



Curlew 1.0: Spatio-temporal implicit geological modelling with neural fields in python

Akshay V. Kamath^{1,5}, Samuel T. Thiele¹, Marie Moulard², Lachlan Grose³, Raimon Tolosana-Delgado¹, Michael J. Hillier⁴, Florian Wellmann^{5,6}, and Richard Gloaguen¹

¹Helmholtz-Zentrum Dresden-Rossendorf, Helmholtz Institute Freiberg, Chemnitz Str. 40, 09599 Freiberg, Germany

²École Nationale Supérieure de Géologie (ENSG), Université de Lorraine, 54000 Nancy, France

³School of Earth, Atmosphere and Environment, Monash University, Clayton, VIC, Australia

⁴Geological Survey of Canada, Natural Resources Canada, 601 Booth Street, Ottawa, ON K1A 0E8, Canada

⁵Institute of Computational Geoscience, Geothermics and Reservoir Geophysics (CG3), RWTH Aachen University, 52074 Aachen, Germany

⁶Fraunhofer Research Institution for Energy Infrastructures and Geothermal Systems (IEG), 44801 Bochum, Germany

Correspondence: Akshay V. Kamath (a.kamath@hzdr.de)

Received: 15 October 2025 – Discussion started: 13 November 2025

Revised: 13 March 2026 – Accepted: 10 April 2026 – Published: 27 April 2026

Abstract. We present *curlew*, an open-source python package for structural geological modelling using neural fields. This modelling framework incorporates various local constraints (value, gradient, tangent and (in)equalities) and tailored global loss functions to ensure data-consistent and geologically realistic predictions. Random Fourier Feature (RFF) encodings are used to improve model convergence and facilitate stochastic uncertainty quantification, while simultaneously improving the model's ability to learn naturally periodic features such as folds. These advances are integrated into a software framework that allows incremental construction of complex geological models through temporally-linked neural fields, each representing a specific deposition, intrusion or faulting event. Significantly, this framework allows semi-supervised learning to integrate diverse unlabelled datasets (e.g., geochemistry, petrophysics), reducing interpretation bias and potentially improving model robustness. We describe and demonstrate these various capabilities using synthetic examples and real data from a faulted stratigraphic digital outcrop model from Newcastle, Australia.

Commons Attribution 4.0 License and the OGL are interoperable and do not conflict with, reduce or limit each other. The co-author Michael J. Hillier is an employee of the Canadian Government and therefore claims Crown copyright for the respective contributions.

© Crown copyright 2026

1 Introduction

Three-dimensional (3D) structural geological models (SGMs) integrate disparate geological and geophysical data to make predictions about the subsurface distribution of rock types (Wellmann and Caumon, 2018). The results can be used to inform our understanding of large-scale processes, like plate tectonics, geomorphology and hydrogeology. For example, SGMs can provide crucial constraints on construction works (such as tunnelling), guide minerals explorationists and mine operators that supply society with metals, and are fundamental to successful groundwater modelling and management, geothermal energy extraction and (potentially) subsurface carbon storage (Pérez-Díaz et al., 2020; Wellmann and Caumon, 2018).

The uncertainty of SGMs (Bjerre et al., 2020; Bond, 2015; Lindsay et al., 2013; Pérez-Díaz et al., 2020; Thiele et al., 2016a; Wellmann and Caumon, 2018; Wellmann and Regenauer-Lieb, 2012) limits geoscientists' ability to disentangle Earth's history and leads to challenging risks for in-

Copyright statement. The works published in this journal are distributed under the Creative Commons Attribution 4.0 License. This license does not affect the Crown copyright work, which is reusable under the Open Government Licence (OGL). The Creative

frastructure, mining or energy projects. Inconsistencies regarding the thickness of a subsurface reservoir unit and its connectivity with surrounding rock types, for example, will significantly influence estimates of its storage capacity (Wellmann and Caumon, 2018), with crucial implications for groundwater management or geothermal energy.

Geological model uncertainty derives from two key challenges: data sparsity, and erroneous constraints. Subsurface data is expensive to collect, meaning that geological problems are nearly always under-constrained. Statistical methods have been developed to quantify this uncertainty (Wellmann and Caumon, 2018), and so mitigate risks or inform future data collection (Lindi et al., 2024), but these cannot account for the subjectivity of drill core or outcrop interpretations from which model inputs are derived (Pérez-Díaz et al., 2020). Additionally, the methods used to generate geological models often do not fully explore the space of geologically plausible solutions, which can result in an under-sampling of the geological uncertainty.

With this contribution we propose a framework for SGM construction using neural fields (see Sect. 2) that allow a better integration of geological constraints and unlabelled quantitative information (e.g., geochemistry or hyperspectral datasets) into interpolation workflows. We have implemented our approach in an open-source python toolbox for neural field modeling, *curlew*, and aim here to document the theory underlying our approach and demonstrate its applicability to a variety of geological structures and settings.

2 Context and theory

Current state-of-the-art SGM approaches, such as Aspen-SKUA (Aspen-SKUA: <https://www.aspentech.com/en/products/sse/aspen-skua>, last access: 10 October 2025) (formerly Gocad-SKUA), 3D-GeoModeller (3D Geomodeller: <https://www.intrepid-geophysics.com/products/geomodeller/>, last access: 10 October 2025), Leapfrog (Leapfrog: <https://www.seequent.com/products-solutions/leapfrog-geo/>, last access: 10 October 2025), GemPy (de la Varga et al., 2019) and LoopStructural (Grose et al., 2021b), represent geological bodies numerically using continuous implicit fields (Wellmann and Caumon, 2018). These fields can be interpolated from sparse data, and isosurfaces that represent geological contacts extracted for visualisation or further analysis. While many interpolation approaches can be used to construct these implicit fields, interpolation using spatial neural networks (hereafter referred to as neural fields) present a highly adaptable new approach (Hillier et al., 2023).

Neural fields (a.k.a implicit neural representations, coordinate-based/spatial neural networks) are neural networks that take spatial coordinates as inputs, and output predictions by learning the “geometry” of the spatial domain (Xie et al., 2022). The flexibility allowed by this parameteri-

sation has been used to address a variety of geoscience problems, including to interpolate potential fields (Smith et al., 2025), full tensor gradiometry data (Kamath et al., 2025), and for geophysical inversions (Xu and Heagy, 2025).

For SGM applications, neural fields have been applied to predict implicit field values, constrained by contact and orientation information (Hillier et al., 2023). Unlike many other interpolation approaches, the loss function used to train a neural field can be adapted to perform customised and multi-objective optimisation. It is this ability that theoretically allows neural field interpolators to better incorporate geological rules, physical constraints (Raissi et al., 2019) and/or diverse unlabelled quantitative data.

Recent studies have laid the groundwork for the use of neural fields in SGM construction (Gao and Wellmann, 2025; Hillier et al., 2023), though many challenges remain. These include difficulties getting neural fields to converge appropriately without pre-fitting and careful initialisation (e.g. Hillier et al., 2023). More complex geological structures (e.g., faults) also remain difficult, although step-function based faults (Calcagno et al., 2008) have been implemented in previous works.

2.1 Local constraints

Most SGMs use two types of constraint: value constraints, which represent a geological contact and specify the value of the underlying implicit field at specific locations, and gradient constraints that constrain the direction of the field’s gradient to match structural (e.g. bedding) measurements (de la Varga et al., 2019). Standard loss functions (e.g., Mean Squared Error) can be used to fit neural field predictions to value constraints by penalising differences between predicted and observed values at the constraint locations. Non-standard loss functions (e.g., involving the signed distance for geological contact information as used in Hillier et al., 2023) can also be implemented for such constraints.

Spatial gradients of the neural field can be obtained through automatic differentiation (Margossian, 2019) with respect to the inputs (coordinates). As shown by Hillier et al. (2023), this gradient can be compared to structural measurements to penalise violations of gradient constraints (i.e. bedding measurements). This comparison can be signed (called gradient constraints), using e.g., Mean-Squared Error (MSE) loss on the normalised gradient vector, or unsigned (called tangent constraints), using absolute cosine similarity as a loss function (Caumon et al., 2013). The former implies a known gradient (“younging”) direction, while the latter minimises the acute angle between the two vectors such that orientation is constrained to a known axis (strike and dip), but with unspecified younging direction.

Finally, as outlined in Sect. 3, we show that quantitative measurements of meaningful rock properties (e.g., features derived from geochemical, hyperspectral or petrophysical data) can be used to directly constrain neural field interpo-

lations. The core assumption here is that an implicit field that accurately represents subsurface geology should explain a maximal amount of the variance in the measured properties. Property constraints can thus be implemented using a reconstruction loss, where predicted implicit field values are passed through a learned forward model (cf. Sect. 3.2) to reconstruct property measurements and penalise errors (using e.g., MSE loss).

2.2 Relational constraints

As shown by Hillier et al. (2023), neural field interpolations can be formulated so that value constraints are replaced by constraints on the relation (equality and inequality) between pairs of observation points. Because the interpolated values in an SGM are typically arbitrary, such approaches find any (under-constrained) solution in which geological contacts have the same value i.e., equality constraints, and points in younger geological units are systematically greater than (or less than, depending on convention) older ones, i.e., inequality constraints. This representation has been suggested to enable significantly more flexibility than traditional value constraints (Hillier et al., 2023).

2.3 Global constraints

The previously mentioned local and relational constraints can be used to generate models that fit a sparse set of data. However, the under-constrained nature of the problem means a typically infinite number of solutions can be found, many of which are geologically implausible. Global constraints are thus typically used to encourage smoother solutions over others, and integrate geological rules that penalise geologically unrealistic geometries.

Smoothness regularisation is used in discrete implicit modelling to constrain the evolution of the interpolated field between geological observations. In a discrete interpolation framework the implicit function is represented on a piecewise support (e.g., tetrahedral mesh or cartesian grid) and geological constraints are applied by adding linear constraints to the element containing each observation. The smoothness constraints are applied globally to the interpolation function by adding additional constraints to locally minimize variations in the function gradient. This can be done by minimising the variations in the second derivatives of the implicit field using finite differences (Irakarama et al., 2018) or by minimising the variation in function gradient projected across the shared elements (Frank et al., 2007). Data supported methods such as co-kriging (Lajaunie et al., 1997) or radial basis function (RBF) interpolation (Hillier et al., 2014) typically use globally smooth basis functions, which naturally yield smooth interpolants. The smoothness can be controlled by the shape parameters of the basis function used. In neural field approaches, the loss functions can be adjusted to implement smoothness regularisation. This is based either

on the magnitude of the gradient vector (e.g. Hillier et al., 2023), or on the second derivative tensor (i.e., the hessian) of the implicit field. These adjustments require some global support (set of points) at which these losses are accumulated.

However, smooth implicit fields can still have geologically impossible isosurfaces. Most standard interpolators operate on the minimum curvature principle and are often implemented via biharmonic splines or discrete smooth interpolation. It implies that the resulting field approximates a solution to the biharmonic equation ($\nabla^4\phi = 0$) (Briggs, 1974; Sandwell, 1987; Mallet, 1989; Smith and Wessel, 1990). Biharmonic functions, while smooth, lack a maximum principle and therefore impose no limitations on the formation of local extrema. This leads the interpolator to generate closed isosurfaces (colloquially known as "bubbles"), that contradict the fundamental principles of original horizontality (Steno and Oldenburg, 1671; Thiele et al., 2016a, b). To encourage geologically realistic, bubble-free stratigraphy without allowing the network to collapse to a trivial solution, we have implemented a set of global geologically-informed loss functions (Sect. 3.2.2).

In many cases it is also desirable to enforce a constant (or approximately constant) gradient magnitude (Laurent, 2016). Changes in gradient magnitude imply variations in the thicknesses of the sedimentary units (which correspondingly imply changes in volume during deformation events), which might be unwanted. For example, Hillier et al. (2023) penalised gradient vectors of non-unit lengths to limit lateral variations in the thickness of interpolated sedimentary units. Such constant gradient constraints can be useful when interpolating true distance fields, such as the distance from a fault surface, but quickly become problematic when modelling units that change thickness (e.g., sedimentary series that change thickness, folds, etc.).

2.4 Deformation fields and time-aware modelling

Many geological objects (folds, faults, intrusions) create discontinuities. Such discontinuities contradict the principles of global and local maximisation of smoothness. One approach to include such discontinuous geological features is to modify the interpolation framework and incorporate the geometries from separate implicit fields in a step-wise fashion, as in e.g., LoopStructural (Grose et al., 2021b, a). LoopStructural models are built using a temporally successive approach where the most recent geological objects are built first, and these objects are used to constrain the geometry of older geological objects through kinematic reconstruction.

In the LoopStructural approach, faults are incorporated into the implicit model in two steps. First, constraints along the fault are used to interpolate an implicit field representing the fault surface. Then, this field is either used to split the interpolator into separate domains (i.e., domain boundary faults), as a step function in the polynomial trend (Calcagno et al., 2008; de la Varga and Wellmann, 2016) or to constrain

a kinematic operator for the fault (Godefroy et al., 2018; Grose et al., 2021a; Laurent et al., 2013). The kinematic approach provides the most control over the faulted surface, but requires a priori knowledge to constrain the displacement and slip direction.

Folds also pose challenges for smoothness regularisation. One approach is to modify the implicit interpolator by reducing the regularisation in the direction orthogonal to the axial surface and along the fold axis, such that high curvature in geologically plausible directions (i.e. fold hinges) is not penalised (Grose et al., 2017; Laurent, 2016).

3 Methods: geological neural fields with *curlew*

We have developed an open-source python package, *curlew*, that provides a flexible toolbox for building SGMs with neural fields. The following sections explore the Random Fourier Feature mapping (Sect. 3.1) used to improve convergence during training, and customisable loss functions (Sect. 3.2) that have been implemented to capture many of the different constraint types reviewed in the previous section. Additionally, we have established a framework (Sect. 3.3) for combining multiple neural fields representing different structures and introducing geological discontinuities.

A SGM in *curlew* is parameterised by a temporally directed graph of neural fields (see Sect. 3.3). Importantly, each neural field can be trained independently to fit specified constraints, and simultaneously to optimize parameters that influence multiple fields (e.g., fault slip). We also introduce a novel approach that, through a learnable forward model, allows the set of neural fields to be jointly fit to compositional data (e.g., drill core geochemistry or hyperspectral data) and/or lithological classifications (i.e., geological maps or logs) to directly update geological geometries. The inclusion of ancillary datasets (e.g., compositional, categorical, ordinal, etc.) into the modelling framework allows the model to decrease the uncertainties associated with interpretative bias and reduce the subjectivity associated with interpreted information.

3.1 Random Fourier feature encoding and neural field architecture

SGMs are defined in 2- or 3-D space. This low dimensionality presents a challenge for the neural fields, which can struggle to converge given the small number of input features. To help mitigate these limitations, Tancik et al. (2020) proposed Random Fourier Feature (RFF) mappings, a method that maps input coordinates into a higher-dimensional feature space by projecting the input coordinates onto randomly chosen direction vectors and passing the resulting lengths through sine and cosine functions. This work was built on the contribution by Rahimi and Recht (2007), who proposed these feature mappings as approximators for large-scale ker-

nel machines. This mapping (1) improves the representation of high-frequency signals in neural fields, (2) accelerates convergence during neural network training, avoiding the problem of spectral bias in neural networks (Rahaman et al., 2019), and (3) allows the network to learn a quasi-Fourier representation of the spatial variability. This approach is particularly effective to accurately capture geological complexities, especially in situations where geological structures display periodic geometries (i.e. folds).

In our implementation, input coordinates are projected onto a set of M 2- or 3-D vectors in which each component is randomly drawn from a Gaussian distribution with zero mean and unit variance. The resulting values are scaled by a frequency term and passed through a sine and a cosine function, to give $2M$ features ranging between -1 and 1 , analogous to amplitudes in a sparsely sampled Fourier domain. The frequencies of the mapping are explicitly scaled with length scale parameters to match the expected scale of variability in the model being constructed. These features effectively seed the network and guide feature learning at desired scales.

For a 3D input, we get a $2M$ dimensional feature vector \mathbf{v}_s for every length scale ℓ_s given by

$$\mathbf{v}_s = [\sin(2\pi \mathbf{W}_s \mathbf{r}), \cos(2\pi \mathbf{W}_s \mathbf{r})], \quad (1)$$

where $\mathbf{W}_s = \ell_s^{-1} \mathbf{W}^{M \times 3}$

where $\mathbf{r} = [x, y, z]^T$, $W_{ij} \sim \mathcal{N}(0, 1)$, and the sine/cosine transformations are applied element-wise followed by concatenation along the feature axis. Hence, for L length scales ($\ell_s \forall s = 1 \dots L$), we get a $2ML$ -dimensional feature vector, which is then fed into the subsequent multi-layer perceptron to get the implicit field value (Fig. 1a).

This RFF approach enhances the model's ability to learn high-frequency geological features. Furthermore, the encoding introduces inherent stochasticity into the interpolation process. Models trained with different seeds will learn different representations of the subsurface geometry for the same constraints, due to the random nature of the projection directions. Therefore, we can generate model ensembles by changing the seed of the random number generator to get different projections for each model. This becomes useful for stochastic uncertainty estimation, as multiple outputs can be generated without perturbing the input data (as is typical in established approaches; e.g., Wellmann and Regenauer-Lieb, 2012). In specific cases, it theoretically also allows known information on e.g., fold wavelength to be indirectly integrated as a model parameter by seeding the Fourier frequencies (and potentially directions) according to measured fold geometry.

The RFF encoding is followed by a standard multi-layer perceptron (MLP). The MLP block applies non-linear activation functions to all hidden layers. Because our framework computes derivatives via automatic differentiation (AD), the chosen activations must be C^2 differentiable (as the network requires a second backward pass for the optimisation). Func-

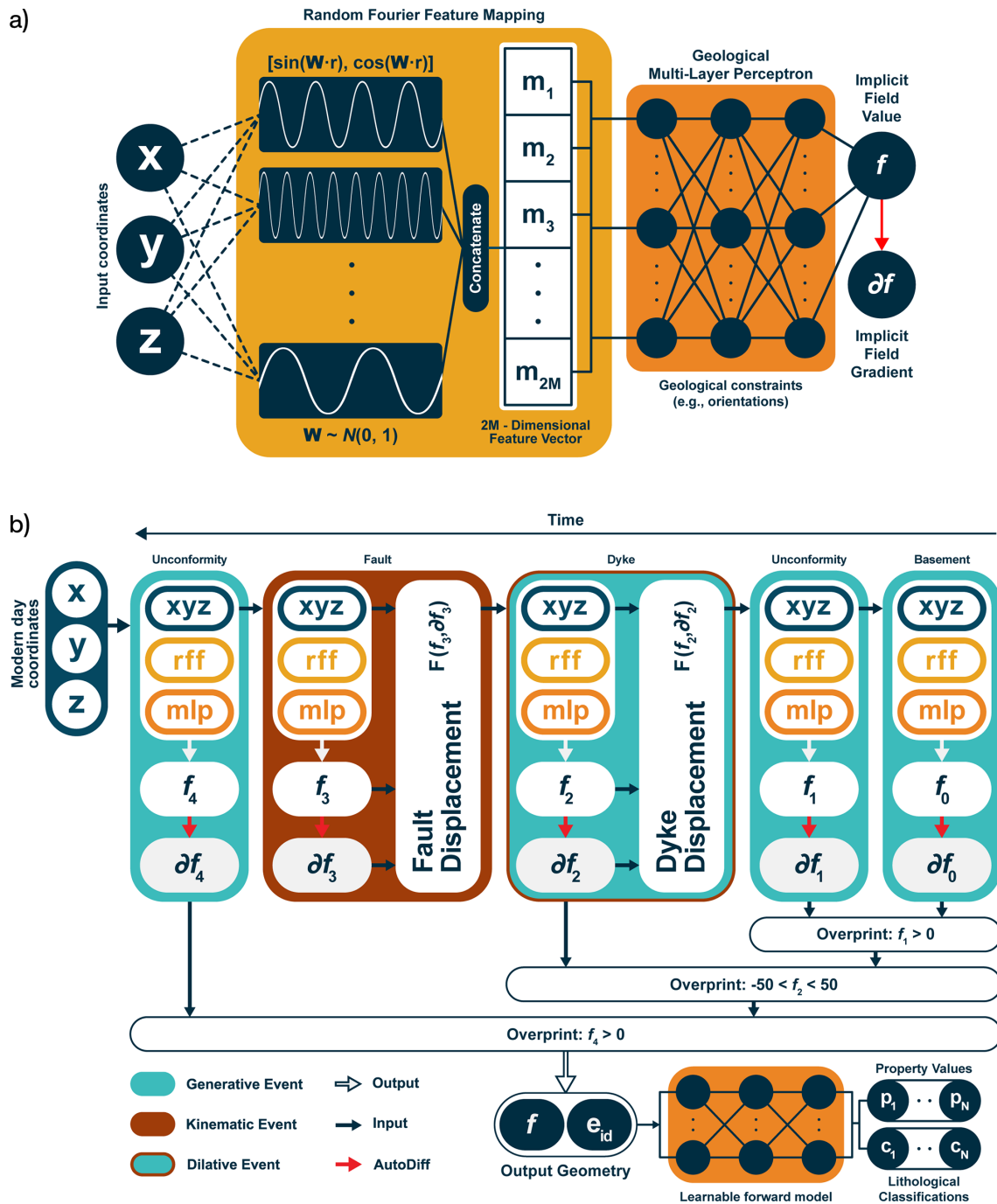


Figure 1. Overview of the *curlew* geological modeling approach, which combines (a) Random Fourier Feature (RFF) mappings, which transform input (spatial) coordinates into higher-dimensional Fourier features, with neural fields (multi-layer perceptrons) to predict implicit field values and gradients, guided by geological constraints. Different geological structures are each parameterised with their own neural field, such that a model is constructed from a temporally sequential combination of events (e.g., unconformities, faults, and dykes). Generative (stratigraphic deposition), kinematic (fault displacement), and dilative (dyke intrusion) events are chained through differentiable relationships, allowing integrated modeling and semi-supervised updates via a learnable forward model informed by quantitative property data and lithological classifications.

tions lacking this property, such as *ReLU*, produce abrupt edges in the resulting interpolation. Even among C^2 differentiable options, performances may vary. For instance, the hyperbolic tangent (*TanH*) is stable, but its second-order derivatives are small and prone to saturation, which can impede convergence. Empirical testing showed that *Swish* (*SiLU*, Ramachandran et al., 2017) and *Mish* (Misra, 2019) provided the best overall results. We have also experimented with options in which the Fourier features are directly connected to the predicted scalar field with a single layer of neurons (i.e. with no hidden layers and no activation functions). Early tests suggest that this minimalist neural field performs well in many situations, so we suggest that this architecture is used as a starting point (and extra complexity is added only if needed).

3.2 Loss functions

A neural field stores the implicit field within the weights and biases of the MLP block. These weights and biases are trained using a combination of local and global losses (the RFF projection remains fixed after initialisation). Local losses are accumulated at specific coordinates for which there is information on the implicit field value, its gradient, or its relation to other points in the model (see Sect. 2.2). Global losses are accumulated either over a grid of points that covers (or extends slightly beyond) the model domain, or (typically) on points that are randomly sampled from a grid during each training epoch. We have implemented loss functions in *curlew* such that each of these aspects can be introduced via corresponding hyperparameters. However, we recommend that any specific neural field should only use two to four of these loss functions (usually a local/relational loss coupled with a global loss), with the remaining terms disabled by setting their corresponding hyperparameter to zero. Every additional loss term adds an extra objective for the optimiser to minimise, quickly turning the interpolation into a complicated multi-objective optimisation problem (see Sect. 5.3).

3.2.1 Local losses – geological data

Each local constraint is defined by a coordinate and a value, represented in *curlew* as pairs of arrays to allow vectorised *pytorch* (Paszke et al., 2019) operations. The input coordinates can be presented in any Euclidean coordinate system (e.g., UTM) as the feature mapping has a normalisation effect. The values can represent known stratigraphic position (value constraints), younging direction (gradient constraints) or strike and dip (tangent constraints). Therefore, the local loss \mathcal{L}_1 is

$$\mathcal{L}_1 = \underbrace{\alpha \|f_{p_1} - f_{c_1}\|_2}_{\text{Value Loss}} + \underbrace{\beta \|\hat{\mathbf{g}}_{p_2} - \hat{\mathbf{g}}_{c_2}\|_2}_{\text{Gradient Loss}} + \underbrace{\gamma (1 - |\hat{\mathbf{g}}_{p_3} \cdot \hat{\mathbf{g}}_{c_3}|)}_{\text{Tangent Loss}} \quad (2)$$

where f refers to the implicit field value and $\hat{\mathbf{g}}$ refers to the normalised implicit field gradient. The subscripts p and c refer to the predicted and constraint (measured) terms, $\|\cdot\|_2$ is the L_2 norm, and α , β , and γ are the corresponding hyperparameters. As with any multi-objective optimisation problem, careful selection of these hyperparameters can be crucial for achieving good results. Also note that where the absolute direction of the gradient is unknown, the tangent loss accumulates errors proportional to the angular distance between the predicted and measured orientation axes by maximising cosine similarity.

Relational constraints (inequalities and equalities) are implemented using a computationally efficient approach that randomly samples P pairs of points from the left- and right-hand sides of each (in)equality. This avoids needing to construct a large pairwise difference matrix, which would be prohibitively expensive for large datasets, and is analogous to widely used mini-batch approaches used when training deep learning models. Thus, if we construct a difference term $\Delta = f_{p \in \text{LHS}} - f_{p \in \text{RHS}}$, the relational loss term \mathcal{L}_r can thus be expressed as:

$$\mathcal{L}_r = \begin{cases} \lambda \|\Delta\|_2 & \forall \text{LHS} = \text{RHS, Contact equality} \\ \lambda \min(\Delta, 0) & \forall \text{LHS} > \text{RHS, Above contact inequality} \\ \lambda \max(\Delta, 0) & \forall \text{LHS} < \text{RHS, Below contact inequality} \end{cases} \quad (3)$$

where LHS and RHS refer to the set of points sampled above and below a specific geological interface (inequalities), or along a single interface (equalities), and λ is the corresponding hyperparameter. The first case in Eq. (3) corresponds to the equality constraints, where N points are drawn randomly from the LHS and RHS sets each (i.e. N pairs), and the absolute difference between the predicted implicit field values at these points is minimized. The remaining cases correspond to the inequality constraints, where the clamp function is used to ignore differences above or below 0, as per the direction of the inequality.

3.2.2 Global losses – geological rules

Global losses control more general properties of the interpolated implicit field. This can be used to e.g., encourage constant gradient (resulting in geological units that minimize changes in thickness) or discourage geologically implausible closed isosurfaces (bubbles). Unlike local losses, they are not related to any specific data, and are evaluated on either a grid of points covering the model domain, or at a set of N points chosen randomly across the model domain during each epoch. We have currently implemented three global losses (\mathcal{L}_g) in *curlew*: flatness loss, thickness loss and monotonicity loss. These can be written as:

$$\mathcal{L}_g = \underbrace{\eta \|\hat{\mathbf{g}}_{p \in G} - \hat{\mathbf{g}}_{\text{trend}}\|_2}_{\text{Flatness Loss}} + \underbrace{\epsilon \left(\frac{\text{Var}(\|\mathbf{g}_{p \in G}\|_2)}{\mathbb{E}[\|\mathbf{g}_{p \in G}\|_2]^2} \right)}_{\text{Thickness Loss}} + \underbrace{\kappa \left| \nabla \cdot (\hat{\mathbf{g}}_{p \in G}) \right|}_{\text{Monotonicity Loss}} \quad (4)$$

where G is the set of grid points sampled in each epoch, $\hat{\mathbf{g}}_{\text{trend}}$ is the user-defined global gradient trend, $\nabla(\cdot)$ is the divergence operator, and η , ϵ , and κ are the corresponding hyperparameters. Flatness loss acts as a (weak) gradient constraint on each grid point to encourage unconstrained parts of the implicit field to have a specific orientation. The thickness loss penalises the relative deviation of gradient magnitudes from their mean, encouraging a uniform and non-zero gradient magnitude (i.e. constant thickness). It is equivalent to minimising the squared coefficient of variation (CV^2) of the gradient norm – that is, the variance normalised by the squared mean – making the penalty scale-invariant.

Finally, we have implemented a loss that discourages local maxima and minima (bubbles) in the interpolated implicit field. This ‘monotonicity’ loss is defined as the divergence of the normalised (unit length) gradient vector field. Rather than directly penalizing the Laplacian (which would aim for an ideal harmonic implicit field), we adopt a curvature-based divergence regulariser, by using normalized gradients. This approach is analogous to total variation regularisation (Rudin et al., 1992) and mean curvature flow formulations (Osher and Sethian, 1988), and in this context helps to preserve geological interfaces and prevents the formation of spurious internal extrema, as recently demonstrated for neural field applications (Gropp et al., 2020; Sitzmann et al., 2020). To save memory, the second derivative (hessian) matrices used to compute the monotonicity loss are estimated numerically. This numerical estimation also allows us to have an external control on the scale of curvatures we want to minimise, by controlling the size of the grid-step used to numerically compute the hessian.

3.3 Generative and kinematic events – a framework for combining neural fields

The aforementioned sections explain the constraints and controls over the interpolation of a single, smooth, and continuous implicit field. However, most SGMs contain multiple geological events, separated by unconformable, intrusive or faulted contacts. Each of these events typically need to be represented using their own interpolated implicit field, before temporally guided intersections are used to derive a combined model (Grose et al., 2021a; Wellmann and Caumon, 2018).

In *curlew*, this is achieved using overprint and/or offset relations (Fig. 1b). Therefore, most neural fields in a model built using *curlew* represent a geological event (Table 1) that

(1) creates new units (generative events, such as e.g., depositions or intrusions) or (2) displaces older units (kinematic events, such as faults). Sheet intrusions (e.g., dykes) are an example of events that do both – generate a new unit (the dyke fill) while displacing the older units (due to opening along the dyke).

Generative events in *curlew* are combined with older ones using an inequality: implicit field values representing deposition above an unconformity “overprint” older values where they exceed a threshold value, and values defining a dyke overprint older values within a specified range (proportional to the dyke aperture). Conversely, kinematic events define vector displacements (based on the gradient-direction and/or tangent plane of the interpolated implicit field) that retro-deform the input coordinates used to constrain (or evaluate) older fields. Importantly these vector displacements are differentiable, allowing the combined training of multiple neural fields to learn additional unknown parameters (e.g., fault slip magnitude).

Overprint relations are implemented using a differentiable sigmoid function, such that values from a younger implicit field replace those of older ones based on an inequality. For unconformities, this inequality specifies a basement value below which the interpolated values are ignored (to retain predictions of basement lithology from older neural fields). An Event ID (e_{id} , see Fig. 1b) is assigned to each overprinted location to keep track of the neural field that generated each implicit field value (and distinguish regions representing e.g., the interior of a dyke from those representing the surrounding stratigraphy). The geological operators outlining the implementation of kinematic events in *curlew* are outlined in the following sections, though terminology for these operations varies across the literature (Grose et al., 2021a, b; Calcagno et al., 2008; Hillier et al., 2023)

3.3.1 Faults

Faults are implemented in *curlew* using displacement fields. These fields modify existing geological structures by offsetting them along geologically plausible slip directions. The displacement direction is determined based on the interaction between the regional stress field and the local geometry of the fault surface.

Specifically, the slip direction vector $\hat{\mathbf{u}}$ is computed by projecting a vector representing the maximum shortening direction $\hat{\mathbf{s}}$ onto the plane perpendicular to the normalised gradient $\hat{\mathbf{g}}$ of the implicit field defining the fault surface:

$$\hat{\mathbf{u}} = \frac{\hat{\mathbf{s}} - (\hat{\mathbf{s}} \cdot \hat{\mathbf{g}}) \hat{\mathbf{g}}}{\|\hat{\mathbf{s}} - (\hat{\mathbf{s}} \cdot \hat{\mathbf{g}}) \hat{\mathbf{g}}\|_2} \quad (5)$$

Here, the term $(\hat{\mathbf{s}} \cdot \hat{\mathbf{g}}) \hat{\mathbf{g}}$ gives the component of the shortening vector normal to the fault plane. Subtracting this from $\hat{\mathbf{s}}$ yields the component tangential to the fault, ensuring that the resulting slip direction is (locally) fault-parallel and consistent with mechanical considerations (Fig. 2a). The resulting

Table 1. Overview of structural fields available in *curlew*, highlighting generative and kinematic components. Domain boundaries partition the model into separate domains, each governed by distinct implicit fields or chains of implicit fields (L , G).

Structural Fields in <i>curlew</i>			
Name	Use Cases	Generative Component	Kinematic Component
Stratigraphy	Layer cakes, folds, unconformities	$f \leq f_0$ or $f \geq f_0$, where f_0 is the boundary isovalue	None
Fault	Faults, shear zones	None	$\mathbf{u} = D \frac{\hat{\mathbf{s}} - (\hat{\mathbf{s}} \cdot \hat{\mathbf{g}}) \hat{\mathbf{g}}}{\ \hat{\mathbf{s}} - (\hat{\mathbf{s}} \cdot \hat{\mathbf{g}}) \hat{\mathbf{g}}\ _2}$
Sheet Intrusion	Dykes, sills	$f_0 \leq f \leq f_1$, where f_0, f_1 are contact isovalues	$\mathbf{u} = A \hat{\mathbf{g}}$
Domain Boundary	Onlap geometries, large offset faults	$M = \begin{cases} L & f \leq f_0 \\ G & f > f_0 \end{cases}$	None

f : Implicit field value; $\hat{\mathbf{g}}$: Implicit field gradient direction; $\hat{\mathbf{s}}$: Shortening direction; D : Fault offset; A : Sheet aperture; \mathbf{u} : Displacement.

vector is then normalised to give a unit slip direction vector $\hat{\mathbf{u}}$ that can be scaled by the desired displacement magnitude (D) to produce the final displacement vector \mathbf{u} .

For infinite faults, the magnitude is defined by a single scalar value (optionally learnable). For finite faults, the magnitude would need to vary as a function of distance from the fault center – following empirical displacement profiles (Grose et al., 2021b, a). Once the displacement vector is determined, its direction (sign) is assigned by applying two sigmoid functions that impose opposite slip directions on either side (hangingwall and footwall) of the fault. The sharpness of the sigmoid can be tuned to simulate either brittle fault behavior (sharp transition) or ductile shear zones (smooth transition), enabling the generation of simple drag folds adjacent to brittle faults.

For highly curved (with respect to displacement magnitude) faults, *curlew* applies an optional gradient correction to correct for curvature in the displacement vector between the initial and displaced positions (Fig. 2b). This correction ensures that displaced points remain (approximately) on the same isosurface of the fault implicit field, and is given by:

$$\mathbf{u}_{\text{corr}} = \mathbf{u} + (f_i - f_u) \hat{\mathbf{g}}_u \quad (6)$$

where $\hat{\mathbf{g}}_u$ is the gradient unit vector at the displaced location, f_i and f_u are the implicit field values at the original and displaced positions, respectively. The term $(f_i - f_u) \hat{\mathbf{g}}_u$ projects the scalar mismatch back along the gradient, correcting for small misalignments introduced by displacement in regions of high curvature or gradient variation.

3.3.2 Sheet intrusions

Sheet intrusions, such as dykes and sills, are implemented using the logic of both generative and kinematic implicit fields. These fields create the new geological units (solidi-

fied magma), and displace older ones to account for dilation during intrusion.

The generative component uses an overprinting relationship defined between an upper and lower scalar bound, assigning new implicit field values (and event IDs) between two isosurfaces that represent the intrusion contacts. An offset function is also defined to move pre-existing structures outward, in the direction of the gradient of the intrusion's implicit field. The gradient is flipped – based on the sign of the implicit field – so that vectors consistently point away from the intrusion's centerplane, which is represented by the zero isosurface. An aperture term scales the gradient's magnitude to reflect the intrusion's thickness.

3.4 Domain boundaries – multi-domain structural models in *curlew*

The final structural element implemented in *curlew* is the domain boundary, which enables the segmentation of geological models into distinct sub-domains, each governed by its own sequence of (older) geological events. Domain boundaries thus partition the model into separate neural field chains, based on inequalities defined on the domain boundary implicit field, and can be combined to construct complex geological structures and relations.

A typical use case arises when sedimentary units are deposited above an unconformity but pinch-out laterally (onlap geometries). Because such deposits do not follow the generative logic of a conformal event (i.e., they are not stratigraphically contiguous with the unconformity), they cannot be represented within a single event chain. Instead, the unconformity is defined as a domain boundary, which splits the model into two stratigraphically and structurally independent domains: one below the unconformity and one above it.

Another key application of domain boundaries is in the modelling of major fault zones. When a fault induces large

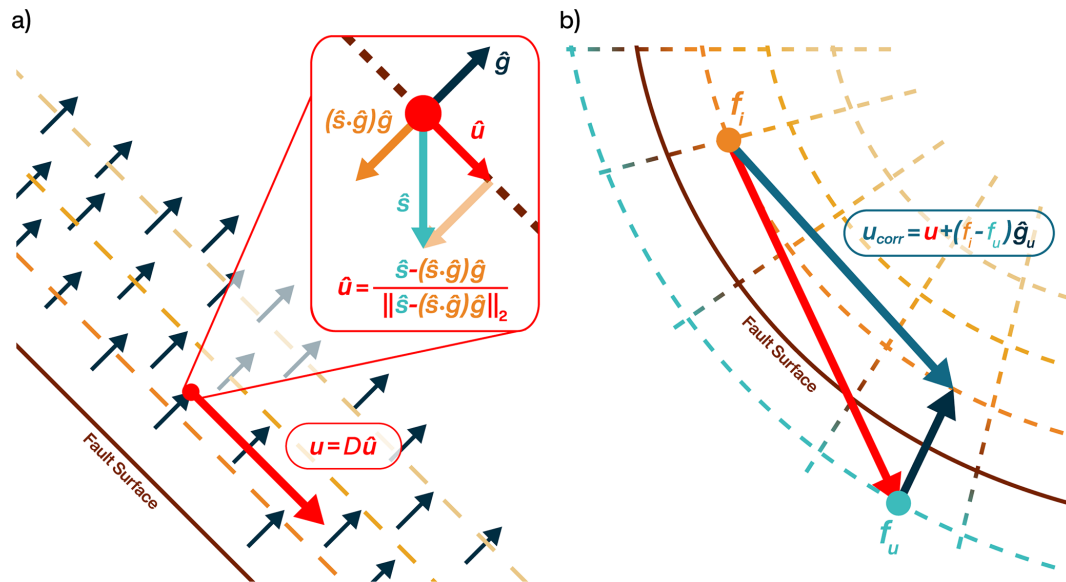


Figure 2. Fault displacement in *curlew*. The displacement direction at every point is calculated by projecting the (pre-defined) shortening direction (\hat{s}) onto the tangent plane of the implicit field that defines the fault geometry (a). The resulting displacement direction vector (\hat{u}) is then scaled by the (optionally learnable) offset magnitude (D) to derive a displacement vector (u). For curved fault systems (i.e., where the curvature of the fault plane is relevant with respect to the fault offset), *curlew* offers an optional correction (b), which uses the difference between the implicit field values at the displaced and initial locations, to project the point back onto the initial isosurface. This approximation results in some change in the displacement magnitude, but for most realistic fault geometries this error is negligible.

displacement, or juxtaposes basement rock against cover sequences, the stratigraphic relationship between the foot- and hanging-wall often becomes unresolvable. For such structurally complex settings, the fault is most simply defined as a domain boundary separating sub-models with different geological histories.

This formalism allows *curlew* to integrate geologically inconsistent or discontinuous regions within a unified framework. It removes the need for users to manually partition their models or construct separate simulations for each geological block, and facilitates instances where domain boundaries are affected by younger deformations (e.g. faults).

4 Results

We have applied *curlew* to a set of synthetic examples that explore the various capabilities of our neural field based interpolation approach, and illustrate different aspects of the underlying loss function. These are presented in the following sections, followed by an application to a real example constrained by data extracted from a digital outcrop model from Newcastle, Australia.

To help document and introduce potential users to specifics of the *curlew* modelling approach, each of these examples is provided as a jupyter notebook (see Code and Data availability section).

4.1 Folded stratigraphy

To illustrate the various loss terms described above, and to showcase the importance of the length scale parameter used during the Fourier feature projection, we have generated a simple test model containing a folded stratigraphic sequence (Fig. 3). This model involves two generations of folding events, a small wavelength small amplitude folding overprinted on large wavelength large amplitude folds. A Fourier series was used to generate similar folds with relatively sharp noses (Fig. 3a). This sequence was then sampled along hypothetical boreholes (Fig. 3b) to simulate gradient, value and tangent constraints. Note that while this synthetic example is 2D for ease of visualisation, all of the aforementioned methods (and the successive results) could be applied in 3D too.

4.1.1 Local, relational and global losses

We used different combinations of constraints and losses to show the effect of each part of our loss function (Eqs. 2–4) on the interpolated result (Fig. 4). To start, we fit a neural field model using each constraint type in isolation (Fig. 4a–d), without any additional global losses. The results show that tangent constraints (Fig. 4a) on their own are (unsurprisingly) extremely weak, since they only encode the axis of the gradient. This results in closed isosurfaces (i.e. “bubbles”), where the implicit field curls back on itself due to the undefined direction of the gradient.

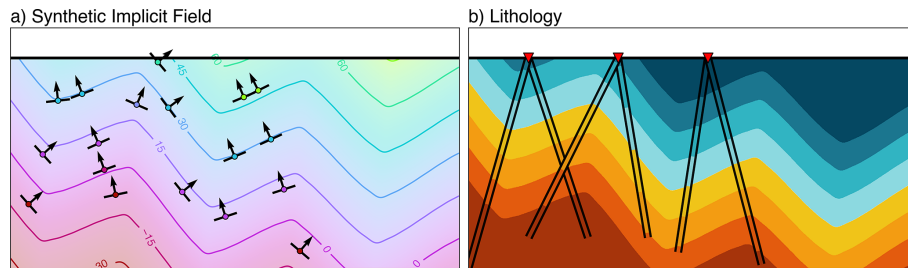


Figure 3. Synthetic (analytical) geological model simulating a folded stratigraphy. The implicit field (a) was sampled to derive gradient orientations (arrows) and implicit field values (circles) as constraints for a subsequent interpolation. Isosurfaces extracted from the implicit field at predefined seed positions also give a stratigraphic classification (b), which were used to define relational constraints. The red triangles represent the locations of the boreholes along which the constraints were sampled.

This is partially resolved when gradient constraints are used (Fig. 4b). The isosurfaces are now close to monotonic, as the directionality encoded in the gradient constraints forces the implicit field to “flow” from the bottom to the top of the model. However, away from the data, the interpolated field behaves randomly, and includes geologically unlikely variations in gradient magnitude (changes in thickness).

The use of equality and inequality (relational) constraints (Fig. 4c) results in a smoothly undulating model that correctly matches all of the contacts sampled by the boreholes, as well as maintaining correct stratigraphic relationships across the domain. Interestingly, however, the use of value constraints results in a very different interpolation (Fig. 4d) in which the neural field accurately reproduced the values at constraint locations but created artefacts (such as bubbles) elsewhere.

Introducing the monotonicity loss removes the bubble artefacts from results obtained with the aforementioned constraint types (Fig. 4e and g), but produces an interpolation with potentially unrealistic variations in unit thickness. The monotonicity loss seems to require further fine tuning in combination with value constraints (Fig. 4i).

Use of the thickness loss term resulted in an interpolation that honours unit thickness (class 1B parallel folds), giving geologically plausible models in combination with gradient and relational constraints. In combination with value constraints, however, the neural field found a solution dominated by angular bubbles that we consider to be unreasonable.

These additional global losses thus prove to be extremely useful in constraining the model in areas lacking local constraints, and can be adapted to enforce various geological rules. Interestingly, the global losses also appear to work best with relational constraints (inequalities), rather than value constraints.

4.1.2 Length scales

The synthetic folds model also allows us to explore one of the most important parameters for our implementation of a neural field interpolator: the length scale used during the Fourier

feature encoding (cf. Sect. 3.1). To show the impact of different length scales, we used relational and gradient constraints in combination with a global monotonicity loss while varying only the length scales used by the neural field (Fig. 5).

The resulting models highlight the importance of choosing appropriate length scales. Scales that are too small (Fig. 5a) result in severe over-fitting and largely meaningless predictions away from the data. Conversely, when the length scale is too large (Fig. 5d), the neural field under-fits and cannot make accurate predictions at the constraint locations, due to its limited capability to learn higher frequency variations.

For this example, the length scales of (150, 500) seems to be a good trade-off between fitting the data and avoiding random overfitting artefacts (Fig. 5b). A comparison of Fig. 5b and c suggest that the results are most heavily influenced by the smaller length scale (while the large length scale serves to reproduce the overall trend and so is less sensitive, albeit still important). As a rule of thumb, the longest length scale should be 4 times the longest model dimension, while the shortest should be determined by the data spacing.

4.2 Combining multiple neural fields

Only the simplest geological models can be represented by a single implicit field. Hence, we develop a second synthetic model that highlights *curlew*'s approach to fault displacement, dyke injection and the use of domain boundaries to model intrusive and onlap relations (Fig. 6a). The implicit fields associated with the depositional and kinematic events of the model are generated using predefined analytic equations, allowing us to construct a completely synthetic model for testing. Seed locations within these implicit fields were used to define isosurfaces that mimic stratigraphic contacts (Fig. 6b). Value and gradient constraints were sampled from each implicit field along vertical drill holes, to be used as synthetic data (Fig. 6d–j) when fitting (training) an interpolated neural field based model. Every individual field in the model influences every field that came before it with displacements, overprinting, or a combination of both actions (Fig. 7c).

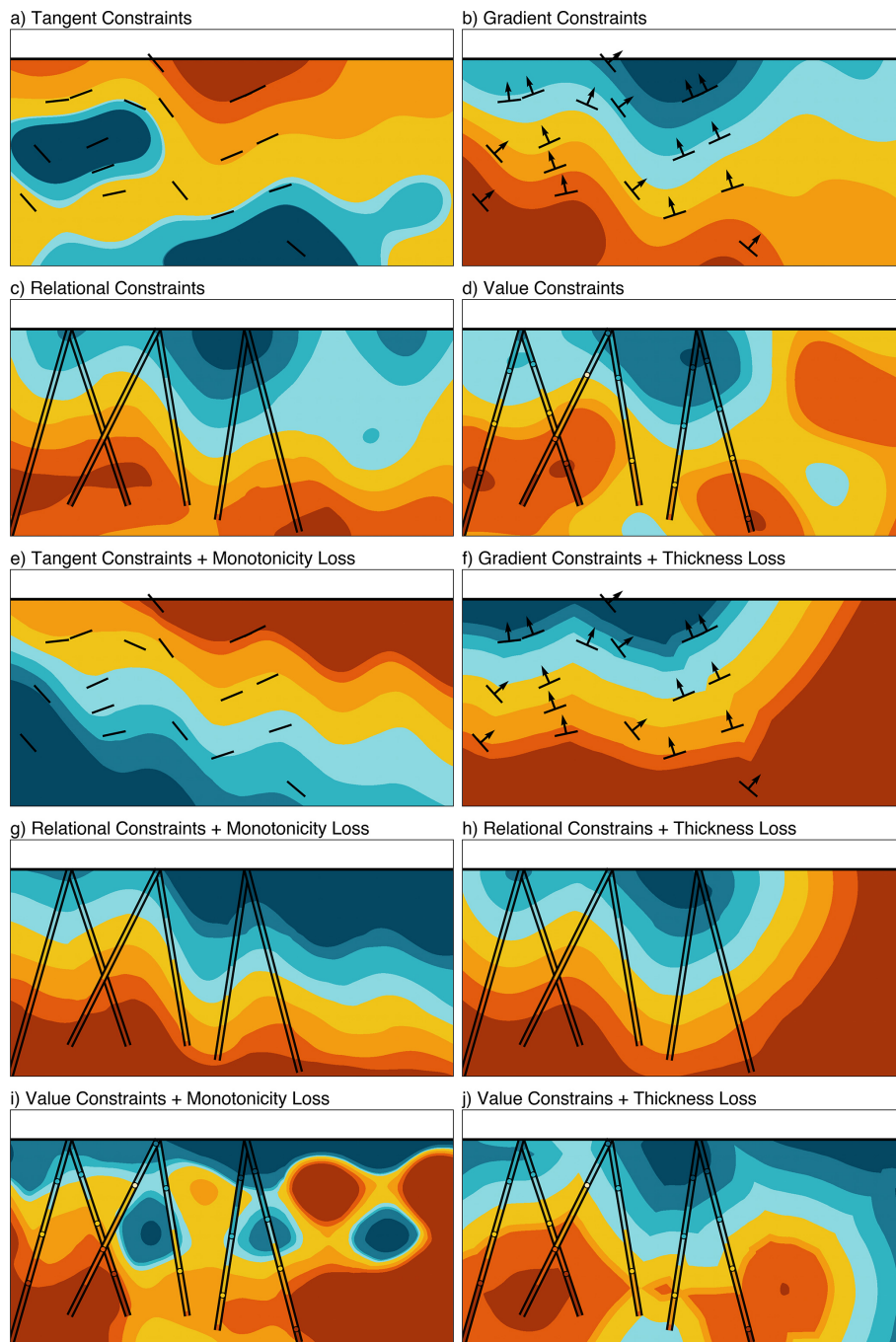


Figure 4. Interpolated stratigraphy for different combinations of constraints and losses, using the same network architecture (1 layer, 512 neurons), 2000 training epochs, and two (150, 1000) length scales. An early stopping criterion was applied to stop training after 100 epochs with less than 10^{-4} improvement in the loss. The individual models show results using tangent constraints (a), gradient constraints (b), relational constraints (c), and value constraints (d) in isolation. The addition of monotonicity loss to tangent constraints (e) and the thickness loss to the gradient constraints (f) improves the resulting interpolation. The flexibility afforded by relational (g, h) rather than value (i, j) constraints also seems to help the model converge to a more geologically realistic solution.

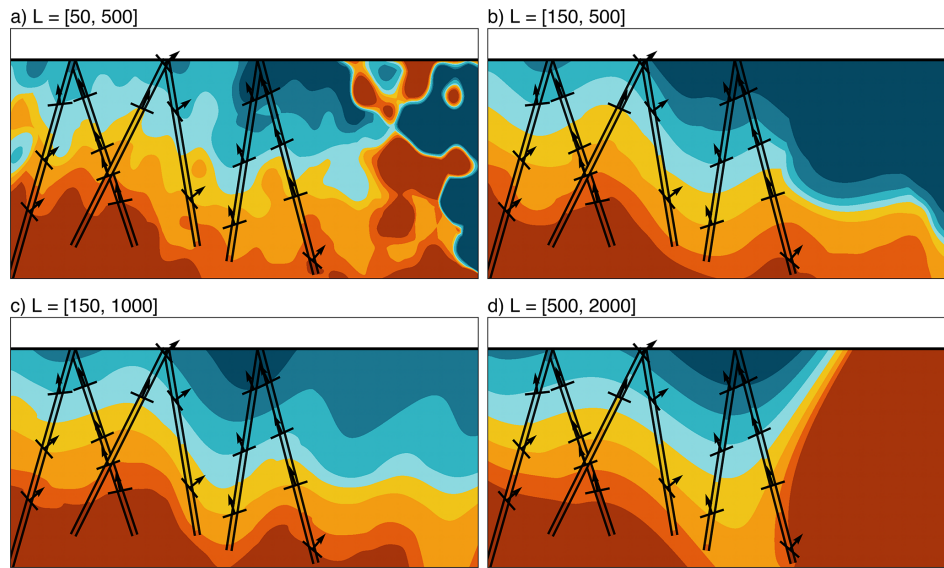


Figure 5. Interpolated stratigraphy for different pairs of length scales (but retaining the same settings as Fig. 4). Small length scales (**a**) result in over-fitting and random artefacts away from the data, whereas large length scales (**d**) result in under-fitting. Note also that the model is most influenced by the smaller length scale (as seen in **b** and **c**), whereas the larger length scale likely has more influence on large scale trends.

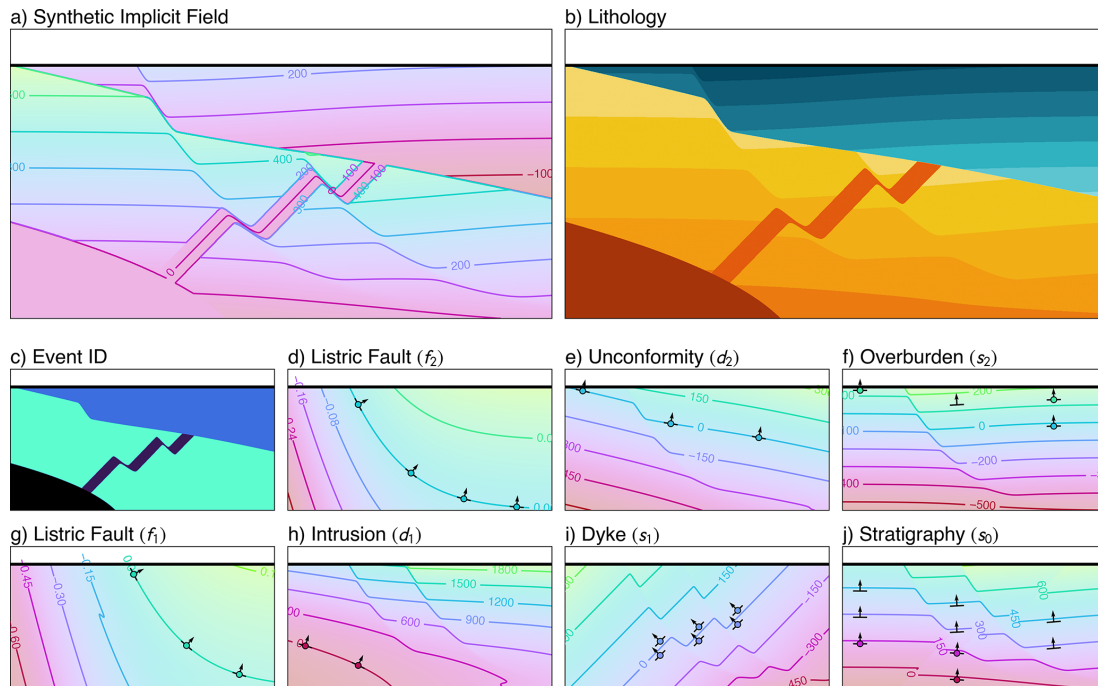


Figure 6. A structurally complex analytical synthetic model built in *curlew*. The resulting implicit field (**a**), corresponding lithology (**b**), and event ID (**c**) are also shown. This model represents a sequence of seven geological events, starting with a “layer cake” stratigraphy (**j**). Subsequently, this stratigraphy is intersected by a dyke (**i**), and intruded by a stock in the lower left corner of the model domain, represented through a domain boundary (**h**). The intrusion is assigned an arbitrary implicit field value (in this case -2) as its boundary is the only relevant entity. A listric fault then cuts through the units (**g**) followed by an unconformity that erodes through the deposited sequence, which is overlaid by a secondary deposition (**f**). As the deposited unit is not conformable to the unconformity surface (i.e., onlap), the unconformity is also modelled as a domain boundary (**e**). Finally, a second listric fault (**d**) cuts through the entire sequence.

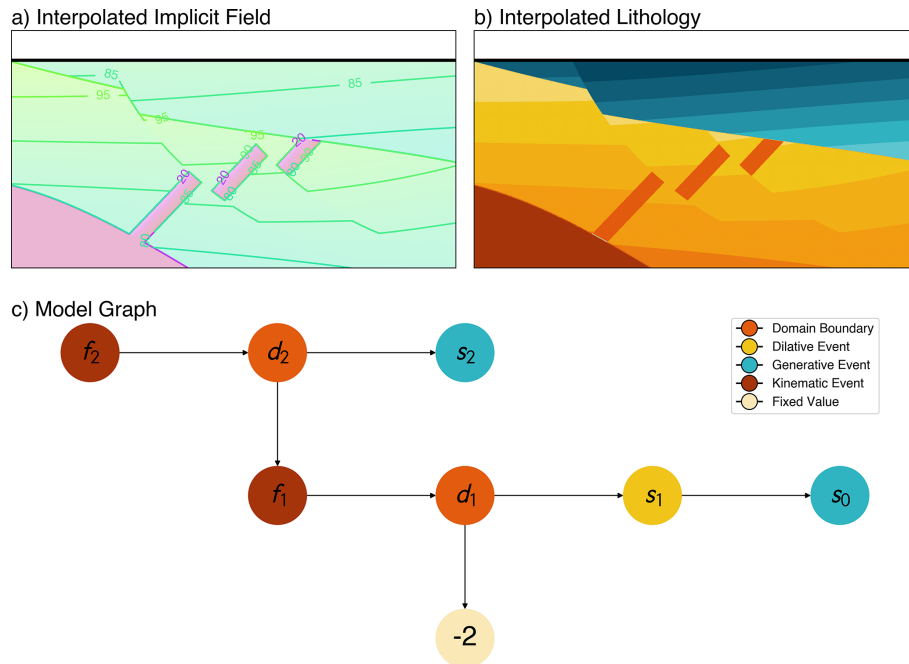


Figure 7. Interpolation results (a) and corresponding lithology classification (b) derived by fitting a network of neural fields (c) to the sparse constraints shown in Fig. 6. As mentioned earlier, the intrusive body is represented by a constant scalar field value (-2 in this case) as only the boundary of the intrusion (defined by the domain boundary field) is of interest.

A neural field was parameterised (with different hyperparameters for various losses) for each set of constraints. The gradient loss term for all the implicit fields was given a higher weight, to give more importance to the shape of the reconstructed model. For the depositional events (s_0 and s_2), the value loss was given a very low weight (as the values are arbitrarily defined and may impede convergence), and the monotonicity loss enabled used to encourage a bubble-free geometry. For the dyke, as the values determine the aperture of the dyke, the value loss was enabled, along with the thickness loss, to ensure it retains a constant thickness in interpolated regions. Finally, the domain boundaries and faults were fit by prioritising value-loss in combination with a thickness loss that acts to keep a constant implicit field gradient. For this example fault-displacement was assumed to be known, so the principal shortening direction (\hat{s}) and slip magnitude were defined a-priori (though the slip magnitude can also be treated as a learnable parameter, as shown in the following section).

After 1000 epochs of training, the interpolated *curlew* model closely matches the original synthetic one, with a few visible differences (Fig. 7). The implicit values for the two models are very different (0–600 vs. 80–100 for s_0), but this is coherent with the disabled value loss function for these two fields and does not necessarily influence the isosurface geometry (given we extract the isosurfaces for the lithologies based on seed positions).

4.3 Digital outcrop reconstruction

Finally, to demonstrate that *curlew* can be applied to real (3D) data, we present a model generated from a digital outcrop in Australia. The digital outcrop consists of a faulted stratigraphic sequence intruded by a dyke. The Compass plugin for CloudCompare (Thiele et al., 2017) was used to extract (1) traces along the contacts between the stratigraphic units, dyke margin and fault surface, and (2) orientation measurements (strike and dip) of each of these geological structures.

Since these traces included a large number of points (due to the high resolution of the point cloud), they were subsampled to acquire evenly-spaced points along the stratigraphic contacts, the fault surface, and the dyke (Fig. 8a). While these traces could be directly used as relational constraints, we used the subsampled points as value constraints, with the values being the approximate height of the layer from the base of the digital outcrop. These constraints were then used to constrain three consecutive neural fields: (1) a stratigraphic field, (2) a fault field, and (3) dyke field.

After fitting each neural field separately to generate an initial geometry, we froze the geometries of the fault and dyke fields by fixing the weights of their corresponding neural fields. The fault slip and stratigraphic interpolation were then fit simultaneously, to find an optimal displacement for the fault (i.e. by finding the fault offset that results, after recon-

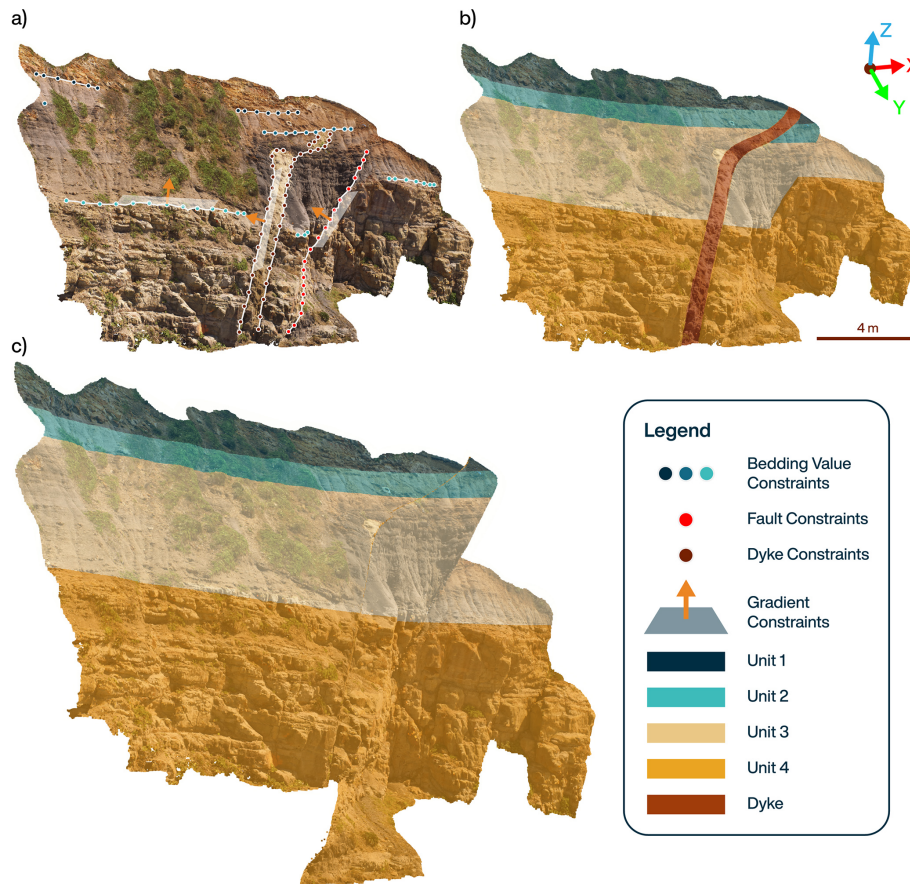


Figure 8. Interface and gradient constraints extracted from a digital outcrop model (a) from Newcastle, Australia. These were used to construct a 3D *curlew* model describing the main geological structures and stratigraphic units and derive a lithological classification (b), plotted here as an overlay on the original RGB colours for clarity. The fitted displacement fields from the *curlew* model were also used to offset these points and remove displacement from the dyke and fault, resulting in an undeformed point cloud closer to the original depositional geometry (c).

struction, in the smoothest interpolation of the older stratigraphy field).

The resulting model (Fig. 8b) and solutions to the displacement fields induced by the fault and the dyke were then used to “undeform” the outcrop and retrieve pre-deformation locations for all of the points in the digital outcrop model (Fig. 8c). Note that while the dyke in the original outcrop is finite, the modelled dyke is infinite. Finite dykes (and faults) are planned to be implemented in future versions of *curlew*.

5 Discussion

We have presented an open-source python package, *curlew*, which leverages neural fields to generate implicit geological representations of structural geological models. The neural field approach employed by *curlew* is highly versatile and adaptable, capable of handling a wide range of constraint types, data modalities, and loss functions. Its representation of structural geological models in a fully differentiable ar-

chitecture, and the ability of the underlying neural fields to arbitrarily adapt the geometry they represent, opens the door to a variety of new approaches to common earth modelling, property field interpolation and geophysical inversion.

Similarly to radial basis functions (RBF) or co-kriging approaches, models built with *curlew* are entirely meshless. Once trained, they can be evaluated at arbitrary locations, facilitating advanced methods for contour extraction or adaptive gridding strategies. Furthermore, given its *pytorch* core, training and inference operations are GPU-parallelised and scalable on large computing infrastructure. This contrasts against established methods such as co-kriging and RBF, which scale poorly with increasing data volume. Specifically, the requirement to invert a dense $N \times N$ covariance or interpolation matrix incurs a cubic computational cost ($\mathcal{O}(N^3)$), making these methods inefficient without numerical simplifications like compactly supported kernels, fast multipole methods, or domain decomposition (Rasmussen and Williams, 2005; Cavoretto et al., 2016; Beatson et al., 2001; Wendland, 1995).

The differentiable nature of the generated models also means model outputs can include the gradient of the underlying implicit fields (i.e. structural orientation measurements), and if necessary, higher order derivatives (e.g., curvature) through automatic differentiation (although the second order derivatives are computed numerically in *curlew* for computational scalability). As shown in Sect. 3.2.2, this has been leveraged to suppress (albeit not eliminate) common interpolation artefacts such as “bubbles”, and further applications are envisaged when interpolating property fields (e.g., permeability, ore grade) given a structural geological model.

Curiously, our experiments also suggest that, in the context of geological modelling with neural fields, the inclusion of value constraints (instead of somewhat equivalent inequality constraints) can make it difficult for neural fields to converge to a geologically realistic geometry. This behaviour might be explained by the under-determined nature of gradient-only constraints – the network’s loss is invariant to adding a constant offset to the field. Thus, the optimisation has a flat direction, yielding a continuum (valley) of equally valid solutions rather than a single narrow optimum. By contrast, specifying exact field values removes this degeneracy, collapsing the solution space to isolated optima that are much harder to find. Prior work in neural network training has noted that having such flat minima or degenerate solution manifolds can make optimisation easier and models generalise better (Li et al., 2017). Through their work on Sobolev Training, Czarnecki et al. (2017), further demonstrate that leveraging derivative information (gradients) in training leads to improved learning efficiency, implying that matching the shape of the target function is often more tractable than matching exact values.

5.1 Random Fourier Features

The inclusion of Random Fourier Feature (RFF) mapping allows *curlew* to train neural fields without using the pre-training protocol outlined by Hillier et al. (2023) for network weight initialisation. Although there are several alternative approaches that could behave similarly (e.g., SiREN; Sitzmann et al., 2020), the RFF mapping has the advantage that length-scales used by the interpolator can be carefully controlled, and potentially seeded for specific geological settings (e.g., in folded rocks with known amplitude and axial plane). However, this sensitivity can also be a disadvantage, as the manual selection of appropriate length scales can be quite challenging.

Interestingly, the randomness inherent to RFF mapping means multiple model realisations can be obtained by training models on the same data but with different RFF seeds (Fig. 9). While model ensembles can also be derived from conventional neural fields, we suspect that the significant differences in (learned) geometric representation that result when randomly sampling different RFFs enable greater exploration of the potential solution space. The precise meaning of these variations remains a topic of future research, but

it could facilitate methods for quantifying the epistemic uncertainty of the interpolated result.

For example, we have generated 25 random realisations of the folded stratigraphy model (see Sect. 4.1) and computed the information entropy (Shannon, 1948) of the resulting lithology predictions (Fig. 9f). This clearly highlights increased uncertainty near fold hinges and away from the constraints, aiding interpretation and potentially facilitating targeted data collection (Tacher et al., 2006; Wellmann et al., 2010, 2014; Wellmann and Caumon, 2018).

A full-variational Bayesian Neural Network (BNN; Goan and Fookes, 2020) provides a first order uncertainty quantification, as the weights and biases of the neural network are not fixed values and can be approximated as probability distributions. Thus, this more comprehensive representation of variability may help to capture epistemic uncertainty arising from the data. However, BNNs are computationally expensive, even more than the ensembles for highly optimised networks, and require more complex training procedures compared to standard neural networks with RFF encoding. Some studies propose Monte-Carlo dropout (Hasan et al., 2022) or Hamiltonian Monte-Carlo simulations within implicit neural representations (Gao et al., 2026) as a simpler alternative to BNNs. These approaches are all theoretically compatible with our *curlew* architecture, and could be explored in future works.

5.2 Learnable forward models

As hinted at previously, the differentiability of the combined network of neural fields can also be leveraged to update the model geometry according to some additional loss that depends on the model as a whole (rather than the individual neural fields). This can potentially be applied to integrate unlabelled auxiliary data through an additional neural network (Fig. 1b), which learns the relationship between the outputs of the structural model (implicit field value and event ID; cf. Fig. 1b) and arbitrary measurements (e.g., geochemistry or petrophysics). The primary role of this neural network is not to accurately predict the measured properties (as it cannot account for lateral variability), but rather to encourage the model geometry to converge on a solution that explains as much property variance as possible (i.e. a common earth model). By jointly training this learnable forward model and the geological geometry fields, geometries that are most predictive of the measured properties will be favoured.

We illustrate this approach using a simple model containing a folded sequence, in which three arbitrary property measurements have been made along the boreholes. These are visualised using a ternary colour mapping that gives a false-colour representation of the measurements (Fig. 10a). An initial model, fit to sparse constraints from these drill holes, does a reasonable job at predicting the field geometry (Fig. 10b) given the available information. But, when this is coupled with the learnable forward model, the added unla-

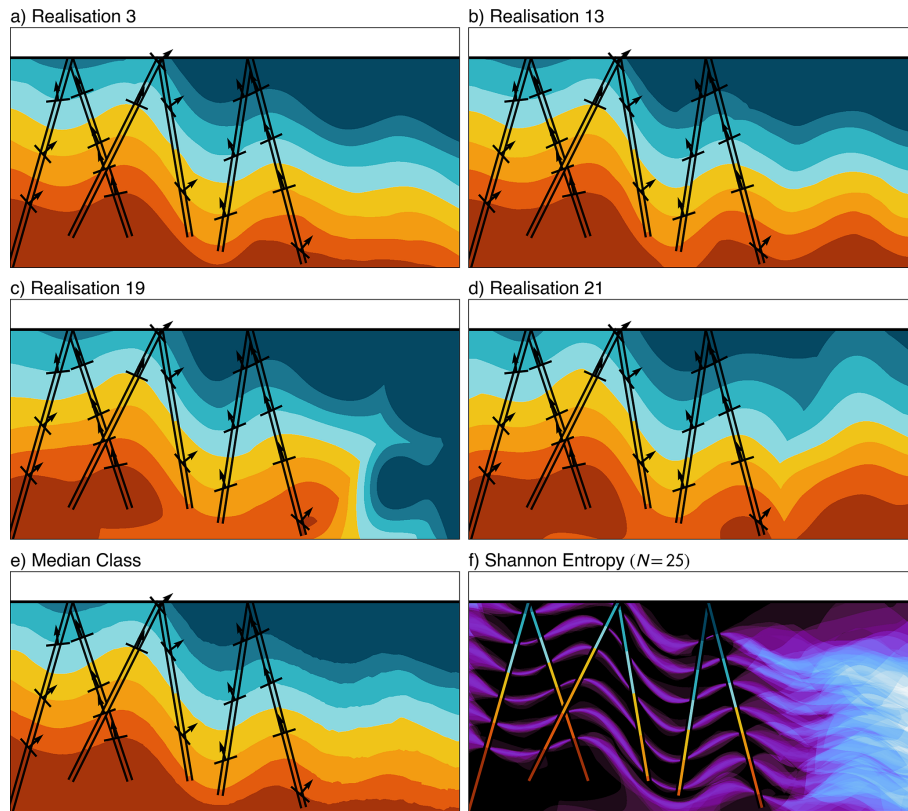


Figure 9. Selected realisations (a–d) from an ensemble of 25 *curlew* models derived using different random seeds for the RFF encoding. The median predicted class (e) highlights the predominance of geologically plausible solutions. The Shannon information entropy of the predicted classes (f) shows high uncertainty (brighter colour) at the fold hinges and away from the drill hole data, suggesting this approach might be used to meaningfully quantify model uncertainty.

belled drillhole information (property measurements) result in a solution much closer to the original (Fig. 10c).

This approach potentially adds a powerful semi-supervised aspect to the modelling process: constraints are used to define an approximate (“supervised”) geometry, which is further refined to find solutions that best explains additional quantitative data or lithological logs, reducing interpretation bias and enabling the rapid integration of new data without explicit interpretation. A similar approach might also be used to integrate geophysical constraints on model geometry, albeit with physical forward models.

5.3 Future directions

The parameterisation of geological models using neural networks opens up several exciting directions for future research and application. The introduction of Random Fourier Features, while powerful, necessitates research into the extraction of appropriate length scales from sparse data, instead of the trial-and-error approach used in the current work. Progressive implicit representations (Landgraf et al., 2022) might also help in mitigating random noise generated by including high frequencies, while fitting sparse data better. Us-

ing the domain of the model and known sparsity of the constraints, one could also generate a power-spectrum of length scales to seed the model. This would effectively eliminate the need for a length scales hyperparameter. However, as the norms of the spatial wavenumbers follow a Chi distribution, the resulting distribution of wavenumbers includes a heavy tail. Careful optimisation is necessary to ensure that no spectral noise overtakes the interpolation process of the model.

We also note that our method of generating multiple model realisations by changing the Random Fourier Feature encoding has several similarities to the turning bands and spectral methods used to simulate Gaussian random fields (Mantoglou and Wilson, 1982), suggesting that a deeper stochastic link to other Gaussian process methods may be possible. The turning bands method uses the covariance matrix to seed the random field generation, which could possibly be adapted for the sparse datasets used as inputs to geological models. However, unlike the turning bands method, our method relies on Bochner’s theorem (Bochner, 1955), which states that a shift invariant kernel i.e., the interpolator that the model is trying to approximate, is defined by the Fourier transform of the probability distribution of the frequencies used to construct the kernel. Using a uniform distribution (as suggested

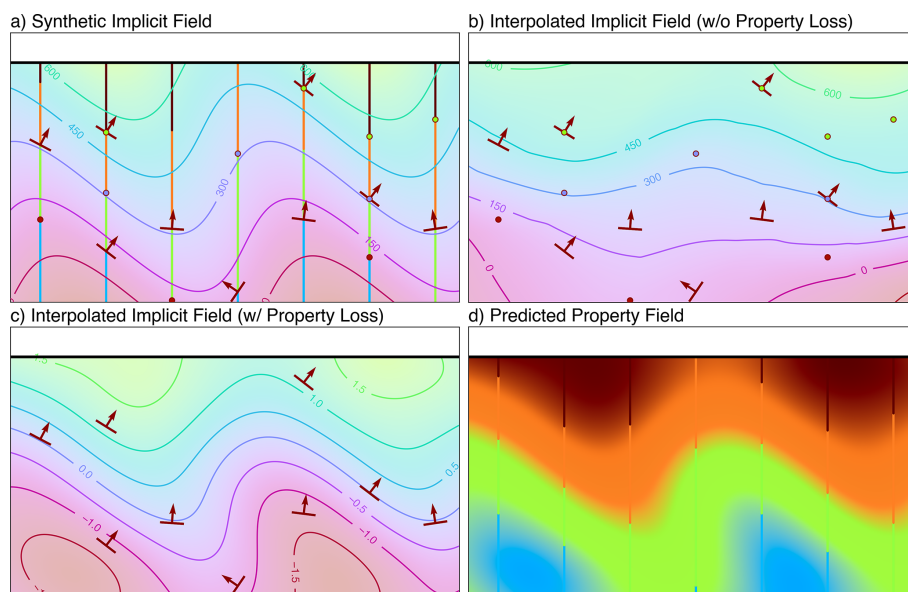


Figure 10. Synthetic (a) and interpolated implicit field derived without (b) and with (c) the learnable forward model. The original implicit field was sampled to extract sparse conventional (value and gradient) constraints and denser property measurements (coloured lines; analogous to geochemistry assays or hyperspectral drill core scans). The predicted model without the use of property constraints shows a wider folding in the basement due to lack of constraints. The addition of the learnable forward model allows the use of extra data, thereby tightening the wavelength of the interpolated model. The learnable forward model can also be evaluated across the entire domain (d) to show the recovered property field.

by Emery and Lantuéjoul, 2006, for turning bands simulations) signifies that the interpolator will behave as a Sinc- or Bessel-type kernel, and could cause ringing artifacts. A careful examination of alternative distributions for drawing these frequencies is also an avenue for future research.

The flexibility of the loss function used to train neural fields opens the door to additional innovative constraints that better encode geological rules or prior knowledge, e.g., topological relationships of the model outputs (Thiele et al., 2016a, b). However, incorporating multiple physical and geometric constraints further increases the complexity of our already complex multi-objective optimisation problem. Because different constraints often have entirely different numerical scales and physical units, each additional loss component requires careful hyperparameter tuning to prevent one objective from dominating the network's gradient updates.

To mitigate this in *curlew*, we implement an optional initial loss normalization strategy in which each individual loss component is divided by its detached initial value at the very start of training. This initial scaling forces all the losses to begin at 1, and can be thought of as an assumption that the losses at initialisation are equally bad. Once the losses are mapped to this common (unitless) scale, users can apply a single, intuitive scaling hyperparameter to explicitly dictate the relative physical or geological importance of one constraint over another. Our tests show that setting the local loss scaling hyperparameters to 1, and the global loss scaling hy-

perparameters between 0.01 and 0.1 serves as a good starting point for most models.

Alternative approaches, such as Gradient Surgery (Yu et al., 2020) and other gradient based algorithms (Zhang et al., 2024), exist to automatically select and adapt hyperparameters during training. However, further work is needed to assess the extent to which these are able to replace manual hyperparameter optimisation in the context of *curlew*. We have implemented one of these, SoftAdapt (Heydari et al., 2019), but with mixed results.

6 Conclusion

To conclude, we suggest that neural fields offer an exciting and flexible method for parameterising structural geological models, and hope that the lightweight, modular, open-source and well-documented software framework provided by *curlew* facilitates further development in this direction. While much work remains (cf. Sect. 5.3), we have applied *curlew* to real (Sect. 4.3) and structurally complex synthetic (Sect. 4.2) data to model geological structures that would challenge established approaches. Our novel loss function helps encode geological rules regarding lateral continuity (i.e. monotonicity loss), and extends earlier work by Hillier et al. (2023) to avoid assigning geological data arbitrary but restrictive values (value constraints, cf. Sect. 2.1), through our implementation of pairwise relational constraints. By

chaining neural fields together, through overprinting and off-setting relationships, *curllew* is also able to represent a variety of geological structures (faults, dykes and domain boundaries) and incrementally build complex geological models. Our results indicate that the flexibility of neural fields, and their ability to incorporate geological rules and unlabelled categorical or quantitative geological data, make them a powerful tool for the next generation of structural geological models. We hope that *curllew* helps unlock some of this potential, and ultimately facilitates more accurate and less biased representations of the subsurface.

Code and data availability. *curllew* is an open-source Python library licensed under the MIT License. It is currently hosted on <https://github.com/hifexplo/curllew> (last access: 24 September 2025), and the version associated with this publication is archived at <https://doi.org/10.5281/zenodo.17187731> (Thiele and Kamath, 2025). Documentation is available within the package and is hosted at <https://samthiele.github.io/curllew/> (last access: 22 April 2025).

The synthetic examples and the digital outcrop model shown in this contribution have been organised into documented jupyter notebooks hosted at https://github.com/k4m4th/curllew_examples (last access: 13 March 2026), and can also be found at <https://doi.org/10.5281/zenodo.19002735> (Kamath and Thiele, 2026). The specific versions of packages used to create the figures in the publication can be found in the *requirements.txt* file within the repository. The original digital outcrop from Newcastle, Australia, can be found at <https://ausgeol.org/sitedetails/?site=NewcastleUAV3> (last access: 13 March 2026).

Author contributions. AVK: Conceptualisation, Formal analysis, Code development, Investigation, Methodology, Visualisation, Writing – original draft; STT: Conceptualisation, Methodology, Code development, Writing – original draft, review and editing, Funding; MM: Code development, Writing – original draft; LG: Code testing, Writing – original draft, review and editing; RTD: Methodology, Writing – review and editing; MJH: Code testing, Writing – review and editing; FW: Writing – review and editing; RG: Writing – review and editing.

Competing interests. The contact author has declared that none of the authors has any competing interests.

Disclaimer. Publisher's note: Copernicus Publications remains neutral with regard to jurisdictional claims made in the text, published maps, institutional affiliations, or any other geographical representation in this paper. The authors bear the ultimate responsibility for providing appropriate place names. Views expressed in the text are those of the authors and do not necessarily reflect the views of the publisher.

Acknowledgements. Akshay V. Kamath and Samuel T. Thiele were partially supported by funding from the European Union's HORIZON Europe Research Council and UK Research and Innovation (UKRI) under grant agreement No. 101058483 (VECTOR). This research has also received funding from the Klaus Tschira Boost Fund, a joint initiative of GSO – Guidance, Skills & Opportunities e.V. and Klaus Tschira Stiftung.

Financial support. This research has been supported by the HORIZON EUROPE European Research Council (grant no. 101058483) and the Klaus Tschira Stiftung (grant-no. GSO/KT 76).

The article processing charges for this open-access publication were covered by the Helmholtz-Zentrum Dresden-Rossendorf (HZDR).

Review statement. This paper was edited by Evangelos Moulas and reviewed by Ítalo Gonçalves and one anonymous referee.

References

- Beatson, R. K., Light, W. A., and Billings, S.: Fast Solution of the Radial Basis Function Interpolation Equations: Domain Decomposition Methods, *SIAM J. Sci. Comput.*, 22, 1717–1740, <https://doi.org/10.1137/s1064827599361771>, 2001.
- Bjerre, E., Kristensen, L. S., Engesgaard, P., and Højberg, A. L.: Drivers and barriers for taking account of geological uncertainty in decision making for groundwater protection, *Sci. Total Environ.*, 746, 141045, <https://doi.org/10.1016/j.scitotenv.2020.141045>, 2020.
- Bochner, S.: *Harmonic Analysis and the Theory of Probability*, University of California Press, <https://doi.org/10.1525/9780520345294>, 1955.
- Bond, C. E.: Uncertainty in structural interpretation: Lessons to be learnt, *J. Struct. Geol.*, 74, 185–200, <https://doi.org/10.1016/j.jsg.2015.03.003>, 2015.
- Briggs, I. C.: Machine contouring using minimum curvature, *Geophysics*, 39, 39–48, <https://doi.org/10.1190/1.1440410>, 1974.
- Calcagno, P., Chilès, J., Courrioux, G., and Guillen, A.: Geological modelling from field data and geological knowledge, *Phys. Earth Planet. In.*, 171, 147–157, <https://doi.org/10.1016/j.pepi.2008.06.013>, 2008.
- Caumon, G., Gray, G., Antoine, C., and Titeux, M.-O.: Three-Dimensional Implicit Stratigraphic Model Building From Remote Sensing Data on Tetrahedral Meshes: Theory and Application to a Regional Model of La Popa Basin, NE Mexico, *IEEE T. Geosci. Remote*, 51, 1613–1621, <https://doi.org/10.1109/tgrs.2012.2207727>, 2013.
- Cavoretto, R., De Rossi, A., and Perracchione, E.: Efficient computation of partition of unity interpolants through a block-based searching technique, *arXiv [preprint]*, <https://doi.org/10.48550/arXiv.1604.04585>, 2016.
- Czarnecki, W. M., Osindero, S., Jaderberg, M., Świrszcz, G., and Pascanu, R.: Sobolev Training for Neural Networks, *arXiv [preprint]*, <https://doi.org/10.48550/arXiv.1706.04859>, 2017.

- de la Varga, M. and Wellmann, J. F.: Structural geologic modeling as an inference problem: A Bayesian perspective, *Interpretation*, 4, SM1–SM16, <https://doi.org/10.1190/int-2015-0188.1>, 2016.
- de la Varga, M., Schaaf, A., and Wellmann, F.: GemPy 1.0: open-source stochastic geological modeling and inversion, *Geosci. Model Dev.*, 12, 1–32, <https://doi.org/10.5194/gmd-12-1-2019>, 2019.
- Emery, X. and Lantuéjoul, C.: TBSIM: A computer program for conditional simulation of three-dimensional Gaussian random fields via the turning bands method, *Comput. Geosci.*, 32, 1615–1628, <https://doi.org/10.1016/j.cageo.2006.03.001>, 2006.
- Frank, T., Tertois, A.-L., and Mallet, J.-L.: 3D-reconstruction of complex geological interfaces from irregularly distributed and noisy point data, *Comput. Geosci.*, 33, 932–943, <https://doi.org/10.1016/j.cageo.2006.11.014>, 2007.
- Gao, K. and Wellmann, F.: Fault representation in structural modelling with implicit neural representations, *Comput. Geosci.*, 199, 105911, <https://doi.org/10.1016/j.cageo.2025.105911>, 2025.
- Gao, K., Hillier, M., and Wellmann, F.: Uncertainty quantification using Hamiltonian Monte Carlo for structural geological modelling with implicit neural representations (INR), *Comput. Geosci.*, 209, 106123, <https://doi.org/10.1016/j.cageo.2026.106123>, 2026.
- Goan, E. and Fookes, C.: *Bayesian Neural Networks: An Introduction and Survey*, Springer International Publishing, 45–87, https://doi.org/10.1007/978-3-030-42553-1_3, 2020.
- Godefroy, G., Caumon, G., Ford, M., Laurent, G., and Jackson, C. A.-L.: A parametric fault displacement model to introduce kinematic control into modeling faults from sparse data, *Interpretation*, 6, B1–B13, <https://doi.org/10.1190/int-2017-0059.1>, 2018.
- Gropp, A., Yariv, L., Haim, N., Atzmon, M., and Lipman, Y.: Implicit Geometric Regularization for Learning Shapes, *arXiv [preprint]*, <https://doi.org/10.48550/arXiv.2002.10099>, 2020.
- Grose, L., Laurent, G., Aillères, L., Armit, R., Jessell, M., and Caumon, G.: Structural data constraints for implicit modeling of folds, *J. Struct. Geol.*, 104, 80–92, <https://doi.org/10.1016/j.jsg.2017.09.013>, 2017.
- Grose, L., Aillères, L., Laurent, G., Caumon, G., Jessell, M., and Armit, R.: Modelling of faults in LoopStructural 1.0, *Geosci. Model Dev.*, 14, 6197–6213, <https://doi.org/10.5194/gmd-14-6197-2021>, 2021a.
- Grose, L., Aillères, L., Laurent, G., and Jessell, M.: LoopStructural 1.0: time-aware geological modelling, *Geosci. Model Dev.*, 14, 3915–3937, <https://doi.org/10.5194/gmd-14-3915-2021>, 2021b.
- Hasan, M., Khosravi, A., Hossain, I., Rahman, A., and Nahavandi, S.: Controlled Dropout for Uncertainty Estimation, *arXiv [preprint]*, <https://doi.org/10.48550/arXiv.2205.03109>, 2022.
- Heydari, A. A., Thompson, C. A., and Mehmood, A.: Soft-Adapt: Techniques for Adaptive Loss Weighting of Neural Networks with Multi-Part Loss Functions, *arXiv [preprint]*, <https://doi.org/10.48550/arXiv.1912.12355>, 2019.
- Hillier, M., Wellmann, F., de Kemp, E. A., Brodaric, B., Schetselaar, E., and Bédard, K.: GeoINR 1.0: an implicit neural network approach to three-dimensional geological modelling, *Geosci. Model Dev.*, 16, 6987–7012, <https://doi.org/10.5194/gmd-16-6987-2023>, 2023.
- Hillier, M. J., Schetselaar, E. M., de Kemp, E. A., and Perron, G.: Three-Dimensional Modelling of Geological Surfaces Using Generalized Interpolation with Radial Basis Functions, *Math. Geosci.*, 46, 931–953, <https://doi.org/10.1007/s11004-014-9540-3>, 2014.
- Irakarama, M., Laurent, G., Renaudeau, J., and Caumon, G.: Finite Difference Implicit Modeling of Geological Structures, in: *Proceedings, EAGE Publications BV, Copenhagen, Denmark*, <https://doi.org/10.3997/2214-4609.201800794>, 2018.
- Kamath, A. and Thiele, S.: k4m4th/curlew_examples: curlew_examples, Zenodo [code], <https://doi.org/10.5281/ZENODO.19002735>, 2026.
- Kamath, A. V., Thiele, S. T., Ugalde, H., Morris, B., Tolosana-Delgado, R., Kirsch, M., and Gloaguen, R.: TensorWeave 1.0: Interpolating geophysical tensor fields with spatial neural networks, *EGUSphere [preprint]*, <https://doi.org/10.5194/egusphere-2025-2345>, 2025.
- Lajaunie, C., Courrioux, G., and Manuel, L.: Foliation fields and 3D cartography in geology: Principles of a method based on potential interpolation, *Math. Geol.*, 29, 571–584, <https://doi.org/10.1007/BF02775087>, 1997.
- Landgraf, Z., Hornung, A. S., and Cabral, R. S.: PINs: Progressive Implicit Networks for Multi-Scale Neural Representations, *arXiv [preprint]*, <https://doi.org/10.48550/arXiv.2202.04713>, 2022.
- Laurent, G.: Iterative Thickness Regularization of Stratigraphic Layers in Discrete Implicit Modeling, *Math. Geosci.*, 48, 811–833, <https://doi.org/10.1007/s11004-016-9637-y>, publisher: Springer Science and Business Media LLC, 2016.
- Laurent, G., Caumon, G., Bouziat, A., and Jessell, M.: A parametric method to model 3D displacements around faults with volumetric vector fields, *Tectonophysics*, 590, 83–93, <https://doi.org/10.1016/j.tecto.2013.01.015>, 2013.
- Li, H., Xu, Z., Taylor, G., Studer, C., and Goldstein, T.: Visualizing the Loss Landscape of Neural Nets, *arXiv [preprint]*, <https://doi.org/10.48550/arXiv.1712.09913>, 2017.
- Lindi, O. T., Aladejare, A. E., Ozoji, T. M., and Ranta, J.-P.: Uncertainty Quantification in Mineral Resource Estimation, *Natural Resources Research*, 33, 2503–2526, <https://doi.org/10.1007/s11053-024-10394-6>, 2024.
- Lindsay, M., Jessell, M., Aillères, L., Perrouy, S., De Kemp, E., and Betts, P.: Geodiversity: Exploration of 3D geological model space, *Tectonophysics*, 594, 27–37, <https://doi.org/10.1016/j.tecto.2013.03.013>, 2013.
- Mallet, J.-L.: Discrete smooth interpolation, *ACM T. Graphic.*, 8, 121–144, <https://doi.org/10.1145/62054.62057>, 1989.
- Mantoglou, A. and Wilson, J. L.: The Turning Bands Method for simulation of random fields using line generation by a spectral method, *Water Resour. Res.*, 18, 1379–1394, <https://doi.org/10.1029/WR018i005p01379>, 1982.
- Margossian, C. C.: A Review of automatic differentiation and its efficient implementation, *WIRES*, 9, e1305, <https://doi.org/10.1002/WIDM.1305>, 2019.
- Misra, D.: Mish: A Self Regularized Non-Monotonic Activation Function, *arXiv [preprint]*, <https://doi.org/10.48550/arXiv.1908.08681>, 2019.

- Osher, S. and Sethian, J. A.: Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations, *J. Comput. Phys.*, 79, 12–49, [https://doi.org/10.1016/0021-9991\(88\)90002-2](https://doi.org/10.1016/0021-9991(88)90002-2), 1988.
- Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S.: PyTorch: An Imperative Style, High-Performance Deep Learning Library, arXiv [preprint], <https://doi.org/10.48550/arXiv.1912.01703>, 2019.
- Pérez-Díaz, L., Alcalde, J., and Bond, C. E.: Introduction: Handling uncertainty in the geosciences: identification, mitigation and communication, *Solid Earth*, 11, 889–897, <https://doi.org/10.5194/se-11-889-2020>, 2020.
- Rahaman, N., Baratin, A., Arpit, D., Draxler, F., Lin, M., Hamprecht, F. A., Bengio, Y., and Courville, A.: On the Spectral Bias of Neural Networks, in: Proceedings of the 36th International Conference on Machine Learning (ICML), in: Proceedings of Machine Learning Research, vol. 97, 5301–5310, <https://doi.org/10.48550/arXiv.1806.08734>, 2019.
- Rahimi, A. and Recht, B.: Random Features for Large-Scale Kernel Machines, in: Advances in Neural Information Processing Systems, edited by: Platt, J., Koller, D., Singer, Y., and Roweis, S., vol. 20, Curran Associates, Inc., https://proceedings.neurips.cc/paper_files/paper/2007/file/013a006f03dbc5392effeb8f18fda755-Paper.pdf, 2007.
- Raissi, M., Perdikaris, P., and Karniadakis, G. E.: Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, *J. Comput. Phys.*, 378, 686–707, <https://doi.org/10.1016/j.jcp.2018.10.045>, 2019.
- Ramachandran, P., Zoph, B., and Le, Q. V.: Searching for Activation Functions, arXiv [preprint], <https://doi.org/10.48550/arXiv.1710.05941>, 2017.
- Rasmussen, C. E. and Williams, C. K. I.: Gaussian Processes for Machine Learning, The MIT Press, <https://doi.org/10.7551/mitpress/3206.001.0001>, 2005.
- Rudin, L. I., Osher, S., and Fatemi, E.: Nonlinear total variation based noise removal algorithms, *Physica D*, 60, 259–268, [https://doi.org/10.1016/0167-2789\(92\)90242-F](https://doi.org/10.1016/0167-2789(92)90242-F), 1992.
- Sandwell, D. T.: Biharmonic spline interpolation of GEOS-3 and SEASAT altimeter data, *Geophys. Res. Lett.*, 14, 139–142, <https://doi.org/10.1029/g1014i002p00139>, 1987.
- Shannon, C. E.: A Mathematical Theory of Communication, *Bell Syst. Tech. J.*, 27, 379–423, <https://doi.org/10.1002/j.1538-7305.1948.tb01338.x>, 1948.
- Sitzmann, V., Martel, J. N. P., Bergman, A. W., Lindell, D. B., and Wetzstein, G.: Implicit Neural Representations with Periodic Activation Functions, arXiv [preprint], <https://doi.org/10.48550/arXiv.2006.09661>, 2020.
- Smith, L. T., Horrocks, T., Akhtar, N., Holden, E.-J., and Wedge, D.: Implicit neural representation for potential field geophysics, *Sci. Rep.*, 15, <https://doi.org/10.1038/s41598-024-83979-z>, 2025.
- Smith, W. H. F. and Wessel, P.: Gridding with continuous curvature splines in tension, *Geophysics*, 55, 293–305, <https://doi.org/10.1190/1.1442837>, 1990.
- Steno, N. and Oldenburg, H.: The prodromus to a dissertation concerning solids naturally contained within solids: laying a foundation for the rendering a rational account both of the frame and the several changes of the masse of the Earth, as also of the various productions in the same, Printed by F. Winter, and are to be sold by Moses Pitt, <https://doi.org/10.5962/bhl.title.145115>, 1671.
- Tacher, L., Pomian-Srzednicki, I., and Parriaux, A.: Geological uncertainties associated with 3-D subsurface models, *Comput. Geosci.*, 32, 212–221, <https://doi.org/10.1016/j.cageo.2005.06.010>, 2006.
- Tancik, M., Srinivasan, P. P., Mildenhall, B., Fridovich-Keil, S., Raghavan, N., Singhal, U., Ramamoorthi, R., Barron, J. T., and Ng, R.: Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains, arXiv [preprint], <https://doi.org/10.48550/arXiv.2006.10739>, 2020.
- Thiele, S. and Kamath, A.: samthiele/curlew: Curlew 1.00, Zenodo [code], <https://doi.org/10.5281/ZENODO.17187731>, 2025.
- Thiele, S. T., Jessell, M. W., Lindsay, M., Ogarko, V., Wellmann, J. F., and Pakyuz-Charrier, E.: The topology of geology 1: Topological analysis, *J. Struct. Geol.*, 91, 27–38, <https://doi.org/10.1016/j.jsg.2016.08.009>, 2016a.
- Thiele, S. T., Jessell, M. W., Lindsay, M., Wellmann, J. F., and Pakyuz-Charrier, E.: The topology of geology 2: Topological uncertainty, *J. Struct. Geol.*, 91, 74–87, <https://doi.org/10.1016/j.jsg.2016.08.010>, 2016b.
- Thiele, S. T., Grose, L., Samsu, A., Micklethwaite, S., Vollgger, S. A., and Cruden, A. R.: Rapid, semi-automatic fracture and contact mapping for point clouds, images and geophysical data, *Solid Earth*, 8, 1241–1253, <https://doi.org/10.5194/se-8-1241-2017>, 2017.
- Wellmann, F. and Caumon, G.: 3-D Structural geological models: Concepts, methods, and uncertainties, *Adv. Geophys.*, 59, 1–121, <https://doi.org/10.1016/bs.agph.2018.09.001>, 2018.
- Wellmann, J. F. and Regenauer-Lieb, K.: Uncertainties have a meaning: Information entropy as a quality measure for 3-D geological models, *Tectonophysics*, 526–529, 207–216, <https://doi.org/10.1016/j.tecto.2011.05.001>, 2012.
- Wellmann, J. F., Horowitz, F. G., Schill, E., and Regenauer-Lieb, K.: Towards incorporating uncertainty of structural data in 3D geological inversion, *Tectonophysics*, 490, 141–151, <https://doi.org/10.1016/j.tecto.2010.04.022>, 2010.
- Wellmann, J. F., Lindsay, M., Poh, J., and Jessell, M.: Validating 3-D Structural Models with Geological Knowledge for Improved Uncertainty Evaluations, *Energ. Proced.*, 59, 374–381, <https://doi.org/10.1016/j.egypro.2014.10.391>, 2014.
- Wendland, H.: Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree, *Adv. Comput. Math.*, 4, 389–396, <https://doi.org/10.1007/bf02123482>, 1995.
- Xie, Y., Takikawa, T., Saito, S., Litany, O., Yan, S., Khan, N., Tombari, F., Tompkin, J., Sitzmann, V., and Sridhar, S.: Neural Fields in Visual Computing and Beyond, *Comput. Graph. Forum*, 41, 641–676, <https://doi.org/10.1111/cgf.14505>, 2022.
- Xu, A. and Heagy, L. J.: Toward Understanding the Benefits of Neural Network Parameterizations in Geophysical Inversions: A Study With Neural Fields, *IEEE T. Geosci. Remote*, 63, 1–14, <https://doi.org/10.1109/tgrs.2025.3583970>, 2025.
- Yu, T., Kumar, S., Gupta, A., Levine, S., Hausman, K., and Finn, C.: Gradient Surgery for Multi-Task Learning, arXiv [preprint], <https://doi.org/10.48550/arXiv.2001.06782>, 2020.

Zhang, X., Zhao, L., Yu, Y., Lin, X., Chen, Y., Zhao, H., and Zhang, Q.: LibMOON: A Gradient-based MultiObjective Optimization Library in PyTorch, arXiv [preprint], <https://doi.org/10.48550/arXiv.2409.02969>, 2024.