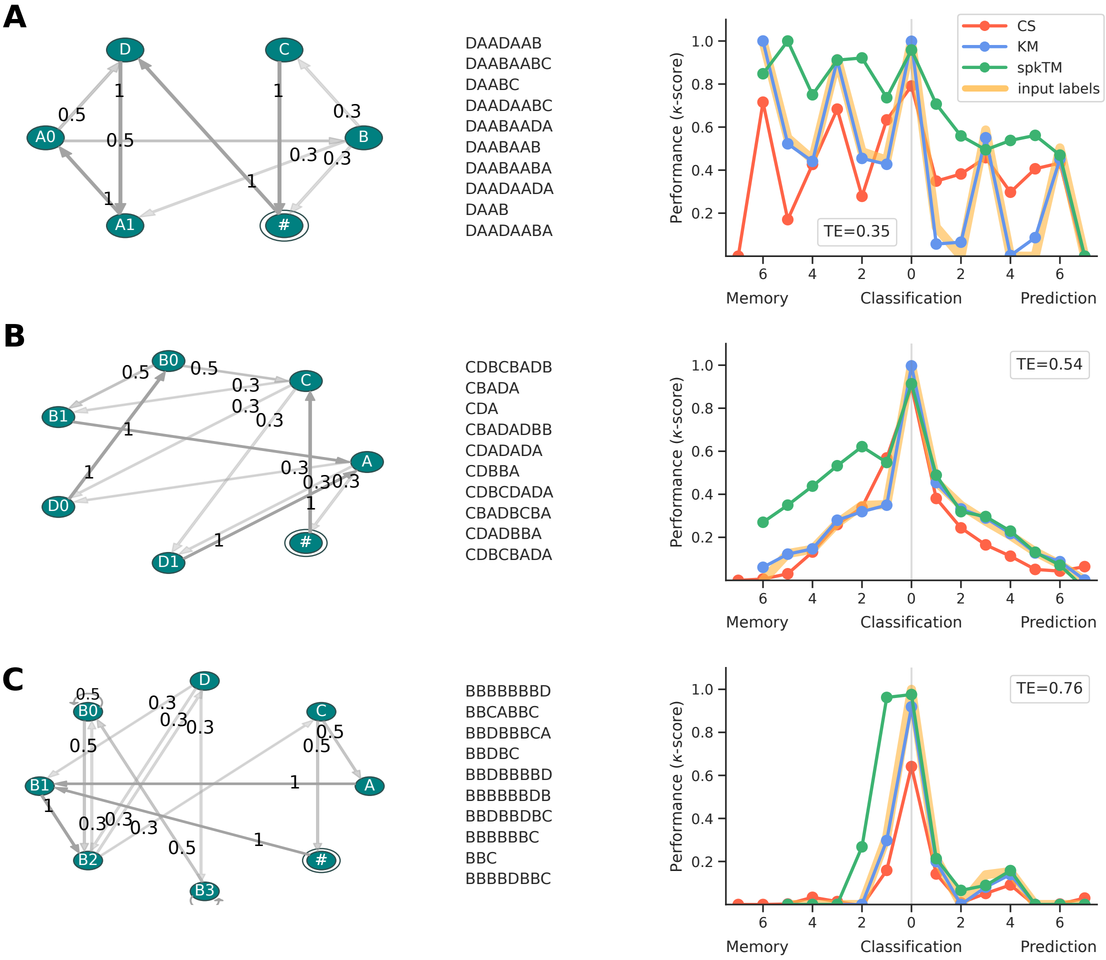


Figure 8.11: **Representative grammars and corresponding task performance.** For each grammar, only a maximum of 10 strings are shown (center panels). Parameters such as the number of initial states (states with transitions from '#') and ambiguities (indexed states) can be read out directly from the transition diagrams. Thick, gold curve (typically behind the blue one) denotes the baseline performance of a readout trained on the input labels (see also Figure 8.8).

An examination of performances on individual grammars allows us to identify the strengths and weaknesses of each model, along with possible limitations of the evaluation approaches and metrics. Figure 8.8 already prefaced these on simple yet illustrative examples. Here we analyze three additional representative grammars in Figure 8.8 to further explore the relation between measured capacity and rule structure. To get a baseline estimate for comparison, we again compute the performance of a readout trained directly on the input labels.

Given that the spiking CS model investigated here should not be able, by design, to handle contextual information beyond a single step, its results are mostly in line with the expectations. For moderate complexity, the model can retain some information from the previous step, but in almost all other cases it performs on par with or worse than the baseline. A notable exception is the prediction in Figure 8.8A, the cause of which requires further investigation. Due to the frequent underperformance, the CS cannot be considered as a baseline that reflects the static mappings within the language defined by the grammar. This is in contrast to KM , the results of which are surprisingly consistent with the baseline. Unfortunately, this also means that in our implementation the model does not really learn anything beyond simple classification.

Building on the previous findings, spkTM demonstrates high memory capacity also for more complex grammars containing two ambiguous elements (Figure 8.11B). For one ambiguous token which can nevertheless appear in four independent contexts (ambiguity of 1 with depth 4, Figure 8.11C), the performance decays close to chance level for $n \geq 2$. However, the model does impressively well to disambiguate the previous token ($n = 1$) even for such complex rules.

The examples further highlight the need for clear reference points that account for the static, recurring patterns in the generated sequences. This is particularly important for low and moderate complexity, whereas higher TE values imply more heterogeneous sequences for which the bias of the readout mechanism is averaged out statistically and measures such as the κ-score yield more accurate scores.

Influence of grammar properties

Until now, we have primarily investigated grammars with an alphabet containing four tokens and a single initial state (with the exception of some examples in Figure 8.11). These parameters, along with other properties such as the density of state transitions, directly influence the grammar complexity and can strongly impact model performance. In particular, different sets of properties may lead to similar complexities as measured using the topological entropy, even if there are core distinctions in the generated sequences. To assess this, we evaluated the spkTM model on four grammar classes containing up to two initial states and alphabet sizes of four and six (see Figure 8.12).

In general, TE increases both with the number of initial states and alphabet size, which control the number of distinct strings. Increasing these parameters shifted the minimum complexity as illustrated by the missing data points in Figure 8.12. The classification accuracy was comparable in all cases. For $|\Sigma| = 4$, the model performed slightly better on grammars with two initial states, except for very simple ones. As the average string length was not controlled for, this large discrepancy may be caused by shorter sequences for $|S_{\text{in}}\ | = 2$.

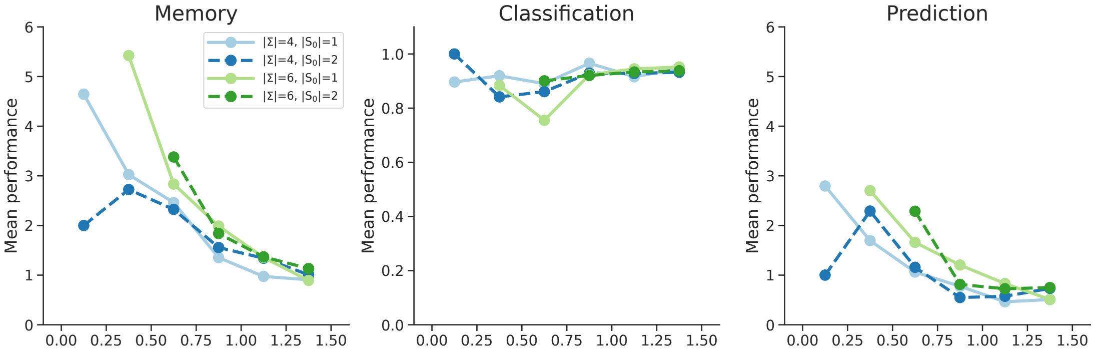


Figure 8.12: **Performance of the spkTM model for grammars containing different number of tokens and initial states.** Data is computed and binned as the histograms in Figure 8.9B, but illustrated as a line plot for readability. Colors code the alphabet size, with solid (dashed) curves representing grammars with one (two) initial states. Minimum string length was two while the maximum was truncated at $2 * |\Sigma| - 1$.

More interestingly, despite longer average strings for $|\Sigma| = 6$, the absolute n-step memory and prediction performance did not improve significantly (compare solid memory curves in Figure 8.12). This is suggestive of a peak contextual memory of around 5-6 items for relatively simple but not trivial grammars. Similarly, peak average performance for prediction was below 4, with slightly better scores for $|S_{\text{in}}| = 2$ on moderately complex grammars (TE < 0.75). However, the performance for larger alphabets is consistently and significantly better for the same range of complexities. Whether this is a consequence of the string lengths (and thus an artifact) or the model indeed benefits from more varied sequences should be evaluated more closely.

Taken together, these results indicate a stronger impact of the alphabet size rather than the number of initial states, but more careful consideration of the role of string lengths is necessary.

8.5 Conclusion

In this chapter, we presented a framework for benchmarking and systematic evaluation of biophysical sequence learning models and applied it to critically review and compare three existing models as a proof-of-concept. Given that this work is still in the early stages, here we focused primarily on the task methodology, topological entropy as a complexity measure, and the challenges for creating a unified framework to handle very different systems. Despite a rather small number of tasks, we nevertheless gained relevant insights into the strengths and weaknesses of the models.

In a first step, we investigated how the temporal properties of the input, such as stimulus duration, inter-stimulus interval and time-warping, affect the ability of the models to represent spatiotemporal patterns that constitute the sequence elements. We identified lower but no upper bounds on the stimulus duration for both CS and KM , while spkTM exhibited consistent robustness. However, these results reflect the classification accuracy of individual tokens, and thus only tell a partial story.

For correctly learning structured sequences, the ISI must be below ≈ 75 ms for the spkTM model (Bouhadjar et al., 2022), while Cone and Shouval (2021) found a maximum element duration of about 1800 ms (which includes the ISI as discussed here) for CS . An upper limit should also exist for KM , as transitions between tokens can only be learned through STDP and therefore through temporal co-activation. To capture these effects, the same task could be performed on structured sequences. This also holds for time-warped inputs, to which only KM was found sensitive when warping the stimulus duration. Moreover, warping the ISIs beyond 75 ms and extended stimulus durations are both likely to negatively impact the performance of spkTM on complex sequences, which is not captured here.

By studying the passive memory properties of the models, we demonstrated that only spkTM can maintain information about past inputs beyond a single element (see Figure 8.4). This capacity depends strongly on the dendritic APs and their properties. In all models considered here, contextual representations do not arise dynamically in the network activity, such as for some recurrent circuits in RC studies, but are learned through prolonged exposure to structured input and engraved in the synaptic connectivity.

This ability to learn rule-based sequences was evaluated and demonstrated on inputs of increasing complexity. Our results indicate that only spkTM can cope with contextual dependencies (see e.g., Figure 8.8), while the other two models are mostly limited to distinguishing individual stimuli. Note that our implementation of KM may be incomplete and more reliable results require further testing. As expected, the complexity measure and the ambiguity parameters chosen for the grammar generation successfully reflect and control the task difficulty, with performance decaying with complexity for all models. This is particularly clear for moderate and high complexity sequences, whereas more simple ones suffer from the readout bias discussed before (see Figure 8.10).

There are multiple ways to counteract such biases. In Figure 8.8 we computed a baseline performance by training a readout directly on the input (see e.g., Klampfl and Maass, 2013), which allowed a clearer distinction between models that are limited to classification and ones that do more sophisticated context-dependent processing. In addition, the sequence length should be carefully accounted for as it impacts the absolute memory and prediction performance, at least as measured in this study. Lazar (2009) overcame this issue by concatenating multiple strings with a delimiter, which worked for predicting the next input thanks to particular sequence structures. However, for sequences with little restriction on where and how often individual tokens appear in it, n-step memory and prediction for $n > 1$ would be affected by cross-sequence references.

Additional issues are posed by model specificities, which make certain tasks difficult to apply or evaluate fairly — for instance, time-warping of the stimulus duration can induce substantial sparsity in the input pattern for KM , whereas the spkTM currently simply cannot handle more than a few input spikes. Moreover, homeostasis plays a central role in the spkTM 's learning process, but we disabled it here for two reasons: it is unclear how the corresponding parameters ought to be chosen given the dependence on the (fixed) sequence length, and any attempt would require extensive testing and possibly substantial modifications to the learning rule.

Even if these difficulties are solved, a deeper understanding of each model's behavior requires additional analysis at the level of single grammars and individual sequences. For example, state-space analysis could help uncover sensitivity to associative chunks and other frequency effects (Meulemans and der Linden, 1997; Robinson, 2005). These should additionally be combined with scrutinization of robustness, such as training stability and effect of dataset sizes. Our goal is to implement these functionalities, alongside several further tasks discussed in the next chapter, and make it available to the community as a software library that enables, through intuitive APIs, to easily benchmark future models.

Chapter 9

Discussion and outlook

As argued in Chapter 1, structure and function are indissociable aspects of neurobiological circuits and should therefore be jointly taken into account, as much as possible, by any mathematical model claiming biological plausibility and aiming to understand the brain. The present thesis is a modest contribution towards this goal, throughout which we sought to investigate and relate architectural and functional characteristics of neural circuits during the processing of noisy information and temporally structured sequences. We pursued these objectives through a combination of software tools, simulation studies and theoretical analysis, as well as a conceptual and computational framework for meta-analysis of biological sequence processing models. The following sections summarize the main findings of each chapter, and place them in a broader context by considering their limitations, potential extensions and scientific implications.

9.1 On the role of structured projections and topographic modularity

Real-time interactions between a dynamic environment and a modular, hierarchical system like the mammalian neocortex strictly require efficient and reliable mechanisms supporting the acquisition and propagation of adequate internal representations. Stable and reliable representations of relevant stimulus features must permeate the system in order to allow it to perform both local and distributed computations online. In Chapters 5 and 6, we have considered models of local microcircuits as state-dependent processing reservoirs whose computations are performed by the systems' high-dimensional transient dynamics (Mante et al., 2013; Sussillo, 2014), acting as a temporal expansion operator, and investigated how the features of long-range connectivity in a modular architecture influence the system's overall computational properties. By treating the network as a large modular reservoir of spiking neurons, composed of multiple, sequentially connected sub-systems, we have explored the role played by biologically-inspired connectivity features (conserved topographic projections) in the reliable information propagation across the network, as well as the underlying dynamics that support the development and maintenance of such internal representations.

9.1.1 Signal representation and denoising

In Chapter 5, we found that random feedforward projections are insufficient for transmitting information about the input beyond the third sub-network, irrespective of the stimulus intensity. While such random connectivity can be powerful for local computations (Buonomano and Maass, 2009), some form of structured projections between processing circuits seems necessary for long-range communication. Although these can also emerge through learning, as in the case of multilayered artificial networks, we demonstrated that even simple topographic projections akin to cortical maps are sufficient to ensure accurate decoding in deeper modules. Such stimulus-specific projections additionally reduce response variability, increase robustness against interference effects, and boost memory capacity.

Expanding on these results, Chapter 6 investigated the dependence of this phenomenon on the modularity — or precision, as a central feature — of topographic projections. Simulation results indicated that not only is some degree of precision required for successfully reconstructing a noisy input signal in very deep networks, but that the performance increases significantly with network depth beyond a critical modularity. This denoising process involves sharpening the spatially encoded input along the topographic map by modulating the E/I balance across the network. Using mean-field approximations, we showed that modularity acts as a bifurcation parameter and derived an analytical expression for its critical value in a simplified model. Further analysis revealed a robust and generic structural property dependent only on the modularity and the presence of recurrent inhibition. The mechanism allows the system to accurately track and denoise rapidly changing signals, requiring that the encoding is locally static/semi-stationary for only a few tens of milliseconds, which is roughly in line with psychophysics studies on the limits of sensory perception (Borghuis et al., 2019).

More generally, topographic modularity, in conjunction with other top-down processes (Kok et al., 2012), could provide the anatomical substrate for the implementation of a number of behaviorally relevant processes. For example, feedforward topographic projections on the visual pathway could contribute, together with various attentional control processes, to the widely observed *pop-out effect* in the later stages of the visual hierarchy (Brefczynski-Lewis et al., 2009; Itti et al., 1998). The pop-out effect, at its core, assumes that in a given context some neurons exhibit sharper selectivity to their preferred stimulus feature than the neighboring regions, which can be achieved through a winner-take-all (WTA) mechanism (Himberger et al., 2018).

However, due to the reliance on increasing inhibitory activity at every stage, we speculate that denoising, as studied here, would not occur in such a system containing a single, shared inhibitory pool with homogeneous connectivity. In this case, inhibition would affect all excitatory populations uniformly, with stronger activity potentially preventing accurate stimulus transmission from the initial sub-networks. Nevertheless, this problem could be alleviated using a more realistic, localized spatial connectivity profile as in (Kumar et al., 2008a), or by adding shadow pools (groups of inhibitory neurons) for each layer of the network, carefully wired in a recurrent or feedforward manner (Aviel et al., 2003, 2005; Vogels and Abbott, 2009). In such networks with non-random or spatially-dependent connectivity, structured (modular) topographic projections onto the inhibitory populations will likely be necessary to maintain stable dynamics and attain the appropriate inhibition-dominated regimes. Alternatively, these could be achieved through additional, targeted inputs from other areas, with feedforward inhibition known to provide a possible mechanism for context-dependent gating or selective enhancement of certain stimulus features (Ferrante et al., 2009; Roberts et al., 2013).

9.1.2 Integrating multiple information streams

The omnipresence of such long-range projections from cortical and sub-cortical areas implies that local circuits are engaged in a continuous integration of distinct information sources. We modeled this scenario by including a second input stream and found that topographic modularity gave rise to different behaviorally-relevant dynamical regimes depending on the relative stimulus intensities (Chapter 6). Whereas asymmetric values led to WTA dynamics and denoising process as observed for single streams, comparable intensities allowed the system to distinguish both inputs for moderately structured projections. Further increasing the topographic precision induced multi-stability (uncertainty) in representations, alternating between two stable fixed points corresponding to the two input signals as found in competition-based winnerless dynamics.

Computation by switching is a functionally relevant principle (McCormick, 2005; Schittler Neves and Timme, 2012), which relies on fluctuation- or input-driven competition between different metastable (unstable) or stable attractor states. Structured projections may thus partially explain the experimentally observed competition between multiple stimulus representations across the visual pathway (Li et al., 2016), and is conceptually similar to an attractor-based model of perceptual bistability (Moreno-Bote et al., 2007). Moreover, this multi-stability across sub-networks can be "exploited" at any stage by control signals, i.e. additional (inhibitory) modulation could suppress one and amplify or bias another.

The nature of these interactions and their impact on the downstream circuitry depends strongly on the clarity of state representations available at the site of aggregation. One of our main results in Chapter 5 suggests that computing locally, within a module, and transmitting the outcome of such computation (local integration scenario) is more effective than transmitting partial information and computing downstream. Accordingly, even a single step of nonlinear transformation on individual inputs (downstream integration scenario) hinders the ability of subsequent modules to exploit non-trivial dependencies and features in the data, which was evaluated on the XOR task. In partial disagreement with previous studies (Rigotti and Fusi, 2016; Barak et al., 2013), which reported higher response dimensionality in populations during nonlinear processing, we found only a negligible correlation between the effective dimensionality and the XOR performance beyond the first modules. The degree of mixed selectivity in early processing stages — rather than dimensionality — proved a better predictor of the task performance in the deeper levels, particularly for nonlinear tasks.

Therefore, it might be more efficient to integrate information and extract relevant features within local microcircuits that can act as individual computational units (e.g. cortical columns, Mountcastle, 1997). Combining the inputs locally may lead to more stable representations, which can then be robustly transferred across multiple modules. We speculate that in hierarchical cortical microcircuits, contextual information (simply modeled as a second input stream) must be present in the early processing stages to enable more accurate computations in the deeper modules. This could, in part, explain the role of feedback connections from higher to lower processing centers. As discussed below, exploring these questions and, ultimately, understanding the core principles of cortical computation, requires bridging cognitively relevant tasks with neuroanatomy and physiology at a level of complexity far beyond the models studied in this thesis.

9.1.3 Beyond simplified models of cortical processing

The classification and XOR problems we have employed here provided a convenient method to investigate information transfer across multiple spiking modules, and allowed us to shed light on the functional implications of the wiring architecture. Our key finding, that the modulation of information processing dynamics and the fidelity of stimulus/feature representations results from the structure of topographic feedforward projections, provides new meaning and functional relevance to the pervasiveness of these projection maps throughout the mammalian neocortex.

Beyond routing feature-specific information from sensory transducers through brainstem, thalamus and into primary sensory cortices (notably tonotopic, retinotopic and somatotopic maps), their maintenance within the neocortex (Patel et al., 2014) ensures that even cortical regions that are not directly engaged with the sensory input (higher-order cortex), can receive faithful representations of it. Moreover, these internal signals, emanating from lower-order cortical areas, can dramatically skew and modulate the circuit's E/I balance and local functional connectivity, resulting in fundamental differences in the systems' responsiveness.

As evidenced by the projected implications of these findings, the simplicity of the model stands in stark contrast with the anatomical and computational intricacies of the brain we drew the reader's attention to in the opening of this thesis. In light of this sheer complexity, finding the right level of description for neuroscientific models is a massive challenge. With the risk of abstracting away crucial aspects, there are many reasons to favor simpler models: they allow building intuition more easily, simplify causal claims about the relation between network properties and their function, are more efficient to simulate, and are inevitable when analytical tractability is desired. Considering that "a computational model should be as simple as possible, but no simpler" (Wilson and Collins, 2019), our results should be interpreted in the context of some critical structural features that were not included.

The denoising phenomenon, as studied here, relies on direct projections between similarly tuned populations in distinct circuits. While such connectivity is a good approximation for the sensory pathways arriving at the primary cortices, information propagation beyond this stage involves stereotypical flows within and between cortical columns. Although this complicates signal transmission, denoising could in principle still occur as long as some form of effective topography is maintained throughout the layers and areas. Despite abundant information on the size of receptive fields (Smith et al., 2001; Liu et al., 2016; Keliris et al., 2019), there is relatively little data on the laminar position and specific connectivity between neurons tuned to related or different stimulus features across distinct circuits. Should such experiments become feasible in the future, our model provides a testable prediction: the projections must be denser (or stronger) between smaller maps to allow robust communication, whereas for larger maps fewer connections may be sufficient (see Chapter 6).

Cortical systems also display an abundance of feedback loops that exhibit, similarly to the feedforward cortico-cortical connections, a high degree of specificity and spatial segregation (Markov et al., 2014a; Markov and Kennedy, 2013). Such feedback connections from more anterior cortical regions (typically associated with more abstract or 'higher' cognitive functions) have been shown to play a central role in top-down control and modulation of sensory processing by providing contextual information and facilitating multisensory integration (see e.g., Markopoulos et al., 2012; Clavagnier et al., 2004; Revina et al., 2018). Important theoretical frameworks of cortical processing, known as predictive coding theories (Friston and Kiebel, 2009; Bastos et al., 2012) place a fundamental importance in the role of such top-down feedback as a pathway through which internal predictions from higher cortical regions are propagated downstream and used as an explicit error signal, guiding and structuring the nature of internal representation in the hierarchically lower cortical modules. Their functional role is not entirely unambiguous and depends on specific functional interpretations, with evidence for both destabilizing (Joglekar et al., 2018) and facilitating (Rezaei et al., 2020) effects on long-range signal propagation. Failure to account for feedback projections may therefore limit the scope and generalizability of our models.

Similarly, we did not consider long-range projections, which directly link distant cortical modules (commonly referred to as skip, or "jump" connections Knösche and Tittgemeyer, 2011). Such projections between non-adjacent areas were found to significantly improve the short-term memory capacity of a biologically realistic spiking network model (Schomers et al., 2017), suggesting that a similar effect could be expected in our model. In the domain of artificial neural networks, an entire class of architectures exploit this principle (residual networks) and demonstrate their functional significance as a way to eliminate singularities during training and ameliorate the problem of vanishing gradients

(Orhan and Pitkow, 2018), as well as improving performance in image standard recognition tasks (He et al., 2016). Even though these aspects were not explicitly explored in this thesis as they would greatly extend its scope, these studies support the crucial role of network architecture and the nature of inter-modular connections in determining the system's computational characteristics.

Finally, mounting experimental evidence suggests that direct cortico-cortical projections are complemented by parallel pathways through higher-order thalamic nuclei (Sherman and Guillery, 2013; Mo and Sherman, 2019). In a brief essay (Zajzon and Morales-Gregorio, 2019), we reviewed recent progress on mapping such trans-thalamic pathways between somatosensory and motor centers and speculated about their possible functional role in cognition. The presence of such projections beyond functionally related areas contradicts previous assumptions (Theyel et al., 2010; Sherman and Guillery, 2013), and suggests that they are a general organizing principle and integral link in cortical communication. These routes may support dynamically constructing task-relevant functional circuits (Nakajima and Halassa, 2017), as well as change the effective connectivity between cortical regions through targeted gain modulation (Jaramillo et al., 2019). Using a simplified spiking network resembling our model, Cortes and van Vreeswijk (2015) demonstrated that such parallel trans-thalamic pathways can help stabilize asynchronous spike propagation over multiple layers, by modulating the feedforward gain to maintain similar rates and dynamical regimes across the network.

Along with our findings, these observations highlight the necessity for incorporating more detailed connectivity patterns and sub-cortical structures in future large-scale, biophysical models of cortical processing. Random (non-specific) projections between different regions, as currently assumed by many multi-area models (Markov et al., 2014b; Schmidt et al., 2018; Joglekar et al., 2018), is unlikely to support the rich dynamics required for challenging tasks. Data-driven approaches that explicitly model thalamic inputs (Billeh et al., 2020; Dura-Bernal et al., 2022) are more promising, particularly if they use naturalistic stimuli and plausible sensory mapping onto the cortex. Integrating these essential ingredients into large-scale models may represent a fertile direction for future exploration of modular topographic maps and the denoising phenomenon investigated here.

9.2 On biologically-plausible models for sequence processing

Irrespective of their size and level of biological detail, computational models of cortical circuits ought to be embedded in a functional context that ideally goes beyond "simple" stimulus representations such as image classification. Although scientific progress has a strong incremental component (which co-exists with the Kuhnian perspective of paradigm shifts; Kuhn, 2012) and there is undeniable value in studying neurobiological phenomena in isolation from higher-level cognitive tasks, ambitious steps are necessary to close the division between low-level or circuit mechanisms and behaviorally relevant functions. Whenever possible, computational models should draw on knowledge from multiple domains and take advantage of existing or contribute to new bridges between these. This certainly applies to models of sequence processing that seek to capture the natural predisposition of the neocortex for detecting temporal regularities, which could, in theory, rely on an incredibly vast literature from cognitive sciences, linguistics or computer science.

Unfortunately, the considered task complexity and interdisciplinary nature of the studies tend to decrease with the degree of biological faithfulness (see Chapter 3). Whereas ANNs tackle problems on all levels of the Chomsky hierarchy and are rigorously tied to formal systems, as recently illustrated in a meta-study by (Delétang et al., 2022), biophysical models are significantly more restricted in their scope and theoretical grounding. The limited capabilities of such models obviously play an important role here, but most studies are nevertheless insufficiently systematic in their evaluation and focus perhaps too narrowly on specific aspects of sequence processing. To bridge this gap, in Chapter 8 we proposed a framework whose objective is twofold: it provides a set of cognitively-inspired tasks and a unified method for evaluating and comparing such models; and by doing so, it creates the scaffold of a meta-analysis of existing models with the larger goal to identify the dynamics, connectivity and learning rules that can support the process. As discussed below, the work undertaken here marks only the initial steps towards both of these objectives due to the multitude of functional aspects, neurobiological implications, and reproducibility challenges that must be taken into account.

9.2.1 Functional considerations

In Chapters 1 and 3, we briefly touched upon the range of tasks that fall under the umbrella of sequence processing. An obstacle to simply applying this battery of tasks consists in the nature of many biologically-plausible models, which are tailored for particular tasks or to illustrate specific mechanisms. Of the three models considered here, one could argue that CS (Cone and Shouval, 2021) was designed to learn and replay the order and duration of simple sequences, KM (Klampfl and Maass, 2013) to reproduce experimentally observed sequential activity, and spkTM (Bouhadjar et al., 2022) to handle context-dependence and prediction while relying on sparse activations and dendritic signals. The limitations posed by spiking networks, for which achieving high memory capacity is still an open issue, is also highlighted by the original SORN model with binary neurons (Lazar, 2009) and its spiking counterpart LIF-SORN (Klos et al., 2018): the original model could solve counting and occluder tasks, while the spiking version modeled a visual experiment involving one simple sequence.

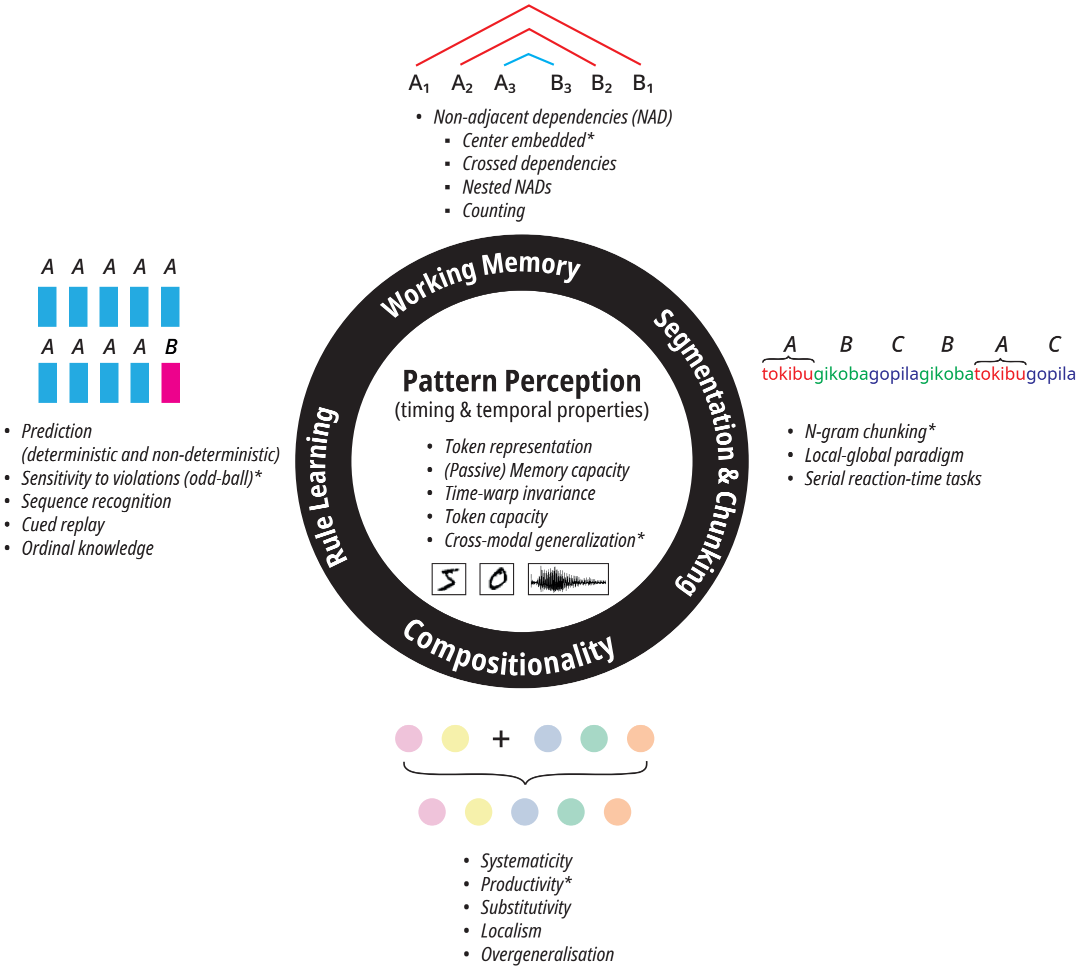


Figure 9.1: **Classes of tasks and paradigms for evaluating different aspects and components of sequence processing.** The *Pattern Perception* tasks (center) target core functionalities that should be displayed by any sequence processor, which can be evaluated both in isolation and as secondary features within the other, more complex tasks. This collection is certainly not exhaustive (e.g., working memory is typically probed using variants of delayed match-to-sample), nor are the individual categories completely independent of each other. Tasks under *Compositionality* were suggested by Hupkes et al. (2019) and graphics (illustrating the tasks marked with *) were adapted, with permission, from Dehaene et al. (2015); de Vries et al. (2012); Hupkes et al. (2019).

Our framework abstracts from model-specific goals and instead treats them as generic sequence processors that are to be evaluated on a variety of established tasks in relevant domains (see Figure 9.1). Currently, we only implemented a subset of the pattern perception and rule learning tasks, but aim to include more in the near future. Decomposing the problem of sequence learning is essential to understand its different aspects: for example, studying the representations of discrete sequence elements (pattern perception) allowed us to identify which temporal features the models are sensitive to, without worrying about the actual input structure (see below).

To investigate the internalization of generative rules, we took inspiration from the artificial grammar learning (AGL) paradigm and developed a simple method for generating sequences of controlled complexity. The abstract nature of such rules means that they do not require existing semantic or syntactic knowledge, making them an ideal tool for studying the neurobiological underpinnings of abstract rule learning independently of the sensory modality (Petersson et al., 2012). Choosing the right level of complexity is key, though, to fully grasp a model's capabilities and not just its limitations. The topological entropy (TE) measure, previously applied to AGs in experimental settings (Schiff and Katan, 2014) but not modeling studies, enabled a fine-grained control over the task difficulty that was also reflected by the model performances. These indicated that only spkTM is equipped for non-trivial processing (Chapter 8).

Decisions about task complexity should be informed by experimental findings and also consider the models' expressive power. Although we explored a wider range of TE values here, the AGs reviewed by Schiff and Katan (2014) were roughly between 0.5 and 0.9. This range not only captured the limits of human capabilities but also exposed strong variations in the results for similar complexities. In these experiments the subjects are typically required to perform binary classification, i.e., decide if a sequence is grammatical or not, which differs from our prediction tests. A more comprehensive and accurate measurement of prediction would require computing the true probability distributions of the tokens and comparing it to the output estimates provided by the readout (Duarte et al., 2014). Nevertheless, while TE may be a good general measure that captures overall trends, task performance depends on a multitude of additional, grammar-specific factors that are beyond the scope of our study (Robinson, 2005; van den Bos and Poletiek, 2008). Considering these aspects, it may be unreasonable to expect better results from models operating within biological constraints.

Striking the right balance between "fairness" to individual models and cognitively relevant functions is therefore challenging. While it may seem unproductive to assess models of sequence learning in the primary sensory areas (e.g., Cone and Shouval, 2021) on psycholinguistic tasks, the computational principles and core neurobiological processes, which underlie widespread sensitivity to temporal structure (Wilson et al., 2018; Henin et al., 2021), are similar across the cortex (Harris and Shepherd, 2015). In this sense, our framework should be understood as a practical and conceptual toolbox that can guide the development and evaluation of future models. Applying it to existing ones and studying their capabilities beyond their original scope can help broaden the understanding of their strengths and shortcomings, both from a functional perspective and at the level of the neurobiological mechanisms they rely on, as we elaborate in the next section.

From a practical perspective, the diversity of these mechanisms, along with the peculiarities of each model, complicates the realization of a unified yet unbiased evaluation method. While the RC approach we adopted here is system-agnostic, it still involves certain choices regarding stimulus encoding, dataset sizes or the sampled state variables, which need careful consideration of the model specificities. Even when these choices are made as fairly as possible, as strived for in our study, there will be certain tasks that are incompatible with a model's characteristics and assumptions or the result can be determined a priori (e.g., token capacity). For instance, thanks to its (questionable) design, the model proposed by Maes et al. (2020) can in principle learn arbitrary sequences (see also Chapter 3) and would thus achieve perfect accuracy on many of the tasks considered here. Does it also elucidate how animals process sequences? Evidently not, which is why in such cases, particularly, a meticulous look at the underlying assumptions and their biological plausibility is warranted.

9.2.2 Neurophysiological implications

In contrast to machine learning or AI models whose primary goal is performance with virtually no restrictions on the architecture or learning algorithms, biophysical models of the brain (should) concentrate on behavioral phenomena whilst adhering to strong constraints imposed by the underlying circuitry. Our model selection criteria ensured that these mostly complied with the latter aspect, even if the constraint on the spiking nature may appear too restrictive. If the information is encoded primarily in the continuous-time firing rates and does not rely on the temporal precision of the spikes, which is the case for the CS model, then perhaps rate models should be treated as equally plausible and included in our evaluations in order not to involuntarily take a stand in the longstanding debate about rate and spike coding (Brette, 2015). Even though systems relying on such dynamics are more flexible, analytically manageable and can support more complex processing (see e.g., the rate version of CS), we argue that their results do not trivially translate to spiking networks. Reasons for this include an incomplete understanding of key mechanisms, particularly memory and learning, as well as the difficulty of maintaining realistic activity statistics using parameters within biological boundaries.

Analyzing the fidelity and identifying contradictions to experimental observations, both at the activity and behavioral level, is in fact a central goal of our framework. Although direct comparisons can be challenging for vastly differing levels of description, there are some general characteristics of sequence processing that plausible models should exhibit. As Figure 9.1 shows, a dedicated set of tasks related to pattern perception (center) could also be applied to more complex processing. For example, real sequences have substantial variability in the distribution of element durations, such as the length or utterance speed of words, which are reflected by various linguistic laws (Torre et al., 2019). Clearly, we are able not only to abstract from these properties (during learning or comprehension), but also manipulate them flexibly (during language production).

Thus, it is tempting to dismiss a model as implausible if it is overly sensitive and rigid to the temporal properties of the input or cannot cope with realistic amounts of noise. Although only partially explored in Chapter 8, this seems to be the case for most systems we considered. However, such conclusions may be imprudent given that sequence learning appears to be temporally specific, at least in some early sensory areas (Gavornik and Bear, 2014). Lacking a behavioral component, such studies are unable to answer a more pertinent question: when does the sequence manipulation transition from only changing the neural representation at a local level to influencing perception and affecting behavior? Finding an answer to this can be hindered by the inability of many species, from which detailed recordings are possible (e.g., mice), to process more complex sequences. Still, the question is relevant to our expectations of the models and their fair assessment. If low-level sequence representations are indeed very fragile, we must continue the search for the right level of complexity that enables robustness, abstraction and generalization.

In certain cases, it can also be unclear what the correct or expected behavior of the model is. Randomized sequences represent a trivial but powerful example of this dilemma. Given the lack of structure in the input, should the model *ignore* or try to learn each sequence? The models considered here will attempt the latter and will likely fail for any reasonable number of stimuli and training time. More importantly, the strengthening of synapses between all stimulus-specific populations, such as the all-to-all connectivity emerging in the CS model, may not be biologically plausible.

Another obvious deficiency of current biophysical models is the lack of hierarchical structure (see Chapter 1), which naturally limits their capabilities as generic sequence processors. For instance, none of the models reviewed in Chapter 3 can operate on multiple timescales, a key signature of complex temporal processing. We must then ask whether these models can be extended towards a hierarchical architecture, or if they rely on more fundamentally incompatible assumptions.

All three of the investigated models incorporate some form of modularity at the network level, which is central to their operation. Although some display flexibility towards more realistic wiring patterns as demonstrated in Chapter 7, the requirement of completely segregated populations tuned to unique stimuli, a core property of the CS and spkTM models, is difficult to reconcile with experimental data. In addition to evidence of mixed selectivity across the cortex including the earliest sensory regions (see Chapters 1 and 5, and de Vries et al., 2020), complex tasks requiring a mixture of representations can not be easily conceptualized in the context of the proposed architectures — a criticism that also applies to the networks considered in Chapters 5 and 6. With one-to-one mappings, compositionality would require exponentially many populations for representing all possible token combinations. Since this obviously cannot be how the cortex operates, future models should focus on mechanisms supporting dynamical feature representations (such as in KM or the model proposed by Asabuki et al., 2022) that can be shared and recycled in new contexts or during new tasks in a flexible manner (Yang et al., 2019; Weidel et al., 2021).

9.3 On software tools and reproducibility

On a final note, the software tools developed and models analyzed and replicated as part of this thesis deserve a brief reflection. In Chapter 4, we presented Functional Neural Architectures (FNA), a toolkit meant to facilitate the creation and evaluation of neural networks using the RC approach. It has a particular focus on symbolic processing and (recurrent) spiking models, but is not limited to these and can be used in virtually any scenario involving an input-driven system. Although there are a number of tools with a similar scope and new ones continue to be developed (see, e.g., Suárez et al., 2023), to the best of our knowledge none of them combine the aspects of symbolic processing, efficient simulation (NEST is used as simulation engine), and the breadth of analysis routines included in FNA. By offering a wide array of methods for input generation and encoding, as well as network activity and performance analysis, FNA also addresses some of the most time-consuming and error-prone parts of developing research software.

As the tool (or parts of it) was used and grew throughout the studies presented here, it became a sort of "all-in-one" codebase where the different components tended towards unnecessarily tight coupling. Minimizing such dependencies is key to increasing flexibility, reducing the time-to-result, as well as reaching a wider user base who may only require some features. These could be achieved by clearer conceptual separation and improved APIs between input generation, network models, and postprocessing. From a software engineering perspective, this means moving from a framework (which FNA technically is not but can "feel" like one) that dictates the flow of a simulation experiment, towards a library where the user can access (call) different components as required (see also "inversion of control"; Fowler, 2005).

Whereas these aspects are more of a cosmetic nature, reproducibility is critical. This is not only guaranteed by FNA, but its usage of an established neural simulator (NEST) also ensures model correctness. The reproducibility and public availability of our models (except Chapter 8, which is ongoing) means that they comply, at least partially, with the FAIR (findable, accessible, interoperable and reusable) principles in the context of reproducible computational workflows (Goble et al., 2020; Eriksson et al., 2022). Steps towards better compliance include model specification in a standardized format (e.g., PyNN; Davison et al., 2008) and integration of a workflow manager such as Snakemake (Köster and Rahmann, 2012).

If the metaphor "standing on the shoulders of giants" - an allusion to scientific advancements building on previous knowledge - is to apply for computational studies, then these must go beyond reproducibility and also be replicable through a comprehensive description and documentation of model properties and parameters (Pauli et al., 2018). Typically, only certain aspects of a model live on to serve as building blocks in future works. Without a full reproduction and in-depth analysis, which is infeasible in most scenarios, this process relies on the accuracy of the original description. The fact that these are often not granted, as demonstrated in Chapter 7, underscores the usefulness and necessity for replication studies even when the code is made available (Simons, 2014; Nosek and Errington, 2020). Besides verifying and consolidating knowledge, there are some errors, inconsistencies or hidden assumptions which can only be discovered during a replication attempt. In more fortunate circumstances like ours, relevant findings can lead to a correction of the original study and codebase (see Chapter 7 and Cone and Shouval, 2023). Is *this* - skeptical, rigorous and self-correcting - not how all science should be (Pulverer, 2015; Bordignon, 2020; Besançon et al., 2022)?

Despite initiatives like ReScience (Rougier et al., 2017) promoting them and a looming confidence crisis surrounding computational neuroscience (Miłkowski et al., 2018), replication studies remain rare due to the significant effort involved and comparatively meager rewards in the current academic system. Considering these aspects, our approach to re-implement all sequence processing models reviewed in Chapter 8 is certainly non-standard. However, our past and current experiences (Pauli et al., 2018; to Brinke et al., 2022; Oberländer et al., 2022) indicate clear benefits, including a profound understanding of each model and providing an open-source and often faster implementation to the community. As a bonus, such undertakings can also help identify weaknesses or missing features in widely-used tools like NEST, driving future development to better serve the needs of researchers.

9.4 On the role and value of computational studies

As data-driven research gains ground on traditional hypothesis- and theory-driven studies and threatens to make them obsolete (Mazzocchi, 2015), it is worth pausing for a moment and reconsider the purpose of computational models, particularly in the context of an interdisciplinary field such as neuroscience. Among the undisputed reasons for brain simulations (Einevoll et al., 2019) - despite this being a highly contentious topic -, is that they enable virtual experiments that would not be possible or ethical on animals. Scanning over vast ranges of biophysical parameters or modifying circuit properties (e.g., to reflect pathological conditions) suddenly become feasible, leading to new predictions and generating testable hypotheses. However, prediction (of neural activity) is by no means the sole or even the most important purpose of a model (Humphries, 2019). Rather, models should be used to better understand and explain observed phenomena, test ideas and theories, and synthesize findings.

We have touched upon several of these aspects throughout this thesis. Initially, Chapter 5 sought to explore the impact of random and structured connectivity on signal propagation, but ended up predicting a deeper functional role for topography. We examined this further in Chapter 6, first crystalizing the potential for denoising through modularity, then validating it through simulations and gradually simplifying the model until the available theoretical tools could help us elucidate the underlying mechanisms. The final two chapters focused on synthesizing insights from models of sequence processing and opening bridges towards disciplines that can provide a scrupulous functional and theoretical framework to assess the capabilities and relevance of such models.

Synthesizing and aggregating knowledge from different fields - across the spectrum "from molecules to culture" - is crucial because there is likely no single best *level of explanation* for cognitive phenomena (Colombo and Knauff, 2020). Given that this endeavor requires a collaborative effort from many communities, we advocate more cross-pollination between approaches focusing on different biological or functional scales. While there is little doubt that computational models are integral to achieving this, trying to build a model of a cortical circuit is a humbling experience in itself. Two of many challenges stand out: the constant need to balance between missing out on potentially critical details (Rubin, 2017) and the desire to capture relevant behavioral functions; and the uncertainty whether a bottom-up (knowing modular topography and questioning its function) or a top-down approach (starting from sequence processing and aiming to work out its implementation) is the most effective way forward. Combining these aspects will likely be essential for any substantial progress, even if the ultimate but perhaps elusive goal of a universal theory for how the brain operates appears rather distant for now.

Bibliography

M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *ArXiv*, 2016.

E. S. Abbot. The Causal Relations between Structure and Function in Biology. *The American Journal of Psychology*, 27(2):245, Apr. 1916. ISSN 00029556. doi: 10.2307/1413176.

L. Abbott and W. R. Regehr. Synaptic Computation. *Nature*, 431(14):796–803, October 2004.

L. F. Abbott, J. A. Varela, K. Sen, and N. S. B. Synaptic Depression and Cortical Gain Control. *Science*, 275:220–223, 10. January 1997.

L. F. Abbott, K. Rajan, and H. Sompolinsky. Interactions between Intrinsic and Stimulus-Evoked Activity in Recurrent Neural Networks. In *The Dynamic Brain: An Exploration of Neuronal Variability and Its Functional Significance*, pages 1–16. 2011. ISBN 9780199897049. doi: 10.1093/acprof:oso/9780195393798.003.0004.

M. Abeles. *Corticonics: Neural Circuits of the Cerebral Cortex.* Cambridge University Press, Cambridge, 1st edition, 1991.

J. Ackerman and G. Cybenko. A survey of neural networks and formal languages, 2020. doi: 10.48550/ARXIV.2006.01338.

R. L. Adler, A. G. Konheim, and M. H. McAndrew. Topological entropy. *Transactions of the American Mathematical Society*, 114(2):309–319, 1965.

N. A. Akar, B. Cumming, V. Karakasis, A. Küsters, W. Klijn, A. Peyser, and S. Yates. Arbor —a morphologically-detailed neural network simulation library for contemporary high-performance computing architectures. In *2019 27th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP)*, pages 274–282. IEEE, 2019.

D. J. Amit and N. Brunel. Model of global spontaneous activity and local structured activity during delay periods in the cerebral cortex. *Cereb. Cortex*, 7:237–252, Apr. 1997. doi: 10.1093/cercor/7.3.237.

S. K. Andersen, S. A. Hillyard, and M. M. Müller. Attention facilitates multiple stimulus features in parallel in human visual cortex. *Current Biology*, 18(13):1006–1009, Jul. 2008. doi: 10.1016/j.cub.2008.06.030.

J. S. Anderson, M. A. Ferguson, M. Lopez-Larson, and D. Yurgelun-Todd. Topographic maps of multisensory attention. *Proceedings of the National Academy of Sciences*, 107 (46):20110–20114, Nov. 2010. doi: 10.1073/pnas.1011616107.

A. Araujo, W. Norris, and J. Sim. Computing receptive fields of convolutional neural networks. *Distill*, 4(11), Nov. 2019. doi: 10.23915/distill.00021.

A. Arieli, A. Sterkin, A. Grinvald, and A. Aertsen. Dynamics of ongoing activity: explanation of the large variability in evoked cortical responses. *Science*, 273(5283): 1868–1871, 1996.

T. Asabuki and T. Fukai. Somatodendritic consistency check for temporal feature segmentation. *Nature Communications*, 11(1):1–13, Mar. 2020. doi: 10.1038/s41467-020-15367-w.

T. Asabuki, P. Kokate, and T. Fukai. Neural circuit mechanisms of hierarchical sequence learning tested on large-scale recording data. *PLOS Computational Biology*, 18(6): e1010214, Jun. 2022. doi: 10.1371/journal.pcbi.1010214.

Y. Aviel, C. Mehring, M. Abeles, and D. Horn. On embedding synfire chains in a balanced network. *Neural Comput.*, 15(6):1321–1340, 2003.

Y. Aviel, D. Horn, and M. Abeles. Memory capacity of balanced networks. *Neural Comput.*, 17(3):691–713, Mar 2005. Comparative Study.

B. Babadi and H. Sompolinsky. Sparseness and expansion in sensory representations. *Neuron*, 83(5):1213–1226, Sep. 2014. doi: 10.1016/j.neuron.2014.07.035.

A. Babloyantz, J. Salazar, and C. Nicolis. Evidence of chaotic dynamics of brain activity during the sleep cycle. *Physics Letters A*, 111(3):152–156, Sep. 1985. doi: 10.1016/0375-9601(85)90444-x.

T. M. Bailey and E. M. Pothos. AGL StimSelect: Software for automated selection of stimuli for artificial grammar learning. *Behavior Research Methods*, 40(1):164–176, Feb. 2008. doi: 10.3758/brm.40.1.164.

F. Balci, D. Freestone, and C. R. Gallistel. Risk assessment in man and mouse. *Proceedings of the National Academy of Sciences*, 106(7):2459–2463, Feb. 2009. doi: 10.1073/pnas.0812709106.

P. Ball. *Shapes: nature's patterns: a tapestry in three parts.* OUP Oxford, 2009.

O. Barak, M. Rigotti, and S. Fusi. The sparseness of mixed selectivity neurons controls the generalization–discrimination trade-off. *Journal of Neuroscience*, 33(9):3844–3856, 2013. ISSN 0270-6474. doi: 10.1523/JNEUROSCI.2753-12.2013.

T. D. Barnes, Y. Kubota, D. Hu, D. Z. Jin, and A. M. Graybiel. Activity of striatal neurons reflects dynamic encoding and recoding of procedural memories. *Nature*, 437 (7062):1158–1161, Oct. 2005. doi: 10.1038/nature04053.

A. M. Bastos, W. M. Usrey, R. A. Adams, G. R. Mangun, P. Fries, and K. J. Friston. Canonical Microcircuits for Predictive Coding. *Neuron*, 76(4):695–711, 11 2012. ISSN 0896-6273. doi: 10.1016/J.NEURON.2012.10.038.

G. Beckers. pyagl: A python library for statistical learning (SL) and artificial grammar learning (AGL) analyses. `https://github.com/gbeckers/pyagl`. [Accessed 19-Nov-2022].

J. A. Bednar and S. P. Wilson. Cortical Maps. *The Neuroscientist*, 22(6):604–617, 2016. ISSN 1073-8584. doi: 10.1177/1073858415597645. PMID: 26290447.

T. Bekolay, J. Bergstra, E. Hunsberger, T. DeWolf, T. C. Stewart, D. Rasmussen, X. Choo, A. R. Voelker, and C. Eliasmith. Nengo: a python tool for building large-scale functional brain models. *Frontiers in Neuroinformatics*, 7, 2013. doi: 10.3389/fninf.2013.00048.

Y. Bengio and Y. LeCun. Scaling Learning Algorithms towards AI. 2007.

Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166, 3 1994. ISSN 10459227. doi: 10.1109/72.279181.

Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intel.*, 35(8):1798–1828, aug 2013. doi: 10.1109/tpami.2013.50.

Y. Bengio, D.-H. Lee, J. Bornschein, T. Mesnard, and Z. Lin. Towards biologically plausible deep learning, 2015. doi: 10.48550/ARXIV.1502.04156.

F. C. Y. Benureau and N. P. Rougier. Re-run, repeat, reproduce, reuse, replicate: Transforming code into scientific contributions. *Frontiers in Neuroinformatics*, 2018. ISSN 1662-5196. doi: 10.3389/fninf.2017.00069.

L. Besançon, E. Bik, J. Heathers, and G. Meyerowitz-Katz. Correction of scientific literature: Too little, too late! *PLOS Biology*, 20(3):e3001572, Mar. 2022. doi: 10.1371/journal.pbio.3001572.

G. Bi and M. Poo. Synaptic modification by correlated activity: Hebb's postulate revisited. *Annu. Rev. Neurosci.*, 24:139–66, 2001.

Y. N. Billeh, B. Cai, S. L. Gratiy, K. Dai, R. Iyer, N. W. Gouwens, R. Abbasi-Asl, X. Jia, J. H. Siegle, S. R. Olsen, et al. Systematic integration of structural and functional data into multi-scale models of mouse primary visual cortex. *Neuron*, 106(3):388–403.e18, 2020. doi: https://doi.org/10.1016/j.neuron.2020.01.040.

T. V. P. Bliss and T. Lomo. Long-lasting potentation of synaptic transmission in the dendate area of anaesthetized rabbit following stimulation of the perforant path. *Journal of Physiology*, 232:331–356, 1973.

E. M. Bollt and M. A. Jones. The Complexity of Artificial Grammars. *Nonlinear Dynamics, Psychology, and Life Sciences*, 4(2):153–168, Apr. 2000. ISSN 1573-6652. doi: 10.1023/A:1009524428448.

F. Bordignon. Self-correction of science: a comparative study of negative citations and post-publication peer review. *Scientometrics*, 124(2):1225–1239, Jun. 2020. doi: 10.1007/s11192-020-03536-z.

B. Borghuis, D. Tadin, M. Lankheet, J. Lappin, and W. van de Grind. Temporal limits of visual motion processing: Psychophysics and neurophysiology. *Vision*, 3(1):5, Jan. 2019. doi: 10.3390/vision3010005.

Y. Bouhadjar, D. J. Wouters, M. Diesmann, and T. Tetzlaff. Sequence learning, prediction, and replay in networks of spiking neurons. *PLOS Comput. Biol.*, 18(6):e1010233, Jun. 2022. doi: 10.1371/journal.pcbi.1010233.

Y. Bouhadjar, D. J. Wouters, M. Diesmann, and T. Tetzlaff. Coherent noise enables probabilistic sequence replay in spiking neuronal networks. *PLOS Computational Biology*, 19(5):e1010989, May 2023. doi: 10.1371/journal.pcbi.1010989.

M. A. Bourjaily and P. Miller. Dynamic afferent synapses to decision-making networks improve performance in tasks requiring stimulus associations and discriminations. *Journal of Neurophysiology*, 108(2):513–527, jul 2012. doi: 10.1152/jn.00806.2011.

V. Braitenberg and A. Schüz. *Anatomy of the Cortex: Statistics and Geometry.* Springer-Verlag, Berlin, Heidelberg, New York, 1991. ISBN 3-540-53233-1.

J. A. Brefczynski-Lewis, R. Datta, J. W. Lewis, and E. A. DeYoe. The topography of visuospatial attention as revealed by a novel visual field mapping technique. *Journal of Cognitive Neuroscience*, 21(7):1447–1460, 2009. doi: 10.1162/jocn.2009.21005. PMID: 18752412.

R. Brette. Philosophy of the spike: Rate-based vs. spike-based theories of the brain. *Frontiers in Systems Neuroscience*, 9:151, 2015. doi: 10.3389/fnsys.2015.00151.

S. P. Brown and S. Hestrin. Intracortical circuits of pyramidal neurons reflect their long-range axonal targets. *Nature*, 457(7233):1133–1136, 2 2009. ISSN 0028-0836. doi: 10.1038/nature07658.

N. Brunel. Dynamics of sparsely connected networks of excitatory and inhibitory spiking neurons. *J. Comput. Neurosci.*, 8(3):183–208, 2000. doi: 10.1023/a:1008925309027.

N. Brunel. Dynamics of neural networks. In *Principles of Neural Coding*, page 489–512. CRC Press, May 2013. doi: 10.1201/b14756-29.

Z. Brzosko, S. B. Mierau, and O. Paulsen. Neuromodulation of spike-timing-dependent plasticity: Past, present, and future. *Neuron*, 103(4):563–581, aug 2019. doi: 10.1016/j.neuron.2019.05.041.

E. Bullmore and O. Sporns. Complex brain networks: graph theoretical analysis of structural and functional systems. *Nature Reviews Neuroscience*, 10(3):186–198, Feb. 2009. ISSN 1471-003X. doi: 10.1038/nrn2575.

D. V. Buonomano and R. Laje. Population clocks: motor timing with neural dynamics. *Trends in Cognitive Sciences*, 14(12):520–527, Dec. 2010. doi: 10.1016/j.tics.2010.09.002.

D. V. Buonomano and W. Maass. State-dependent computations: spatiotemporal processing in cortical networks. *Nature Reviews Neuroscience*, 10(2):113–125, Jan. 2009. doi: 10.1038/nrn2558.

A. N. Burkitt. A review on the integrate-and-fire neuron model: I. homogenous synaptic input. *Biol. Cybern.*, 95(1):1–19, 2006.

G. Buzsáki, C. Geisler, D. A. Henze, and X.-J. Wang. Interneuron Diversity series: Circuit complexity and axon wiring economy of cortical interneurons. *Trends in Neurosciences*, 27(4):186–193, 4 2004. ISSN 01662236. doi: 10.1016/j.tins.2004.02.007.

R. Cahuantzi, X. Chen, and S. Güttel. A comparison of lstm and gru networks for learning symbolic sequences, 2021. doi: 10.48550/ARXIV.2107.02248.

C. B. Calderon, T. Verguts, and M. J. Frank. Thunderstruck: The ACDC model of flexible sequences and rhythms in recurrent neural circuits. *PLOS Computational Biology*, 18(2):e1009854, Feb. 2022. doi: 10.1371/journal.pcbi.1009854.

J. Cannon, N. Kopell, T. Gardner, and J. Markowitz. Neural sequence generation using spatiotemporal patterns of inhibition. *PLOS Computational Biology*, 11(11):e1004581, Nov. 2015. doi: 10.1371/journal.pcbi.1004581.

M. Carandini and D. Heeger. Normalization as a canonical neural computation. *Nature Reviews Neuroscience*, 13(November):1–12, 2012. ISSN 1471-003X. doi: 10.1038/nrn3136.

F. Charlier, M. Weber, D. Izak, E. Harkin, M. Magnus, J. Lalli, L. Fresnais, M. Chan, N. Markov, O. Amsalem, S. Proost, A. Krasoulis, getzze, and S. Repplinger. Statannotations, Oct. 2022. doi: 10.5281/zenodo.7213391.

K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *ArXiv*, 2014.

N. Chomsky. Three models for the description of language. *IEEE Transactions on Information Theory*, 2(3):113–124, Sep. 1956. doi: 10.1109/tit.1956.1056813.

N. Chomsky. On certain formal properties of grammars. *Information and Control*, 2(2): 137–167, jun 1959. doi: 10.1016/s0019-9958(59)90362-6.

S. Clavagnier, A. Falchier, and H. Kennedy. Long-distance feedback projections to area v1: Implications for multisensory integration, spatial awareness, and visual consciousness. *Cognitive, Affective, & Behavioral Neuroscience*, 4(2):117–126, Jun 2004. ISSN 1531-135X. doi: 10.3758/CABN.4.2.117.

C. Clopath, L. Büsing, E. Vasilaki, and W. Gerstner. Connectivity reflects coding: a model of voltage-based STDP with homeostasis. *Nat. Neurosci.*, 13:344–352, 2010.

J. Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46, Apr. 1960. doi: 10.1177/001316446002000104.

M. Colombo and M. Knauff. Editors' review and introduction: Levels of explanation in cognitive science: From molecules to culture. *Topics in Cognitive Science*, 12(4): 1224–1240, May 2020. doi: 10.1111/tops.12503.

I. Cone and H. Z. Shouval. Learning precise spatiotemporal sequences via biophysically realistic learning rules in a modular, spiking network. *eLife*, 10:e63751, 2021.

I. Cone and H. Z. Shouval. Correction: Learning precise spatiotemporal sequences via biophysically realistic learning rules in a modular, spiking network. *eLife*, 12, Mar. 2023. doi: 10.7554/elife.87507.

M. T. Cook, C. M. Chubala, and R. K. Jamieson. AGSuite: Software to conduct feature analysis of artificial grammar learning performance. *Behavior Research Methods*, 49 (5):1639–1651, Jun. 2017. doi: 10.3758/s13428-017-0899-1.

R. Cooper and J. Fox. COGENT: A visual design environment for cognitive modeling. *Behavior Research Methods, Instruments and Computers*, 30(4):553–564, Dec. 1998. doi: 10.3758/bf03209472.

B. Corominas-Murtra, J. Goñi, R. V. Solé, and C. Rodríguez-Caso. On the origins of hierarchy in complex networks. *Proceedings of the National Academy of Sciences*, 110 (33):13316–13321, Jul. 2013. doi: 10.1073/pnas.1300832110.

N. Cortes and C. van Vreeswijk. Pulvinar thalamic nucleus allows for asynchronous spike propagation through the cortex. *Frontiers in Computational Neuroscience*, 9, May 2015. doi: 10.3389/fncom.2015.00060.

T. M. Cover and J. A. Thomas. *Elements of Information Theory.* Wiley, New York, 1991.

B. Cramer, Y. Stradmann, J. Schemmel, and F. Zenke. The heidelberg spiking data sets for the systematic evaluation of spiking neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 33(7):2744–2757, Jul. 2022. doi: 10.1109/tnnls.2020.3044364.

F. Crick. The recent excitement about neural networks. *Nature*, 337(6203):129–132, 1989.

A. Davison, D. Brüderle, J. Eppler, J. Kremkow, E. Muller, D. Pecevski, L. Perrinet, and P. Yger. PyNN: a common interface for neuronal network simulators. *Frontiers in Neuroinformatics*, 2(11), 2008. doi:10.3389/neuro.11.011.2008.

M. de Kamps, V. Baier, J. Drever, M. Dietz, L. Mösenlechner, and F. van der Felde. The state of MIIND. *Neural Networks*, 21(8):1164–1181, Oct. 2008. doi: 10.1016/j.neunet.2008.07.006.

M. H. de Vries, K. M. Petersson, S. Geukes, P. Zwitserlood, and M. H. Christiansen. Processing multiple non-adjacent dependencies: evidence from sequence learning. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 367(1598):2065–2076, Jul. 2012. doi: 10.1098/rstb.2011.0414.

S. E. J. de Vries, J. A. Lecoq, M. A. Buice, P. A. Groblewski, G. K. Ocker, M. Oliver, D. Feng, N. Cain, P. Ledochowitsch, D. Millman, K. Roll, M. Garrett, T. Keenan, L. Kuan, S. Mihalas, S. Olsen, C. Thompson, W. Wakeman, J. Waters, D. Williams, C. Barber, N. Berbesque, B. Blanchard, N. Bowles, S. D. Caldejon, L. Casal, A. Cho, S. Cross, C. Dang, T. Dolbeare, M. Edwards, J. Galbraith, N. Gaudreault, T. L. Gilbert, F. Griffin, P. Hargrave, R. Howard, L. Huang, S. Jewell, N. Keller, U. Knoblich, J. D. Larkin, R. Larsen, C. Lau, E. Lee, F. Lee, A. Leon, L. Li, F. Long, J. Luviano, K. Mace, T. Nguyen, J. Perkins, M. Robertson, S. Seid, E. Shea-Brown,

J. Shi, N. Sjoquist, C. Slaughterbeck, D. Sullivan, R. Valenza, C. White, A. Williford, D. M. Witten, J. Zhuang, H. Zeng, C. Farrell, L. Ng, A. Bernard, J. W. Phillips, R. C. Reid, and C. Koch. A large-scale standardized physiological survey reveals functional organization of the mouse visual cortex. *Nature Neuroscience*, 23(1):138–151, Jan 2020. ISSN 1546-1726. doi: 10.1038/s41593-019-0550-9.

D. Debanne, Y. Inglebert, and M. Russier. Plasticity of intrinsic neuronal excitability. *Current Opinion in Neurobiology*, 54:73–82, Feb. 2019. doi: 10.1016/j.conb.2018.09.001.

S. Dehaene, F. Meyniel, C. Wacongne, L. Wang, and C. Pallier. The neural representation of sequences: from transition probabilities to algebraic patterns and linguistic trees. *Neuron*, 88(1):2–19, 2015.

N. Dehghani, A. Peyrache, B. Telenczuk, M. Le Van Quyen, E. Halgren, S. S. Cash, N. G. Hatsopoulos, and A. Destexhe. Dynamic balance of excitation and inhibition in human and monkey neocortex. *Scientific Reports*, 6, Mar. 2016. doi: 10.1038/srep23176.

G. Delétang, A. Ruoss, J. Grau-Moya, T. Genewein, L. K. Wenliang, E. Catt, C. Cundy, M. Hutter, S. Legg, J. Veness, and P. A. Ortega. Neural networks and the chomsky hierarchy, 2022. doi: 10.48550/ARXIV.2207.02098.

S. Denève and C. K. Machens. Efficient codes and balanced networks. *Nat. Neurosci.*, 19(3):375–382, 2016. doi: 10.1038/nn.4243.

A. Destexhe. Self-sustained asynchronous irregular states and up/down states in thalamic, cortical and thalamocortical networks of nonlinear integrate-and-fire neurons. *J. Comput. Neurosci.*, 27:493–506, 2009.

A. Destexhe and E. Marder. Plasticity in single neuron and circuit computations. *Nature*, 431:789–795, oct 14th 2004.

A. Destexhe, M. Rudolph, and D. Pare. The high-conductance state of neocortical neurons in vivo. *Nat. Rev. Neurosci.*, 4:739–751, 2003.

P. Deutsch. Rfc1951: Deflate compressed data format specification version 1.3, 1996.

M. Diesmann and M.-O. Gewaltig. NEST: An environment for neural systems simulations. In T. Plesser and V. Macho, editors, *Forschung und wisschenschaftliches Rechnen, Beiträge zum Heinz-Billing-Preis 2001*, volume 58 of *GWDG-Bericht*, pages 43–70. Ges. für Wiss. Datenverarbeitung, Göttingen, 2002.

M. Diesmann, M.-O. Gewaltig, and A. Aertsen. Stable propagation of synchronous spiking in cortical neural networks. *Nature*, 402(6761):529–533, 1999.

P. F. Dominey. Recurrent temporal networks and language acquisition—from corticostriatal neurophysiology to reservoir computing. *Frontiers in Psychology*, 4, 2013. doi: 10.3389/fpsyg.2013.00500.

R. J. Douglas and K. A. C. Martin. Neuronal circuits of the neocortex. *Annu. Rev. Neurosci.*, 27:419–451, 2004.

R. Duarte. Expansion and State-Dependent Variability along Sensory Processing Streams. *The Journal of Neuroscience*, 35(19):7315–7316, May 2015. ISSN 0270-6474. doi: 10.1523/JNEUROSCI.0874-15.2015.

R. Duarte. Technical report: Functional neural architectures, 2021. Unpublished.

R. Duarte and A. Morrison. Leveraging heterogeneity for neural computation with fading memory in layer 2/3 cortical microcircuits. *PLOS Comput. Biol.*, 15(4):e1006781, 2019.

R. Duarte, P. Series, and A. Morrison. Self-Organized Artificial Grammar Learning in Spiking Neural Networks Self-Organized Artificial Grammar Learning in Spiking Neural Networks. *Proceedings of the Annual Meeting of the Cognitive Science Society*, 36(36), 2014.

R. Duarte, A. Seeholzer, K. Zilles, and A. Morrison. Synaptic patterning and the timescales of cortical dynamics. *Current Opinion in Neurobiology*, 43:156–165, 4 2017a. ISSN 18736882. doi: 10.1016/j.conb.2017.02.007.

R. Duarte, B. Zajzon, and A. Morrison. Neural Microcircuit Simulation And Analysis Toolkit. *Zenodo*, 2017b. doi: 10.5281/ZENODO.582645.

R. Duarte, M. Uhlmann, D. den van Broek, H. Fitz, K. M. Petersson, and A. Morrison. Encoding symbolic sequences with spiking neural reservoirs. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, July 2018. ISBN 9781509060146. doi: 10.1109/IJCNN.2018.8489114.

R. Duarte, B. Zajzon, T. Schulte to Brinke, and A. Morrison. Functional neural architectures, 2021. doi: 10.5281/ZENODO.5752597.

R. C. Duarte and A. Morrison. Dynamic stability of sequential stimulus representations in adapting neuronal networks. *Frontiers in Computational Neuroscience*, 8(October): 124, 2014. ISSN 1662-5188. doi: 10.3389/fncom.2014.00124.

S. Dura-Bernal, E. Y. Griffith, A. Barczak, M. N. O'Connell, T. McGinnis, C. E. Schroeder, W. W. Lytton, P. Lakatos, and S. A. Neymotin. Data-driven multiscale model of macaque auditory thalamocortical circuits reproduces in vivo dynamics. *bioRxiv*, page 2022.02.03.479036, Apr. 2022.

A. Ecker, B. Bagi, E. Vértes, O. Steinbach-Németh, M. R. Karlócai, O. I. Papp, I. Miklós, N. Hájos, T. F. Freund, A. I. Gulyás, and S. Káli. Hippocampal sharp wave-ripples and the associated sequence replay emerge from structured synaptic interactions in a network model of area CA3. *eLife*, 11, Jan. 2022. doi: 10.7554/elife.71850.

A. S. Ecker, P. Berens, G. A. Keliris, M. Bethge, and N. K. Logothetis. Decorrelated neuronal firing in cortical microcircuits. *Science*, 327(5965):584–587, Jan. 2010. doi: 10.1126/science.1179867.

G. T. Einevoll, A. Destexhe, M. Diesmann, S. Grün, V. Jirsa, M. de Kamps, M. Migliore, T. V. Ness, H. E. Plesser, and F. Schürmann. The Scientific Case for Brain Simulations. *Neuron*, 102(4):735–744, 2019. ISSN 0896-6273. doi: 10.1016/j.neuron.2019.03.027.

C. Eliasmith. *How to Build a Brain: A Neural Architecture for Biological Cognition*. Oxford University Press, 06 2013. ISBN 9780199794546. doi: 10.1093/acprof:oso/9780199794546.001.0001.

C. Eliasmith and C. H. Anderson. *Neural engineering: Computation, representation, and dynamics in neurobiological systems*. MIT press, 2003.

C. Eliasmith, T. C. Stewart, X. Choo, T. Bekolay, T. DeWolf, Y. Tang, and D. Rasmussen. A large-scale model of the functioning brain. *Science*, 338(6111):1202–1205, Nov. 2012. doi: 10.1126/science.1225266.

K. O. Ellefsen, J.-B. Mouret, and J. Clune. Neural modularity helps organisms evolve to learn new skills without forgetting old skills. *PLOS Computational Biology*, 11(4): e1004128, Apr. 2015. doi: 10.1371/journal.pcbi.1004128.

J. L. Elman. Finding Structure in Time. *Cognitive Science*, 14(2):179–211, 3 1990. ISSN 03640213. doi: 10.1207/s15516709cog1402_1.

J. L. Elman. Distributed representations, simple recurrent networks, and grammatical structure. *Machine Learning*, 7(2-3):195–225, Sep. 1991. doi: 10.1007/bf00114844.

P. Enel, E. Procyk, R. Quilodran, and P. F. Dominey. Reservoir computing properties of neural dynamics in prefrontal cortex. *PLOS Computational Biology*, 12(6):e1004967, Jun. 2016. doi: 10.1371/journal.pcbi.1004967.

O. Eriksson, U. S. Bhalla, K. T. Blackwell, S. M. Crook, D. Keller, A. Kramer, M.-L. Linne, A. Saudargienė, R. C. Wade, and J. H. Kotaleski. Combining hypothesis- and data-driven neuroscience modeling in FAIR workflows. *eLife*, 11, Jul. 2022. doi: 10.7554/elife.69013.

A. A. Faisal, L. P. Selen, and D. M. Wolpert. Noise in the nervous system. *Nat. Rev. Neurosci.*, 9(4):292–303, 2008.

T. Fardet, S. B. Vennemo, J. Mitchell, H. Mørk, S. Graber, J. Hahne, S. Spreizer, R. Deepu, G. Trensch, P. Weidel, J. Jordan, J. M. Eppler, D. Terhorst, A. Morrison, C. Linssen, A. Antonietti, K. Dai, A. Serenko, B. Cai, P. Kubaj, R. Gutzen, H. Jiang, I. Kitayama, B. Jürgens, and H. E. Plesser. NEST 2.20.0, 2020. doi: 10.5281/ZENODO.3605514.

M. Fauth and C. Tetzlaff. Opposing effects of neuronal activity on structural plasticity. *Frontiers in Neuroanatomy*, 10, jun 2016. doi: 10.3389/fnana.2016.00075.

M. S. Fee, P. P. Mitra, and D. Kleinfeld. Variability of extracellular spike waveforms of cortical neurons. *J. Neurophysiol.*, 76(6):3823–3833, Dec 1996.

D. J. Felleman and D. C. Van Essen. Distributed hierarchical processing in the primate cerebral cortex. *Cereb. Cortex*, 1:1–47, 1991.

M. Ferrante, M. Migliore, and G. A. Ascoli. Feed-forward inhibition as a buffer of the neuronal input-output relation. *Proceedings of the National Academy of Sciences*, 106 (42):18004–18009, 2009. ISSN 0027-8424. doi: 10.1073/pnas.0904784106.

I. R. Fiete, W. Senn, C. Z. H. Wang, and R. H. R. Hahnloser. Spike-time-dependent plasticity and heterosynaptic competition organize networks to produce long scale-free sequences of neural activity. *Neuron*, 65:563–576, 2010.

E. Fino and R. Yuste. Dense inhibitory connectivity in neocortex. *Neuron*, 69(6):1188–1203, 2011. ISSN 0896-6273. doi: 10.1016/j.neuron.2011.02.025.

D. Fioravante and W. G. Regehr. Short-term forms of presynaptic plasticity. *Current Opinion in Neurobiology*, 21(2):269–274, apr 2011. doi: 10.1016/j.conb.2011.02.003.

W. T. Fitch and A. D. Friederici. Artificial grammar learning meets formal language theory: an overview. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 367(1598):1933–1955, Jul. 2012. doi: 10.1098/rstb.2012.0103.

W. T. Fitch and M. D. Martins. Hierarchical processing in music, language, and action: Lashley revisited. *Annals of the New York Academy of Sciences*, 1316(1):87–104, Apr. 2014. doi: 10.1111/nyas.12406.

W. T. Fitch, A. D. Friederici, and P. Hagoort. Pattern perception and computational complexity: introduction to the special issue. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 367(1598):1925–1932, Jul. 2012. doi: 10.1098/rstb.2012.0099.

H. Fitz, M. Uhlmann, D. van den Broek, R. Duarte, P. Hagoort, and K. M. Petersson. Neuronal spike-rate adaptation supports working memory in language processing. *Proceedings of the National Academy of Sciences*, 117(34):20881–20889, Aug. 2020. doi: 10.1073/pnas.2000222117.

J. A. Fodor and Z. W. Pylyshyn. Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28(1-2):3–71, Mar. 1988. doi: 10.1016/0010-0277(88)90031-5.

J. Fonollosa, E. Neftci, and M. Rabinovich. Learning of chunking sequences in cognition and behavior. *PLOS Computational Biology*, 11(11):e1004592, Nov. 2015. doi: 10.1371/journal.pcbi.1004592.

D. J. Foster and M. A. Wilson. Reverse replay of behavioural sequences in hippocampal place cells during the awake state. *Nature*, 440(7084):680–683, 2006.

N. Fourcaud and N. Brunel. Dynamics of the firing probability of noisy integrate-and-fire neurons. *Neural Comput.*, 14:2057–2110, 2002. doi: 10.1162/089976602320264015.

M. Fowler. Bliki: Inversion of control, Jun. 2005.

N. Frémaux and W. Gerstner. Neuromodulated spike-timing-dependent plasticity, and theory of three-factor learning rules. *Frontiers in Neural Circuits*, 9, Jan. 2016. doi: 10.3389/fncir.2015.00085.

N. Frémaux, H. Sprekeler, and W. Gerstner. Functional requirements for reward-modulated spike-timing-dependent plasticity. *J. Neurosci.*, 30(40):13326–13337, 2010.

U. Frey and R. Morris. Synaptic tagging and long-term potentiation. *Nature*, 385: 533–536, 1997.

K. Friston. Beyond Phrenology: What Can Neuroimaging Tell Us About Distributed Circuitry? *Annual Review of Neuroscience*, 25(1):221–250, 3 2002. ISSN 0147-006X. doi: 10.1146/annurev.neuro.25.112701.142846.

K. Friston and S. Kiebel. Predictive coding under the free-energy principle. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 364(1521):1211–1221, 2009.

R. C. Froemke. Plasticity of cortical excitatory-inhibitory balance. *Annual Review of Neuroscience*, 38(1):195–219, Jul. 2015. doi: 10.1146/annurev-neuro-071714-034002.

N. C. Gajic and E. Shea-Brown. Neutral stability, rate propagation, and critical branching in feedforward networks. *ArXiv*, pages 1210.8406v1 [q–bio.NC], 2012.

M. I. Garrido, J. M. Kilner, K. E. Stephan, and K. J. Friston. The mismatch negativity: A review of underlying mechanisms. *Clinical Neurophysiology*, 120(3):453–463, Mar. 2009. doi: 10.1016/j.clinph.2008.11.029.

M. Garzon and S. Franklin. Neural computability. In O. Omidvar, editor, *Progress In Neural Networks*, volume 1, pages 127–146. Ablex Publishing Corporation, 1991.

J. P. Gavornik and M. F. Bear. Learned spatiotemporal sequence recognition and prediction in primary visual cortex. *Nat. Neurosci.*, 17(5):732, 2014.

J. Gerhart and M. Kirschner. The theory of facilitated variation. *Proceedings of the National Academy of Sciences*, 104(suppl_1):8582–8589, May 2007. doi: 10.1073/pnas.0701035104.

F. Gers and E. Schmidhuber. LSTM recurrent networks learn simple context-free and context-sensitive languages. *IEEE Transactions on Neural Networks*, 12(6):1333–1340, 2001. ISSN 10459227. doi: 10.1109/72.963769.

W. Gerstner and R. Naud. How good are neuron models? *Science*, 326(5951):379–380, 2009.

W. Gerstner, H. Sprekeler, and G. Deco. Theory and simulation in neuroscience. *Science*, 338(6103):60–65, 2012.

W. Gerstner, W. M. Kistler, R. Naud, and L. Paninski. *Neuronal Dynamics. From Single Neurons to Networks and Models of Cognition.* Cambridge University Press, Cambridge, 2014.

W. Gerstner, M. Lehmann, V. Liakoni, D. Corneil, and J. Brea. Eligibility traces and plasticity on behavioral time scales: Experimental support of NeoHebbian three-factor learning rules. *Front. Neural Circuits*, 12, jul 2018. doi: 10.3389/fncir.2018.00053.

G. G. Globus. Toward a noncomputational cognitive neuroscience. *J. Cogn. Neurosci.*, 4(4):299–300, 1992.

F. Gobet, P. C. Lane, S. Croker, P. C.-H. Cheng, G. Jones, I. Oliver, and J. M. Pine. Chunking mechanisms in human learning. *Trends in Cognitive Sciences*, 5(6):236–243, Jun. 2001. doi: 10.1016/s1364-6613(00)01662-4.

C. Goble, S. Cohen-Boulakia, S. Soiland-Reyes, D. Garijo, Y. Gil, M. R. Crusoe, K. Peters, and D. Schober. FAIR computational workflows. *Data Intelligence*, 2(1-2): 108–121, Jan. 2020. doi: 10.1162/dint_a_00033.

M. Goldman. Memory without feedback in a neural network. *Neuron*, 61(4):621–634, 2009.

V. Goudar and D. V. Buonomano. Encoding sensory and motor patterns as time-invariant trajectories in recurrent neural networks. *eLife*, 7, Mar. 2018. doi: 10.7554/elife.31134.

S. Haeusler and W. Maass. A statistical analysis of information-processing properties of lamina-specific cortical microcircuit models. *Cereb. Cortex*, 17(1):149–162, Jan 2007. doi: 10.1093/cercor/bhj132.

D. J. Hagler and M. I. Sereno. Spatial maps in frontal and prefrontal cortex. *NeuroImage*, 29(2):567–577, jan 2006. ISSN 10538119. doi: 10.1016/j.neuroimage.2005.08.058.

R. H. Hahnloser, A. A. Kozhevnikov, and M. S. Fee. An ultra-sparse code underlies the generation of neural sequences in a songbird. *Nature*, 419(6902):65–70, Sep 2002.

B. Haider, A. Duque, A. R. Hasenstaub, and D. A. McCormick. Neocortical network activity in vivo is generated through a dynamic balance of excitation and inhibition. *J. Neurosci.*, 26(17):4535–4545, 2006.

T. F. Hansen. Is modularity necessary for evolvability? *Biosystems*, 69(2-3):83–94, May 2003. doi: 10.1016/s0303-2647(02)00132-6.

N. F. Hardy and D. V. Buonomano. Neurocomputational models of interval and pattern timing. *Current Opinion in Behavioral Sciences*, 8:250–257, Apr. 2016. doi: 10.1016/j.cobeha.2016.01.012.

N. F. Hardy and D. V. Buonomano. Encoding time in feedforward trajectories of a recurrent neural network model. *Neural Computation*, 30(2):378–396, Feb. 2018. doi: 10.1162/neco_a_01041.

K. D. Harris and T. D. Mrsic-Flogel. Cortical connectivity and sensory coding. *Nature*, 503(7474):51–8, 2013. ISSN 1476-4687. doi: 10.1038/nature12654.

K. D. Harris and G. M. G. Shepherd. The neocortical circuit: themes and variations. *Nat. Neurosci.*, 18(2):170–181, Jan. 2015. ISSN 1097-6256. doi: 10.1038/nn.3917.

K. D. Harris and A. Thiele. Cortical state and attention. *Nat. Rev. Neurosci.*, 12: 509–523, September 2011. doi: 10.1038/nrn3084.

E. Hay, S. Hill, F. Schuermann, H. Markram, and I. Segev. Models of neocortical layer 5b pyramidal cells capturing a wide range of dendritic and perisomatic active properties. *PLOS Comput. Biol.*, 7(7), july 2011.

K. He, M. Huertas, S. Z. Hong, X. Tie, J. W. Hell, H. Shouval, and A. Kirkwood. Distinct eligibility traces for LTP and LTD in cortical synapses. *Neuron*, 88(3):528–538, nov 2015. doi: 10.1016/j.neuron.2015.09.037.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

D. O. Hebb. *The organization of behavior: A neuropsychological theory.* John Wiley & Sons, New York, 1949.

J. Hegdé and D. J. Felleman. Reappraising the functional implications of the primate visual anatomical hierarchy. *The Neuroscientist*, 13(5):416–421, Oct. 2007. doi: 10.1177/1073858407305201.

S. Heitmann, M. J. Aburn, and M. Breakspear. The brain dynamics toolbox for matlab. *Neurocomputing*, 315:82–88, 2018.

M. Helias, T. Tetzlaff, and M. Diesmann. Echoes in correlated neural systems. *New J. Phys.*, 15:023002, 2013. doi: 10.1088/1367-2630/15/2/023002.

S. Henin, N. B. Turk-Browne, D. Friedman, A. Liu, P. Dugan, A. Flinker, W. Doyle, O. Devinsky, and L. Melloni. Learning hierarchical sequence representations across human cortex and hippocampus. *Science Advances*, 7(8), Feb. 2021. doi: 10.1126/sciadv.abc4530.

A. Henry, F. Monéger, A. Samal, and O. C. Martin. Network function shapes network structure: the case of the arabidopsis flower organ specification genetic network. *Molecular BioSystems*, 9(7):1726, 2013. doi: 10.1039/c3mb25562j.

M. A. Herman, B. R. Aiello, J. D. DeLong, H. Garcia-Ruiz, A. L. González, W. Hwang, C. McBeth, E. A. Stojković, M. A. Trakselis, and N. Yakoby. A Unifying Framework for Understanding Biological Structures and Functions Across Levels of Biological Organization. *Integrative and Comparative Biology*, 61(6):2038–2047, Dec. 2021. ISSN 1540-7063. doi: 10.1093/icb/icab167.

C. C. Hilgetag and A. Goulas. Is the brain really a small-world network? *Brain Struct Funct*, 221(4):2361–2366, apr 2015. doi: 10.1007/s00429-015-1035-6.

C. C. Hilgetag and A. Goulas. 'hierarchy' in the organization of brain networks. *Phil. Trans. R. Soc. B*, 375(1796):20190319, feb 2020. doi: 10.1098/rstb.2019.0319.

C. C. Hilgetag, G. A. Burns, M. A. O'Neill, J. W. Scannell, and M. P. Young. Anatomical connectivity defines the organization of clusters of cortical areas in the macaque monkey and the cat. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, 355(1393):91–110, 1 2000. ISSN 0962-8436. doi: 10.1098/rstb.2000.0551.

K. D. Himberger, H.-Y. Chien, and C. J. Honey. Principles of temporal processing across the cortical hierarchy. *Neuroscience*, 389:161–174, 2018. ISSN 0306-4522. doi: https://doi.org/10.1016/j.neuroscience.2018.04.030. Sensory Sequence Processing in the Brain.

X. Hinaut and P. F. Dominey. Real-time parallel processing of grammatical structure in the fronto-striatal system: A recurrent network simulation study using reservoir computing. *PLoS ONE*, 8(2):e52946, Feb. 2013. doi: 10.1371/journal.pone.0052946.

X. Hinaut, F. Lance, C. Droin, M. Petit, G. Pointeau, and P. F. Dominey. Corticostriatal response selection in sentence production: Insights from neural network simulation with reservoir computing. *Brain and Language*, 150:54–68, Nov. 2015. doi: 10.1016/j.bandl.2015.08.002.

M. Hines and N. T. Carnevale. The NEURON simulation environment. *Neural Comput.*, 9:1179–1209, 1997.

S. Hochreiter. Untersuchungen zu dynamischen neuronalen Netzen. *Diploma, Technische Universität München*, 91, 1991.

S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8): 1735–1780, 1997.

A. L. Hodgkin and A. F. Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *Journal of Physiology*, 117:500–544, 1952.

G. Holmes. DISTURBANCES OF VISION BY CEREBRAL LESIONS. *British Journal of Ophthalmology*, 2(7):353–384, Jul. 1918. doi: 10.1136/bjo.2.7.353.

C. J. Honey, R. Kötter, M. Breakspear, and O. Sporns. Network structure of cerebral cortex shapes functional connectivity on multiple time scales. *Proc. Natl. Acad. Sci. U. S. A.*, 104(24):10240–10245, Jun. 2007.

J. E. Hopcroft and J. D. Ullman. *An introduction to automata theory, languages, and computation.* Addison-Wesley series in computer science. Pearson, Upper Saddle River, NJ, Jan. 1979.

J. J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proc. Natl. Acad. Sci. USA*, 79:2554–2558, 1982.

J. C. Horton and D. L. Adams. The cortical column: a structure without a function. *Philosophical Transactions of the Royal Society B*, 360(1456):837–862, Apr 2005.

M. A. Huertas, M. G. H. Shuler, and H. Z. Shouval. A simple network architecture accounts for diverse reward time responses in primary visual cortex. *Journal of Neuroscience*, 35(37):12659–12672, 2015.

M. A. Huertas, S. E. Schwettmann, and H. Z. Shouval. The role of multiple neuromodulators in reinforcement learning that is based on competition between eligibility traces. *Front. Synaptic Neurosci.*, 8, dec 2016. doi: 10.3389/fnsyn.2016.00037.

M. Humphries. Why model the brain? The Spike, Jul 2019. `https://medium.com/the-spike/why-model-the-brain-c7a8e160e566` [Accessed 01.08.2023].

D. Hupkes, V. Dankers, M. Mul, and E. Bruni. Compositionality decomposed: how do neural networks generalise? *arXiv e-prints*, art. arXiv:1908.08351, Aug. 2019.

B. Iglewicz and D. C. Hoaglin. *How to detect and handle outliers*, volume 16. Asq Press, 1993.

Y. Ikegaya, G. Aaron, R. Cossart, D. Aronov, I. Lampl, D. Ferster, and R. Yuste. Synfire chains and cortical songs: temporal modules of cortical activity. *Science*, 304(5670): 559–564, 2004.

M. Inubushi and K. Yoshimura. Reservoir computing beyond memory-nonlinearity trade-off. *Scientific Reports*, 7(1), Aug. 2017. doi: 10.1038/s41598-017-10257-6.

L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20 (11):1254–1259, 1998.

E. Izhikevich. Which model to use for cortical spiking neurons? *IEEE Transactions on Neural Networks*, 5(15):1063–1070, Sep. 2004. doi: 10.1109/TNN.2004.832719.

H. Jaeger. The “echo state” approach to analysing and training recurrent neural networks. Technical Report GMD Report 148, German National Research Center for Information Technology, St. Augustin, Germany, 2001.

R. K. Jamieson and D. J. K. Mewhort. The influence of grammatical, local, and organizational redundancy on implicit learning: An analysis using information theory. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 31(1):9–23, 2005. doi: 10.1037/0278-7393.31.1.9.

S. Janson, S. Lonardi, and W. Szpankowski. On average sequence complexity. *Theoretical Computer Science*, 326(1-3):213–227, Oct. 2004. doi: 10.1016/j.tcs.2004.06.023.

J. Jaramillo, J. F. Mejias, and X.-J. Wang. Engagement of Pulvino-cortical Feedforward and Feedback Pathways in Cognitive Computations. *Neuron*, 101(2):321–336.e9, Jan. 2019. ISSN 0896-6273. doi: 10.1016/j.neuron.2018.11.023.

S. Jarvis, S. Rotter, and U. Egert. Extending stability through hierarchical clusters in Echo State Networks. *Frontiers in Neuroinformatics*, 4:11, 7 2010. ISSN 16625196. doi: 10.3389/fninf.2010.00011.

M. R. Joglekar, J. F. Mejias, G. R. Yang, and X.-J. Wang. Inter-areal balanced amplification enhances signal propagation in a large-scale circuit model of the primate cortex. *Neuron*, 98(1):222–234, 2018.

J. Jordan, H. Mørk, S. B. Vennemo, D. Terhorst, A. Peyser, T. Ippen, R. Deepu, J. M. Eppler, A. van Meegen, S. Kunkel, A. Sinha, T. Fardet, S. Diaz, A. Morrison, W. Schenck, D. Dahmen, J. Pronold, J. Stapmanns, G. Trensch, S. Spreizer, J. Mitchell, S. Graber, J. Senk, C. Linssen, J. Hahne, A. Serenko, D. Naoumenko, E. Thomson, I. Kitayama, S. Berns, and H. E. Plesser. *NEST 2.18.0*, Jun. 2019. doi: 10.5281/zenodo.2605422. Zenodo.

A. K. Joshi, K. V. Shanker, and D. Weir. The convergence of mildly context-sensitive grammar formalisms. Technical report, 1990.

A. Joulin and T. Mikolov. Inferring algorithmic patterns with stack-augmented recurrent nets. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

J. K. Jun and D. Z. Jin. Development of neural circuitry for precise temporal sequences through spontaneous activity, axon remodeling, and synaptic plasticity. *PLOSONE*, 2(8):e723, 2007.

J. H. Kaas. Topographic maps are fundamental to sensory processing. *Brain research bulletin*, 44(2):107–112, 1997.

J. Kadmon and H. Sompolinsky. Optimal architectures in a solvable model of deep networks. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 4781–4789. Curran Associates, Inc., 2016.

M. Kaiser. Hierarchy and dynamics of neural networks. *Frontiers in Neuroinformatics*, 4, 2010a. doi: 10.3389/fninf.2010.00112.

M. Kaiser. Optimal hierarchical modular topologies for producing limited sustained activation of neural networks. *Frontiers in Neuroinformatics*, 2010b. doi: 10.3389/fninf.2010.00008.

M. Kaiser, M. Görner, and C. C. Hilgetag. Criticality of spreading dynamics in hierarchical cluster networks without inhibition. *New Journal of Physics*, 9(5):110–110, May 2007. doi: 10.1088/1367-2630/9/5/110.

S. Kastner, K. DeSimone, C. S. Konen, S. M. Szczepanski, K. S. Weiner, and K. A. Schneider. Topographic Maps in Human Frontal Cortex Revealed in Memory-Guided Saccade and Spatial Working-Memory Tasks. *Journal of Neurophysiology*, 97(5): 3494–3507, 3 2007. ISSN 0022-3077, 1522-1598. doi: 10.1152/jn.00010.2007. PMID: 17360822.

G. A. Keliris, Q. Li, A. Papanikolaou, N. K. Logothetis, and S. M. Smirnakis. Estimating average single-neuron visual receptive field sizes by fmri. *Proceedings of the National Academy of Sciences*, 116(13):6425–6434, 2019. ISSN 0027-8424. doi: 10.1073/pnas.1809612116.

G. Keller, T. Bonhoeffer, and M. Hübener. Sensorimotor mismatch signals in primary visual cortex of the behaving mouse. *Neuron*, 74(5):809–815, 2012. ISSN 0896-6273. doi: https://doi.org/10.1016/j.neuron.2012.03.040.

R. M. Keller. *Classifiers, Acceptors, Transducers, and Sequencers*, pages 471–545. Harvey Mudd College, 2001.

S. Klampfl and W. Maass. Emergence of dynamic memory traces in cortical microcircuit models through stdp. *J. Neurosci.*, 33(28):11515–11529, 2013.

S. Klampfl, S. V. David, P. Yin, S. A. Shamma, and W. Maass. A quantitative analysis of information about past and present stimuli encoded by spikes of a1 neurons. *Journal of Neurophysiology*, 108(5):1366–1380, Sep. 2012. doi: 10.1152/jn.00935.2011.

S. C. Kleene. Representation of events in nerve nets and finite automata. In C. E. S. . J. J. McCarthy, editor, *Automata studies*, 1956.

C. Klos, D. Miner, and J. Triesch. Bridging structure and function: A model of sequence learning and prediction in primary visual cortex. *PLOS Comput. Biol.*, 14(6):e1006187, 2018.

T. R. Knösche and M. Tittgemeyer. The role of long-range connectivity for the characterization of the functional–anatomical organization of the cortex. *Frontiers in systems neuroscience*, 5:58, 2011.

T. Kohonen. Self-organized formation of topologically correct feature maps. *Biol. Cybern.*, 43:59–69, 1982.

P. Kok, J. F. Jehee, and F. P. de Lange. Less is more: Expectation sharpens representations in the primary visual cortex. *Neuron*, 75(2):265–270, 2012. ISSN 0896-6273. doi: https://doi.org/10.1016/j.neuron.2012.04.034.

A. N. Kolmogorov. On tables of random numbers. *Sankhyā: The Indian Journal of Statistics, Series A*, pages 369–376, 1963.

S. A. Korsky and R. C. Berwick. On the Computational Power of RNNs, Jun. 2019. doi: 10.48550/arXiv.1906.06349. arXiv:1906.06349 [cs].

J. Köster and S. Rahmann. Snakemake—a scalable bioinformatics workflow engine. *Biochemistry*, 28(19):2520–2522, 2012.

J. Kremkow, A. Aertsen, and A. Kumar. Gating of signal propagation in spiking neural networks by balanced and correlated excitation and inhibition. *Journal of Neuroscience*, 30(47):15760–15768, nov 2010. ISSN 0270-6474. doi: 10.1523/JNEUROSCI.3874-10.2010.

T. Kreuz, D. Chicharro, C. Houghton, R. G. Andrzejak, and F. Mormann. Monitoring spike train synchrony. *Journal of Neurophysiology*, 109(5):1457–1472, Mar. 2013. doi: 10.1152/jn.00873.2012.

T. S. Kuhn. *The structure of scientific revolutions*. University of Chicago Press, Chicago, IL, May 2012.

A. Kumar, S. Rotter, and A. Aertsen. Conditions for propagating synchronous spiking and asynchronous firing rates in a cortical network model. *J. Neurosci.*, 28(20):5268–5280, 2008a.

A. Kumar, S. Schrader, A. Aertsen, and S. Rotter. The high-conductance state of cortical networks. *Neural Comput.*, 20(1):1–43, 2008b. doi: 10.1162/neco.2008.20.1.1.

A. Kumar, S. Rotter, and A. Aertsen. Spiking activity propagation in neuronal networks: reconciling different perspectives on neural coding. *Nat. Rev. Neurosci.*, 11:615–627, 2010a.

A. Kumar, S. Rotter, and A. Aertsen. Spiking activity propagation in neuronal networks: reconciling different perspectives on neural coding. *Nat. Rev. Neurosci.*, 11:615–627, 2010b.

S. Kunkel and W. Schenck. The NEST dry-run mode: Efficient dynamic analysis of neuronal network simulation code. *Frontiers in Neuroinformatics*, 11:40, 2017. ISSN 1662-5196. doi: 10.3389/fninf.2017.00040.

F. Lagzi and S. Rotter. Dynamics of competition between subnetworks of spiking neuronal networks in the balanced state. *PLoS ONE*, 10(9):1–28, 09 2015. ISSN 19326203. doi: 10.1371/journal.pone.0138947.

F. Lagzi, F. M. Atay, and S. Rotter. Bifurcation analysis of the dynamics of interacting subnetworks of a spiking network. *Scientific reports*, 9(1):1–17, 2019.

J. E. Laird. *The soar cognitive architecture the soar cognitive architecture*. The MIT Press. MIT Press, London, England, Aug. 2012.

R. Laje and D. V. Buonomano. Robust timing and motor patterns by taming chaos in recurrent neural networks. *Nat. Neurosci.*, 16(7):925–933, 2013.

L. Lapicque. Recherches quantitatives sur l'excitation electrique des nerfs traitee comme une polarization. *Journal de Physiologie et de Pathologie Générale*, 9:620–635, 1907.

K. S. Lashley. *The problem of serial order in behavior*, volume 21. Bobbs-Merrill, 1951.

M. Layer, J. Senk, S. Essink, K. Korvasová, A. van Meegen, H. Bos, J. Schuecker, and M. Helias. Lif meanfield tools, Feb. 2020. doi: 10.5281/zenodo.3661413.

A. Lazar. SORN: a self-organizing recurrent neural network. *Frontiers in Computational Neuroscience*, 3, 2009. doi: 10.3389/neuro.10.023.2009.

Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

H. Lee, E. Margalit, K. M. Jozwik, M. A. Cohen, N. Kanwisher, D. L. K. Yamins, and J. J. DiCarlo. Topographic deep artificial neural networks reproduce the hallmarks of the primate inferior temporal cortex face processing network. Jul. 2020. doi: 10.1101/2020.07.09.185116.

J. Leugering, P. Nieters, and G. Pipa. Dendritic plateau potentials can process spike sequences across multiple time-scales. *Frontiers in Cognition*, 2, Feb. 2023. doi: 10.3389/fcogn.2023.1044216.

W. B. Levy and D. Steward. Temporal contiguity requirements for long-term associative potentiation/depression in the hippocampus. *Neuroscience*, 8:791–797, 1983.

K. Li, V. Kozyrev, S. Kyllingsbæk, S. Treue, S. Ditlevsen, and C. Bundesen. Neurons in primate visual cortex alternate between responses to multiple stimuli in their receptive field. *Frontiers in Computational Neuroscience*, 10:141, 2016. ISSN 1662-5188. doi: 10.3389/fncom.2016.00141.

T. P. Lillicrap, A. Santoro, L. Marris, C. J. Akerman, and G. Hinton. Backpropagation and the brain. *Nature Reviews Neuroscience*, 21(6):335–346, Apr. 2020. doi: 10.1038/s41583-020-0277-3.

Z. C. Lipton, J. Berkowitz, and C. Elkan. A critical review of recurrent neural networks for sequence learning, 2015. doi: 10.48550/ARXIV.1506.00019.

J. Lisman, K. Cooper, M. Sehgal, and A. J. Silva. Memory formation depends on both synapse-specific modifications of synaptic strength and cell-specific increases in excitability. *Nature Neuroscience*, 21(3):309–314, Feb. 2018. doi: 10.1038/s41593-018-0076-6.

A. Litwin-Kumar and B. Doiron. Slow dynamics and high variability in balanced cortical networks with clustered connections. *Nat. Neurosci.*, 15(11):1498–1505, Sep. 2012. doi: 10.1038/nn.3220.

A. Litwin-Kumar and B. Doiron. Formation and maintenance of neuronal assemblies through synaptic plasticity. *Nature Communications*, 5(1):1–12, 2014.

C.-H. Liu, J. E. Coleman, H. Davoudi, K. Zhang, and M. G. H. Shuler. Selective activation of a putative reinforcement signal conditions cued interval timing in primary visual cortex. *Current Biology*, 25(12):1551–1561, Jun. 2015. doi: 10.1016/j.cub.2015.04.028.

J. K. Liu and D. V. Buonomano. Embedding multiple trajectories in simulated recurrent neural networks in a self-organizing manner. *J. Neurosci.*, 29(42):13172–13181, 2009.

L. Liu, L. She, M. Chen, T. Liu, H. D. Lu, Y. Dan, and M.-m. Poo. Spatial structure of neuronal receptive field in awake monkey secondary visual cortex (v2). *Proceedings of the National Academy of Sciences*, 113(7):1913–1918, 2016. ISSN 0027-8424. doi: 10.1073/pnas.1525505113.

M. A. Long, D. Z. Jin, and M. S. Fee. Support for a synaptic chain model of neuronal sequence generation. *Nature*, 468:394–399, 2010.

R. Lorente de Nó. Studies on the structure of the cerebral cortex. i. the area entorhinalis. *Journal für Psychologie und Neurologie*, (45):381–438, 1933.

D. M. Lorenz, A. Jeng, and M. W. Deem. The emergence of modularity in biological systems. *Physics of Life Reviews*, Feb. 2011. doi: 10.1016/j.plrev.2011.02.003.

M. Lukoševičius and H. Jaeger. Reservoir computing approaches to recurrent neural network training. *Computer Science Review*, 3(3):127–149, 2009.

W. Maass and H. Markram. On the computational power of circuits of spiking neurons. *Journal of Computer and System Sciences*, 69(4):593–616, Dec. 2004. doi: 10.1016/j.jcss.2004.04.001.

W. Maass, T. Natschläger, and H. Markram. Real-time computing without stable states: a new framework for neural computation based on perturbations. *Neural Comput.*, 14 (11):2531–2560, 2002. doi: 10.1162/089976602760407955.

W. Maass, T. Natschläger, and H. Markram. Fading memory and kernel properties of generic cortical microcircuit models. *Journal of Physiology Paris*, 98(4-6 SPEC. ISS.): 315–330, 7 2004. ISSN 09284257. doi: 10.1016/j.jphysparis.2005.09.020.

E. Macaluso. Multisensory processing in sensory-specific cortical areas. *The Neuroscientist*, 12(4):327–338, Aug. 2006. doi: 10.1177/1073858406287908.

W. E. Mackey, J. Winawer, and C. E. Curtis. Visual field map clusters in human frontoparietal cortex. *eLife*, 6, Jun. 2017. doi: 10.7554/elife.22974.

A. Maes, M. Barahona, and C. Clopath. Learning spatiotemporal signals using a recurrent spiking network that discretizes time. *PLOS Comput. Biol.*, 16(1):e1007606, 2020.

A. Maes, M. Barahona, and C. Clopath. Learning compositional sequences with multiple time scales through a hierarchical network of spiking neurons. *PLOS Computational Biology*, 17(3):e1008866, Mar. 2021. doi: 10.1371/journal.pcbi.1008866.

G. A. Manley. Comparative auditory neuroscience: Understanding the evolution and function of ears. *Journal of the Association for Research in Otolaryngology*, 18(1): 1–24, Aug. 2016. doi: 10.1007/s10162-016-0579-3.

V. Mante, D. Sussillo, K. V. Shenoy, and W. T. Newsome. Context-dependent computation by recurrent dynamics in prefrontal cortex. *Nature*, 503(7474):78–84, 2013.

G. F. Marcus. *The algebraic mind: Integrating connectionism and cognitive science.* MIT press, 2003.

E. Marder. Neuromodulation of neuronal circuits: Back to the future. *Neuron*, 76(1): 1–11, Oct. 2012. doi: 10.1016/j.neuron.2012.09.010.

F. Markopoulos, D. Rokni, D. H. Gire, and V. N. Murthy. Functional properties of cortical feedback projections to the olfactory bulb. *Neuron*, 76(6):1175–1188, 2012. ISSN 0896-6273. doi: https://doi.org/10.1016/j.neuron.2012.10.028.

N. T. Markov and H. Kennedy. The importance of being hierarchical. *Current Opinion in Neurobiology*, 23(2):187–194, 4 2013. ISSN 0959-4388. doi: 10.1016/j.conb.2012.12.008. Macrocircuits.

N. T. Markov, M. M. Ercsey-Ravasz, A. R. Ribeiro Gomes, C. Lamy, L. Magrou, J. Vezoli, P. Misery, A. Falchier, R. Quilodran, M. A. Gariel, J. Sallet, R. Gamanut, C. Huissoud, S. Clavagnier, P. Giroud, D. Sappey-Marinier, P. Barone, C. Dehay, Z. Toroczkai, K. Knoblauch, D. C. Van Essen, and H. Kennedy. A weighted and directed interareal connectivity matrix for macaque cerebral cortex. *Cereb. Cortex*, 24 (1):17–36, 2014a. doi: 10.1093/cercor/bhs270.

N. T. Markov, J. Vezoli, P. Chameau, A. Falchier, R. Quilodran, C. Huissoud, C. Lamy, P. Misery, P. Giroud, S. Ullman, P. Barone, C. Dehay, K. Knoblauch, and H. Kennedy. Anatomy of hierarchy: Feedforward and feedback pathways in macaque visual cortex. *Journal of Comparative Neurology*, 522(1):225–259, 2014b. ISSN 1096-9861. doi: 10.1002/cne.23458.

H. Markram, J. Lübke, M. Frotscher, and B. Sakmann. Regulation of synaptic efficacy by coincidence of postsynaptic APs and EPSPs. *Science*, 275:213–215, 10. January 1997.

H. Markram, Y. Wang, and M. Tsodyks. Differential signaling via the same axon of neocortical pyramidal neurons. *Proc. Natl. Acad. Sci. USA*, 95(9):5323–5328, 1998.

M. I. Marrufo-Pérez, D. del Pilar Sturla-Carreto, A. Eustaquio-Martín, and E. A. Lopez-Poveda. Adaptation to noise in human speech recognition depends on noise-level statistics and fast dynamic-range compression. *The Journal of Neuroscience*, 40(34): 6613–6623, Jul. 2020. doi: 10.1523/jneurosci.0469-20.2020.

M. Mascaro and D. J. Amit. Effective neural response function for collective population states. *Network: Computation in Neural Systems*, 10(4):351–373, 1999.

M. Matsumura, T. Cope, and E. Fetz. Sustained excitatory synaptic input to motor cortex neurons in awake animals revealed by intracellular recording of membrane potentials. *Experimental Brain Research*, 70(3):463–469, 5 1988. ISSN 0014-4819. doi: 10.1007/BF00247594.

F. Mazzocchi. Could big data be the end of theory in science? *EMBO reports*, 16(10): 1250–1255, Sep. 2015. doi: 10.15252/embr.201541001.

L. Mazzucato, A. Fontanini, and G. La Camera. Dynamics of multistable states during ongoing and evoked cortical activity. *J. Neurosci.*, 35(21):8214–8231, 2015. ISSN 0270-6474. doi: 10.1523/JNEUROSCI.4819-14.2015.

L. Mazzucato, A. Fontanini, and G. La Camera. Stimuli reduce the dimensionality of cortical activity. *Frontiers in Systems Neuroscience*, 10(11), 2016. ISSN 16625137. doi: 10.3389/fnsys.2016.00011.

D. A. McCormick. Neuronal networks: Flip-flops in the brain. *Current Biology*, 15(8): R294–R296, 2005.

W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, 5(4):115–133, Dec. 1943. ISSN 0007-4985. doi: 10.1007/BF02478259.

M. J. McGinley, M. Vinck, J. Reimer, R. Batista-Brito, E. Zagha, C. R. Cadwell, A. S. Tolias, J. A. Cardin, and D. A. McCormick. Waking state: Rapid variations modulate neural and behavioral responses. *Neuron*, 87(6):1143–1161, Sep. 2015. doi: 10.1016/j.neuron.2015.09.012.

M. L. McHugh. Interrater reliability: the kappa statistic. *Biochemia Medica*, pages 276–282, 2012. doi: 10.11613/bm.2012.031.

T. McLaughlin and D. D. O'Leary. MOLECULAR GRADIENTS AND DEVELOPMENT OF RETINOTOPIC MAPS. *Annual Review of Neuroscience*, 28(1):327–355, 7 2005. ISSN 0147-006X. doi: 10.1146/annurev.neuro.28.061604.135714.

P. McLeod, K. Plunkett, and E. T. Rolls. *Introduction to Connectionist Modelling of Cognitive Processes.* Oxford University Press, Oxford, 1998.

H. Mengistu, J. Huizinga, J.-B. Mouret, and J. Clune. The evolutionary origins of hierarchy. *PLOS Computational Biology*, 12(6):e1004829, Jun. 2016. doi: 10.1371/journal.pcbi.1004829.

W. Merrill, G. Weiss, Y. Goldberg, R. Schwartz, N. A. Smith, and E. Yahav. A formal hierarchy of RNN architectures. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.acl-main.43.

T. Meulemans and M. V. der Linden. Associative chunk strength in artificial grammar learning. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 23 (4):1007–1028, Jul. 1997. doi: 10.1037/0278-7393.23.4.1007.

D. Meunier, R. Lambiotte, A. Fornito, K. D. Ersche, and E. T. Bullmore. Hierarchical modularity in human brain functional networks. *Frontiers in neuroinformatics*, 3:37, 2009. ISSN 1662-5196. doi: 10.3389/neuro.11.037.2009.

D. Meunier, R. Lambiotte, and E. Bullmore. Modular and hierarchically modular organization of brain networks. *Frontiers in Neuroscience*, 4:200, 2010. ISSN 1662-453X. doi: 10.3389/fnins.2010.00200.

C. Michaelis. Pelenet: A reservoir computing framework for loihi, 2020. doi: 10.48550/ARXIV.2011.12338.

S. Michau. Dynamic memory traces for sequence learning in spiking networks. Bachelor's thesis, RWTH Aachen University, 2022.

S. Mihalas and E. Niebur. A generalized linear integrate-and-fire neural model produces diverse spiking behaviors. *Neural Comput.*, 21(3):704–718, 2010.

M. Miłkowski, W. M. Hensel, and M. Hohol. Replicability or reproducibility? on the replication crisis in computational neuroscience and sharing only relevant detail. *Journal of Computational Neuroscience*, 45(3):163–172, Oct. 2018. doi: 10.1007/s10827-018-0702-z.

G. A. Miller. *The psychology of communication: seven essays*. Basic Books, 1967.

J. L. Miller, F. Grosjean, and C. Lomanto. Articulation rate and its variability in spontaneous speech: A reanalysis and some implications. *Phonetica*, 41(4):215–225, Jul. 1984. doi: 10.1159/000261728.

C. Mo and S. M. Sherman. A Sensorimotor Pathway via Higher-Order Thalamus. *Journal of Neuroscience*, 39(4):692–704, Jan. 2019. ISSN 0270-6474. doi: 10.1523/JNEUROSCI.1467-18.2018.

F. Mölder, K. P. Jablonski, B. Letcher, M. B. Hall, C. H. Tomkins-Tinch, V. Sochat, J. Forster, S. Lee, S. O. Twardziok, A. Kanitz, A. Wilm, M. Holtgrewe, S. Rahmann, S. Nahnsen, and J. Köster. Sustainable data analysis with snakemake. *F1000Research*, 10:33, Apr. 2021. doi: 10.12688/f1000research.29032.2.

G. Mongillo, O. Barak, and M. Tsodyks. Synaptic theory of working memory. *Science*, 319:1543–1546, 2008.

R. Moreno-Bote, J. Rinzel, and N. Rubin. Noise-induced alternations in an attractor network model of perceptual bistability. *Journal of Neurophysiology*, 98(3):1125–1139, 2007. doi: 10.1152/jn.00116.2007. PMID: 17615138.

A. Morrison, M. Diesmann, and W. Gerstner. Phenomenological models of synaptic plasticity based on spike-timing. *Biol. Cybern.*, 98(6):459–478, 2008. doi: 10.1007/s00422-008-0233-1.

H. Mostafa and G. Indiveri. Sequential activity in asymmetrically coupled winner-take-all circuits. *Neural Computation*, 26(9):1973–2004, Sep. 2014. doi: 10.1162/neco_a_00619.

V. B. Mountcastle. The columnar organization of the neocortex. *Brain*, 120:701–722, 1997.

V. B. Mountcastle and T. P. Powell. Neural mechanisms subserving cutaneous sensibility, with special reference to the role of afferent inhibition in sensory perception and discrimination. *Bull Johns Hopkins Hosp*, 105:201–232, Oct 1959.

D. Muir, F. Bauer, and P. Weidel. Rockpool documentaton, Mar. 2022. doi: 10.5281/zenodo.6381006.

J. D. Murray, A. Bernacchia, D. J. Freedman, R. Romo, J. D. Wallis, X. Cai, C. Padoa-Schioppa, T. Pasternak, H. Seo, D. Lee, et al. A hierarchy of intrinsic timescales across primate cortex. *Nat. Neurosci.*, 17(12):1661–1663, 2014.

J. M. Murray and G. S. Escola. Learning multiple variable-speed sequences in striatum via cortical tutoring. *eLife*, 6, May 2017. doi: 10.7554/elife.26084.

W. F. Młynarski and A. M. Hermundstad. Adaptive coding for dynamic sensory inference. *eLife*, 7:e32055, jul 2018. ISSN 2050-084X. doi: 10.7554/eLife.32055.

Z. Nádasdy, H. Hirase, A. Czurkó, J. Csicsvari, and G. Buzsáki. Replay and time compression of recurring spike sequences in the hippocampus. *J. Neurosci.*, 19(21): 9497–9507, Nov. 1999. doi: 10.1523/jneurosci.19-21-09497.1999.

M. Nakajima and M. M. Halassa. Thalamic control of functional cortical connectivity. *Current Opinion in Neurobiology*, 44:127–131, Jun. 2017. ISSN 1873-6882. doi: 10.1016/j.conb.2017.04.001.

A. Newell and H. A. Simon. Computer science as empirical inquiry: symbols and search. In *ACM Turing Award Lectures*. Association of Computing Machinery, 1975. doi: 10.1145/1283920.1283930.

M. E. J. Newman. The structure and function of complex networks. *SIAM Review*, 45 (2):167–256, 2003.

M. E. J. Newman. Random graphs with clustering. *Physical Review Letters*, 103(5), Jul. 2009. doi: 10.1103/physrevlett.103.058701.

D. Nikolic, S. Haeusler, W. Singer, and W. Maass. Distributed fading memory for stimulus properties in the primary visual cortex. *PLOS Biology*, 7(12):e1000260, 2009. doi: 10.1371/journal.pbio.1000260.

E. Nordlie, M.-O. Gewaltig, and H. E. Plesser. Towards reproducible descriptions of neuronal network models. *PLOS Comput. Biol.*, 5(8):e1000456, Aug. 2009. doi: 10.1371/journal.pcbi.1000456.

B. A. Nosek and T. M. Errington. What is replication? *PLOS Biology*, 18(3):e3000691, Mar. 2020. doi: 10.1371/journal.pbio.3000691.

J. Oberländer, Y. Bouhadjar, and A. Morrison. Learning and replaying spatiotemporal sequences: A replication study. *Frontiers in Integrative Neuroscience*, 16, Oct. 2022. doi: 10.3389/fnint.2022.974177.

G. Orbán, J. Fiser, R. N. Aslin, and M. Lengyel. Bayesian learning of visual chunks by human observers. *Proceedings of the National Academy of Sciences*, 105(7):2745–2750, Feb. 2008. doi: 10.1073/pnas.0708424105.

E. Orhan and X. Pitkow. Skip connections eliminate singularities. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018.

S. Ostojic. Two types of asynchronous activity in networks of excitatory and inhibitory spiking neurons. *Nat. Neurosci.*, 17:594–600, Feb. 2014. doi: 10.1038/nn.3658.

Y. Pan and M. Monje. Activity shapes neural circuit form and function: A historical perspective. *The Journal of Neuroscience*, 40(5):944–954, jan 2020. doi: 10.1523/jneurosci.0740-19.2019.

H.-J. Park and K. Friston. Structural and Functional Brain Networks: From Connections to Cognition. *Science*, 342(6158):1238411–1238411, 11 2013. ISSN 0036-8075. doi: 10.1126/science.1238411.

T. Parr, A. W. Corcoran, K. J. Friston, and J. Hohwy. Perceptual awareness and active inference. *Neuroscience of Consciousness*, 2019(1), 09 2019. ISSN 2057-2107. doi: 10.1093/nc/niz012. niz012.

A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

G. H. Patel, D. M. Kaplan, and L. H. Snyder. Topographic organization in the brain: searching for general principles. *Trends in cognitive sciences*, 18(7):351–363, 2014.

R. Pauli, P. Weidel, S. Kunkel, and A. Morrison. Reproducing polychronization: a guide to maximizing the reproducibility of spiking network models. *Frontiers in Neuroinformatics*, 12(46), 2018. doi: 10.3389/fninf.2018.00046.

P. Perruchet and A. Vinter. Learning and development: The implicit knowledge assumption reconsidered. 1998.

K.-M. Petersson, V. Folia, and P. Hagoort. What artificial grammar learning reveals about the neurobiology of syntax. *Brain and Language*, 120(2):83–95, Feb. 2012. doi: 10.1016/j.bandl.2010.08.003.

J.-P. Pfister and W. Gerstner. Triplets of spikes in a model of spike timing-dependent plasticity. *J. Neurosci.*, 26:9673–9682, 2006.

G. Piccinini and S. Bahar. Neural computation and the computational theory of cognition. *Cognitive Science*, 37(3):453–488, nov 2012. doi: 10.1111/cogs.12012.

G. Piccinini and A. Scarantino. Information processing, computation, and cognition. *J. Biol. Phys.*, 37(1):1–38, Jan. 2011.

H. E. Plesser. Reproducibility vs. replicability: A brief history of a confused terminology. *Frontiers in Neuroinformatics*, 11, Jan. 2018. doi: 10.3389/fninf.2017.00076.

E. M. Pothos. Theories of artificial grammar learning. *Psychological Bulletin*, 133(2): 227–244, Mar. 2007. doi: 10.1037/0033-2909.133.2.227.

A. Pouget, S. Deneve, J.-C. Ducom, and P. E. Latham. Narrow versus wide tuning curves: What's best for a population code? *Neural Computation*, 11(1):85–90, 1999. doi: 10.1162/089976699300016818.

N. Pradhan, S. Dasgupta, and S. Sinha. Modular organization enhances the robustness of attractor network dynamics. *EPL (Europhysics Letters)*, 94(3):38004, Apr. 2011. doi: 10.1209/0295-5075/94/38004.

B. Pulverer. When things go wrong: correcting the scientific record. *The EMBO Journal*, 34(20):2483–2485, Oct. 2015. doi: 10.15252/embj.201570080.

F. Pulvermüller and Y. Shtyrov. Spatiotemporal signatures of large-scale synfire chains for speech processing as revealed by MEG. *Cereb. Cortex*, 19:79–88, 2009.

M. Rabinovich, R. Huerta, and G. Laurent. Transient dynamics for neural processing. *Science*, 321(5885):48–50, Jul. 2008. doi: 10.1126/science.1155564.

K. Rajan and L. F. Abbott. Eigenvalue spectra of random matrices for neural networks. *Phys. Rev. Lett.*, 97:188104, 2006. doi: 10.1103/PhysRevLett.97.188104.

K. Rajan, C. D. Harvey, and D. W. Tank. Recurrent network models of sequence generation and memory. *Neuron*, 90(1):128–142, 2016.

S. Ramón y Cajal. Estructura de los centros nerviosos de las aves. *Rev. Trim. Histol. Norm. Pat.*, (1):1–10, 1888.

A. S. Reber. Implicit learning of artificial grammars. *Journal of Verbal Learning and Verbal Behavior*, 6(6):855–863, dec 1967. doi: 10.1016/s0022-5371(67)80149-x.

A. Renart and C. K. Machens. Variability in neural activity and behavior. *Curr. Opin. Neurobiol.*, 25:211–220, 2014.

A. Renart and M. C. W. van Rossum. Transmission of population-coded information. *Neural Computation*, 24(2):391–407, Feb. 2012. doi: 10.1162/neco_a_00227.

A. Renart, R. Moreno-Bote, X.-J. Wang, and N. Parga. Mean-driven and fluctuation-driven persistent activity in recurrent networks. *Neural Computation*, 19(1):1–46, Jan. 2007. doi: 10.1162/neco.2007.19.1.1.

A. Renart, J. De La Rocha, P. Bartho, L. Hollender, N. Parga, A. Reyes, and K. D. Harris. The asynchronous state in cortical circuits. *Science*, 327:587–590, Jan. 2010. doi: 10.1126/science.1179850.

Y. Revina, L. S. Petro, and L. Muckli. Cortical feedback signals generalise across different spatial frequencies of feedforward inputs. *NeuroImage*, 180:280–290, 2018. ISSN 1053-8119. doi: https://doi.org/10.1016/j.neuroimage.2017.09.047. New advances in encoding and decoding of brain signals.

H. Rezaei, A. Aertsen, A. Kumar, and A. Valizadeh. Facilitating the propagation of spiking activity in feedforward networks by including feedback. *PLOS Computational Biology*, 16(8):e1008033, Aug. 2020. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1008033.

M. Rigotti and S. Fusi. Estimating the dimensionality of neural responses with fmri repetition suppression. *arXiv preprint arXiv:1605.03952*, 2016.

M. Rigotti, O. Barak, M. R. Warden, X.-J. Wang, N. D. Daw, E. K. Miller, and S. Fusi. The importance of mixed selectivity in complex cognitive tasks. *Nature*, 497(7451): 585–590, 2013. doi: 10.1038/nature12160.

H. Risken. *The Fokker-Planck Equation*. Springer Verlag Berlin Heidelberg, 1996. doi: 10.1007/978-3-642-61544-3_4.

M. T. Roberts, S. C. Seeman, and N. L. Golding. A mechanistic understanding of the role of feedforward inhibition in the mammalian sound localization circuitry. *Neuron*, 78(5): 923–935, 2013. ISSN 0896-6273. doi: https://doi.org/10.1016/j.neuron.2013.04.022.

C. Robinson. *Dynamical systems: stability, symbolic dynamics, and chaos*. CRC press, 1998.

P. Robinson. COGNITIVE ABILITIES, CHUNK-STRENGTH, AND FREQUENCY EFFECTS IN IMPLICIT ARTIFICIAL GRAMMAR AND INCIDENTAL l2 LEARNING: REPLICATIONS OF REBER, WALKENFELD, AND HERNSTADT (1991) AND KNOWLTON AND SQUIRE (1996) AND THEIR RELEVANCE FOR SLA. *Studies in Second Language Acquisition*, 27(02), Jun. 2005. doi: 10.1017/s0272263105050126.

P. Rodriguez and J. Wiles. Recurrent Neural Networks Can Learn to Implement Symbol-Sensitive Counting. In *Advances in Neural Information Processing Systems*, volume 10. MIT Press, 1997.

F. Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychol Rev*, 65:386–408, 1958.

T. Rost, M. Deger, and M. P. Nawrot. Winnerless competition in clustered balanced networks: inhibitory assemblies do the trick. *Biol. Cybern.*, 112(1):81–98, 2018.

N. P. Rougier, K. Hinsen, F. Alexandre, T. Arildsen, L. A. Barba, F. C. Benureau, C. T. Brown, P. de Buyl, O. Caglayan, A. P. Davison, M.-A. Delsuc, G. Detorakis, A. K. Diem, D. Drix, P. Enel, B. Girard, O. Guest, M. G. Hall, R. N. Henriques, X. Hinaut, K. S. Jaron, M. Khamassi, A. Klein, T. Manninen, P. Marchesi, D. McGlinn, C. Metzner, O. Petchey, H. E. Plesser, T. Poisot, K. Ram, Y. Ram, E. Roesch, C. Rossant, V. Rostami, A. Shifman, J. Stachelek, M. Stimberg, F. Stollmeier, F. Vaggi, G. Viejo,

J. Vitay, A. E. Vostinar, R. Yurchak, and T. Zito. Sustainable computational science: the ReScience initiative. *PeerJ Computer Science*, 3:e142, Dec. 2017. doi: 10.7717/peerj-cs.142.

P. J. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

J. E. Rubin. Computational models of basal ganglia dysfunction: the dynamics is in the details. *Current Opinion in Neurobiology*, 46:127–135, Oct. 2017. doi: 10.1016/j.conb.2017.08.011.

M. Rubinov, O. Sporns, C. van Leeuwen, and M. Breakspear. Symbiotic relationship between brain structure and dynamics. *BMC Neuroscience*, 10(1):55, Jun. 2009. ISSN 1471-2202. doi: 10.1186/1471-2202-10-55.

M. Rubinov, J. Lizier, M. Prokopenko, and M. Breakspear. Maximized directed information transfer in critical neuronal networks. *BMC Neuroscience*, 12(Suppl 1):P18, Jun. 2011. ISSN 1471-2202. doi: 10.1186/1471-2202-12-S1-P18.

D. E. Rumelhart, J. L. McClelland, and S. D. P. R. G. University of California. *Parallel distributed processing : explorations in the microstructure of cognition.* MIT Press, 1986a. ISBN 026268053X.

E. Rumelhart, David, E. Hinton, Geoffrey, and J. Williams, Ronald. Learning representations by back-propagating errors. *Nature*, 323:533–536, 1986b. doi: 10.1038/323533a0.

P. Sanz Leon, S. Knock, M. Woodman, L. Domide, J. Mersmann, A. McIntosh, and V. Jirsa. The virtual brain: a simulator of primate brain network dynamics. *Frontiers in Neuroinformatics*, 7:10, 2013. doi: 10.3389/fninf.2013.00010.

A. Sanzeni, B. Akitake, H. C. Goldbach, C. E. Leedy, N. Brunel, and M. H. Histed. Inhibition stabilization is a widespread property of cortical networks. *eLife*, 9:e54875, 2020.

N. Schaetti. Echotorch: Reservoir computing with pytorch. `https://github.com/nschaetti/EchoTorch`, 2018.

A. C. Schapiro, T. T. Rogers, N. I. Cordova, N. B. Turk-Browne, and M. M. Botvinick. Neural representations of events arise from temporal community structure. *Nature Neuroscience*, 16(4):486–492, Feb. 2013. doi: 10.1038/nn.3331.

M. T. Schaub*, Y. Billeh*, C. A. Anastassiou, C. Koch, and M. Barahona. Emergence of slow-switching assemblies in structured neuronal networks. *PLOS Comput. Biol.*, 11(7):e1004196, 2015. doi: 10.1371/journal.pcbi.1004196.

R. Schiff and P. Katan. Does complexity matter? meta-analysis of learner performance in artificial grammar tasks. *Frontiers in Psychology*, 5, Sep. 2014. doi: 10.3389/fpsyg.2014.01084.

F. Schittler Neves and M. Timme. Computation by switching in complex networks of states. *Phys. Rev. Lett.*, 109:018701, Jul 2012. doi: 10.1103/PhysRevLett.109.018701.

M. Schmidt, R. Bakker, K. Shen, G. Bezgin, M. Diesmann, and S. J. van Albada. A multi-scale layer-resolved spiking network model of resting-state dynamics in macaque visual cortical areas. *PLOS Comput. Biol.*, 14(10):e1006359, 2018. doi: 10.1371/journal.pcbi.1006359.

B. Schölkopf and A. J. Smola. *Learning with Kernels – Support Vector Machines, Regularization, Optimization and Beyond.* The MIT Press, Cambridge,Massachusetts, 2002. ISBN 0-262-19475-9.

M. R. Schomers, M. Garagnani, and F. Pulvermüller. Neurocomputational consequences of evolutionary connectivity changes in perisylvian language cortex. *Journal of Neuroscience*, 37(11):3045–3055, 2017. ISSN 0270-6474. doi: 10.1523/JNEUROSCI.2693-16.2017.

J. Schuecker, M. Diesmann, and M. Helias. Modulated escape from a metastable state driven by colored noise. *Phys. Rev. E*, 92:052119, Nov. 2015. doi: 10.1103/PhysRevE.92.052119.

J. Schuecker, M. Schmidt, S. J. van Albada, M. Diesmann, and M. Helias. Fundamental activity constraints lead to specific interpretations of the connectome. *PLOS Computational Biology*, 13(2):e1005179, Feb. 2017. doi: 10.1371/journal.pcbi.1005179.

J. R. Searle. Minds, brains, and programs. *Behavioral and Brain Sciences*, 3(3):417–424, sep 1980. doi: 10.1017/s0140525x00005756.

L. Sennhauser and R. Berwick. Evaluating the ability of LSTMs to learn context-free grammars. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Association for Computational Linguistics, 2018. doi: 10.18653/v1/w18-5414.

P. Seriès, P. E. Latham, and A. Pouget. Tuning curve sharpening for orientation selectivity: coding efficiency and the impact of correlations. *Nature Neuroscience*, 7(10): 1129–1135, Sep. 2004. doi: 10.1038/nn1321.

M. N. Shadlen and W. T. Newsome. Noise, neural codes and cortical organization. *Curr. Opin. Neurobiol.*, 4(4):569–579, 1994. doi: 10.1016/0959-4388(94)90059-0.

M. N. Shadlen and W. T. Newsome. The variable discharge of cortical neurons: Implications for connectivity, computation, and information coding. *J. Neurosci.*, 18(10): 3870–3896, 1998. doi: 10.1523/jneurosci.18-10-03870.1998.

D. R. Shanks and T. Johnstone. Evaluating the relationship between explicit and implicit knowledge in a sequential reaction time task. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 25(6):1435–1451, 1999. doi: 10.1037/0278-7393.25.6.1435.

C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27(3):379–423, Jul. 1948. doi: 10.1002/j.1538-7305.1948.tb01338.x.

T. O. Sharpee, C. A. Atencio, and C. E. Schreiner. Hierarchical representations in the auditory cortex. *Current Opinion in Neurobiology*, 21(5):761–767, Oct. 2011. doi: 10.1016/j.conb.2011.05.027.

C. J. Shatz. The developing brain. *Scientific American*, 267(3):60–67, Sep. 1992. doi: 10.1038/scientificamerican0992-60.

J. S. Sherfey, A. E. Soplata, S. Ardid, E. A. Roberts, D. A. Stanley, B. R. Pittman-Polletta, and N. J. Kopell. Dynasim: a matlab toolbox for neural modeling and simulation. *Frontiers in Neuroinformatics*, 12:10, 2018.

S. M. Sherman and R. W. Guillery. *Functional Connections of Cortical Areas.* The MIT Press, Aug. 2013. doi: 10.7551/mitpress/9780262019309.001.0001.

C. S. Sherrington. *Integrative action of the nervous system.* Yale University Press, New Haven, 1906.

S. Shinomoto, H. Kim, T. Shimokawa, N. Matsuno, S. Funahashi, K. Shima, I. Fujita, H. Tamura, T. Doi, K. Kawano, N. Inaba, K. Fukushima, S. Kurkin, K. Kurata, M. Taira, K.-I. Tsutsui, H. Komatsu, T. Ogawa, K. Koida, J. Tanji, and K. Toyama. Relating neuronal firing patterns to functional differentiation of cerebral cortex. *PLOS Comput. Biol.*, 5(7):e1000433, 2009.

H. T. Siegelmann and E. D. Sontag. Turing computability with neural nets. *Applied Mathematics Letters*, 4(6):77–80, 1991.

H. T. Siegelmann and E. D. Sontag. On the computational power of neural nets. In *Proceedings of the fifth annual workshop on Computational learning theory.* ACM, Jul. 1992. doi: 10.1145/130385.130432.

M. A. Silver and S. Kastner. Topographic maps in human frontal and parietal cortex, 11 2009. ISSN 13646613. doi: 10.1016/j.tics.2009.08.005.

H. A. Simon. *Models of Discovery*, volume 54 of *Boston Studies in the Philosophy of Science*. Springer Netherlands, Dordrecht, 1977. ISBN 978-90-277-0970-7 978-94-010-9521-1. doi: 10.1007/978-94-010-9521-1.

D. J. Simons. The value of direct replication. *Perspectives on Psychological Science*, 9 (1):76–80, Jan. 2014. doi: 10.1177/1745691613514755.

W. Singer. Cortical dynamics revisited. *Trends in Cognitive Sciences*, 17(12):616–626, Dec. 2013. doi: 10.1016/j.tics.2013.09.006.

P. Sjöström, G. Turrigiano, and S. Nelson. Rate, timing, and cooperativity jointly determine cortical synaptic plasticity. *Neuron*, 32:1149–1164, 2001.

A. Smith, K. Singh, A. Williams, and M. Greenlee. Estimating Receptive Field Size from fMRI Data in Human Striate and Extrastriate Visual Cortex. *Cerebral Cortex*, 11(12):1182–1190, 12 2001. ISSN 1047-3211. doi: 10.1093/cercor/11.12.1182.

W. R. Softky and C. Koch. The highly irregular firing of cortical cells is inconsistent with temporal integration of random EPSPs. *J. Neurosci.*, 13(1):334–350, 1993. doi: 10.1523/jneurosci.13-01-00334.1993.

S. Song, K. D. Miller, and L. F. Abbott. Competitive Hebbian learning through spike-timing-dependent synaptic plasticity. *Nat. Neurosci.*, 3(9):919–926, 2000.

H. Spencer. *The Principles of biology v. 1, 1866*, volume 1. D. Appleton & Company, 1866.

O. Sporns. *Networks of the brain*. MIT Press, Cambridge, Mass, 2011. ISBN 978-0-262-01469-4. OCLC: ocn551342282.

O. Sporns and R. F. Betzel. Modular brain networks. *Annual Review of Psychology*, 67 (1):613–640, Jan. 2016. doi: 10.1146/annurev-psych-122414-033634.

M. Steriade, I. Timofeev, and F. Grenier. Natural waking and sleep states: A view from inside neocortical neurons. *J. Neurophysiol.*, 85:1969–1985, 2001.

M. Stimberg, R. Brette, and D. F. Goodman. Brian 2, an intuitive and efficient neural simulator. *eLife*, 8, Aug. 2019. doi: 10.7554/elife.47314.

M. Stimberg, D. F. M. Goodman, and T. Nowotny. Brian2genn: accelerating spiking neural network simulations with graphics hardware. *Scientific Reports*, 10(1), Jan. 2020. doi: 10.1038/s41598-019-54957-7.

L. E. Suárez, A. Mihalik, F. Milisav, K. Marshall, M. Li, P. E. Vértes, G. Lajoie, and B. Misic. conn2res: A toolbox for connectome-based reservoir computing. Jun. 2023. doi: 10.1101/2023.05.31.543092.

D. Sussillo. Neural circuits as computational dynamical systems. *Curr. Opin. Neurobiol.*, 25:156–163, 2014.

D. Sussillo and L. F. Abbott. Generating coherent patterns of activity from chaotic neural networks. *Neuron*, 63(4):544–557, 2009.

M. Suzgun, Y. Belinkov, S. Shieber, and S. Gehrmann. LSTM networks can perform dynamic counting. In *Proceedings of the Workshop on Deep Learning and Formal Languages: Building Bridges*. Association for Computational Linguistics, 2019. doi: 10.18653/v1/w19-3905.

N. Swindale and H. Bauer. Application of kohonen's self–organizing feature map algorithm to cortical maps of orientation and direction preference. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 265(1398):827–838, May 1998. doi: 10.1098/rspb.1998.0367.

T. Tetzlaff, M. Buschermöhle, T. Geisel, and M. Diesmann. The spread of rate and correlation in stationary cortical networks. *Neurocomputing*, 52–54:949–954, 2003.

T. Tetzlaff, M. Helias, G. T. Einevoll, and M. Diesmann. Decorrelation of neural-network activity by inhibitory feedback. *PLOS Comput. Biol.*, 8(8):e1002596, 2012. doi: 10.1371/journal.pcbi.1002596.

B. B. Theyel, D. A. Llano, and S. M. Sherman. The corticothalamocortical circuit drives higher-order cortex in the mouse. *Nature Neuroscience*, 13(1):84–88, Jan. 2010. doi: 10.1038/nn.2449.

J. P. Thivierge and G. F. Marcus. The topographic brain: from neural connectivity to cognition. *Trends in Neurosciences*, 30(6):251–259, 6 2007. ISSN 01662236. doi: 10.1016/j.tins.2007.04.004.

H. K. Titley, N. Brunel, and C. Hansel. Toward a neurocentric view of learning. *Neuron*, 95(1):19–32, Jul. 2017. doi: 10.1016/j.neuron.2017.05.021.

G. Tkačik, J. S. Prentice, V. Balasubramanian, and E. Schneidman. Optimal population coding by noisy spiking neurons. *Proceedings of the National Academy of Sciences*, 107(32):14419–14424, 2010. ISSN 0027-8424. doi: 10.1073/pnas.1004906107.

T. S. to Brinke, R. Duarte, and A. Morrison. Characteristic columnar connectivity caters to cortical computation: Replication, simulation, and evaluation of a microcircuit model. *Frontiers in Integrative Neuroscience*, 16, Oct. 2022. doi: 10.3389/fnint.2022.923468.

R. Tomasello, M. Garagnani, T. Wennekers, and F. Pulvermüller. A neurobiologically constrained cortex model of semantic grounding with spiking neurons and brain-like connectivity. *Frontiers in Computational Neuroscience*, 12, Nov. 2018. doi: 10.3389/fncom.2018.00088.

G. Tononi, O. Sporns, and G. M. Edelman. A measure for brain complexity: relating functional segregation and integration in the nervous system. *Proceedings of the National Academy of Sciences*, 91(11):5033–5037, May 1994. doi: 10.1073/pnas.91.11.5033.

I. G. Torre, B. Luque, L. Lacasa, C. T. Kello, and A. Hernández-Fernández. On the physical origin of linguistic laws and lognormality in speech. *Royal Society Open Science*, 6(8):191023, Aug. 2019. doi: 10.1098/rsos.191023.

T. Toyoizumi. Nearly Extensive Sequential Memory Lifetime Achieved by Coupled Nonlinear Neurons. *Neural Comput.*, 24(10):2678–2699, Oct. 2012. ISSN 0899-7667. doi: 10.1162/NECO_a_00324.

R. Tremblay, S. Lee, and B. Rudy. GABAergic Interneurons in the Neocortex: From Cellular Properties to Circuits. *Neuron*, 91(2):260–292, 2016. ISSN 10974199. doi: 10.1016/j.neuron.2016.06.033.

S. J. Tripathy, S. D. Burton, M. Geramita, R. C. Gerkin, and N. N. Urban. Brain-wide analysis of electrophysiological diversity yields novel categorization of mammalian neuron types. *Journal of Neurophysiology*, 113(10):3474–3489, Jun. 2015. doi: 10.1152/jn.00237.2015.

N. Trouvain, L. Pedrelli, T. T. Dinh, and X. Hinaut. ReservoirPy: an Efficient and User-Friendly Library to Design Echo State Networks. In *ICANN 2020 - 29th International Conference on Artificial Neural Networks*, Bratislava, Slovakia, Sep. 2020.

A. Tsao, S. A. Yousefzadeh, W. H. Meck, M.-B. Moser, and E. I. Moser. The neural bases for timing of durations. *Nature Reviews Neuroscience*, 23(11):646–665, Sep. 2022. doi: 10.1038/s41583-022-00623-3.

M. Tsodyks, K. Pawelzik, and H. Markram. Neural networks with dynamic synapses. *Neural computation*, 10(4):821–835, 1998. ISSN 0899-7667. doi: 10.1162/089976698300017502.

M. V. Tsodyks and T. Sejnowski. Rapid state switching in balanced cortical network models. *Network: Computation in Neural Systems*, 6:111–124, 1995.

A. M. Turing. On computable numbers, with an application to the entscheidungsproblem. *Proceedings of the London Mathematical Society*, 2(1):230–265, 1937.

A. M. Turing. The chemical basis of morphogenesis. *Philosophical Transactions of the Royal Society*, 237:37–72, Aug. 1952. doi: 10.1098/rstb.1952.0012.

G. Turrigiano. Homeostatic synaptic plasticity: Local and global mechanisms for stabilizing neuronal function. *Cold Spring Harbor Perspectives in Biology*, 4(1):a005736–a005736, Nov. 2011. doi: 10.1101/cshperspect.a005736.

G. Turrigiano, L. F. Abbott, and M. E. Activity-dependent changes in the intrinsic properties of pyramidal neurons. *Science*, 264:974–977, 1994.

J. Uddén and C. Männel. Artificial Grammar Learning and Its Neurobiology in Relation to Language Processing and Development. In S.-A. Rueschemeyer and M. G. Gaskell, editors, *The Oxford Handbook of Psycholinguistics*, pages 754–783. Oxford University Press, Aug. 2018. ISBN 978-0-19-878682-5. doi: 10.1093/oxfordhb/9780198786825.013.33.

E. Vaadia, I. Haalman, M. Abeles, H. Bergman, Y. Prut, H. Slovin, and A. Aertsen. Dynamics of neuronal interactions in monkey cortex in relation to behavioural events. *Nature*, 373(6514):515–518, 2 1995. ISSN 0028-0836. doi: 10.1038/373515a0.

S. J. van Albada, M. Helias, and M. Diesmann. Scalability of asynchronous networks is limited by one-to-one mapping between effective connectivity and correlations. *PLOS Computational Biology*, 11(9):e1004490, Sep. 2015. doi: 10.1371/journal.pcbi.1004490.

E. van den Bos and F. H. Poletiek. Effects of grammar complexity on artificial grammar learning. *Memory & Cognition*, 36(6):1122–1131, Sep. 2008. doi: 10.3758/mc.36.6.1122.

D. van den Broek, M. Uhlmann, H. Fitz, R. Duarte, P. Hagoort, and K. M. Petersson. The best spike filter kernel is a neuron. In *Cognitive Computational Neuroscience*, volume 1, 2017.

M. C. W. van Rossum, G. G. Turrigiano, and S. B. Nelson. Fast propagation of firing rates through layered networks of noisy neurons. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 22(5):1956–1966, 10 2002. ISSN 1529-2401. doi: 22/5/1956[pii].

C. van Vreeswijk and H. Sompolinsky. Chaos in neuronal networks with balanced excitatory and inhibitory activity. *Science*, 274:1724–1726, Dec. 1996. doi: 10.1126/science.274.5293.1724.

C. van Vreeswijk and H. Sompolinsky. Chaotic balanced state in a model of cortical circuits. *Neural Comput.*, 10(6):1321–1371, 1998. doi: 10.1162/089976698300017214.

R. VanRullen and C. Koch. Is perception discrete or continuous? *Trends in Cognitive Sciences*, 7(5):207–213, May 2003. doi: 10.1016/s1364-6613(03)00095-0.

V. Vapnik. The support vector method of function estimation. In *Nonlinear Modeling*, pages 55–85. Springer, 1998.

E. Vasilaki, N. Frémaux, R. Urbanczik, W. Senn, and W. Gerstner. Spike-based reinforcement learning in continuous state and action space: When policy gradient methods fail. *PLOS Comput. Biol.*, 5(12):e1000586. doi:10.1371/journal.pcbi.1000586, 2009.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Adv. Neural. Inf. Process. Syst.*, 30, 2017.

D. Vecchia and D. Pietrobon. Migraine: a disorder of brain excitatory–inhibitory balance? *Trends in Neurosciences*, 35(8):507–520, Aug. 2012. doi: 10.1016/j.tins.2012.04.007.

D. Verstraeten, B. Schrauwen, M. d'Haene, and D. Stroobandt. An experimental unification of reservoir computing methods. *Neural Networks*, 20(3):391–403, 2007.

P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. doi: 10.1038/s41592-019-0686-2.

T. Vogels, H. Sprekeler, F. Zenke, C. Clopath, and W. Gerstner. Inhibitory plasticity balances excitation and inhibition in sensory pathways and memory networks. *Science*, 334(6062):1569–1573, 2011. doi: 10.1126/science.1211095.

T. P. Vogels and L. F. Abbott. Signal propagation and logic gating in networks of integrate-and-fire neurons. *J. Neurosci.*, 25(46):10786–10795, 2005.

T. P. Vogels and L. F. Abbott. Gating multiple signals through detailed balance of excitation and inhibition in spiking networks. *Nat. Neurosci.*, 12(4):483–491, 2009.

T. P. Vogels, K. Rajan, and L. F. Abbott. Neural network dynamics. *Annu. Rev. Neurosci.*, 28:357–376, 2005.

C. von der Malsburg. Self-organization of orientation selective cells in the striate cortex. *Kybernetik*, 14:85–100, 1973.

K. M. M. Walker, J. K. Bizley, A. J. King, and J. W. H. Schnupp. Multiplexed and robust representations of sound features in auditory cortex. *Journal of Neuroscience*, 31(41):14565–14576, Oct. 2011. doi: 10.1523/jneurosci.2074-11.2011.

B. A. Wandell, A. A. Brewer, and R. F. Dougherty. Visual field map clusters in human cortex. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 360 (1456):693–707, Apr. 2005. doi: 10.1098/rstb.2005.1628.

B. A. Wandell, S. O. Dumoulin, and A. A. Brewer. Visual field maps in human cortex. *Neuron*, 56(2):366–383, Oct. 2007. doi: 10.1016/j.neuron.2007.10.012.

M. R. Warden and E. K. Miller. Task-dependent changes in short-term memory in the prefrontal cortex. *Journal of Neuroscience*, 30(47):15801–15810, 2010.

R. Warren and P. J. Schroeder. Topological entropy measure of artificial grammar complexity for use in designing experiments on Human performance in intelligence, surveilance and reconnaissance (ISR) tasks. Technical report, 2015.

P. Weidel, R. Duarte, and A. Morrison. Unsupervised learning and clustered connectivity enhance reinforcement learning in spiking neural networks. *Frontiers in Computational Neuroscience*, 15, Mar. 2021. doi: 10.3389/fncom.2021.543872.

G. Weiss, Y. Goldberg, and E. Yahav. On the practical computational power of finite precision rnns for language recognition, 2018. doi: 10.48550/ARXIV.1805.04908.

N. Weisz, C. Wienbruch, S. Hoffmeister, and T. Elbert. Tonotopic organization of the human auditory cortex probed with frequency-modulated tones. *Hearing Research*, 191(1-2):49–58, 5 2004. ISSN 03785955. doi: 10.1016/j.heares.2004.01.012.

Welch. A technique for high-performance data compression. *Computer*, 17(6):8–19, Jun. 1984. doi: 10.1109/mc.1984.1659158.

P. Werbos. Backpropagation through time: what it does and how to do it. *Procedings of the IEEE*, 78(10):1550–1560, 1990. doi: 10.1109/5.58337.

J. C. Whittington and R. Bogacz. Theories of error back-propagation in the brain. *Trends. Cogn. Sci.*, 23(3):235–250, 2019. doi: https://doi.org/10.1016/j.tics.2018.12.005.

J. Wiles and J. Elman. Learning to count without a counter: A case study of dynamics and activation landscapes in recurrent networks. In *Proceedings of the seventeenth annual conference of the cognitive science society*, number s 482, page 487. MIT Press Cambridge, MA, 1995.

R. J. Williams and D. Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*, 1(2):270–280, Jun. 1989. doi: 10.1162/neco.1989.1.2.270.

B. Wilson, M. Spierings, A. Ravignani, J. L. Mueller, T. H. Mintz, F. Wijnen, A. Kant, K. Smith, and A. Rey. Non-adjacent dependency learning in humans and other animals. *Topics in Cognitive Science*, 12(3):843–858, Sep. 2018. doi: 10.1111/tops.12381.

R. C. Wilson and A. G. Collins. Ten simple rules for the computational modeling of behavioral data. *eLife*, 8, Nov. 2019. doi: 10.7554/elife.49547.

K.-F. Wong. A recurrent network mechanism of time integration in perceptual decisions. *Journal of Neuroscience*, 26(4):1314–1328, Jan. 2006. doi: 10.1523/jneurosci.3733-05.2006.

Y. K. Wu and F. Zenke. Nonlinear transient amplification in recurrent neural networks with short-term plasticity. *eLife*, 10, dec 2021. doi: 10.7554/elife.71263.

S. Xu, W. Jiang, M. M. Poo, and Y. Dan. Activity recall in a visual cortical ensemble. *Nat. Neurosci.*, 15(3):449–455, mar 2012. ISSN 10976256. doi: 10.1038/nn.3036.

Y. Yamaguti and I. Tsuda. Mathematical modeling for evolution of heterogeneous modules in the brain. *Neural Networks*, 62:3–10, Feb. 2015. doi: 10.1016/j.neunet.2014.07.013.

T. Yamazaki and S. Tanaka. The cerebellum as a liquid state machine. *Neural Networks*, 20(3):290–297, Apr. 2007. doi: 10.1016/j.neunet.2007.04.004.

G. R. Yang, M. R. Joglekar, H. F. Song, W. T. Newsome, and X.-J. Wang. Task representations in neural networks trained to perform many cognitive tasks. *Nature Neuroscience*, 22(2):297–306, Jan. 2019. doi: 10.1038/s41593-018-0310-2.

O. Yizhar, L. E. Fenno, M. Prigge, F. Schneider, T. J. Davidson, D. J. O'Shea, V. S. Sohal, I. Goshen, J. Finkelstein, J. T. Paz, K. Stehfest, R. Fudim, C. Ramakrishnan, J. R. Huguenard, P. Hegemann, and K. Deisseroth. Neocortical excitation/inhibition balance in information processing and social dysfunction. *Nature*, 477(7363):171–178, Jul. 2011. doi: 10.1038/nature10360.

J. M. Zacks. Event perception and memory. *Annual Review of Psychology*, 71(1):165–191, Jan. 2020. doi: 10.1146/annurev-psych-010419-051101.

J. M. Zacks and B. Tversky. Event structure in perception and conception. *Psychological Bulletin*, 127(1):3–21, 2001. doi: 10.1037/0033-2909.127.1.3.

B. Zajzon and A. Morales-Gregorio. Trans-thalamic Pathways: Strong Candidates for Supporting Communication between Functionally Distinct Cortical Areas. *Journal of Neuroscience*, 39(36):7034–7036, Sep. 2019. ISSN 0270-6474. doi: 10.1523/JNEUROSCI.0656-19.2019.

B. Zajzon, S. Mahmoudian, A. Morrison, and R. Duarte. Passing the message: Representation transfer in modular balanced networks. *Frontiers in Computational Neuroscience*, 13(December):79, 2019. ISSN 1662-5188. doi: 10.3389/fncom.2019.00079.

B. Zajzon, D. Dahmen, A. Morrison, and R. Duarte. Signal denoising through topographic modularity of neural circuits. *Zenodo*, 2022. doi: 10.5281/ZENODO.6326497.

B. Zajzon, D. Dahmen, A. Morrison, and R. Duarte. Signal denoising through topographic modularity of neural circuits. *eLife*, 12, Jan. 2023. doi: 10.7554/elife.77009.

F. Zenke and S. Ganguli. SuperSpike: Supervised learning in multilayer spiking neural networks. *Neural Computation*, 30(6):1514–1541, Jun. 2018. doi: 10.1162/neco_a_01086.

F. Zenke and W. Gerstner. Limits to high-speed simulations of spiking neural networks using general-purpose computers. *Frontiers in Neuroinformatics*, 8:76, 2014.

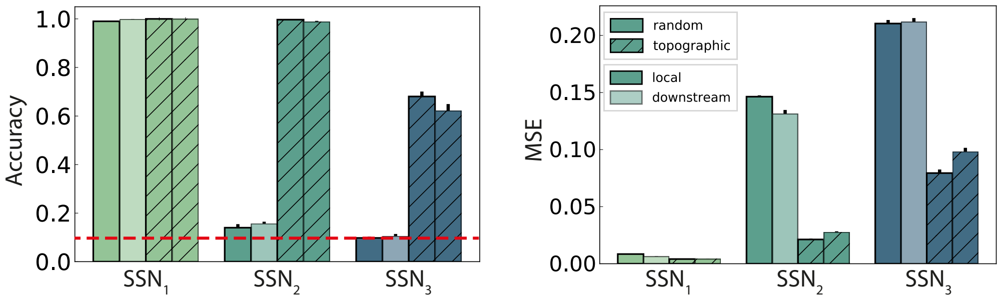
F. Zenke, E. J. Agnes, and W. Gerstner. Diverse synaptic plasticity mechanisms orchestrated to form and retrieve memories in spiking neural networks. *Nat Commun*, 6(1), apr 2015. doi: 10.1038/ncomms7922.

J. Zylberberg. Untuned But Not Irrelevant: A Role For Untuned Neurons In Sensory Information Coding. *bioRxiv*, 13(4):1–35, 04 2017. doi: 10.1371/journal.pcbi.1005497.

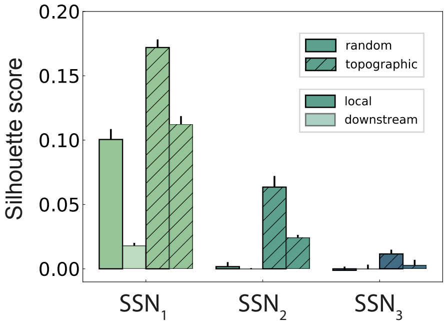
Appendix A

Supplementary materials for Chapter 5

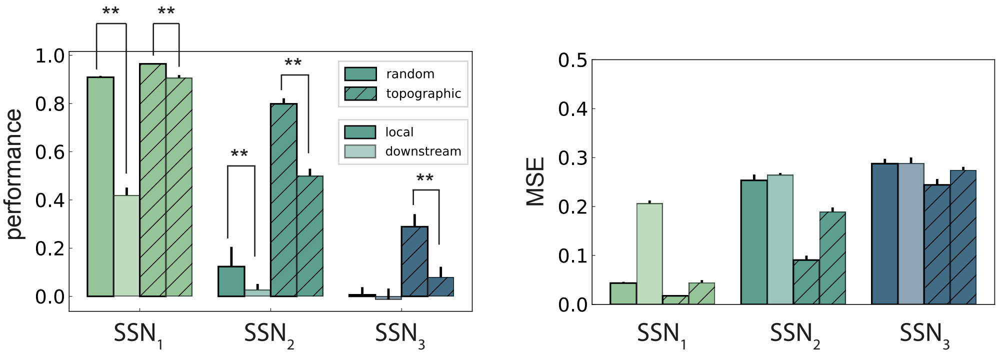
A.1 Supplementary figures



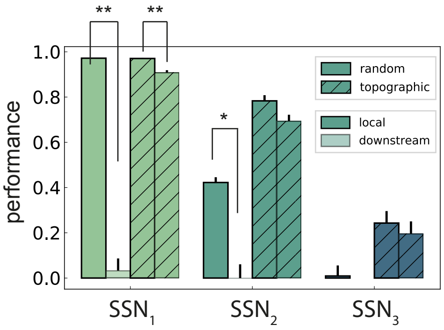
Supplementary Figure A.1: Classification accuracy and MSE computed using 10 stimuli from the second input stream, S'. There are no significant differences between local and downstream integration.



Supplementary Figure A.2: Silhouette score quantifying cluster separability in the XOR task. Scores are calculated in the space spanned by the first ten PCs, using the low-pass filtered spike trains.



Supplementary Figure A.3: XOR performance computed on the low-pass filtered spike trains. The differences in performance are statistically significant, with local integration proving to be consistently more beneficial. These results are in agreement with the values computed using the membrane potentials as state variables.



Supplementary Figure A.4: XOR performance for networks with non-scaled feed-forward projections between $\mathrm{SSN}_0 \rightarrow \mathrm{SSN}_1$ and $\mathrm{SSN}_0' \rightarrow \mathrm{SSN}_1$ in the downstream integration scenario (Figure 5.6B). The denser connectivity (p_{ff} instead of $p_{\mathrm{ff}}/2$ as in Figure 5.7A) does not significantly alter the relative differences between local and downstream integration.

A.2 Supplementary tables

A: Model Summary	
Populations	Multiple modules, each composed of 1 excitatory and 1 inhibitory subpopulation
Connectivity	Sparse, random recurrent connectivity with random or topographically structured feed-forward projections
Neuron Model	Leaky integrate-and-fire, fixed voltage threshold, fixed absolute refractory time, no adaptation
Synapse Model	Conductance-based, exponential, no plasticity
Input	Stochastic background spikes and inhomogeneous Poisson spikes onto 10% E and 10% I neurons

B: Populations		
Name	**Elements**	**Size**
E_i, E_0'	`iaf_cond_exp`	8000
I_i, I_0'	`iaf_cond_exp`	2000

C: Neuron Models	
Name	Leaky integrate-and-fire neuron (`iaf_cond_exp`)

Subthreshold Dynamics	if $(t > t^* + \tau_\mathrm{ref})$
	$C_\mathrm{m}\frac{dV_i}{dt} = \mathrm{g}_\mathrm{leak}(V_\mathrm{rest} - V_i(t)) + I_i^\mathrm{E}(t) + I_i^\mathrm{I}(t) + I_i^\mathrm{x}(t)$
	else
	$V(t) = V_\mathrm{reset}$

Synaptic Transmission	$I_\mathrm{ij}^\mathrm{syn}(t) = g_\mathrm{ij}^\mathrm{syn}(V_\mathrm{syn} - V_i(t))$
Spiking	If $V(t-) < V_\mathrm{th}$ OR $V(t+) \geq V_\mathrm{th}$ 1. set $t^* = t$ 2. emit spike with time stamp t^*

D: Synapse Models	
Synaptic Conductance	$\frac{dg_\mathrm{ij}(t)}{dt} = -\frac{g_\mathrm{ij}(t)}{\tau_\beta} + \bar{g}^\beta \sum_{t_j} \delta(t - t_j - d)$

E: Input		
Type	**Target**	**Description**
poisson_generator	E_0, I_0	Total rate $\nu_\mathrm{X} \cdot \mathrm{K_X}$
poisson_generator	E_i, I_i for $i > 0$	Total rate $0.25 \cdot \nu_\mathrm{X} \cdot \mathrm{K_X}$
Inhomogeneous Poisson generator	$\mathrm{E}_0^{(k)}$, $\mathrm{I}_0^{(k)}$ for $S_k \in S$ $\mathrm{E}_0^{'(j)}$, $\mathrm{I}_0^{'(j)}$ for $S_j' \in S'$	Inhomogeneous Poisson process with rate ν_{stim}, changing every 200 ms

Supplementary Table A.1: Tabular description of network model after Nordlie et al. (2009).

A: Populations		
Name	**Value**	**Description**
N^{E}	8000	Excitatory population size in each module
N^{I}	2000	Excitatory population size in each module

B: Connectivity		
Name	**Value**	**Description**
d	1.5 ms	Synaptic transmission delay
$\bar{g}_{\mathrm{E}}$	1 nS	Excitatory synaptic conductance
$\bar{g}_{\mathrm{I}}$	$\gamma\bar{g}_{\mathrm{E}}$ nS	Inhibitory synaptic conductance
γ	16	Scaling factor for the inhibitory synapses
ϵ	0.1	Baseline connection probability
p_{x}	ϵ	Connection probability for background noise input in SSN_0
	0.25ϵ	Scaled connection probability for background input in $\mathrm{SSN}_i, i > 0$
p_{ff}	0.75ϵ	Feed-forward connection probability within topographic maps

B: Neuron Model		
Name	**Value**	**Description**
C_{m}	250 pF	Membrane capacitance
E_L	−70 mV	Resting membrane potential
τ_{m}	15 ms	Membrane time constant
V_{th}	−50 mV	Membrane potential threshold for action-potential firing
V_{reset}	−60 mV	Reset potential
τ_{ref}	2 ms	Absolute refractory period
g_{L}	16.7 nS	Leak conductance

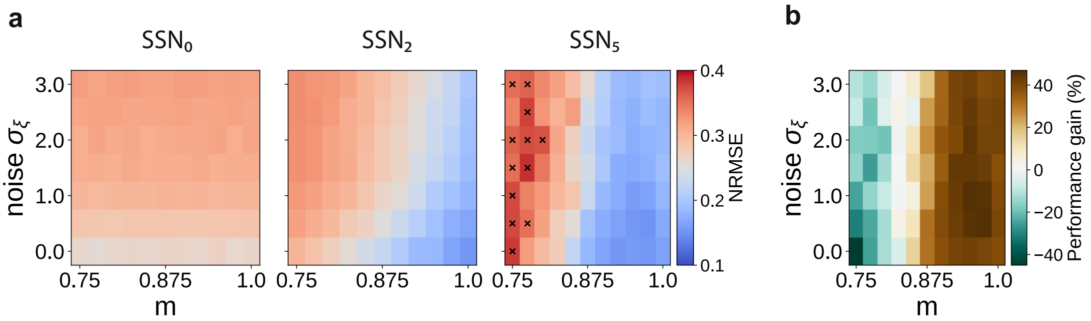
C: Synapse Model		
τ_{E}	5 ms	Synaptic decay time constant for excitatory synapses
τ_{I}	10 ms	Synaptic decay time constant for inhibitory synapses
V_{E}	0 mV	Excitatory reversal potential
V_{I}	−80 mV	Inhibitory reversal potential

Supplementary Table A.2: Summary of all the numerical experiments that can be run using the provided source code.

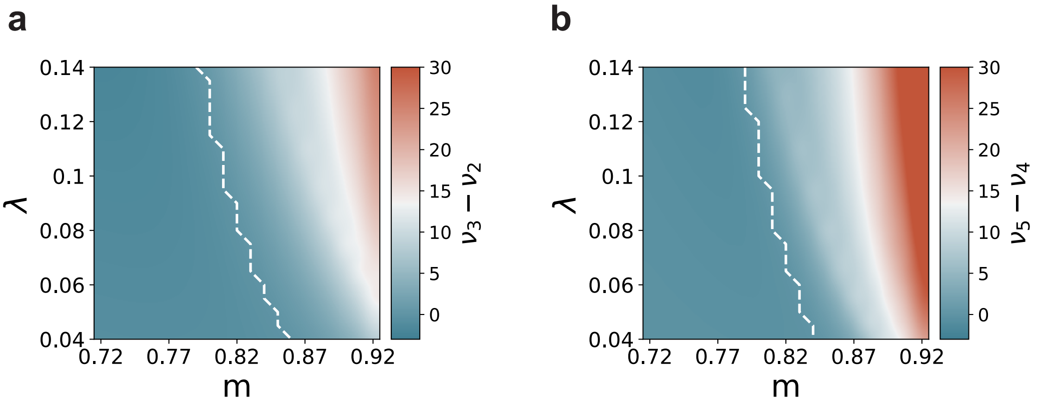
Appendix B

Supplementary materials for Chapter 6

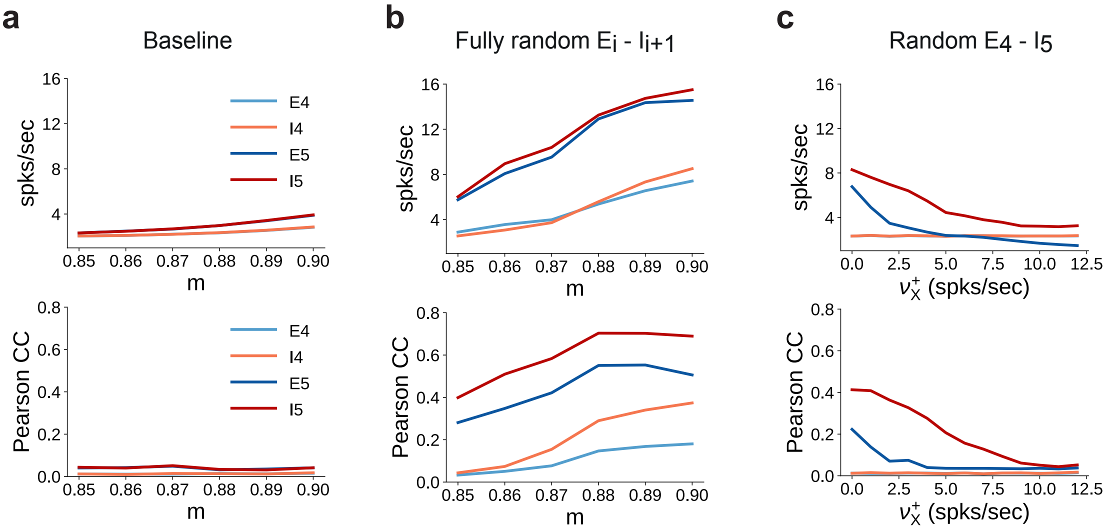
B.1 Supplementary figures



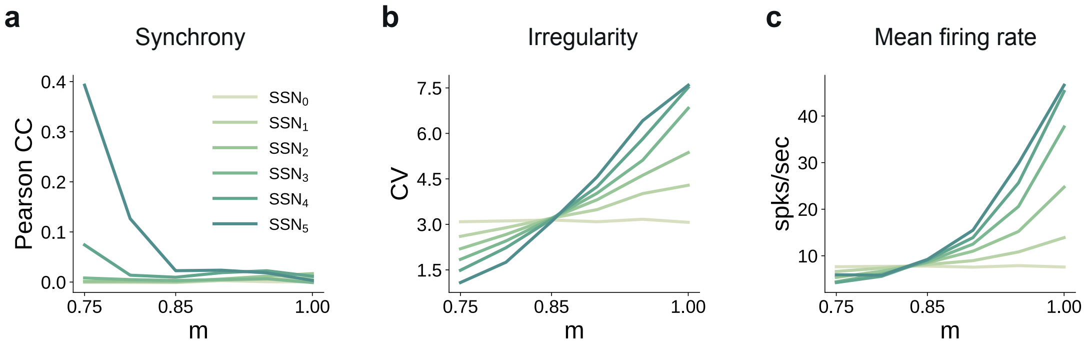
Supplementary Figure B.1: **Sequential denoising effect. (a)** Reconstruction error (NRMSE) in 3 different sub-networks as a function of modularity (m) and noise amplitude (σ_ϵ). The points marked in the rightmost panel correspond to chance-level reconstruction accuracy. **(b)** Relative reconstruction performance gain in SSN$_5$ compared to SSN$_0$, expressed as percentage of error decrease.



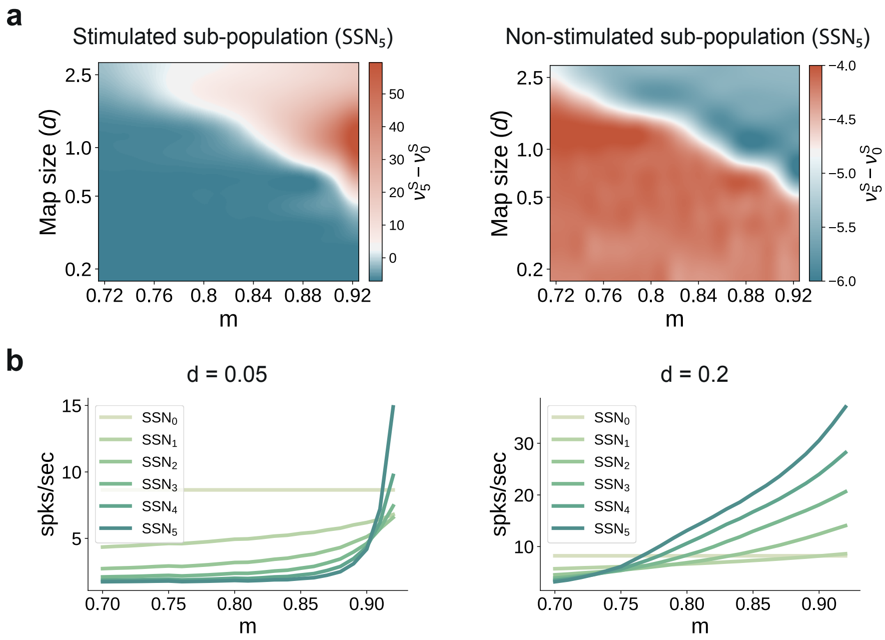
Supplementary Figure B.2: Mean-field predictions for the gain in the firing rates of stimulated sub-populations (**(a)** $\nu_3^S - \nu_2^S$ and **(b)** $\nu_5^S - \nu_4^S$), as a function of modularity m and input intensity, scaled by λ (see Section 6.3). Dashed lines demarcate the transition to positive gain.



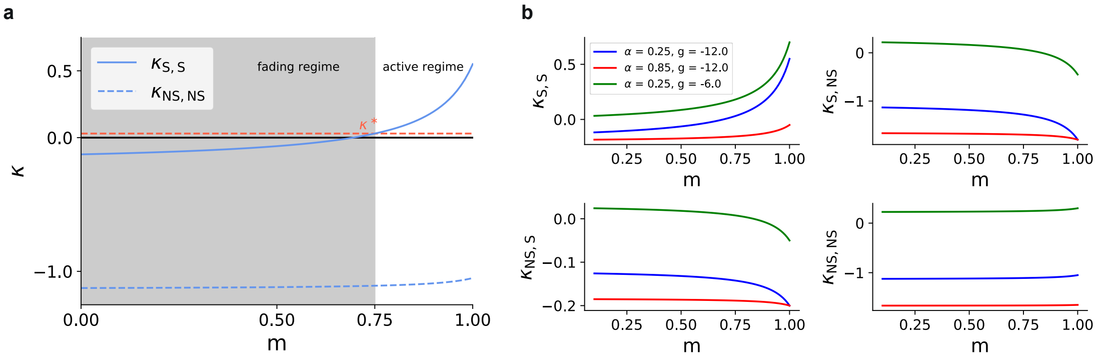
Supplementary Figure B.3: **Spiking statistics for different feedforward wiring to inhibitory neurons.** **(a)** Mean firing rates (top panel) and synchrony (Pearson's correlation coefficient, computed pairwise over spikes binned into 2 ms bins and averaged across 500 pairs, lower panel) in SSN_4 and SSN_5, as a function of modularity. **(b)** Same as **(a)**, except with random feedforward projections to the inhibitory pools, i.e., $m = 0$ for all $\mathrm{E_i} \to \mathrm{I_{i+1}}$ connections, $i = \{0..4\}$. **(c)** Same as the baseline network in **(a)**, with $m = 0$ only for $\mathrm{E}_4 \to \mathrm{I}_5$. In addition, each neuron in I_5 receives further excitatory background input with intensity $\nu'_\mathrm{X} = \nu_\mathrm{X} + \nu_\mathrm{X}^+$. Statistics are computed as a function of the additional rate ν_X^+.



Supplementary Figure B.4: **Spiking statistics for the conductance-based model: (a)** synchrony (Pearson's correlation coefficient, computed pairwise over spikes binned into 2 ms bins and averaged across 500 pairs); **(b)** irregularity measured as the coefficient of variation (CV); **(c)** mean firing rate across the excitatory populations. All depicted statistics were averaged over five simulations, each lasting 5 s, with 10 input stimuli.

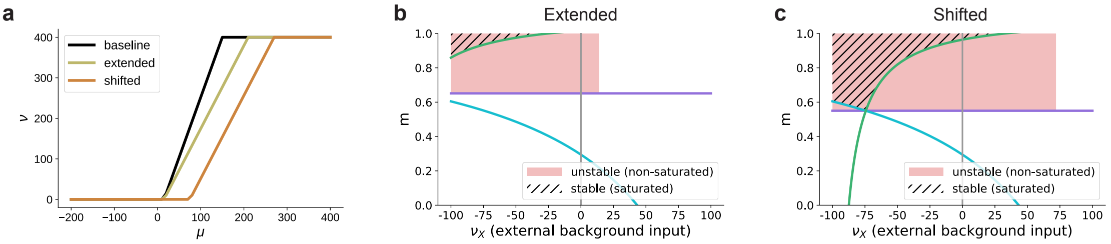


Supplementary Figure B.5: **Transition point in modularity decreases with larger map sizes.** **(a)** Mean-field predictions for the stationary firing rates of the stimulated (left) and non-stimulated sub-populations (right) in SSN$_5$, as a function of modularity (m) and fixed map size (parametrized by d, see Section 6.3) across the modules ($\delta = 0$). To limit the impact of additional parameters when varying the map sizes (e.g., overlap), the number of stimulus-specific sub-populations and d where chosen such that every neuron in each population belonged to exactly one stimulus-specific sub-population (see main text for more details). **(b)** Predicted firing rates in the stimulated sub-populations of the different sub-networks, for $d = 0.05$ (left) and $d = 0.2$ (right), with $\delta = 0$.

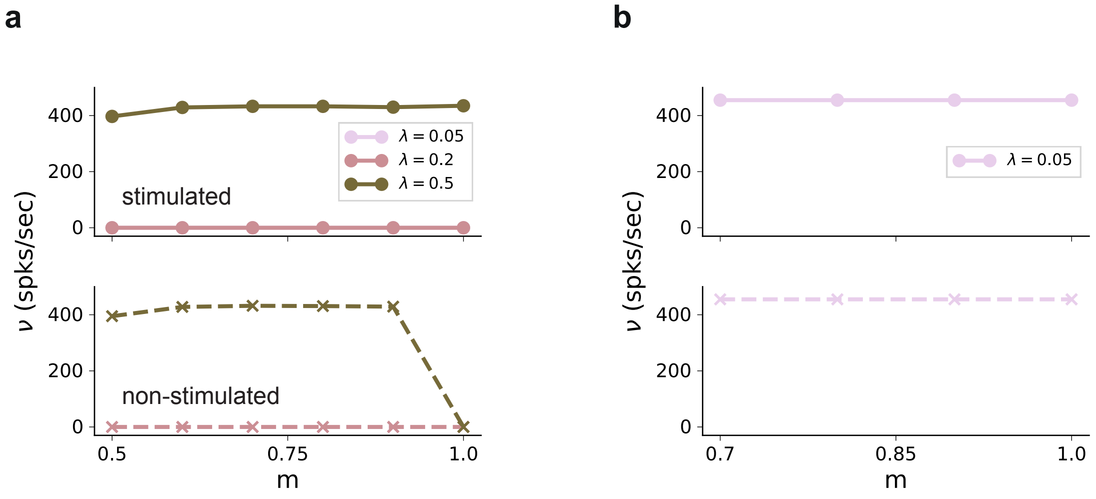


Supplementary Figure B.6: **Effective couplings.** **(a)** The effective coupling between stimulated sub-populations, $\kappa_{\mathrm{S,S}}$ increases with modularity, with κ^* marking the critical transition point between the fading and active regime. Dashed red line represents this κ^* value. **(b)** $\kappa_{\mathrm{S,S}}$ (top left panel), $\kappa_{\mathrm{S,NS}}$ (top right), $\kappa_{\mathrm{NS,S}}$ (bottom left), $\kappa_{\mathrm{NS,NS}}$ (bottom right), as a function of modularity for different values of α and E/I weight ratio g. The parameters used in this study (blue) yield $\kappa_{\mathrm{S,S}} > 0$ only for large modularity, with the other couplings being negative for all m. Increasing the signal-to-noise ratio to $\mathrm{SSN}_{\geq 1}$ (red), i.e., increasing the background external input while reducing the feedforward connection density (directly coupled, see Section 6.3), destroys bistability (all couplings are negative for all values of m) and leads to extinction of activity in all sub-populations in the limit of very deep networks. Decreasing inhibition (green) also creates possible bistability for non-stimulated sub-populations ($\kappa_{\mathrm{NS,NS}} > 0$) such that their activity might approach a high-activity fixed point, leading to destruction of task performance. Note that in all panels, the values κ are scaled by $1/\mathcal{J}$ to highlight the transitions around zero (see Eq. B.7 in Appendix B)

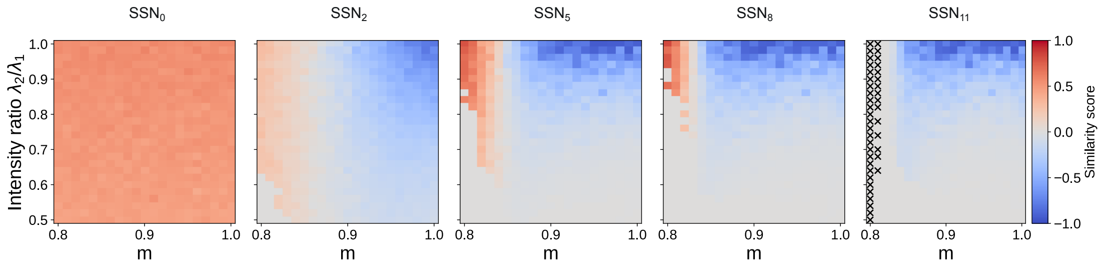
.



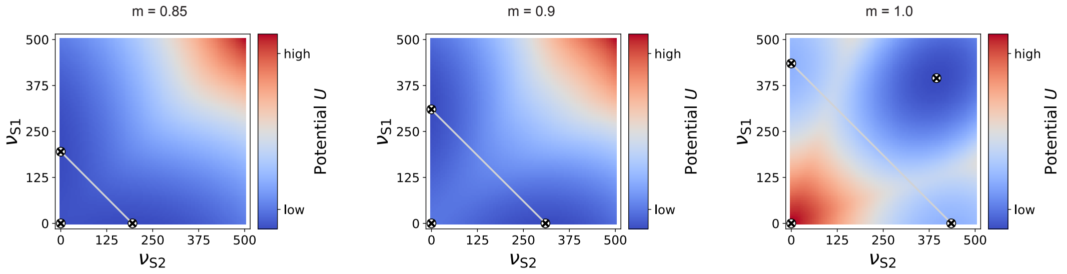
Supplementary Figure B.7: **Influence of the activation function's dynamic range on the bifurcation behavior in excitation-dominated networks** ($g = -3$, see also Figure 6.8E). **(a)** The baseline dynamic range $[15, 150]$ is extended to $[15, 210]$ or shifted to $[75, 210]$. **(b)** Given that μ_{max} does not enter the lower bound on the modularity determined by Eq. 6.3 (green curve), extending the dynamic range (see panel **(a)**) does not affect the region of stable fixed points in the parameter space. For positive background input, there are no stable fixed points, only unstable ones at non-saturated rates for low values of ν_{X}, due to excitatory recurrent fluctuations in the activity. For stronger background input, no fixed points exist where $\nu_{\text{S}} > \nu_{\text{NS}}$. In this case, the activity of the non-stimulated populations (non-zero) dominates the recurrent dynamics and denoising can not be achieved. **(c)** Shifting the dynamic range altogether (see panel **(a)**) leads to the emergence of stable fixed points at saturated rates also for positive external input, but the region in which denoising occurs is still significantly smaller than for networks with recurrent inhibition (see Figure 6.8c). For these values, $\nu_{\text{NS}} = 0$ is ensured because the total input to the non-stimulated populations remains below the shifted dynamical range, in contrast to just extending the dynamic range where even low inputs can lead to non-zero activity. Moreover, the shifted activation function requires a biologically implausible strength of input for activation. The firing threshold of biological neurons is typically $15 - 20$mV above the resting membrane potential, which is much less than the shifted μ_{min}. Note that similarly to Figure 6.8, here we plot only the fixed points for $\nu_{\text{S}} > 0$ and $\nu_{\text{NS}} = 0$.



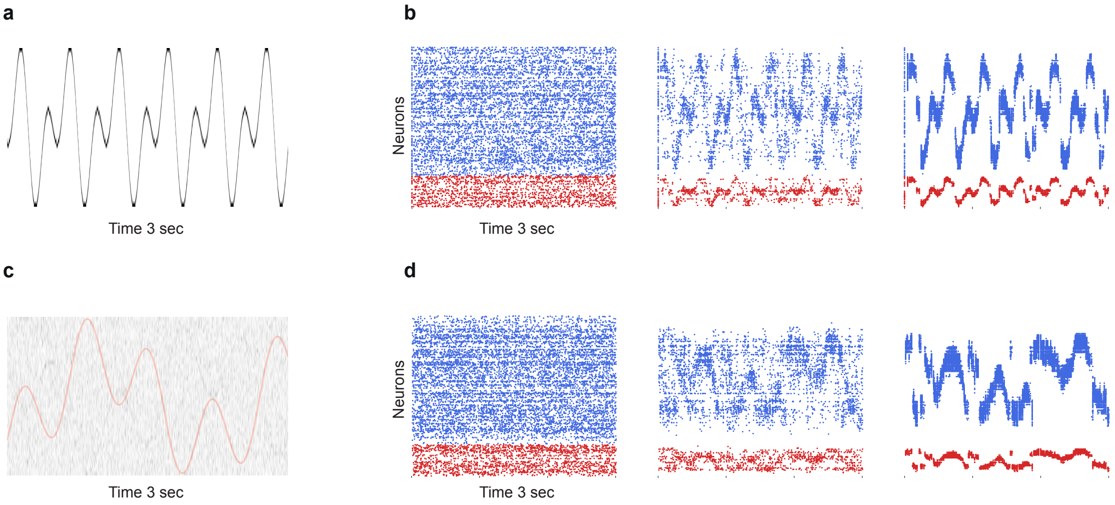
Supplementary Figure B.8: **Firing rates in** SSN_5 **in the absence of external background noise** ($\nu_X = 0$)**.** **(a)** Firing rates of stimulated (top) and non-stimulated (bottom) populations obtained from simulations of a network without any recurrent connections, for different input intensities λ. Successful denoising only occurs for the extreme case of $m = 1$, in which case the pathways are completely segregated. Note that the input rate was unchanged, $\nu_{\text{in}} = 12\lambda K_E$. **(b)** Same as **(a)**, but for excitatory recurrence ($g = -3$). Recurrent excitation spreads the input from the stimulated pathway to non-stimulated neurons. Results shown only for $\lambda = 0.05$, with larger values leading to similar results.



Supplementary Figure B.9: **Evolution of similarity score for 12 sub-networks.** Correlation-based similarity score illustrates the three dynamical regimes observed across the different sub-networks, for two input streams: coexisting (CoEx, red area and positive score), winner-take-all behavior (grey, score near 0) and winnerless competition (WLC, blue and negative score). As predicted by the mean-field analysis, the CoEx region vanishes with increasing network depth. The calculation of the similarity score is detailed in Section 6.3. If either stimuli could not be decoded, we set the score to 0. In SSN_{11}, 'X' indicates parameter combinations where none of the stimuli could be decoded.



Supplementary Figure B.10: **Potential landscape for two input streams.** For intermediate modularity ($m = 0.85$, left and $m = 0.9$, right), there are two high-activity fixed points (circled cross markers) in addition to the low-activity one near zero (marker added manually here, as it is not observable due to the larger integration step of 5 spks/sec used here). If the projections are almost fully modular ($m \approx 1$), an additional high-activity fixed point can be observed for identical ν^{S1} and ν^{S2}. In this case, the two stimulated sub-populations can be considered as one larger population, for which the common $\kappa_{S,S}$ becomes positive, as in the case of a single input stream (see Supplementary Figure B.6), just for larger m. Grey, anti-diagonal lines represent the one-dimensional sections illustrated in Figure 6.9e.



Supplementary Figure B.11: **Limits of denoising for rapidly changing and noisy dynamical inputs.** **(a)** A fast changing input signal $x(t) = \sin(24t)+\cos(12t)$ sampled at $dt = 1$ms, with no additional noise ($\sigma_\xi = 0$). **(b)** While portions of the signal can be successfully transmitted and denoised in SSN$_5$, there are significant periods (steep slopes) where the signal representation is lost. **(c)** Slower signal $x(t) = \sin(10t) + \cos(3t)$ with significant noise corruption ($\sigma_\xi = 2\nu_{\text{in}}$). Continuous red curve denotes the input signal $u(t)$. **(d)** Strong noise in the input leads to heavy fluctuations in the activity of the deeper populations, corrupting the signal representations.

B.2 Supplementary tables

A: Model Summary	
Populations	Multiple modules, each composed of 1 excitatory and 1 inhibitory sub-population
Connectivity	Sparse, random recurrent connectivity with random or topographically structured feed-forward projections (fixed in-degrees)
Neuron Model	Leaky integrate-and-fire, fixed voltage threshold, no adaptation
Synapse Model	Exponentially decaying postsynaptic currents, static synaptic weights, fixed delays
Input	Stochastic background spikes and inhomogeneous Poisson spikes onto $d_0 N^{\mathrm{E}}$ excitatory and $d_0 N^{\mathrm{E}}$ inhibitory neurons in SSN_0

B: Populations		
Name	**Elements**	**Size**
E_i / I_i	`iaf_psc_exp`	8000 / 2000

C: Neuron Models	
Name	Leaky integrate-and-fire (LIF) neuron (`iaf_psc_exp`)

Subthreshold Dynamics	if $(t > t^* + \tau_{\mathrm{ref}})$
	$\tau_{\mathrm{m}} \frac{dV_i(t)}{dt} = (V_{\mathrm{rest}} - V_i(t)) + R_{\mathrm{m}} \left(I_i^{\mathrm{E}}(t) + I_i^{\mathrm{I}}(t) + I_i^{\mathrm{X}}(t) \right)$
	else
	$V(t) = V_{\mathrm{reset}}$

Spiking	If $V(t-) < V_{\mathrm{th}}$ OR $V(t+) \geq V_{\mathrm{th}}$ 1. set $t^* = t$ 2. emit spike with time stamp t^*

D: Synapse Models	
Synaptic Transmission	$\tau_\beta \frac{dI_i^\beta(t)}{dt} = -I_i(t) + \tau_\beta \hat{I}_\beta \sum_j \sum_k \delta(t - t_j^k)$ with postsynaptic potential $\mathrm{PSP}_{ij}(t) = \hat{I}_\beta \frac{R_{\mathrm{m}} \tau_\beta}{\tau_\beta - \tau_{\mathrm{m}}} \left(e^{-t/\tau_\beta} - e^{-t/\tau_{\mathrm{m}}} \right) \Theta(t)$ and Heaviside function $\Theta(t)$. The synaptic efficacy (weight) corresponds to the PSP amplitude: $J_\beta = \hat{I}_\beta R_{\mathrm{m}} \frac{\tau_\beta}{\tau_\beta - \tau_{\mathrm{m}}} \left(\left[\frac{\tau_{\mathrm{m}}}{\tau_\beta} \right]^{\frac{-\tau_m}{\tau_m - \tau_\beta}} - \left[\frac{\tau_{\mathrm{m}}}{\tau_\beta} \right]^{\frac{-\tau_\beta}{\tau_{\mathrm{m}} - \tau_\beta}} \right)$

E: Input		
Type	**Target**	**Description**
poisson_generator	E_0, I_0	Total rate $\nu_{\mathrm{X}} \cdot \mathrm{K}_{\mathrm{X}}$
poisson_generator	E_i, I_i for $i > 0$	Total rate $0.25 \cdot \nu_{\mathrm{X}} \cdot \mathrm{K}_{\mathrm{X}}$
Inhomogeneous Poisson generator	$\mathrm{E}_0^{(k)}$, $\mathrm{I}_0^{(k)}$ for $S_k \in S$ $\mathrm{E}_0^{'(j)}$, $\mathrm{I}_0^{'(j)}$ for $S'_j \in S'$	Inhomogeneous Poisson process with rate ν_{stim}, changing every 200 ms

Supplementary Table B.1: Tabular description of current-based (baseline) network model after Nordlie et al. (2009).

A: Populations

Name	Value	Description
N^{E}	8000	Excitatory population size in each module
N^{I}	2000	Inhibitory population size in each module

B: Connectivity

Name	Value	Description
ϵ	0.1	Baseline connection probability
α	0.25	Connection scaling factor for $\mathrm{SSN}_{i>0}$
p_{x}	ϵ	Connection probability for background noise input in SSN_0
	$\alpha\epsilon$	Scaled connection probability for background input in $\mathrm{SSN}_i, i > 0$
σ_{i}	$(1-\alpha) * \epsilon$	Fixed density of feed-forward projection matrices
p_{c}	$(1-m) * p_0$	Feed-forward connection probability within topographic maps
p_0	$(1-m) * p_{\mathrm{c}}$	Feed-forward connection probability between SPs on different topographic maps

B: Neuron Model

Name	Value	Description
C_{m}	250 pF	Membrane capacitance
E_L	-70 mV	Resting membrane potential
τ_{m}	20 ms	Membrane time constant
V_{th}	-55 mV	Membrane potential threshold for action-potential firing
V_{reset}	-60 mV	Reset potential
τ_{ref}	2 ms	Absolute refractory period

C: Synapse Model

τ_{E}	2 ms	Synaptic decay time constant for excitatory synapses
τ_{I}	2 ms	Synaptic decay time constant for inhibitory synapses
d	1.5 ms	Synaptic transmission delay
$\hat{I}_{\mathrm{E}}$	32.78 pA	Peak excitatory current
$\hat{I}_{\mathrm{I}}$	g32.78 pA	Peak inhibitory current
J_{E}	0.2 mV	EPSP amplitude
J_{I}	g0.2 mV	IPSP amplitude
g	-12	Scaling factor for the inhibitory synapses

Supplementary Table B.2: Summary of all the model parameters for the current-based network. For more details, see Zajzon et al. (2019).

A: Populations

Name	Value	Description
N^{E}	8000	Excitatory population size in each module
N^{I}	2000	Excitatory population size in each module

B: Connectivity

Name	Value	Description
d	1.5 ms	Synaptic transmission delay
$\bar{g}_{\mathrm{E}}$	1nS	Excitatory synaptic conductance
$\bar{g}_{\mathrm{I}}$	$g\bar{g}_{\mathrm{E}}$nS	Inhibitory synaptic conductance
g	-16	Scaling factor for the inhibitory synapses
ϵ	0.1	Baseline connection probability
α	0.25	Connection scaling factor for $\mathrm{SSN}_{i>0}$
p_{x}	ϵ	Connection probability for background noise input in SSN_0
	$\alpha\epsilon$	Scaled connection probability for background input in $\mathrm{SSN}_i, i > 0$
σ_{i}	$(1-\alpha) * \epsilon$	Fixed density of feed-forward projection matrices
p_{c}	$(1-m) * p_0$	Feed-forward connection probability within topographic maps
p_0	$(1-m) * p_{\mathrm{c}}$	Feed-forward connection probability between SPs on different topographic maps

B: Neuron Model

Name	Value	Description
C_{m}	250 pF	Membrane capacitance
E_L	-70 mV	Resting membrane potential
τ_{m}	15 ms	Membrane time constant
V_{th}	-50 mV	Membrane potential threshold for action-potential firing
V_{reset}	-60 mV	Reset potential
τ_{ref}	2 ms	Absolute refractory period
g_{L}	16.7nS	Leak conductance

C: Synapse Model

τ_{E}	5 ms	Synaptic decay time constant for excitatory synapses
τ_{I}	10 ms	Synaptic decay time constant for inhibitory synapses
V_{E}	0 mV	Excitatory reversal potential
V_{I}	-80 mV	Inhibitory reversal potential

Supplementary Table B.3: Parameter values for the conductance-based model.

A: Model Summary	
Populations	Multiple modules, each one composed of 1 excitatory and 1 inhibitory sub-population
Topology	None
Connectivity	Sparse, random recurrent connectivity with modular topographic feed-forward projections (fixed in-degrees)
Neuron Model	Rate neuron with shifted tanh gain function
Synapse Model	Delayed rate connection
Plasticity	None
Input	Uniformly distributed input onto $d_0 N^{\mathrm{E}}$ excitatory and $d_0 N^{\mathrm{E}}$ inhibitory neurons in SSN$_0$
Measurements	Unit output (rate)

B: Populations		
Name	**Elements**	**Size**
E$_i$	rate neuron	2400
I$_i$	rate neuron	600

C: Neuron Models	
Name	Rate neuron
Differential equation	$\boldsymbol{\tau_x} \frac{\mathrm{d}\boldsymbol{x}}{\mathrm{dt}} = -\boldsymbol{x} + W\boldsymbol{r} + W^{\mathrm{in}}\boldsymbol{u} - \boldsymbol{b}^{\mathrm{rec}} + \sqrt{2\boldsymbol{\tau_x}}\sigma_{\mathrm{X}}\xi$ $\boldsymbol{r} = 0.5(1 + \tanh(\boldsymbol{x}))$

D: Input		
Type	**Target**	**Description**
random uniform distribution	E$_0$, I$_0$	Step signal input to SSN$_0$, changing every 200 ms, with amplitude 0.8
Gaussian white noise	E$_i$, I$_i$ for $i \in \{0..5\}$	Intrinsic unit noise

Supplementary Table B.4: Description of the rate model.

<table>
<tr><th colspan="3">A: Populations</th></tr>
<tr><td><b>Name</b></td><td><b>Value</b></td><td><b>Description</b></td></tr>
<tr><td>N^{E}</td><td>2400</td><td>Number of excitatory units in each module</td></tr>
<tr><td>N^{I}</td><td>600</td><td>Number of inhibitory units in each module</td></tr>
<tr><th colspan="3">B: Connectivity</th></tr>
<tr><td><b>Name</b></td><td><b>Value</b></td><td><b>Description</b></td></tr>
<tr><td>d</td><td>1 ms</td><td>Synaptic transmission delay</td></tr>
<tr><td>ϵ</td><td>0.2</td><td>Baseline connection probability</td></tr>
<tr><td>w_{in}</td><td>$\sim U(0.9, 1.0)$</td><td>Input weights</td></tr>
<tr><td>w_{in}</td><td>$\sim N_{\mathrm{tr}}(0, 1/\sqrt{\epsilon N}) > 0$</td><td>Recurrent and feed-forward weights drawn from a normal distribution truncated to positive values</td></tr>
<tr><td>g</td><td>-6</td><td>Scaling factor for the inhibitory synapses</td></tr>
<tr><th colspan="3">B: Neuron Model</th></tr>
<tr><td><b>Name</b></td><td><b>Value</b></td><td><b>Description</b></td></tr>
<tr><td>τ</td><td>10 ms</td><td>Unit time constant</td></tr>
<tr><td>b</td><td>1</td><td>Bias term</td></tr>
<tr><td>σ_{X}</td><td>1.5</td><td>Scaling term for unit noise</td></tr>
</table>

Supplementary Table B.5: Rate model parameters.

B.3 Constraints on feedforward connectivity

This section expands on the limitations arising from the definitions of topographic modularity and map sizes used in this study. By imposing a fixed connection density on the feed-forward connection matrices, the projection probabilities between neurons tuned to the same (p_c) and different (p_0) stimuli are uniquely determined by the modularity m and the parameter d_0 and δ, which control the size of stimulus-specific sub-populations (see Methods). For notational simplicity, here we consider the merged excitatory and inhibitory sub-populations tuned to a particular stimulus in a given sub-network SSN$_i$, with a total size $C_\mathrm{i} = C_\mathrm{i}^\mathrm{E} + C_\mathrm{i}^\mathrm{I}$.

Under the constraints applied in this work, the total density of a feed-forward adjacency matrix between SSN$_\mathrm{i}$ and SSN$_\mathrm{i+1}$ can be computed as:

$$\sigma_\mathrm{i} = \frac{p_\mathrm{c} U_\mathrm{c}^\mathrm{i} + p_0 U_0^\mathrm{i}}{N^2} \tag{B.1}$$

where U_0^i and U_c^i are the number of *realizable* connections between similarly and differently tuned sub-populations, respectively. Since $U_\mathrm{c}^\mathrm{i} = N^2 - U_0^\mathrm{i}$, we can simplify the notation and focus only on U_0^i. We distinguish between the cases of non-overlapping and overlapping stimulus-specific sub-populations:

$$U_0^\mathrm{i} = \begin{cases} N^2 - N_\mathrm{C} C_\mathrm{i} C_\mathrm{i+1} & \text{if } d_\mathrm{i} < \frac{1}{N_\mathrm{C}} \\ \frac{N_\mathrm{C}}{N_\mathrm{C}-1}(N - C_\mathrm{i})(N - C_\mathrm{i+1}) & \text{if } d_\mathrm{i} \geq \frac{1}{N_\mathrm{C}} \end{cases},$$

where each potential synapse is counted only once, regardless of whether the involved neurons belong to any or multiple overlapping sub-populations. This ensures consistency with the definitions of the probabilities p_c and p_0. Alternatively, we can express U_0^i as:

$$U_0^\mathrm{i} = \frac{N^2 N_\mathrm{stim}}{N_\mathrm{stim} - 1}(1 - i\delta - d_0)(1 - (i-1)\delta - d_0)$$

For the case with no overlap, we can derive an additional constraint on the minimum sub-populations size C_i for the required density σ_i to be satisfied, which we define in relation to the total number of sub-populations N_C:

$$d_\mathrm{i} \geq \sqrt{\frac{\sigma_\mathrm{i}}{N_\mathrm{C}}} \tag{B.2}$$

The equality holds in the case of $m = 1$ and all-to-all feed-forward connectivity between similarly tuned sub-populations, i.e., $p_\mathrm{c} = 1$.

B.4 Mean-field analysis of network dynamics

For an analytical investigation of the role of topographic modularity on the network dynamics, we used mean field theory (Fourcaud and Brunel, 2002; Helias et al., 2013; Schuecker et al., 2015). Under the assumptions that each neuron receives a large number of small amplitude inputs at every time step, the synaptic time constants τ_s are small compared to the membrane time constant τ_m, and that the network activity is sufficiently asynchronous and irregular, we can make use of theoretical results obtained from the diffusion approximation of the LIF neuron model to determine the stationary population dynamics. The equations in this section were partially solved using a modified version of the LIF Meanfield Tools library (Layer et al., 2020).

B.4.1 Stationary firing rates and fixed points

In the circumstances described above, the total synaptic input to each neuron can be replaced by a Gaussian white noise process (independent across neurons) with mean $\mu(t)$ and variance $\sigma^2(t)$. In the stationary state, these quantities, along with the firing rates of each afferent, can be well approximated by their constant time average. The stationary firing rate of the LIF neuron in response to such input is:

$$\nu = \left(\tau_\mathrm{ref} + \sqrt{\pi}\tau_\mathrm{eff}\int_{y_\mathrm{r}}^{y_\theta} \exp(u^2)\left[1 + \mathrm{erf}\,(u)\right] du\right)^{-1} \tag{B.3}$$

where erf is the error function and the integration limits are defined as $y_\mathrm{r} = (V_\mathrm{reset} - \mu)/\sigma + \frac{q}{2}\sqrt{\tau_\mathrm{s}/\tau_\mathrm{eff}}$ and $y_\theta = (\theta - \mu)/\sigma + \frac{q}{2}\sqrt{\tau_\mathrm{s}/\tau_\mathrm{eff}}$, with $q = \sqrt{2}|\zeta(1/2)|$ and Riemann zeta function ζ (see Fourcaud and Brunel (2002), eq. 4.33). As we will see below, the mean μ and variance σ^2 of the input also depend on the stationary firing rate ν, rendering Eq. B.3 an implicit equation that needs to be solved self-consistently using fixed-point iteration.

For simplicity, throughout the mean-field analyses we consider perfectly partitioned networks where each neuron belongs to exactly one topographic map, that is, to one of the N_C stimulus-specific, identically sized sub-populations *SP* (no overlap condition). We denote the firing rate of a neuron in the currently stimulated SP (receiving stimulus input in $\mathrm{SSN_0}$) in sub-network $\mathrm{SSN_i}$ by $\nu_\mathrm{i}^\mathrm{S}$, and by $\nu_\mathrm{i}^\mathrm{NS}$ that of neurons not associated with the stimulated pathway. Since the firing rates of excitatory and inhibitory neurons are equal (due to identical synaptic time constants and input statistics), we can write the constant mean synaptic input to neurons in the input sub-network as

$$\mu_0^{\mathrm{S}} = \left(\overbrace{K_{\mathrm{X}} J_{\mathrm{X}} \nu_{\mathrm{X}}}^{\text{noise}} + \overbrace{(\frac{1}{N_{\mathrm{C}}} K_{\mathrm{E}} J_{\mathrm{E}} + \frac{1}{N_{\mathrm{C}}} K_{\mathrm{I}} J_{\mathrm{I}}) \nu_0^{\mathrm{S}}}^{\text{rec. stimulated}} + \overbrace{(N_{\mathrm{C}} - 1)(\frac{1}{N_{\mathrm{C}}} K_{\mathrm{E}} J_{\mathrm{E}} + \frac{1}{N_{\mathrm{C}}} K_{\mathrm{I}} J_{\mathrm{I}}) \nu_0^{\mathrm{NS}}}^{\text{rec. non-stimulated}} \right) \tau_{\mathrm{m}}$$

$$+ \overbrace{J_{\mathrm{X}} \nu_{\mathrm{in}}}^{\text{stimulus}} \tau_{\mathrm{m}}$$

$$\mu_0^{\mathrm{NS}} = \left(\overbrace{K_{\mathrm{X}} J_{\mathrm{X}} \nu_{\mathrm{X}}}^{\text{noise}} + \overbrace{(\frac{1}{N_{\mathrm{C}}} K_{\mathrm{E}} J_{\mathrm{E}} + \frac{1}{N_{\mathrm{C}}} K_{\mathrm{I}} J_{\mathrm{I}}) \nu_0^{\mathrm{S}}}^{\text{rec. stimulated}} + \overbrace{(N_{\mathrm{C}} - 1)(\frac{1}{N_{\mathrm{C}}} K_{\mathrm{E}} J_{\mathrm{E}} + \frac{1}{N_{\mathrm{C}}} K_{\mathrm{I}} J_{\mathrm{I}}) \nu_0^{\mathrm{NS}}}^{\text{rec. non-stimulated}} \right) \tau_{\mathrm{m}}, \tag{B.4}$$

The variances $(\sigma_0^{\mathrm{S}})^2$ and $(\sigma_0^{\mathrm{NS}})^2$ can be obtained by squaring each weight J in the above equation. To derive these equations for the deeper sub-networks $\mathrm{SSN}_{\mathrm{i}>0}$, it is helpful to include auxiliary variables K_{S} and K_{NS}, representing the number of feed-forward inputs to a neuron in $\mathrm{SSN}_{\mathrm{i}}$ from its own SP in $\mathrm{SSN}_{\mathrm{i-1}}$, and from one different SP (there are $N_{\mathrm{C}} - 1$ such sub-populations), respectively. Both K_{S} and K_{NS} are uniquely defined by the modularity m and projection density d, and $K_{\mathrm{NS}} = (1 - m) K_{\mathrm{S}} = (1 - m)(1 - \alpha) K_{\mathrm{E}}$ holds as well. The mean synaptic inputs to the neurons in the deeper sub-networks can thus be written as:

$$
\begin{aligned}
\mu_{\mathrm{i}}^{\mathrm{S}} = &\Bigg(\overbrace{\alpha K_{\mathrm{X}} J_{\mathrm{X}} \nu_{\mathrm{X}}}^{\text{noise}} + \overbrace{(\frac{1}{N_{\mathrm{C}}} K_{\mathrm{E}} J_{\mathrm{E}} + \frac{1}{N_{\mathrm{C}}} K_{\mathrm{I}} J_{\mathrm{I}}) \nu_{\mathrm{i}}^{\mathrm{S}}}^{\text{rec. stimulated}} \\
&+ \overbrace{(N_{\mathrm{C}} - 1)(\frac{1}{N_{\mathrm{C}}} K_{\mathrm{E}} J_{\mathrm{E}} + \frac{1}{N_{\mathrm{C}}} K_{\mathrm{I}} J_{\mathrm{I}}) \nu_{\mathrm{i}}^{\mathrm{NS}}}^{\text{rec. non-stimulated}} \\
&+ \overbrace{K_{\mathrm{S}} J_{\mathrm{E}} \nu_{\mathrm{i-1}}^{\mathrm{S}}}^{\text{stimulated FF}} + \overbrace{(N_{\mathrm{C}} - 1) K_{\mathrm{NS}} J_{\mathrm{E}} \nu_{\mathrm{i-1}}^{\mathrm{NS}}}^{\text{non-stimulated FF}} \Bigg) \tau_{\mathrm{m}} \\
\mu_{\mathrm{i}}^{\mathrm{NS}} = &\Bigg(\overbrace{\alpha K_{\mathrm{X}} J_{\mathrm{X}} \nu_{\mathrm{X}}}^{\text{noise}} + \overbrace{(\frac{1}{N_{\mathrm{C}}} K_{\mathrm{E}} J_{\mathrm{E}} + \frac{1}{N_{\mathrm{C}}} K_{\mathrm{I}} J_{\mathrm{I}}) \nu_{\mathrm{i}}^{\mathrm{S}}}^{\text{rec. stimulated}} \\
&+ \overbrace{(N_{\mathrm{C}} - 1)(\frac{1}{N_{\mathrm{C}}} K_{\mathrm{E}} J_{\mathrm{E}} + \frac{1}{N_{\mathrm{C}}} K_{\mathrm{I}} J_{\mathrm{I}}) \nu_{\mathrm{i}}^{\mathrm{NS}}}^{\text{rec. non-stimulated}} \\
&+ K_{\mathrm{NS}} J_{\mathrm{E}} \nu_{1}^{\mathrm{S}} + ((N_{\mathrm{C}} - 2) K_{\mathrm{NS}} + K_{\mathrm{S}}) J_{\mathrm{E}} \nu_{\mathrm{i-1}}^{\mathrm{NS}} \Big) \tau_{\mathrm{m}}
\end{aligned}
\tag{B.5}
$$

Again, one can obtain the variances by squaring each weight J. The stationary firing rates for the stimulated and non-stimulated sub-populations in all sub-networks are then found by first solving Eq. B.4 and Eq. B.3 for the first sub-network and then Eq. B.5 and Eq. B.3 successively for deeper sub-networks.

For very deep networks, one can ask the question, whether firing rates approach fixed points across sub-networks. If there are multiple fixed points, the initial condition, that is the externally stimulated activity of sub-populations in the first sub-network, decides in which of the fixed points the rates evolve, in a similar spirit as in recurrent networks after a start-up transient. For a fixed point, we have $\nu_{i-1} = \nu_i$. In effect, we can re-group terms in Eq. B.5 that have the same rates such that formally we obtain an effective new group of neurons from the excitatory and inhibitory SPs of the current sub-network and the corresponding excitatory SPs of the previous sub-network, as indicated by the square brackets in the following formulas:

$$\mu^{\mathrm{S}} = \alpha\beta\mathcal{J}\nu_x + \underbrace{\mathcal{J}\left[\frac{1}{N_{\mathrm{C}}}(1+\gamma g) + (1-\alpha)\frac{1}{(N_{\mathrm{C}}-1)(1-m)+1}\right]}_{\kappa_{\mathrm{S,S}}}\nu^{\mathrm{S}} \tag{B.6}$$

$$+ \underbrace{\mathcal{J}\left[\frac{N_{\mathrm{C}}-1}{N_{\mathrm{C}}}(1+\gamma g) + (1-\alpha)\frac{(N_{\mathrm{C}}-1)(1-m)}{(N_{\mathrm{C}}-1)(1-m)+1}\right]}_{\kappa_{\mathrm{S,NS}}}\nu^{\mathrm{NS}}$$

$$\mu^{\mathrm{NS}} = \alpha\beta\mathcal{J}\nu_x + \underbrace{\mathcal{J}\left[\frac{1}{N_{\mathrm{C}}}(1+\gamma g) + (1-\alpha)\frac{(1-m)}{(N_{\mathrm{C}}-1)(1-m)+1}\right]}_{\kappa_{\mathrm{NS,S}}}\nu^{\mathrm{S}} \tag{B.7}$$

$$+ \underbrace{\mathcal{J}\left[\frac{N_{\mathrm{C}}-1}{N_{\mathrm{C}}}(1+\gamma g) + (1-\alpha)\frac{1+(N_{\mathrm{C}}-2)(1-m)}{(N_{\mathrm{C}}-1)(1-m)+1}\right]}_{\kappa_{\mathrm{NS,NS}}}\nu^{\mathrm{NS}}$$

with $\beta = K_{\mathrm{X}}/K_{\mathrm{E}}$, $\gamma = K_{\mathrm{I}}/K_{\mathrm{E}}$ and $\mathcal{J} = \tau K_{\mathrm{E}} J$.

For the parameters g and γ chosen here, $\kappa_{\mathrm{S,NS}}$, $\kappa_{\mathrm{NS,S}}$ and $\kappa_{\mathrm{NS,NS}}$ in Eq. B.6 and Eq. B.7 are always negative for any modularity m due to the large recurrent inhibition. Therefore, for the non-stimulated group, $\kappa < 0$ in Eq. 5 (see main text), such that one always finds a single fixed point, which, as desired, is at a low rate. Interestingly, the excitatory feed-forward connections can switch the sign of $\kappa_{\mathrm{S,S}}$ from negative to positive for large values of m, thereby rendering the active group effectively excitatory, leading to a saddle-node bifurcation and the emergence of a stable high-activity fixed point (see Fig. 7b in the main text).

The structure of fixed points can also be understood by studying the potential landscape of the system: Eq. B.3 can be regarded as the fixed-point solution of the following evolution equations for the stimulated and non-stimulated sub-populations (Wong, 2006; Schuecker et al., 2017)

$$\tau_{\mathrm{S}}\frac{d\nu^{\mathrm{S}}}{dt} = -\nu^{\mathrm{S}} + \Phi_{\mathrm{S}}(\nu^{\mathrm{S}}, \nu^{\mathrm{NS}}), \tag{B.8}$$

$$\tau_{\mathrm{NS}}\frac{d\nu^{\mathrm{NS}}}{dt} = -\nu^{\mathrm{NS}} + \Phi_{\mathrm{NS}}(\nu^{\mathrm{S}}, \nu^{\mathrm{NS}}), \tag{B.9}$$

where Φ_{S} and Φ_{NS} are defined via the right-hand side of Eq. B.3 with μ^{S} and μ^{NS} inserted as defined in Eq. B.6 and Eq. B.7 (and likewise for σ^{S} and σ^{NS}). Due to the asymmetry in connections between stimulated and non-stimulated sub-populations, the right-hand side of Eq. B.8 and Eq. B.9 cannot be interpreted as a conservative force. Following the idea of effective response functions (Mascaro and Amit, 1999), a potential $U(\nu^{\mathrm{S}})$ for the stimulated sub-population alone can, however, be defined by inserting the solution $\nu^{\mathrm{NS}} = f(\nu^{\mathrm{S}})$ of Eq. B.9 into Eq. B.8

$$\tau_{\mathrm{S}} \frac{d\nu^{\mathrm{S}}}{dt} = -\nu^{\mathrm{S}} + \Phi_{\mathrm{S}}(\nu^{\mathrm{S}}, f(\nu^{\mathrm{S}})) \tag{B.10}$$

and interpreting the right-hand side as a conservative force $F = -\frac{dU}{d\nu^{\mathrm{S}}}$ (Litwin-Kumar and Doiron, 2012). The potential then follows from integration as

$$U(\nu^{\mathrm{S}}) - U(0) = \frac{1}{2}(\nu^{\mathrm{S}})^2 - \int_0^{\nu^{\mathrm{S}}} \Phi_{\mathrm{S}}(\nu, f(\nu)) d\nu\,, \tag{B.11}$$

where $U(0)$ is an inconsequential constant. We solved the latter integral numerically using the `scipy.integrate.trapz` function of `SciPy` (Virtanen et al., 2020). The minima and maxima of the resulting potential correspond to locally stable and unstable fixed points, respectively. Note that while this single-population potential is useful to study the structure of fixed points, the full dynamics of all populations and global stability cannot be straight-forwardly infered from this reduced picture (Mascaro and Amit, 1999; Rost et al., 2018), here for two reasons : 1. For spiking networks, Eq. B.8 and Eq. B.9 do not describe the real dynamics of the mean activity. Their right hand side only defines the stationary state solution. 2. The global stability of fixed points also depends on the time constants of all sub-populations' mean activities (here τ_{S} and τ_{NS}), but the temporal dynamics of the non-stimulated sub-populations is neglected here.

B.4.2 Mean-field analysis for two input streams

In the case of two simultaneously active stimuli (see Section "Input integration and multi-stability"), if the stimulated group 1 is in the high-activity state with rate ν^{S1}, the second stimulated group 2 will receive an additional non-vanishing input of the form

$$\left[\frac{1}{N_{\mathrm{C}}}(1+\gamma g) + (1-\alpha)\frac{(1-m)}{(N_{\mathrm{C}}-1)(1-m)+1}\right]\nu^{\mathrm{S1}} < 0, \tag{B.12}$$

which is negative for all values of m and can therefore lead to the silencing of group 2. If the stimuli are similarly strong, network fluctuations can dynamically switch the roles of the stimulated groups 1 and 2.

The dynamics and fixed-point structure in deep sub-networks can be studied using a two-dimensional potential landscape that is defined via the following evolution equations

$$\frac{d\nu^{\mathrm{S1}}}{dt} = -\nu^{\mathrm{S1}} + \Phi_{\mathrm{S1}}(\nu^{\mathrm{S1}}, \nu^{\mathrm{S2}}, f(\nu^{\mathrm{S1}}, \nu^{\mathrm{S2}}))\,, \tag{B.13}$$

$$\frac{d\nu^{\mathrm{S2}}}{dt} = -\nu^{\mathrm{S2}} + \Phi_{\mathrm{S2}}(\nu^{\mathrm{S1}}, \nu^{\mathrm{S2}}, f(\nu^{\mathrm{S1}}, \nu^{\mathrm{S2}}))\,, \tag{B.14}$$

where $f(\nu^{\mathrm{S1}}, \nu^{\mathrm{S2}}) = \nu^{\mathrm{NS}}$ is the fixed-point of the non-stimulated sub-populations for given rates $\nu^{\mathrm{S1}}, \nu^{\mathrm{S2}}$ of the two stimulated sub-populations, respectively. The functions

Φ_{S1} and Φ_{S2} are again defined via the right-hand side of Eq. B.3 with inserted μ^{S1}, μ^{S2} and μ^{NS} that are defined as follows (derivation analogous to the single-input case):

$$\mu^{\mathrm{S1}} = \alpha\mathcal{J}\nu_x + \underbrace{\mathcal{J}\left[\frac{1}{N_{\mathrm{C}}}(1+\gamma g) + (1-\alpha)\frac{1}{(N_{\mathrm{C}}-1)(1-m)+1}\right]}_{\kappa_{\mathrm{S1,S1}}}\nu^{\mathrm{S1}} \tag{B.15}$$

$$+\underbrace{\mathcal{J}\left[\frac{1}{N_{\mathrm{C}}}(1+\gamma g) + (1-\alpha)\frac{1-m}{(N_{\mathrm{C}}-1)(1-m)+1}\right]}_{\kappa_{\mathrm{S1,S2}}}\nu^{\mathrm{S2}}$$

$$+\underbrace{\mathcal{J}\left[\frac{N_{\mathrm{C}}-2}{N_{\mathrm{C}}}(1+\gamma g) + (1-\alpha)\frac{(N_{\mathrm{C}}-2)(1-m)}{(N_{\mathrm{C}}-1)(1-m)+1}\right]}_{\kappa_{\mathrm{S1,NS}}}\nu^{\mathrm{NS}}$$

$$\mu^{\mathrm{S2}} = \alpha\mathcal{J}\nu_x + \underbrace{\mathcal{J}\left[\frac{1}{N_{\mathrm{C}}}(1+\gamma g) + (1-\alpha)\frac{1-m}{(N_{\mathrm{C}}-1)(1-m)+1}\right]}_{\kappa_{\mathrm{S2,S1}}}\nu^{\mathrm{S1}} \tag{B.16}$$

$$+\underbrace{\mathcal{J}\left[\frac{1}{N_{\mathrm{C}}}(1+\gamma g) + (1-\alpha)\frac{1}{(N_{\mathrm{C}}-1)(1-m)+1}\right]}_{\kappa_{\mathrm{S2,S2}}}\nu^{\mathrm{S2}}$$

$$+\underbrace{\mathcal{J}\left[\frac{N_{\mathrm{C}}-2}{N_{\mathrm{C}}}(1+\gamma g) + (1-\alpha)\frac{(N_{\mathrm{C}}-2)(1-m)}{(N_{\mathrm{C}}-1)(1-m)+1}\right]}_{\kappa_{\mathrm{S1,NS}}}\nu^{\mathrm{NS}}$$

$$\mu^{\mathrm{NS}} = \alpha\mathcal{J}\nu_x + \underbrace{\mathcal{J}\left[\frac{1}{N_{\mathrm{C}}}(1+\gamma g) + (1-\alpha)\frac{(1-m)}{(N_{\mathrm{C}}-1)(1-m)+1}\right]}_{\kappa_{\mathrm{NS,S1}}}\nu^{\mathrm{S1}} \tag{B.17}$$

$$+\underbrace{\mathcal{J}\left[\frac{1}{N_{\mathrm{C}}}(1+\gamma g) + (1-\alpha)\frac{(1-m)}{(N_{\mathrm{C}}-1)(1-m)+1}\right]}_{\kappa_{\mathrm{NS,S2}}}\nu^{\mathrm{S2}} \tag{B.18}$$

$$+\underbrace{\mathcal{J}\left[\frac{N_{\mathrm{C}}-2}{N_{\mathrm{C}}}(1+\gamma g) + (1-\alpha)\frac{1+(N_{\mathrm{C}}-3)(1-m)}{(N_{\mathrm{C}}-1)(1-m)+1}\right]}_{\kappa_{\mathrm{NS,NS}}}\nu^{\mathrm{NS}}$$

Due to the symmetry between the two stimulated sub-populations, the right-hand side of Eq. B.13 and Eq. B.14 can be viewed as a conservative force $\boldsymbol{F}$ of the potential $U(\nu^{\mathrm{S1}}, \nu^{\mathrm{S2}}) = -\int_{\mathcal{C}} \boldsymbol{F}\, ds$, where we parameterized the line integral along the path $\nu : [0,1] \to \mathcal{C}, t \mapsto t \cdot (\nu^{\mathrm{S1}}, \nu^{\mathrm{S2}})$, which yields

$$\begin{aligned}U(\nu^{\text{S1}},\nu^{\text{S2}}) &= \frac{1}{2}(\nu^{\text{S1}})^2+\frac{1}{2}(\nu^{\text{S2}})^2\\ &-\int_0^{\nu^{\text{S1}}}\Phi_{\text{S1}}\left(\nu,\nu\frac{\nu^{\text{S2}}}{\nu^{\text{S1}}},f(\nu,\nu\frac{\nu^{\text{S2}}}{\nu^{\text{S1}}})\right)\\ &-\int_0^{\nu^{\text{S2}}}\Phi_{\text{S2}}\left(\nu\frac{\nu^{\text{S1}}}{\nu^{\text{S2}}},\nu,f(\nu\frac{\nu^{\text{S1}}}{\nu^{\text{S2}}},\nu)\right)\ .\end{aligned} \tag{B.19}$$

The numerical evaluation of this two-dimensional potential is shown in Supplementary Figure B.10, whereas sketches in Figure 6.11E show a one-dimensional section (grey lines in Supplementary Figure B.10) that goes anti-diagonal through the two minima corresponding to one population being in the high-activity state and the other one being in the low-activity state.

B.4.3 Critical modularity for piecewise linear activation function

To obtain a closed-form analytic solution for the critical modularity, we in the following consider a neuron model with piecewise linear activation function

$$\nu(\mu)=\nu_{\max}\frac{\mu-\mu_{\min}}{\mu_{\max}-\mu_{\min}} \tag{B.20}$$

for $\mu\in[\mu_{\min},\mu_{\max}]$, $\nu(\mu)=0$ for $\mu<\mu_{\min}$ and $\nu(\mu)=\nu_{\max}$ for $\mu>\mu_{\max}$ (Figure 6.8a). Successful denoising requires the non-stimulated sub-populations to be silent, $\nu^{NS}=0$, and the stimulated sub-populations to be active, $\nu^{S}>0$. We first study solutions where $0<\nu^{S}<\nu_{\max}$ and afterwards the case where $\nu^{S}=\nu_{\max}$. Inserting Eq. B.20 into Eq. 6.9 and Eq. 6.10, we obtain

$$\begin{aligned}\mu^{\text{S}} &= \alpha\mathcal{J}\nu_x+\kappa_{\text{S,S}}(m)\,\nu_{\max}\frac{\mu_S-\mu_{\min}}{\mu_{\max}-\mu_{\min}}\,,\\ \mu^{\text{NS}} &= \alpha\mathcal{J}\nu_x+\kappa_{\text{NS,S}}(m)\,\nu_{\max}\frac{\mu_S-\mu_{\min}}{\mu_{\max}-\mu_{\min}}\,.\end{aligned}$$

The first equation can be solved for μ^{S}

$$\frac{\mu^{\text{S}}}{\mu_{\min}}=1+\frac{\alpha\mathcal{J}\nu_x-\mu_{\min}}{\mu_{\min}-\kappa_{\text{S,S}}(m)\,\nu_{\max}\frac{\mu_{\min}}{\mu_{\max}-\mu_{\min}}}\,, \tag{B.21}$$

which holds for

$$\mu_{\min} \leq \mu^{\mathrm{S}} \leq \mu_{\max}\,, \tag{B.22}$$

$$\mu^{\mathrm{NS}} \leq \mu_{\min}\,. \tag{B.23}$$

Requirement (Eq. B.22) is equivalent to an inequality for m

$$0 \leq \frac{\alpha\mathcal{J}\nu_x - \mu_{\min}}{\mu_{\max} - \frac{\mathcal{J}}{N_{\mathrm{C}}}\left(1+\gamma g\right)\nu_{\max} - \frac{(1-\alpha)\mathcal{J}\nu_{\max}}{(N_{\mathrm{C}}-1)(1-m)+1} - \mu_{\min}} \leq 1$$

that, depending on the dynamic range of the neuron, the strength of the external background input and the recurrence, yields

$$m = \frac{N_{\mathrm{C}}}{N_{\mathrm{C}}-1} - \frac{1}{N_{\mathrm{C}}-1}\frac{(1-\alpha)\mathcal{J}\nu_{\max}}{\mu_{\max} - \alpha\mathcal{J}\nu_x - \frac{\mathcal{J}}{N_{\mathrm{C}}}\left(1+\gamma g\right)\nu_{\max}} \tag{B.24}$$

as an upper or lower bound for the modularity (Figure 6.8). Requirement (Eq. B.23) with the solution (Eq. B.21) for μ^{S} inserted yields a further lower bound

$$m \geq \frac{(\mu_{\max} - \mu_{\min})N_{\mathrm{C}}}{(1-\alpha)\mathcal{J}\nu_{\max} + (\mu_{\max} - \mu_{\min})(N_{\mathrm{C}}-1)} \tag{B.25}$$

for the modularity that is required for denoising. This criterion is independent of the external background input and the recurrence of the SSN.

Now we turn to the saturated scenario $\nu^{\mathrm{S}} = \nu_{\max}$ and $\nu^{\mathrm{NS}} = 0$ and obtain

$$\mu^{\mathrm{S}} = \alpha\mathcal{J}\nu_x + \kappa_{\mathrm{S,S}}(m)\,\nu_{\max}\,,$$

$$\mu^{\mathrm{NS}} = \alpha\mathcal{J}\nu_x + \kappa_{\mathrm{NS,S}}(m)\,\nu_{\max}\,,$$

with the criteria

$$\mu^{\mathrm{S}} \geq \mu_{\max}\,, \tag{B.26}$$

$$\mu^{\mathrm{NS}} \leq \mu_{\min}\,. \tag{B.27}$$

The first criterion (Eq. B.26) yields the same critical value (Eq. B.24) that for $\mu_{\max} - \alpha\mathcal{J}\nu_x - \frac{\mathcal{J}}{N_{\mathrm{C}}}\left(1+\gamma g\right)\nu_{\max} \geq 0$ is a lower bound and otherwise an upper bound. The second criterion (Eq. B.27) yields an additional lower bound for $\mathcal{J}(1-\alpha)\nu_{\max} - (N_{\mathrm{C}}-1)\left(\mu_{\min} - \alpha\mathcal{J}\nu_x - \frac{\mathcal{J}}{N_{\mathrm{C}}}\left(1+\gamma g\right)\nu_{\max}\right) \geq 0$ (Figure 6.8):

$$m \geq 1 - \frac{\left(\mu_{\min} - \alpha\mathcal{J}\nu_x - \frac{\mathcal{J}}{N_{\mathrm{C}}}\left(1+\gamma g\right)\nu_{\max}\right)}{\mathcal{J}(1-\alpha)\nu_{\max} - (N_{\mathrm{C}}-1)\left(\mu_{\min} - \alpha\mathcal{J}\nu_x - \frac{\mathcal{J}}{N_{\mathrm{C}}}\left(1+\gamma g\right)\nu_{\max}\right)}\,. \tag{B.28}$$

The above criteria yield necessary conditions for the existence of a fixed point with $\nu^S > 0$ and $\nu^{NS} = 0$. Next, we study the stability of such solutions. This works analogously to the stability in the spiking models discussed in Section 6.3.5 by studying the spectrum of the effective connectivity matrix. For the model Eq. B.20, the effective connectivity is given by

$$w_{ij} = \frac{\partial \nu_i}{\partial \nu_j} = \nu'(\mu_i)\frac{\partial \mu_i}{\partial \nu_j} = \nu'(\mu_i)\mathcal{J}_{ij} \tag{B.29}$$

with $\nu'(\mu) = \frac{d\nu}{d\mu}(\mu)$ and $\mathcal{J}_{ij} = \tau_x J_{ij}$. On the level of stimulated and non-stimulated sub-populations across layers, the effective connectivity becomes

$$W = \begin{pmatrix} \kappa_{\mathrm{S,S}}(m)\nu'(\mu^{\mathrm{S}}) & \kappa_{\mathrm{S,NS}}(m)\nu'(\mu^{\mathrm{NS}}) \\ \kappa_{\mathrm{NS,S}}(m)\nu'(\mu^{\mathrm{S}}) & \kappa_{\mathrm{NS,NS}}(m)\nu'(\mu^{\mathrm{NS}}) \end{pmatrix} \tag{B.30}$$

with eigenvalues

$$\begin{aligned}\lambda_\pm = {} & \frac{\kappa_{\mathrm{S,S}}(m)\nu'(\mu^{\mathrm{S}}) + \kappa_{\mathrm{NS,NS}}(m)\nu'(\mu^{\mathrm{NS}})}{2} \\ & \pm \sqrt{\left(\frac{\kappa_{\mathrm{S,S}}(m)\nu'(\mu^{\mathrm{S}}) + \kappa_{\mathrm{NS,NS}}(m)\nu'(\mu^{\mathrm{NS}})}{2}\right)^2 - X}\,,\end{aligned} \tag{B.31}$$

where X is

$$X = \left(\kappa_{\mathrm{S,S}}(m)\nu'(\mu^{\mathrm{S}})\kappa_{\mathrm{NS,NS}}(m)\nu'(\mu^{\mathrm{NS}}) - \kappa_{\mathrm{S,NS}}(m)\nu'(\mu^{\mathrm{NS}})\kappa_{\mathrm{NS,S}}(m)\nu'(\mu^{\mathrm{S}})\right). \tag{B.32}$$

The saturated fixed point $\nu^S = \nu_{\max}$ and $\nu^{NS} = 0$ has $\nu'(\mu^{\mathrm{S}}) = \nu'(\mu^{\mathrm{NS}}) = 0$, leading to $\lambda_\pm = 0$. This fixed point is always stable. The non-saturated fixed point also has $\nu'(\mu^{\mathrm{NS}}) = 0$. Consequently, Eq. B.31 simplifies to $\lambda_- = 0$ and

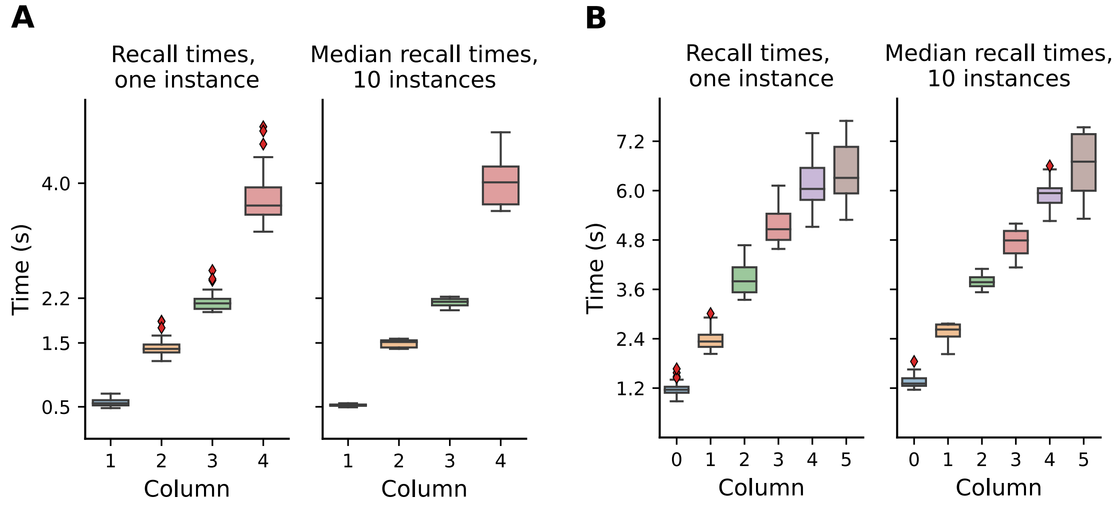
$$\lambda_+ = \frac{\nu_{\max}}{\mu_{\max} - \mu_{\min}}\kappa_{\mathrm{S,S}}(m)\,. \tag{B.33}$$

For $\lambda > 1$ fluctuations in the stimulated sub-population are being amplified. These fluctuations also drive fluctuations of the non-stimulated sub-population via the recurrent coupling. The fixed point thus becomes unstable and the necessary distinction between the stimulated and non-stimulated sub-populations vanishes. For inhibition-dominated recurrence, $\kappa_{\mathrm{S,S}}(m)$ is small enough to obtain stable fixed points at non-saturated rates (Figure 6.8c). In the case of no recurrence or excitation-dominated recurrence, $\kappa_{\mathrm{S,S}}(m)$ is much larger, typically driving λ_+ across the line of instability and preventing non-saturated fixed points to be stable. In such networks, only the saturated fixed point at $\nu^S = \nu_{\max}$ is stable and reachable (Figure 6.9A,B).

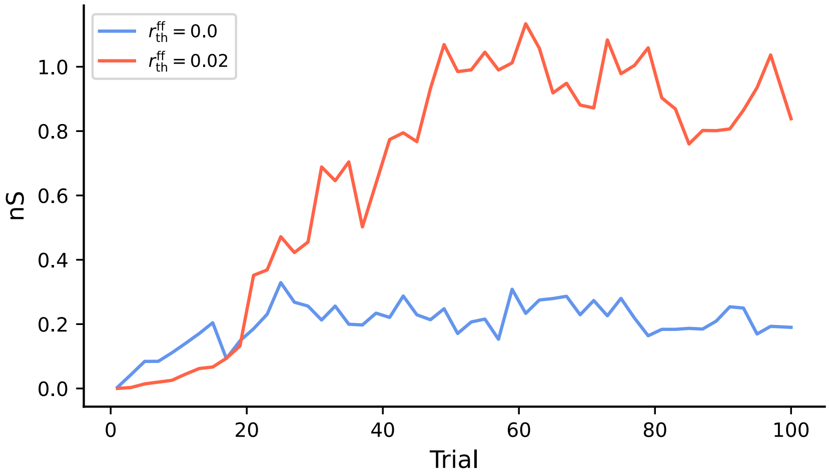
Appendix C

Supplementary materials for Chapter 7

C.1 Supplementary figures



Supplementary Figure C.1: **Fluctuations in learning and recall increase with sequence complexity and number of elements. (A)** The network was trained on a sequence of four elements: 500, 1000, 700, 1800 ms. Left: recall times for 30 trials after learning, for one network instance. Right: distribution of the median recall times over 10 network instances, with the median in each network calculated over 30 replay trials. **(B)** Same as (A), for a sequence of six elements with a duration of 1200 ms each.



Supplementary Figure C.2: **Hebbian threshold impacts learning convergence of cross-columnar connections.** In the baseline network, learning succeeds even in the absence of a Hebbian threshold $r_{\mathrm{th}}^{\mathrm{ff}}$. While a non-zero threshold leads to larger synaptic weights after convergence, it also increases the variability between trials.

C.2 Parameters of the baseline model

A: Model Summary	
Populations	Multiple columns, each one composed of an excitatory Timer (layer L_5) and Messenger ($L_{2/3}$) population, with one inhibitory population in each layer
Connectivity	Sparse, random recurrent connectivity
Neuron Model	Leaky integrate-and-fire, fixed voltage threshold, fixed absolute refractory time, no adaptation
Synapse Model	Conductance-based, exponentially decaying PSCs, static and plastic synaptic weights, fixed delays
Plasticity	Reward-based plasticity, short-term adaptation
Input	Stochastic background current and inhomogeneous Poisson spikes onto stimulus-specific T and I_T

B: Populations		
Name	**Elements**	**Size**
$T^i, I^i_\mathrm{T}, M^i, I^i_\mathrm{M}$ in column C_i	LIF neuron	100

C: Neuron Models	
Subthreshold Dynamics	if $(t > t^* + \tau_\mathrm{ref})$ $C_\mathrm{m} \frac{dV_i}{dt} = g_\mathrm{L}\left(V_\mathrm{rest} - V_i(t)\right) + I_i^\mathrm{E}(t) + I_i^\mathrm{I}(t) + \xi(t)$ else $V(t) = V_\mathrm{reset}$ $I_\mathrm{ij}^\mathrm{syn}(t) = g_\mathrm{ij}^\mathrm{syn}(E_\mathrm{\,syn} - V_i(t))$
Spiking	If $V(t-) < V_\mathrm{th}$ OR $V(t+) \geq V_\mathrm{th}$ 1. set $t^* = t$ 2. emit spike with time stamp t^*

D: Synapse Models	
Synaptic trace	$\frac{\mathrm{ds_i}}{dt} = -\frac{\mathrm{s_i}}{\tau_\mathrm{s}} + \rho\left(1 - \mathrm{s_i}\right)\sum_\mathrm{k}\delta\left(\mathrm{t} - \mathrm{t_k^i}\right)$
Name	Reward-based, with separate LTP and LTD eligibility traces
Trace update rule	$\tau^a \frac{dT^a_{ij}(t)}{dt} = -T^a_{ij}(t) + \eta^a_{(\mathrm{ff})} H_{ij}(t)\left(T^a_\mathrm{max} - T^a_{ij}(t)\right), a \in \{p, d\}$ $H_{ij}(t) = \begin{cases} r_i(t)r_j(t) & \text{if } r_i(t)r_j(t) > r_\mathrm{th}^{(\mathrm{ff})} \\ 0 & \text{otherwise} \end{cases}$
Online update rule	$\frac{dw_{ij}}{dt} = \eta_{(\mathrm{ff})} R(t)\left(T^p_{ij} - T^d_{ij}\right)$ $R(t) = \delta(t - t_\mathrm{reward} - d_\mathrm{reward})$

E: Input		
Type	**Target**	**Description**
poisson_generator	T^i and I^i_T in C_i	Total rate ν_in for a duration of 50 ms

Supplementary Table C.1: Tabular description of network model after Nordlie et al. (2009).

A: Populations

Name	Value	Source	Description
N	100	paper	Population size of every population, excitatory and inhibitory

B: Connectivity

Name	Value	Source	Description
d	1 ms	code	Synaptic transmission delay
φ	0.26	code*	Connection probability for all populations
w_{in}	100 nS	code	Synaptic strength of input connections
$w_{T\to M}$	0.2 nS	code*	Intracolumnar T to M excitatory synaptic strength
$w_{I_\mathrm{T}\to M}$	70 nS *	code*	Inhibitory synaptic strength from I_T in L_5 to M
$w_{I^i_\mathrm{T}\to T^j}$	100 nS *	code*	Inhibitory synaptic strength from I^i_T in C_i to T^j in C_j
$w_{I^i_\mathrm{M}\to M^j}$	100 nS *	code*	Inhibitory synaptic strength from I^i_M in C_i to M^j in C_j
$w_{T\to I_\mathrm{T}}$	0.2 nS *	code*	Intracolumnar T to I_T excitatory synaptic strength
$w_{M\to I_\mathrm{M}}$	1 nS *	code*	Intracolumnar M to I_M excitatory synaptic strength

B: Neuron Model

Name	Value	Source	Description
C_m	200 pF	paper	Membrane capacitance
τ_m	10 ms	paper	Membrane time constant
g_L	10 nS	paper	Leak conductance
E_L	-60 mV	paper	Resting membrane potential
V^E_th	-55 mV	paper	Spiking threshold for excitatory neurons
V^I_th	-50 mV	code*	Spiking threshold for inhibitory neurons
V_reset	-60 mV	code*	Reset potential
τ_ref	3 ms	code*	Absolute refractory period
σ_ξ	100	code*	Standard deviation of Gaussian white noise
ν_in	30 Hz	code*	Rate of Poisson stimulus input

C: Synapse Model

Name	Value	Source	Description
E_E	0 mV	paper	Excitatory reversal potential
E_I	-70 mV	paper	Inhibitory reversal potential
$\tau^\mathrm{exc,inp}_\mathrm{syn}$	10 ms	code*	Excitatory synaptic time constant of the input connections
$\tau^\mathrm{exc}_\mathrm{syn}$	80 ms	paper	Excitatory synaptic time constant
$\tau^\mathrm{inh}_\mathrm{syn}$	10 ms	paper	Inhibitory synaptic time constant
ρ	1/7	paper	Fractional change of synaptic activation

Supplementary Table C.2: Tabular description of the neuron, synapse and connectivity parameters. Parameters marked with * were additionally jittered with a randomly drawn value from $\mathcal{N}(0, 0.1)$. Parameters marked with * had different values in the code than reported in the paper.

A: Learning Parameters			
Name	**Value**	**Source**	**Description**
τ^p	2000 ms	paper	LTP eligibility trace time constant (intracolumnar connections)
τ^d	1000 ms	paper	LTD eligibility trace time constant (intracolumnar connections)
τ^p_{ff}	200 ms	paper	LTP eligibility trace time constant (cross-columnar connections)
τ^d_{ff}	800 ms	paper	LTD eligibility trace time constant (cross-columnar connections)
$T^p_{\max}$	0.0033	code*	Saturation level of LTP trace (intracolumnar connections)
$T^d_{\max}$	0.00345	code*	Saturation level of LTD trace (intracolumnar connections)
$T^{p,\mathrm{ff}}_{\max}$	0.0034	code*	Saturation level of LTP trace (cross-columnar connections)
$T^{d,\mathrm{ff}}_{\max}$	0.00345	code*	Saturation level of LTD trace (cross-columnar connections)
η^p	$45 \times 3500\ \mathrm{ms}^{-1}$	code*	Activation rate of LTP trace (intracolumnar connections)
η^d	$25 \times 3500\ \mathrm{ms}^{-1}$	code*	Activation rate of LTD trace (intracolumnar connections)
η^p_{ff}	$20 \times 3500\ \mathrm{ms}^{-1}$	code*	Activation rate of LTP trace (cross-columnar connections)
η^d_{ff}	$15 \times 3500\ \mathrm{ms}^{-1}$	code*	Activation rate of LTD trace (cross-columnar connections)
r_{th}	10 Hz	code*	Hebbian activation threshold (recurrent connections)
$r^{\mathrm{ff}}_{\mathrm{th}}$	20 Hz	code*	Hebbian activation threshold (feedforward connections)
η	$0.16\ \mathrm{ms}^{-1}$	code*	Learning rate $T \rightarrow T$ connections
η	$20\ \mathrm{ms}^{-1}$	code*	Learning rate $M \rightarrow T$ connections
T_{reward}	25 ms	paper	Duration of neuromodulator presentation upon change in stimulus
T_{tr}	25 ms	paper	Duration of refractory period for traces following neuromodulator presentation
d_{reward}	25 ms	paper	Reward delay

Supplementary Table C.3: Tabular description of learning parameters. Parameters marked with * had different values in the code than reported in the paper.

C.3 Parameters of the scaled model

A: Parameters for standard scaling		
Name	**Value**	**Description**
N'	400	Number of neurons in each population (scaled)
$w'_{T\to M}$	$w_{T\to M}/2$	Intracolumnar T to M excitatory synaptic strength
$w'_{I_\mathrm{T}\to M}$	$w_{I_\mathrm{T}\to M}/2\ ^\star$	Inhibitory synaptic strength from I_T in L_5 to M
$w'_{I^i_\mathrm{T}\to T^j}$	$w_{I^i_\mathrm{T}\to T^j}/2\ ^\star$	Inhibitory synaptic strength from I^i_T in C_i to T^j in C_j
$w'_{I^i_\mathrm{M}\to M^j}$	$w_{I^i_\mathrm{M}\to M^j}/2\ ^\star$	Inhibitory synaptic strength from I^i_M in C_i to M^j in C_j
$w'_{T\to I_\mathrm{T}}$	$w_{T\to I_\mathrm{T}}/2\ ^\star$	Intracolumnar T to I_T excitatory synaptic strength
$w'_{M\to I_\mathrm{M}}$	$w_{M\to I_\mathrm{M}}/2\ ^\star$	Intracolumnar M to I_M excitatory synaptic strength

B: Parameters for manually tuned scaling		
Name	**Value**	**Description**
N''	400	Number of neurons in each population (scaled)
$w''_{T\to M}$	$w'_{T\to M}\cdot 1.2$	Intracolumnar T to M excitatory synaptic strength
$w''_{I_\mathrm{T}\to M}$	$w'_{I_\mathrm{T}\to M}\cdot 2\ ^\star$	Inhibitory synaptic strength from I_T in L_5 to M
$w''_{I^i_\mathrm{T}\to T^j}$	$w'_{I^i_\mathrm{T}\to T^j}\cdot 0.02\ ^\star$	Inhibitory synaptic strength from I^i_T in C_i to T^j in C_j
$w''_{I^i_\mathrm{M}\to M^j}$	$w'_{I^i_\mathrm{M}\to M^j}\cdot 2\ ^\star$	Inhibitory synaptic strength from I^i_M in C_i to M^j in C_j
$w''_{T\to I_\mathrm{T}}$	$w'_{T\to I_\mathrm{T}}\cdot 0.02\ ^\star$	Intracolumnar T to I_T excitatory synaptic strength
$w''_{M\to I_\mathrm{M}}$	$w'_{M\to I_\mathrm{M}}\cdot 2\ ^\star$	Intracolumnar M to I_M excitatory synaptic strength
σ''_ξ	$\sigma_\xi/2$	Standard deviation of Gaussian white noise

Supplementary Table C.4: Tabular description of the modified parameters in the scaled network models. For the standard scaling, the values are obtained by applying a scaling factor of $1/\sqrt{N'/N}$ to the original values (see Section 7.2). Parameters marked with * were additionally jittered with a randomly drawn value from $\mathcal{N}(0, 0.1)$.

C.3.1 Parameters of the alternative model with local inhibition

A: Parameters for Network with Local Inhibition		
Name	**Value**	**Description**
N	100	Number of neurons in each population (as in baseline model)
$w_{T\to M}$	0.2 nS	Intracolumnar T to M excitatory synaptic strength
$w_{I_\mathrm{T}\to T}$	70 nS*	Inhibitory synaptic strength from I_T in L_5 to T
$w_{I_\mathrm{M}\to M}$	70 nS*	Inhibitory synaptic strength from I_M in $L_{2/3}$ to M
$w_{T\to I_\mathrm{M}}$	0.2 nS*	Excitatory synaptic strength from T to I_M
$w_{T^i\to I^j_\mathrm{T}}$	0.2 nS*	Excitatory synaptic strength from T in column C_i to I_T in $C_j, i \neq j$
$w_{M^i\to I^j_\mathrm{M}}$	0.5 nS*	Excitatory synaptic strength from M in column C_i to I_M in $C_j, i \neq j$

Supplementary Table C.5: Tabular description of the modified parameters in the model with rewired local inhibition. Parameters marked with * were additionally jittered with a randomly drawn value from $\mathcal{N}(0, 0.1)$.

List of Figures

List of Tables