Applied Energy 409 (2026) 127478

Contents lists available at ScienceDirect

Applied Energy

journal homepage: www.elsevier.com/locate/apen

# Leveraging the capabilities of physics-informed neural networks for channel optimization in PEM fuel cells

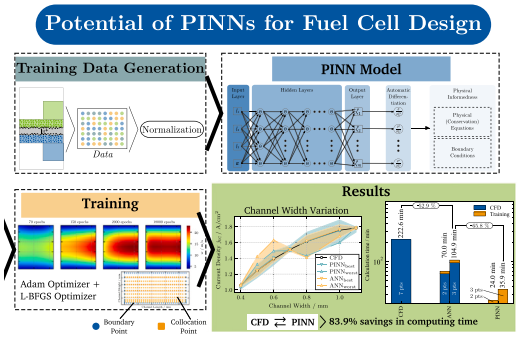
Julian Nicolas Toussaint [a,*], Sebastian Pieper [a], Max Paul Mally [b], Stefan Pischinger [a]

[a] Chair of Thermodynamics of Mobile Energy Conversion Systems (TME), RWTH Aachen University, Forckenbeckstraße 4, 52074 Aachen, Germany
[b] FEV Europe GmbH, Neuenhofstraße 181, 52078 Aachen, Germany

## HIGHLIGHTS

- Mesh-free PINN framework for PEMFC channel flow.
- Accurate PINN and ANN predictions for unseen channel widths.
- PINN outperforms conventional ANNs in generalization.
- PINN reduces computational effort by ~83.9% vs. CFD.
- PINN accelerates design and optimization of fuel cell channels.

## GRAPHICAL ABSTRACT




## ARTICLE INFO

*Keywords:*
Fuel cell
Machine learning
Physics-informed neural networks
PEMFC channel design

## ABSTRACT

Physics-informed neural networks (PINNs) offer a promising alternative to computationally expensive CFD-based design studies of proton exchange membrane fuel cells (PEMFCs). However, their suitability for modeling realistic fuel cell channel flows and their ability to generalize across different geometric designs have not yet been systematically assessed. This work addresses this gap by developing a mesh-free PINN framework that embeds the governing transport equations of multicomponent channel flow, including oxygen consumption, directly into the learning process. Unlike conventional artificial neural networks (ANNs), the proposed PINN does not require labelled CFD training data. In this work, CFD data are used for benchmarking and validation, while the PINN training itself does not rely on CFD field solutions in the interior of the computational domain, and the PINN training itself is driven solely by the embedded governing equations without labelled CFD target fields. The model is evaluated with respect to two core research questions: (i) Can a PINN accurately reproduce the coupled flow and species transport fields in a PEMFC gas channel? and (ii) Can the learned physics be leveraged to predict performance for previously unseen channel widths? Results show that the PINN reproduces CFD reference fields with high accuracy and substantially outperforms ANNs in generalizing to channel widths not included in training. Moreover, the PINN-based design study achieves a computational time reduction of approximately 83.9% compared to full CFD simulations. These findings demonstrate that PINNs provide a data-efficient and computationally lightweight surrogate model suitable for accelerating iterative PEMFC channel design, here exemplarily demonstrated for variations in channel width.

* Corresponding author.
*Email addresses:* toussaint@tme.rwth-aachen.de (J.N. Toussaint), pieper_s@tme.rwth-aachen.de (S. Pieper), mally@fev.com (M.P. Mally), pischinger_s@tme.rwth-aachen.de (S. Pischinger).
*URL:* http://www.tme.rwth-aachen.de (J.N. Toussaint).

https://doi.org/10.1016/j.apenergy.2026.127478
Received 24 November 2025; Received in revised form 6 January 2026; Accepted 29 January 2026
Available online 6 February 2026

**Nomenclature**

| | |
|---|---|
| ACL | Anode Catalyst Layer |
| AD | Automatic Differentiation |
| AI | Artificial Intelligence |
| ANN | Artificial Neural Network |
| ANFIS | Adaptive Neuro-Fuzzy Inference System |
| BPP | Bipolar Plate |
| CCL | Cathode Catalyst Layer |
| CFD | Computational Fluid Dynamics |
| CL | Catalyst Layer |
| CPU | Central Processing Unit |
| FC | Fuel Cell |
| GC | Gas Channel |
| GDL | Gas Diffusion Layer |
| GPU | Graphics Processing Unit |
| HOR | Hydrogen Oxidation Reaction |
| MEA | Membrane Electrode Assembly |
| ML | Machine Learning |
| MMP | Mixture Multiphase |
| MPL | Microporous Layer |
| MSE | Mean Squared Error |
| NN | Neural Network |
| NSE | Navier-Stokes Equation |
| OP | Operating Point |
| ORR | Oxygen Reduction Reaction |
| PDE | Partial Differential Equation |
| PEM | Proton Exchange Membrane |
| PEMFC | Proton Exchange Membrane Fuel Cell |
| PINN | Physics-Informed Neural Network |
| RMSE | Root-Mean-Square Error |
| $A_{\text{geom}}$ | Geometric area of the membrane electrode assembly |
| $A_{\text{VC}}$ | Area of the anode voltage collector |
| $a_0$ | Constant coefficient of resistance polynomial |
| $a_1$ | Linear coefficient of resistance polynomial |
| $a_2$ | Quadratic coefficient of resistance polynomial |
| $a_3$ | Cubic coefficient of resistance polynomial |
| $a_4$ | Quartic coefficient of resistance polynomial |
| $a_5$ | Quintic coefficient of resistance polynomial |
| $a_{\text{CL}}$ | Platinum loading in the catalyst layer |
| $a_i(b_t)$ | Coefficient of resistance polynomial as function of $b_t$ |
| $\alpha$ | Scenario-specific input parameter (BCs, inlet, geometry) |
| $\alpha_a$ | Anodic transfer coefficient |
| $\alpha_c$ | Cathodic transfer coefficient |
| $b$ | Gas channel width |
| $b_t$ | Channel width at data point $t$ |
| $c_{\text{O}_2}$ | Oxygen concentration in the catalyst layer |
| $c_{\text{O}_2,\text{ref}}$ | Reference oxygen concentration |
| $c^{\text{CFD}}_{\text{O}_2,t}$ | Oxygen concentration at data point $t$ from CFD |
| $D_i$ | Diffusion coefficient of species $i$ |
| $E$ | Activation energy |
| $F$ | Faraday constant |
| $j$ | Current density |
| $j_t$ | Current density at data point $t$ |
| $j_0$ | Exchange current density |
| $j_{0,t}$ | Exchange current density at data point $t$ |
| $j_0^{\text{ref}}$ | Reference exchange current density |
| $j_{\text{VC}}$ | Normalized current density, $j \cdot A_{\text{geom}}/A_{\text{VC}}$ |
| $\bar{\kappa}$ | Averaged physical property over all CFD data points |
| $L$ | Number of hidden layers in the NN |
| $L_{\text{ref}}$ | Reference length in main flow direction (z) |
| $\mathcal{L}_i$ | Physics-informed loss term from residuals |
| $\mathcal{L}_{\text{Data}}$ | Data loss term (prediction vs. reference) |
| $\mathcal{L}_{\text{BC}}$ | Boundary condition loss term |
| $\mathcal{L}_l^r$ | Physical loss of type $l$ at collocation point $r$ |
| $\mathcal{L}^s_{\text{BC},l}$ | Boundary condition loss of type $l$ at $s$ |
| $\mathcal{L}_{\text{Tot}}$ | Total loss (weighted sum of all) |
| $M$ | Molar mass of the gas mixture |
| $M_{\text{O}_2}$ | Molar mass of oxygen |
| $M_t$ | Mixture molar mass at data point $t$ |
| $\mu$ | Dynamic viscosity |
| $\mu_\xi$ | Mean of the input variable |
| $\mu_\zeta$ | Mean of the output variable |
| $N$ | Number of training data points |
| $N_S$ | Number of boundary points in boundary loss |
| $N_{\mathcal{T}}$ | Number of data points in data set $\mathcal{T}$ |
| $N_{\text{Batch}}$ | Mini-batch size |
| $N_{\text{Batch},k}$ | Number of points in mini-batch $k$ |
| $N_{\text{Layers}}$ | Number of hidden layers in NN |
| $N_{\text{Neurons}}$ | Neurons per hidden layer |
| $p$ | Pressure |
| $p_{\text{O}_2}$ | Oxygen partial pressure |
| $p_{\text{O}_2,t}$ | Partial pressure of oxygen at $t$ |
| $p_{\text{ref}}$ | Reference pressure |
| $\text{Pe}_i$ | Péclet number for species $i$ |
| $\rho$ | Density of the ideal gas mixture |
| Re | Reynolds number |
| $R$ | Universal gas constant |
| $\mathcal{R}$ | Set of collocation points (PDE residuals) |
| $R_{\text{el}}$ | Electrical resistance of the cell |
| $R_{\text{el},t}$ | Electronic resistance at $t$ |
| $\mathbf{S}$ | Set of boundary points |
| $T$ | Temperature |
| $T_{\text{ref}}$ | Reference temperature |
| $\mathcal{T}$ | CFD data set (all points) |
| $\mathcal{T}_{\text{cat}}$ | Catalyst layer data set for reactions |
| $\mathcal{T}_{\text{Batch},k}$ | Catalyst layer data in batch $k$ |
| $u$ | Velocity component in $x$-direction |
| $u_r$ | $x$-component of velocity at collocation point $r$ |
| $v$ | Velocity component in $y$-direction |
| $\mathbf{v}$ | Velocity vector $(u, v, w)^T$ |
| $w$ | Velocity component in $z$-direction |
| $w_r$ | $z$-component of velocity at collocation point $r$ |
| $w_{\text{ref}}$ | Reference velocity in $z$-direction |
| $X_{\text{O}_2}$ | Oxygen mole fraction |
| $X_{\text{O}_2,t}$ | Oxygen mole fraction at $t$ |
| $Y_i$ | Mass fraction of species $i$ |
| $Y_{i,r}$ | Mass fraction of species $i$ at collocation point $r$ |
| $Y^{\text{CFD}}_{\text{species},t}$ | Mass fraction of species $i$ at $t$ from CFD |
| $y$ | Coordinate in channel width direction |
| $z$ | Coordinate in flow direction |
| $\xi$ | Denormalized input variable |
| $\hat{\zeta}$ | Normalized output variable |
| $\zeta$ | Denormalized output variable |
| $\zeta_k$ | Predicted NN output vector |
| $\zeta_k^{\text{CFD}}$ | Reference vector from CFD/experiments |
| $\eta$ | Overpotential |
| $\eta_t$ | Surface overpotential at $t$ |
| $\eta_{\text{LR}}$ | Learning rate |
| $U_{\text{eq}}$ | Equilibrium potential |

| | | | |
|---|---|---|---|
| $U_{\mathrm{eq},t}^{\mathrm{CFD}}$ | Equilibrium potential at $t$ from CFD | $\omega_{\mathrm{BC},l}$ | Weight for BC loss type $l$ |
| $\omega_i$ | Weight for physics-informed loss term | $\omega_{\mathrm{Surf}}$ | Weight for surface overpotential loss |
| $\omega_{\mathrm{BC},i}$ | Weight for BC loss terms | $\omega_{\mathrm{Rea}}$ | Weight for Tafel reaction loss |
| $\omega_{\mathrm{Data}}$ | Weight for data loss term | $\omega_{\mathrm{Width}}$ | Weight for channel width loss |
| $\omega_l$ | Weight for physical loss type $l$ | $t_{\mathrm{train}}$ | Training time for given hyperparameters |

## 1. Introduction

The ever-increasing number of vehicles emitting greenhouse gases, primarily carbon dioxide ($CO_2$), demands the development of more sustainable propulsion systems. Since $CO_2$ emissions from the transport sector have risen continuously since 1990, the European Parliament decided in 2023 to enforce stricter regulations. Starting from 2035, only emission-free light commercial vehicles will be permitted for registration [1]. In addition to regulations targeting light-duty vehicles, the EU has also introduced stringent $CO_2$ emission standards for heavy-duty vehicles. As of 2024, heavy-duty trucks, buses, and trailers must achieve fleet-wide emission reductions of 45 % by 2030, 65 % by 2035, and 90 % by 2040 compared to 1990. Furthermore, 90 % of new urban buses must be zero-emission by 2030, rising to 100 % by 2035 [2].

Fuel Cells (FCs) offer a promising technology for sustainable propulsion in mobile systems, including heavy-duty, public transport, and light-duty vehicles, due to their fast start-up capabilities and high power density. Among these, the Proton Exchange Membrane Fuel Cell (PEMFC) is particularly suited for automotive applications. A PEMFC is an electrochemical device that converts the chemical energy of a fuel, typically hydrogen, and an oxidizing agent, usually oxygen from air, directly into electrical energy, heat, and water through electrochemical reactions without combustion [3–5]. It consists of two spatially separated electrodes, electrically connected by an electrolyte. In automotive and heavy-duty applications, a Proton Exchange Membrane (PEM) acts as both separator and an electrolyte [6]. Catalyst Layers (CLs) applied on either side of the PEM form the Membrane Electrode Assembly (MEA), which is enveloped by porous layers commonly made from carbon paper, cloth, or fibers [7]. The Gas Diffusion Layer (GDL) ensures homogeneous gas transport but can suffer from water condensation on the cathode side, reducing diffusivity and permeability. This issue is mitigated by the incorporation of a Microporous Layer (MPL) with smaller pore sizes [8]. Surrounding these functional layers, the Bipolar Plates (BPPs) form the outer boundaries of the cell. They are crucial components in Proton Exchange Membrane Fuel Cells (PEMFCs), primarily responsible for the distribution of reactants and removal of products, while also ensuring electrical conduction. They constitute approximately 30 % of the total cost and significantly influence maintenance intervals [9].

A detailed understanding of the resulting flow and transport phenomena is therefore indispensable, which has driven the use of Computational Fluid Dynamics (CFD) to investigate these processes. CFD has significantly advanced flow simulation by numerically solving the compressible and incompressible Navier-Stokes equations (NSEs) through methods such as finite element, spectral, and meshless approaches [10,11]. Nevertheless, mesh generation and solving Partial Differential Equations (PDEs) for every mesh cell remain time-consuming and computationally expensive, especially in industrial applications. Ferziger et al. [12] To overcome these limitations, alternative modeling strategies have been developed.

Traditional physics-based parametric models aim to replicate real-world processes, while purely data-driven methods train neural networks directly on measured or simulated data using supervised learning. Haykin [13]

Artificial Neural Networks (ANNs) are computational models inspired by biological neural structures. They consist of layers of interconnected neurons that process inputs through weighted sums and nonlinear activation functions. Training adjusts weights and biases via optimization algorithms like backpropagation to minimize prediction errors. This allows ANNs to approximate complex, nonlinear relationships between inputs and outputs, making them powerful tools for modeling physical systems from data. Haykin [13] Among these data-driven approaches, artificial neural networks (ANNs) have emerged as the most widely used class of models.

Physics-Informed Neural Networks (PINNs) combine traditional physics modeling with data-driven approaches to improve accuracy. Introduced by Raissi et al. [14–17], PINNs embed physical laws directly into the neural network's loss function as regularization terms. This constrains the solution space by penalizing deviations from governing equations, resulting in more physically consistent approximations.

Although PINNs are promising, research on their application, especially to PEMFCs, remains limited:

- Temperature fluctuation prediction in PEM electrolyzers [18].
- Hydrodynamic cyclo-voltammetry applications [19,20].
- Remaining useful life prediction of PEMFCs [21].

Other approaches include:

- Adaptive Neuro-Fuzzy Inference System (ANFIS) models for PEMFC performance, which combine neural networks and fuzzy logic for enhanced modeling capabilities. [22,23].
- Artificial Neural Network (ANN)-based modeling of hydrogen flow and voltage characteristics [24–26].

Beyond the above approaches, several studies have employed machine-learning models as surrogates for PEMFC performance and degradation. Feed-forward artificial neural networks have been used to predict stack voltage and current from operating conditions for commercial PEM fuel cell systems and laboratory cells, demonstrating that data-driven models can capture nonlinear performance characteristics with good accuracy [27,28]. More advanced architectures, including recurrent neural networks such as LSTM and transformer-based models, have been applied to forecast PEMFC performance and degradation trajectories from time-series data [29–31]. These works underline the potential of machine learning for fast PEMFC performance prediction, but they typically focus on system-level or stack-level metrics and require extensive labelled data for each operating configuration, rather than providing spatially resolved information inside the gas channels or generalizing across different geometries.

In parallel, physics-informed neural networks have been developed for a range of transport and fluid-mechanical problems. For incompressible flows, NSFnets and related PINN formulations have been proposed to solve the Navier–Stokes equations directly, covering laminar and turbulent regimes [32,33]. Further studies have extended PINNs to advection–dispersion and convection-dominated convection–diffusion equations, with tailored loss functionals designed to improve stability and accuracy in challenging transport regimes [34,35]. PINNs and other physics-informed architectures have also been used to reconstruct high-resolution flow fields from sparse measurement data, for example for cylinder wakes, and more broadly, machine learning has been explored for surrogate-based optimization in porous media and PEMFC flow-field design [36,37]. However, these studies do not address reactive, multi-component gas transport in PEMFC channels nor the ability of PINNs to generalize across varying channel geometries, which is essential for their use as surrogate models in bipolar plate design.

Despite the significant progress made in CFD-based analysis of PEMFC channel flows, the current literature exhibits several limitations that hinder efficient geometric design. First, CFD remains computationally expensive and is not well suited for rapid design iterations. Second, existing surrogate modeling approaches—primarily based on ANNs, ANFIS, or purely data-driven models—depend heavily on CFD or experimental training data and lack the ability to generalize beyond the geometries they were trained on. These models therefore offer limited applicability for design exploration involving parameter variations such as channel width.

Recent work has shown that physics-informed neural networks (PINNs) can reduce the dependence on labelled training data by enforcing PDEs directly in the loss function. PINNs have been extensively studied for Navier–Stokes flows, convection–diffusion problems, and complex transport processes. However, these studies either focus on simplified 1D or 2D physics, non-reactive flows, or system-level behavior rather than spatially resolved species and momentum transport in PEMFC channels. Crucially, existing studies provide no evidence regarding the ability of PINNs to generalize across geometric variations – a fundamental requirement for surrogate models used in design optimization. Moreover, no work to date has demonstrated mesh-free prediction of multicomponent, reactive PEMFC gas-channel flow fields using PINNs.

Based on this critical assessment of the literature, a clear research gap emerges: There is currently no geometry-generalizable, mesh-free surrogate model capable of predicting multicomponent reactive transport in PEMFC gas channels without requiring CFD training data for every geometry.

To address this gap, we propose the following hypothesis: A physics-informed neural network that embeds the governing transport equations can learn the underlying physics sufficiently well to generalize to unseen channel widths, thereby reducing the computational cost associated with full CFD simulations.

From this hypothesis, we derive the central research question of this work: *Can a physics-informed neural network accurately predict spatially resolved flow and species fields in PEMFC gas channels for geometries that were not included during training, and can it do so with substantially reduced computational cost compared to CFD?*

To answer this question, this study makes the following contributions: (1) Development of a mesh-free PINN framework for multicomponent, reactive PEMFC channel flow that embeds species transport, oxygen consumption, and momentum conservation directly into the loss function. (2) A systematic evaluation of the PINN's generalization capability by predicting reaction rates, mole fractions, and pressure fields for channel widths not included during training. (3) A quantitative comparison against conventional ANNs trained on CFD data, demonstrating superior extrapolation behavior of the PINN.

Although the PINN framework itself does not require labelled CFD data, a small CFD subset is used in this study solely for benchmarking and for demonstrating the generalization capability to unseen channel widths.

## 2. Methodology

The overall structure of the employed PINN and the training procedure is shown in Fig. 1. The structure of this section follows the illustration to provide a clear and coherent description of the workflow used to build and implement the PINN model. The content of the petrol-colored block *Training Data Generation* and the underlying CFD model are described in detail in Section 2.1, where the required *Normalization* methodology (white-colored block) is also explained. The blue block *PINN Model* comprises a description of the implemented *Deep Neural Network* (see Section 2.2). This model contains the *Physical Conservation Equations* for the respective problem to be solved, which are further specified for the two physical mechanisms considered in this study in Section 2.4. Finally, the options for constructing the *Loss Function* (green block) and the *Training Algorithms* used are discussed in Section 2.3.

### 2.1. Training data generation

Starting at the top left of Fig. 1, two distinct data sets extracted from the CFD model are imported. Rather than being combined, each data set is used to train an individual neural network, with one dedicated to the gas channel and the other to the catalyst layer. The first data set contains the data from the cathode Gas Channel (GC), where the flow is modeled as laminar. Inside the GC of the cathode, a multicomponent mixture consisting of $N_2$, $O_2$, and $H_2O$ is modeled according to the ideal gas law. Since the amount of data in the gas channel is comparatively large, a single representative segment along the primary flow direction ($z$-direction) is considered for analysis, while maintaining a fixed $y$-position of $6 \cdot 10^{-4}$ m, which corresponds to the middle of the channel. The second data set contains data from the Catalyst Layer (CL), which is modeled as an infinitely thin interface between the membrane

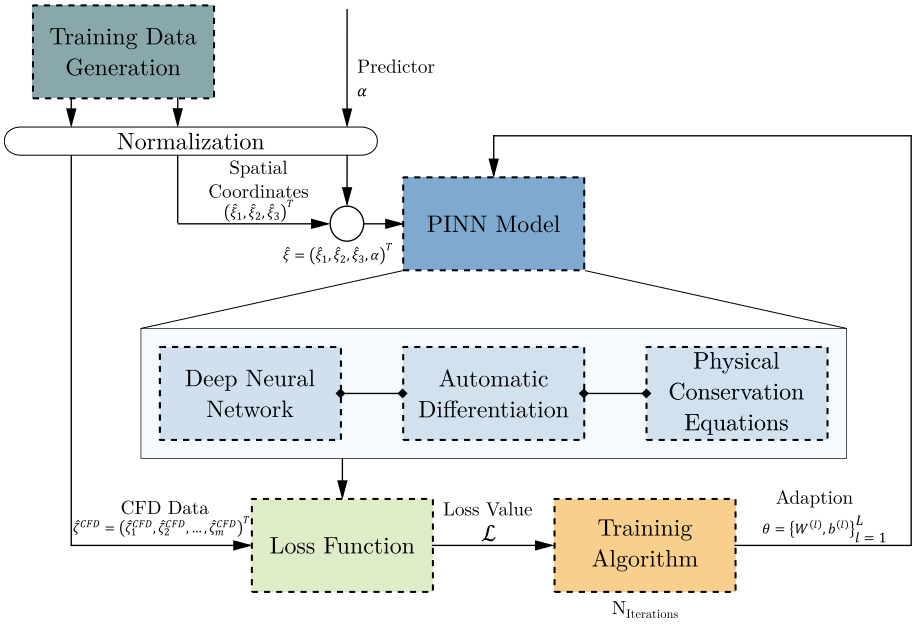



**Fig. 1.** Structure of the physics-informed neural network (PINN).

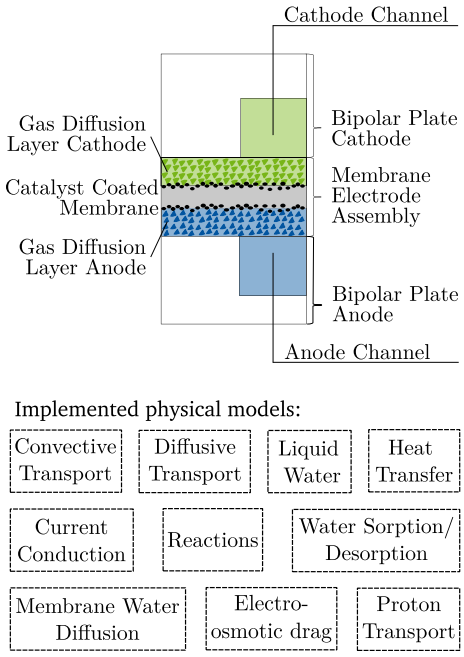



**Fig. 2.** Schematic of the investigated PEMFC.

and the MPL [38]. For training the catalyst layer model, only one segment is used, which makes it possible to neglect the dependency of the electrical resistance on the channel length ($z$-direction).

After importing the data it is normalized according to the z-score

$$\hat{\xi} = \frac{\xi - \mu_\xi}{\sigma_\xi} \tag{1}$$

$$\hat{\zeta} = \frac{\zeta - \mu_\zeta}{\sigma_\zeta} \tag{2}$$

where $\xi$ denotes an arbitrary input and $\zeta$ an arbitrary output variable, respectively. $\mu_\xi$ and $\mu_\zeta$ are the respective mean values, with $\sigma_\xi$ and $\sigma_\zeta$ being the corresponding standard deviations. By normalizing the data, the stability of the training is improved [39]. For the quantitative evaluations of the current density and the overpotential reported in this work for the predictive power of the PINN, the root-mean-square error (RMSE) is computed after de-normalizing the network outputs. Concretely, the network predictions $\hat{\zeta}$ are first transformed back to physical units via $\zeta = \mu_\zeta + \sigma_\zeta \hat{\zeta}$ (cf. Eq. 17), and the RMSE is then calculated with respect to the corresponding CFD reference quantity in physical units.

The training data sets are based on the CFD model of a PEMFC channel developed by Toussaint et al. [38] for geometry optimization in the design process. The model accounts for all relevant physical mechanisms of a PEMFC (see Fig. 2). Owing to the methodology proposed by Toussaint et al., the model enables flexible validation against different MEA materials through two fitting parameters. Furthermore, each geometry parameter of the MEA and channel can be freely specified, making the model well suited as a high-resolution data basis for the surrogate PINN model developed in this work. The model is set up to operate in a stationary regime and only considers half of the gas channel due to the symmetry boundary condition.

This detailed model captures the transport and electrochemical processes within the channel. In the present study, a hydrogen-air fuel cell is considered, where hydrogen serves as the reducing agent (fuel) at the anode and oxygen from air as the oxidizing agent at the cathode. The net reaction of the PEMFC can be expressed as [6]:

$$H_2 + \frac{1}{2} O_2 = H_2O. \tag{3}$$

Both hydrogen and air continuously flow through their respective gas channels of the bipolar plate. Their pressure distribution and flow dynamics are described by the continuity equation and the momentum balance [40]:

$$\nabla \cdot \boldsymbol{v} = 0 \tag{4}$$

$$\rho\, \boldsymbol{v} \cdot \nabla \boldsymbol{v} = -\nabla p + \mu\, \nabla^2 \boldsymbol{v} \tag{5}$$

Here, $\nabla$ denotes the Nabla operator, $\rho$ the density of the ideal gas mixture, $\boldsymbol{v} = (u, v, w)^\top$ the velocity vector, $p$ the pressure, and $\mu$ the dynamic viscosity. Due to the stationary operating point in the CFD model, the temporal dependencies of Eqs. (4) and (5) have already been omitted and will be omitted in further equations.

While the fluids flow through the GC, they diffuse through their GDL and MPL toward the CL. The diffusion process inside the GC is captured by the species transport balance

$$\boldsymbol{v} \cdot \nabla Y_i = D_i\, \nabla^2 Y_i \quad , \tag{6}$$

where $Y_i$ denotes the mass fraction of species $i$. Bird and Stewart [40]

At the CL the electrochemical reactions occur. At the anode, hydrogen undergoes the hydrogen oxidation reaction (Hydrogen Oxidation Reaction (HOR)), producing protons and electrons as described in Eq. (7). The proton-conductive PEM allows for the diffusion of $H^+$ from the Anode Catalyst Layer (ACL) to the Cathode Catalyst Layer (CCL). The electrons are conducted by the porous layers and Bipolar Plate (BPP) in the opposite direction of the protons from ACL to CCL (cf. Fig. 2). The Oxygen Reduction Reaction (ORR) on the cathode (Eq. 8) recombines the electrons and protons with oxygen to form water $H_2O$ [6,41,42].

$$H_2 \longrightarrow 2\,H^+ + 2\,e^- \tag{7}$$

$$\frac{1}{2}\,O_2 + 2\,H^+ + 2\,e^- \longrightarrow H_2O \tag{8}$$

The kinetics of the electrochemical reactions (cf. Eqs. (7) and (8)) are modeled by the Butler-Volmer equation which relates the current density $j$ with the overpotential $\eta$. For the cathode, the Butler-Volmer equation takes the following form [43]:

$$j = j_0 \left( \frac{c_{O_2}}{c_{O_2,\mathrm{ref}}} \right)^{\gamma_{O_2}} \cdot \left[ \exp\left( \frac{\alpha_a F\, \eta}{R T} \right) - \exp\left( -\frac{\alpha_c F\, \eta}{R T} \right) \right] \tag{9}$$

Here, $j_0$ denotes the exchange current density, $T$ the temperature, $c_{O_2}$ the oxygen concentration at the CL, and $c_{O_2,\mathrm{ref}} = 1\ \mathrm{kmol/m^{-3}}$ its reference value. Because temperature variations are small in the operating regime considered, the temperature is set to a constant value $T = 353.15$ K. $F$ and $R$ are the Faraday constant and the ideal gas constant, respectively. $\gamma_{O_2}$ denotes the reaction order, and $\alpha_a$ as well as $\alpha_c$ are modeling parameters to balance the backward and forward reaction directions. Here, only the cathode is considered, as it has the lower reaction rate and therefore limits the current density.

Since the Operating Point (OP) is not close to equilibrium, the second exponential term in Eq. (9) can be neglected, resulting in the Tafel equation [44]

$$j = j_0 \left( \frac{c_{O_2}}{c_{O_2,\mathrm{ref}}} \right)^{\gamma_{O_2}} \exp\left( \frac{\alpha_a F\, \eta}{R T} \right). \tag{10}$$

The exchange current density $j_0$ is modeled by [45,46]

$$j_0 = j_0^{\mathrm{ref}}\, a_{\mathrm{CL}} \left( \frac{p_{O_2}}{p_{\mathrm{ref}}} \right)^{\gamma} \cdot \exp\left( -\frac{E}{R T} \left[ 1 - \frac{T}{T_{\mathrm{ref}}} \right] \right), \tag{11}$$

an Arrhenius equation with the activation energy $E = 72.4\ \mathrm{kJ\ mol^{-1}}$ and reference temperature $T_{\mathrm{ref}} = 298.15$ K. The reference exchange current density is determined based on the methodology developed by Toussaint

et al. [38] to $j_0^{\text{ref}} = 3.32\ \text{A/cm}^{-2}$. Both the platinum loading $a_{\text{CL}}$ and the pressure dependence $\gamma$ equal unity. The reference pressure is $p_{\text{ref}} = 1.013 \cdot 10^5$ Pa. $p_{O_2}$, the oxygen partial pressure, can be determined based on the pressure $p$ and the molar fraction of oxygen $X_{O_2}$ (see detailed calculation in the appendix in Appendix A.1)

Since the Tafel equation (cf. Eq. 10) combines two unknowns, the definition of the overpotential $\eta$ is used

$$\eta = -U_{\text{eq}} - R_{\text{el}}\, A_{\text{geom}}\, j \tag{12}$$

to enable the determination of $\eta$ and $j$, provided that $j_0$, $c_{O_2}$, and $U_{\text{eq}}$ are known. Here, $U_{\text{eq}}$ denotes the equilibrium potential, $R_{\text{el}}$ the electrical resistance of the cell, and $A_{\text{geom}}$ the geometric area of the MEA. The electrical resistance $R_{\text{el}}$ is not a single material property but results from the series connection of several partial resistances within the cell. It is composed of the ohmic resistance of the BPPs and the GDL, as well as the contact resistances at the interfaces between these components.

When analyzed, the electrical resistance showed a significant dependence on the cell width ($y$). This analysis was conducted by first rewriting Eq. (12)

$$R_{\text{el}} = -\frac{\eta^{\text{CFD}} + U_{\text{eq}}^{\text{CFD}}}{A_{\text{geom}}\, j^{\text{CFD}}} \tag{13}$$

and plotting the resulting distribution of the electrical resistance in Fig. 3. It should be noted that the data used in Eq. (13) is directly provided by the CFD.

Fig. 3(a) shows the distribution of the electrical resistance across the entire CL. The solid black line separates regions where the GDL is in contact with either the gas channel or the rib. In Fig. 3 and (b), the segment used for training is highlighted. Within this segment, the strong dependence of the electrical resistance on the cell width $y$ is clearly visible. The higher values beneath the ribs originate mainly from the increased contact resistance between the GDL and the bipolar plate in these regions, which dominates the total resistance [47]. In contrast, the contact resistance beneath the channels is lower, resulting in smaller overall values. A weak dependence on the channel length $z$ can also be observed, but it is neglected in the following, although it is more pronounced in the full distribution shown in Fig. 3(a).

To model the electrical resistance, its values are plotted as a function of the cell width $y$ for $z = 125$ mm in Fig. 3(c). This data can be perfectly approximated by a fifth-order polynomial

$$R_{\text{el}}(b, y) = a_5(b)\, y^5 + a_4(b)\, y^4 + a_3(b)\, y^3 + a_2(b)\, y^2 + a_1(b)\, y + a_0(b) \tag{14}$$

in $y$ direction, since the data and model in fig. 3(c) are in perfect agreement. The coefficients of the polynomial $a_0, \ldots, a_5$ depend on the channel width $b$. Some of them are summarized in Table 1. The entirety of the coefficients can be found in Table A.7.

### 2.2. PINN model

In the subsequent step, the normalized data is used to train the PINN. It consists of an input layer that receives the directions $\xi = (\xi_1, \xi_2, \xi_3, \alpha)^T = (x, y, z, \alpha)^T$ as inputs. To enable the PINN to represent a wide range of boundary or input conditions, an additional input parameter $\alpha$ is introduced. The parameter $\alpha$ acts as an explicit predictor that encodes scenario-specific settings, such as boundary values, inlet velocity profiles, or geometric properties. In practice, $\alpha$ can be provided as a single scalar value that uniquely identifies the respective scenario. This allows the network to distinguish between data sets originating from different operating conditions, so that variations in, for example, boundary conditions or inlet velocities can be represented within a single trained model. The outputs of the input layer then pass through a defined number of hidden layers and neurons in each. At the end, the output layer provides the predicted variables $\zeta_1, \zeta_2, \ldots, \zeta_m$ which represent the physical quantities of interest depending on the specific problem. The output layer can include the pressure $p$, velocities $u, v, w$, oxygen mass fraction $Y_{O_2}$, overpotential $\eta$, and more quantities such as the reaction rate if the catalyst layer is investigated. This architecture is illustrated in Fig. 4.

The PINN model itself is also saved and subsequently used both to visualize its progress in replicating the supplied data and to predict new data.

The activation function for each neuron is the *hyperbolic tangent* tanh, as it is smooth, infinitely differentiable, and symmetric around zero,

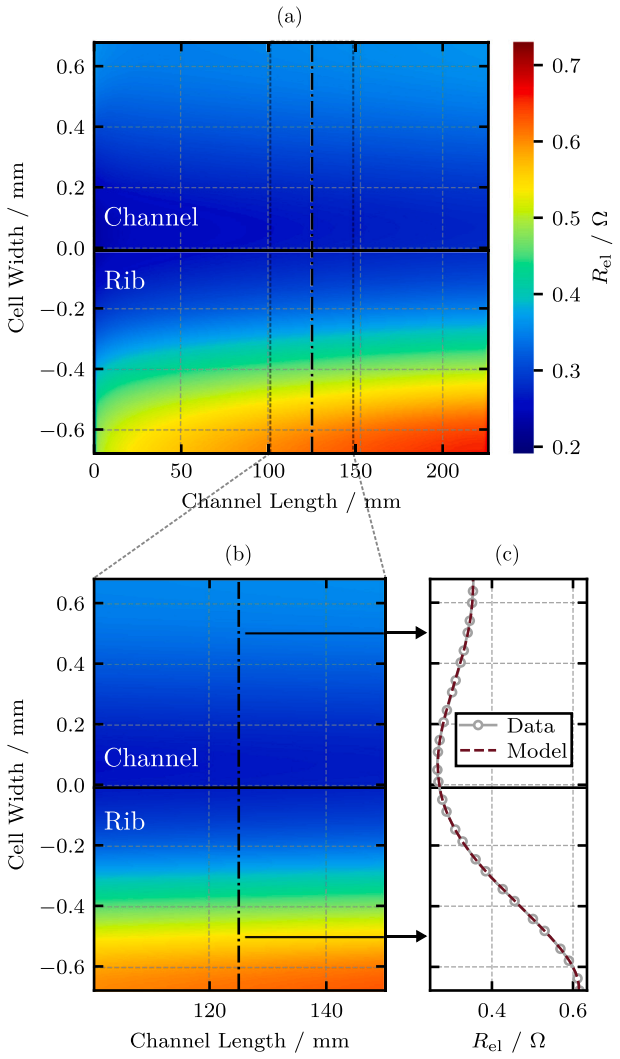



**Fig. 3.** Electrical resistance for (a) the entire CL, (b) the training segment, and (c) the electrical resistance and the model values at channel length $z = 125$ mm for the CFD data set with channel width $b = 0.687388$ mm.

**Table 1**
Coefficients for the resistance polynomial in Eq. (14) for different channel widths $b$. Note that the channel width $b = 0.69$ mm represents the CFD data for $b = 0.687388$ mm, as all values are rounded to the second decimal place. The full-precision coefficients for all channel widths $b$ can be found in Table A.7.

| Coefficient | Channel width $b$ [mm] | | |
|---|---|---|---|
| | 0.4 | 0.69 | 1.1 |
| $a_5$ | $2.77 \cdot 10^{15}$ | $8.65 \cdot 10^{14}$ | $6.45 \cdot 10^{14}$ |
| $a_4$ | $-1.52 \cdot 10^{12}$ | $-1.06 \cdot 10^{12}$ | $-2.13 \cdot 10^{11}$ |
| $a_3$ | $-1.32 \cdot 10^{9}$ | $-5.68 \cdot 10^{8}$ | $-8.05 \cdot 10^{8}$ |
| $a_2$ | $1.69 \cdot 10^{6}$ | $9.53 \cdot 10^{5}$ | $2.26 \cdot 10^{5}$ |
| $a_1$ | $-9.86 \cdot 10^{2}$ | $-1.14 \cdot 10^{2}$ | $5.00 \cdot 10^{2}$ |
| $a_0$ | $5.82 \cdot 10^{-1}$ | $2.70 \cdot 10^{-1}$ | $3.79 \cdot 10^{-1}$ |

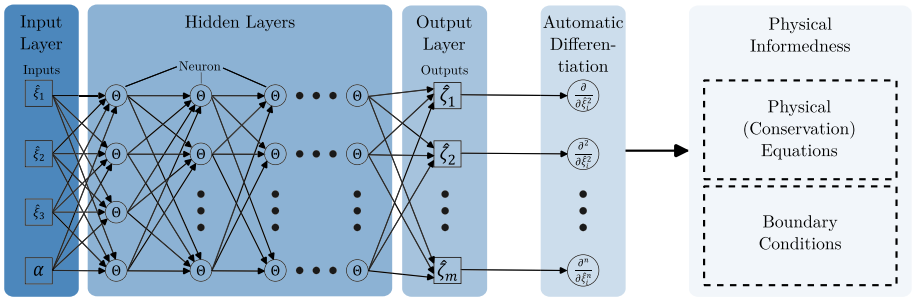



**Fig. 4.** Architecture of the PINN model.

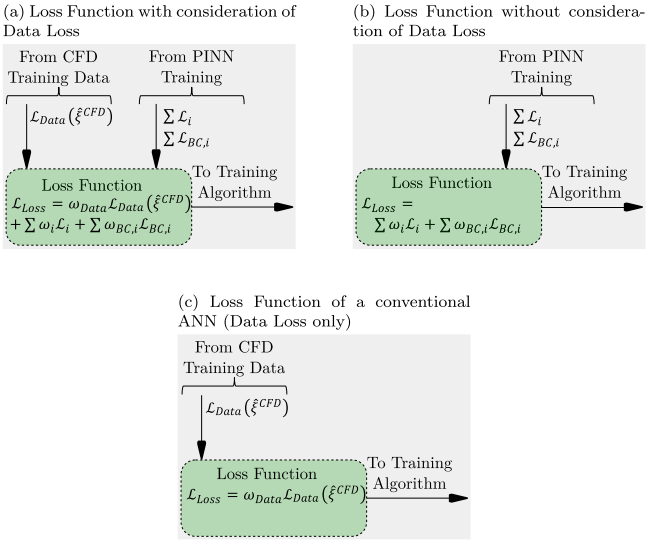



**Fig. 5.** Methodology to build the loss function.

which ensures stable gradient computation and efficient training when computing higher-order derivatives via automatic differentiation. Its bounded range and smooth derivatives make it particularly suitable for representing solutions to PDEs without introducing numerical instabilities. Gnanasambandam et al. [48] The network weights are initialized with the Glorot uniform initializer to maintain balanced variance across layers, and biases are initialized to zero. L2 regularization with a factor of 0.01 is applied to mitigate overfitting. The architecture comprises $L$ fully connected hidden layers. Glorot and Bengio [49]

### 2.3. Loss function and training algorithm

In the context of solving governing equations of physical systems with neural networks, three different strategies for constructing the loss function can be distinguished (cf. Fig. 5):

*Loss function with combined losses (inverse formulation).* In this formulation [50–52] (see Fig. 5a), the loss function consists of two contributions: a physics-informed loss term and a data loss term.

The physics-informed loss term $\mathcal{L}_i$ is defined via the residuals of the governing equations, which depend on the specific physical application (e.g., gas flow in the channels (cf. Section 2.4.1) or electrochemical reactions in the catalyst layer (cf. Section 2.4.2)). $\mathcal{L}_{BC,i}$ quantifies the contribution of the physical loss arising at the boundaries. These residuals are evaluated by inserting the gradients of the network outputs with respect to the inputs, computed efficiently and exactly via automatic differentiation, into the corresponding equations.

The data loss $\mathcal{L}_{\text{Data}}$ quantifies the discrepancy between the predicted network outputs $\hat{\boldsymbol{\zeta}}_k = (p, u, v, w, Y_{\text{O}_2}, \eta, \dots)^\top$ and the reference data $\hat{\boldsymbol{\zeta}}_k^{CFD}$ (typically obtained from high-fidelity CFD simulations or experiments) and is defined as

$$\mathcal{L}_{\text{Data}} = \frac{1}{N} \sum_{i=1}^{N} \left\| \hat{\boldsymbol{\zeta}}_k^{CFD} - \hat{\boldsymbol{\zeta}}_k \right\|_2^2 \tag{15}$$

where $N$ denotes the number of data points.

While this strategy allows the network to simultaneously exploit data and physics, it leads to a competition between both contributions: the data loss pushes the model toward reproducing the reference data, whereas the physics-informed part enforces consistency with the governing laws. Finding a suitable balance remains a challenge. [53,54]

*Loss function with physical conservation equations (forward formulation).* In this approach (see Fig. 5b), the network is trained exclusively on the residuals of the governing equations and boundary conditions, thereby avoiding the competition problem of the combined formulation. Together with the boundary values $\mathcal{L}_{BC}$ at the specified domain, they form the physics-informed loss term, which serves as the fundamental loss function in this training process [17,55,56]. This purely physics-based forward formulation is therefore adopted in the present work. CFD simulation data are only employed for (i) model validation and (ii) benchmarking against conventional ANN models, which rely on supervised learning using CFD reference fields. This distinction clarifies that the PINN learns the physics directly, whereas the ANN requires CFD-based training data.

*Loss function with data loss only.* For completeness, a purely data-driven network is also considered (see Fig. 5c), where the loss function consists solely of the data loss term Eq. (15). This conventional approach disregards physical consistency but provides a baseline to highlight the advantages of physics-informed formulations. Zhou and Xu [57]

*Training algorithm and weighting factors.* Independent of the chosen loss formulation, the training process is divided into two phases using different optimizers and hardware: Initially, the Adam optimizer [58] runs on the Graphics Processing Unit (GPU), adapting weights and biases via backpropagation on mini-batches. This accelerates convergence by frequently updating the loss gradient. Once convergence is reached, training is continued with the L-BFGS optimizer [59] on the entire dataset using the CPU, allowing for more precise parameter refinement. To ensure reproducibility of all reported results, a fixed random seed of 12,345 was used for network initialization, batch shuffling, and optimizer-related stochastic processes in both the ANN and PINN training.

During the Adam phase, training is performed using mini-batches to reduce GPU memory consumption and enable efficient parallelization. For the gas channel model, mini-batches consist exclusively of collocation points, while all boundary points are evaluated in every epoch.

Batch sizes are chosen as large as possible within the available GPU memory limits (cf. Table 4). In contrast, the L-BFGS optimizer operates on the full dataset and is therefore executed on the CPU, as it requires repeated evaluation of the complete loss function and its gradients, which is not memory-efficient on the GPU.

In PINNs, the different loss contributions (data, physics residuals, boundary values) can vary significantly in scale. To prevent certain terms from dominating training, weighting factors $\omega_i$, $\omega_{BC,i}$, and $\omega_{\text{Data}}$ are introduced. These weights balance the contributions, stabilize convergence, and improve the predictive accuracy of the model [60,61].

For both the ANN and PINN, training of the catalyst layer models is terminated after reaching a predefined maximum number of iterations. For the gas channel models, training with the Adam optimizer is likewise stopped after the maximum number of iterations, whereas the subsequent L-BFGS optimization is terminated once the total loss drops below $10^{-4}$.

### 2.4. Application-specific implementations

The PINN framework described in the previous sections is designed to be highly general and capable of representing a wide range of physical scenarios. In the following, two specific aspects relevant to the PEMFC are presented to demonstrate how the generic model can be adapted to particular modeling tasks. The first application focuses on gas flow modeling within the cell, while the second addresses the electrochemical reaction processes in the catalyst layer.

It is important to distinguish between the role of CFD data in the gas channel and in the catalyst layer modeling. For the gas channel, CFD data are used exclusively to prescribe boundary conditions at the domain boundaries. The interior solution of the gas channel is learned solely from the residuals of the governing equations using collocation points, without employing CFD field data as training targets.

#### 2.4.1. Gas flow modeling

The gas flow inside the GC is described by the continuity equation, momentum balance, and species transport (cf. Eqs. 4–6). However, to make the solution of these PDEs unique, boundary conditions need to be supplied as well. For that, the entire CFD data set $\mathcal{T}$ containing $N_{\mathcal{T}} = |\mathcal{T}|$ data points is split into two disjoint sets $\mathcal{T} = \mathcal{R} \cup \mathcal{S}$, as shown in Fig. 6. The set of collocation points $\mathcal{R}$ is used to evaluate the loss of the PDEs. The boundary conditions are enforced on all boundary points of their respective set $\mathcal{S}$. For the gas channel data set, the CFD mesh consists of 4950 points in total. Of these, 990 points (approximately 20 %) are classified as boundary points $\mathcal{S}$, while the remaining 3960 points are used as collocation points $\mathcal{R}$. Boundary points are identified based on their proximity to the domain extrema, where all points located within 1 % of the minimum or maximum coordinate in each spatial direction are considered boundary points.

To combine the PINN with these physical equations, a loss-term is formulated for each equation. As aforementioned, the inputs and outputs of the neural network are normalized according to the z-score (cf. Eqs. (1) and 2). Therefore, the partial derivatives determined with Automatic Differentiation (AD) share the normalized scale of the PINNs outputs. To maintain the consistency of the physical equations, the partial derivatives need to be scaled. Rewriting Eqs. (1) and (2) leads to

$$\xi = \mu_\xi + \sigma_\xi \hat{\xi} \tag{16}$$

$$\zeta = \mu_\zeta + \sigma_\zeta \hat{\zeta} \tag{17}$$

which results in the following scaling for the first partial derivatives:

$$\frac{\partial \zeta}{\partial \xi} = \frac{\partial\left(\mu_\zeta + \sigma_\zeta \hat{\zeta}\right)}{\partial\left(\mu_\xi + \sigma_\xi \hat{\xi}\right)} = \frac{\sigma_\zeta}{\sigma_\xi} \frac{\partial \hat{\zeta}}{\partial \hat{\xi}} \tag{18}$$

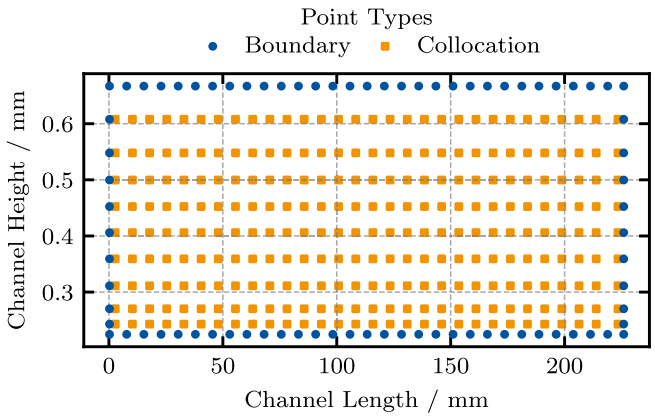



**Fig. 6.** Split CFD data set for the gas channel into boundary and collocation points.

Similarly, the second partial derivatives are scaled:

$$\frac{\partial^2 \zeta}{\partial \xi^2} = \frac{\sigma_\zeta}{\sigma_\xi^2} \frac{\partial^2 \hat{\zeta}}{\partial \hat{\xi}^2} \tag{19}$$

Rewriting Eqs. (4)–6 such that they each equate to zero and combining them with Eqs. (18) and (19) results in the following loss terms for all collocation points $r \in \mathcal{R}$:

$$\mathcal{L}^r_{\text{Con}} = \frac{\sigma_u}{\sigma_x} \frac{\partial \hat{u}_r}{\partial \hat{x}_r} + \frac{\sigma_w}{\sigma_z} \frac{\partial \hat{w}_r}{\partial \hat{z}_r} \tag{20}$$

$$\mathcal{L}^r_{\text{Mom},x} = \frac{\sigma_x}{\sigma_p} \sigma_u \left[\frac{u_r}{\sigma_x} \frac{\partial \hat{u}_r}{\partial \hat{x}_r} + \frac{w_r}{\sigma_z} \frac{\partial \hat{u}_r}{\partial \hat{z}_r}\right] - \frac{\sigma_x}{\sigma_p} \frac{\sigma_u}{\text{Re}} \left[\frac{1}{\sigma_x^2} \frac{\partial^2 \hat{u}_r}{\partial \hat{x}_r^2} + \frac{1}{\sigma_z^2} \frac{\partial^2 \hat{u}_r}{\partial \hat{z}_r^2}\right] + \frac{\partial \hat{p}_r}{\partial \hat{x}_r} \tag{21}$$

$$\mathcal{L}^r_{\text{Mom},z} = \frac{\sigma_z}{\sigma_p} \sigma_w \left[\frac{u_r}{\sigma_x} \frac{\partial \hat{w}_r}{\partial \hat{x}_r} + \frac{w_r}{\sigma_z} \frac{\partial \hat{w}_r}{\partial \hat{z}_r}\right] - \frac{\sigma_z}{\sigma_p} \frac{\sigma_w}{\text{Re}} \left[\frac{1}{\sigma_x^2} \frac{\partial^2 \hat{w}_r}{\partial \hat{x}_r^2} + \frac{1}{\sigma_z^2} \frac{\partial^2 \hat{w}_r}{\partial \hat{z}_r^2}\right] + \frac{\partial \hat{p}_r}{\partial \hat{z}_r} \tag{22}$$

$$\mathcal{L}^r_{\text{Spe}} = \frac{u_r}{\sigma_x} \frac{\partial \hat{Y}_{i,r}}{\partial \hat{x}_r} + \frac{w_r}{\sigma_z} \frac{\partial \hat{Y}_{i,r}}{\partial \hat{z}_r} - \frac{1}{\text{Pe}_i} \left[\frac{1}{\sigma_x^2} \frac{\partial^2 \hat{Y}_{i,r}}{\partial \hat{x}_r^2} + \frac{1}{\sigma_z^2} \frac{\partial^2 \hat{Y}_{i,r}}{\partial \hat{z}_r^2}\right] \tag{23}$$

Since the data set for the GC is constant in $y$, its derivatives with respect to $y$ are not shown here. Note that Eqs. (21) and (22) were each multiplied by the inverse scaling of the pressure gradient, i.e., $\sigma_x/\sigma_p$ and $\sigma_z/\sigma_p$, to reduce the order of magnitude of the loss. Furthermore, the Reynolds-number $\text{Re} = \bar{\rho}\, w_{\text{ref}}\, L_{\text{ref}}/\bar{\mu}$ and the Péclet-number $\text{Pe}_i = w_{\text{ref}}\, L_{\text{ref}}/\bar{D}_i$ were introduced to balance the scales of the convective and diffusive terms in Eqs. (21)–(23). $w_{\text{ref}} = 10\ \text{m/s}^{-1}$ and $L_{\text{ref}} = 5 \times 10^{-2}$ m denote the reference velocity and reference length in the main flow direction ($z$-direction), respectively. They both use average parameters $\bar{\kappa}$ ($\kappa = \rho, \mu, D_i$) that are averaged over the $N_{\mathcal{T}}$ data points of the CFD data set:

$$\bar{\kappa} = \frac{1}{N_{\mathcal{T}}} \sum_{t=1}^{N_{\mathcal{T}}} \kappa_t \tag{24}$$

The boundary conditions are evaluated for each boundary point $s \in \mathcal{S}$ as follows:

$$\mathcal{L}^s_{\text{BC},u} = u_s^{\text{CFD}} - u_s \tag{25}$$

$$\mathcal{L}^s_{\text{BC},w} = w_s^{\text{CFD}} - w_s \tag{26}$$

$$\mathcal{L}^s_{\text{BC},p} = p_s^{\text{CFD}} - p_s \tag{27}$$

$$\mathcal{L}^s_{\text{BC},Y_{\text{O}_2}} = Y_{\text{O}_2,s}^{\text{CFD}} - Y_{\text{O}_2,s} \tag{28}$$

**Table 2**
Loss weighting for Eq. (31). Note that the weights in the code $\lambda_l$ are the inverse of the weighting factors presented here $\lambda_l = \omega_l^{-1}$. This is because in the code the weighting is done by dividing the loss-Mean Squared Errors (MSEs) by the weights instead of multiplying them.

| Loss Component $l$ | Weighting factor $\omega_l$ |
|---|---|
| Con | $1 \cdot 10^{-1}$ |
| Mom, $x$ | $1 \cdot 10^{2}$ |
| Mom, $z$ | $1 \cdot 10^{2}$ |
| Spec | $1 \cdot 10^{-3}$ |
| BC, $u$ | $5 \cdot 10^{1}$ |
| BC, $w$ | 5 |
| BC, $p$ | $1 \cdot 10^{-4}$ |
| BC, $Y_{O_2}$ | $1 \cdot 10^{5}$ |

The physical losses and boundary losses for all of their respective data points are then squared and averaged using the Mean Squared Error (MSE). For the physical losses subsets of the collocation points $\mathcal{R}_{\text{Batch},k} \subseteq \mathcal{R}$, $k = 1, \dots, n$ with their respective number of data points $N_{\text{Batch},k} = |\mathcal{R}_{\text{Batch},k}|$ are used for the mini-batches during each epoch of the training with the Adam optimizer.

$$\mathcal{L}_l = \frac{1}{N_{\text{Batch},k}} \sum_{r=1}^{N_{\text{Batch},k}} \left(\mathcal{L}_l^r\right)^2, \tag{29}$$

with: $r \in \mathcal{R}_{\text{Batch},k}$, $k = 1, \dots, n$

Here, $l$ denotes the different losses. The optimization with L-BFGS uses the entire set of collocation data points, i.e., $\mathcal{R}_{\text{Batch}} = \mathcal{R}$.

During the entire training, the MSE of the boundary losses uses all boundary points $N_S = |S|$.

$$\mathcal{L}_{\text{BC},l} = \frac{1}{N_S} \sum_{s=1}^{N_S} \left(\mathcal{L}_{\text{BC},l}^s\right)^2 \tag{30}$$

By training with the boundary loss during each epoch and iteration of Adam and L-BFGS, respectively, the PINN is able to learn the correct solution more easily, since fulfilling the boundaries is a vital part of learning said solution.

At last, the total loss is determined as a weighted sum of the physical losses and the boundary losses:

$$\begin{aligned}\mathcal{L}_{\text{Tot}} = {} & \omega_{\text{Con}} \mathcal{L}_{\text{Con}} + \omega_{\text{Mom},x} \mathcal{L}_{\text{Mom},x} + \omega_{\text{Mom},z} \mathcal{L}_{\text{Mom},z} + \omega_{\text{Spec}} \mathcal{L}_{\text{Spec}} \\ & + \omega_{\text{BC},u} \mathcal{L}_{\text{BC},u} + \omega_{\text{BC},w} \mathcal{L}_{\text{BC},w} + \omega_{\text{BC},p} \mathcal{L}_{\text{BC},p} + \omega_{\text{BC},Y_{O_2}} \mathcal{L}_{\text{BC},Y_{O_2}}\end{aligned} \tag{31}$$

Here, weighting factors $\omega_l$ and $\omega_{BC,l}$ are introduced to balance the contributions of the individual loss components. Their values were determined heuristically by trial and error and are summarized in Table 2. In PINNs, individual loss components such as data, physical, and boundary losses can vary greatly in scale. Without proper weighting, some terms may dominate training, leading to an imbalance. Weighting factors $\omega$ balance these terms, promoting stable convergence and improved model accuracy.

Fig. 7 illustrates the training process for the entire gas channel, where the learning of CFD data is solely prescribed through the boundary conditions, as described above. The remaining collocation points are trained purely based on physics, relying on the minimization of conservation equation residuals. The imposed boundary conditions enable the PINN to learn the velocity gradients (in the vicinity of the cells walls) early on. This can be seen by the well captured boundary layer (dark blue) by iteration 70 (cf. Fig. 7(a). In contrast, the core flow requires more time during the training process to be accurately learned from the underlying physics. Furthermore, it is noticeable that the flow is first learned in the outlet region (epochs 150–2000 (cf. Figs. 7b and c), before the remaining inlet region is captured (up to epoch 18,000 (cf. Fig. 7d).

### 2.4.2. Electrochemical reaction modeling

The electrochemical reactions take place on the catalyst layer. Since only algebraic equations are used to model the electrochemical reactions, the entire CFD data set $\mathcal{T}_{\text{cat}}$ is used for the training. Similar to the process in the GC, Eqs. (10) and (12) are rewritten to evaluate to zero. The losses for all $t \in \mathcal{T}_{\text{cat}}$ then read:

$$\mathcal{L}_{\text{Surf}}^t = \eta_t + U_{\text{eq},t}^{\text{CFD}} + R_{\text{el},t}\, A_{\text{geom}}\, j_t \tag{32}$$

$$\mathcal{L}_{\text{Rea}}^t = \ln\left(\frac{j_t}{j_{0,t}}\right) - \gamma_{O_2} \ln\left(\frac{c_{O_2,t}^{\text{CFD}}}{c_{O_2,\text{ref}}}\right) - \frac{\alpha_a F \eta_t}{RT} \tag{33}$$

$$\mathcal{L}_{\text{Width}}^t = b_t^{\text{CFD}} - b_t \tag{34}$$

The loss containing the channel width $b$ (cf. Eq. (34)) is included to enforce the learning of the width dependencies on Eqs. (32) and (33) if more than one width is supplied during training. Since only the algebraic equations (cf. Eqs. 32–34) are used to train the PINN to accurately model $j$ and $\eta$, the other quantities, such as the equilibrium voltage $U_{\text{eq},t}^{\text{CFD}}$, exchange current density $j_{0,t}$, and oxygen concentration $c_{O_2,t}^{\text{CFD}}$ are directly supplied by CFD data. Eqs. (11), (14) and (A.1)–(A.3) are solved at each data point $t$ with the CFD data as well:

$$j_{0,t} = j_0^{\text{ref}} \left(\frac{p_{O_2,t}}{p_{\text{ref}}}\right)^{\gamma} \exp\left(-\frac{E}{RT}\left[1 - \frac{T}{T_{\text{ref}}}\right]\right) \tag{35}$$

$$p_{O_2,t} = p_t^{\text{CFD}}\, X_{O_2,t} \tag{36}$$

$$X_{O_2,t} = \frac{M_t}{M_{O_2}}\, Y_{O_2,t}^{\text{CFD}} \tag{37}$$

$$M_t = \left(\frac{Y_{H_2,t}^{\text{CFD}}}{M_{H_2}} + \frac{Y_{O_2,t}^{\text{CFD}}}{M_{O_2}} + \frac{Y_{H_2O,t}^{\text{CFD}}}{M_{H_2O}} + \frac{Y_{N_2,t}^{\text{CFD}}}{M_{N_2}}\right)^{-1} \tag{38}$$

$$R_{\text{el},t}(b_t, y_t) = a_5(b_t)\, y_t^5 + a_4(b_t)\, y_t^4 + a_3(b_t)\, y_t^3 + a_2(b_t)\, y_t^2 + a_1(b_t)\, y_t + a_0(b_t) \tag{39}$$

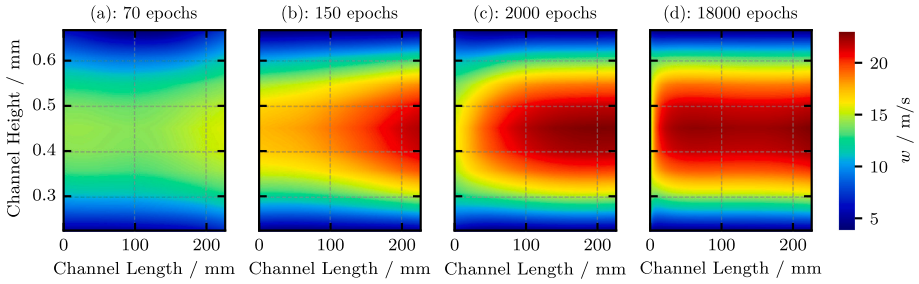



**Fig. 7.** Prediction of the main flow velocity $w$ by the PINN after (a) 70 epochs, (b) 150 epochs, (c) 2000 epochs, and (d) 18,000 epochs.

**Table 3**
Loss weighting for Eq. (41). Note that the weights in the code $\lambda_l$ are the inverse of the weighting factors presented here $\lambda_l = \omega_l^{-1}$. This is because in the code the weighting is done by dividing the loss-MSEs by the weighting factors instead of multiplying them.

| Loss Component $l$ | Weighting factor $\omega_l$ |
|---|---|
| Surf | 1 |
| Rea | 1 |
| Width | $1 \cdot 10^8$ |

**Table 4**
Best hyperparameter values found in the hyperparameter studies in Section 3.1 for both datasets.

| Hyperparameter | Value | |
|---|---|---|
| | Gas channel | Catalyst layer |
| Epochs Adam Optimizer | 12000 | 200 |
| Iterations L-BFGS Optimizer | 6090 | 50 |
| Batch size | 1000 | 1400 |
| Learning rate | $1 \cdot 10^{-3}$ | $1 \cdot 10^{-3}$ |
| Layers | 10 | 1 |
| Neurons | 15 | 5 |

Analogously to the GC, all losses are first squared and averaged using the MSE

$$\mathcal{L}_l = \frac{1}{N_{\text{Batch},k}} \sum_{t=1}^{N_{\text{Batch},k}} \left(\mathcal{L}_l^t\right)^2 \quad , \tag{40}$$

with: $t \in \mathcal{T}_{\text{Batch},k},\ k = 1, \dots, m$

with $\mathcal{T}_{\text{Batch},k} \subseteq \mathcal{T}_{\text{cat}}$ for $k = 1, \dots, m$ and $N_{\text{Batch},k} = |\mathcal{T}_{\text{Batch},k}|$. The equality holds for $\mathcal{T}_{\text{Batch}} = \mathcal{T}$ when using the L-BFGS optimizer. The Adam optimizer, again, works with mini-batches. The total loss combines the individual MSEs of the loss terms into a weighted sum:

$$\mathcal{L}_{\text{Tot}} = \omega_{\text{Surf}}\,\mathcal{L}_{\text{Surf}} + \omega_{\text{Rea}}\,\mathcal{L}_{\text{Rea}}\,\left(+\omega_{\text{Width}}\,\mathcal{L}_{\text{Width}}\right) \tag{41}$$

The term in parentheses is only included when predictions across multiple channel widths are performed, whereas it is omitted for reproduction cases. The determined weighting factors are summarized in Table 3. The comparatively large weighting factor of the channel width results from the different orders of magnitude of the involved variables. While the width itself is in the range of $10^{-4}$–$10^{-3}$ m, the prediction error is already small ($10^{-7}$–$10^{-8}$), whereas the residuals of the other equations are typically $10^{0}$–$10^{-1}$. Similar discrepancies occur for the gas channel boundary conditions, with pressure in the order of $10^{-4}$ and oxygen mass fraction around $10^{5}$. Hence, the scaling factors primarily reflect the physical magnitudes of the variables and the formulation of the equations.

For the catalyst layer, the PINN is trained using algebraic physical relations describing the electrochemical reactions. Some quantities entering these relations, such as the equilibrium potential and the oxygen concentration at the catalyst layer, are taken directly from CFD data. This is necessary since the corresponding transport processes inside the porous layers are not explicitly modeled in the present study.

## 3. Results

### 3.1. Hyperparameter studies

Parameters are considered hyperparameters of a Neural Network (NN) if they can be set by the modeler to change the behavior of the ANN. Two types of hyperparameters exist. Algorithm hyperparameters characterize the optimizer. They include the number of iterations, batch size, and learning rate. Model hyperparameters describe the topology of the ANN, namely the number of hidden layers and the neurons per hidden layer. Here, all of them are varied separately (cf. Fig. 8). Figs. 8(a) and (b) show the final loss and training time for PINN models of the GC and CL for different hyperparameter combinations.

The choice of hyperparameters does not only affect the training dynamics of an NN, but also strongly depends on the underlying physical modeling task. For example, training an PINN to approximate gas flow including species transport is considerably more demanding than modeling the Butler–Volmer kinetics in the catalyst layer, since transport equations involve more complex spatial relationships. Consequently, the optimal parameter set is problem-specific, and a universal configuration of hyperparameters cannot be defined.

Generally, a smaller learning rate (i.e., of the order of magnitude $10^{-3}$) increases the stability of the training. This is because too high values (i.e., of the order of magnitude 0.01) for the learning rate lead to the optimizer overshooting the optimal solution. Conversely, very low learning rates (i.e., of the order of magnitude $10^{-7}$) slow down convergence, since each step toward the optimal solution linearly decreases with the learning rate. Thus, the final loss can remain fairly high if the number of epochs and/or iterations is not increased simultaneously. The investigations further revealed that the optimal choice of learning rate depends on the size of the neural network (number of hidden layers and number of neurons per hidden layer). Regarding the batch size, the largest possible value can be chosen depending on the available hardware (GPU).

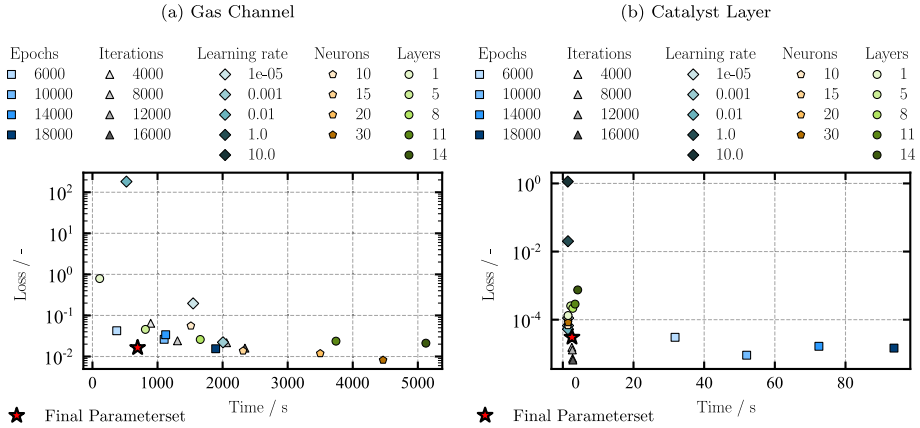



**Fig. 8.** Hyperparameter Variation of (a) GC and (b) CL.

Dividing the data into equally sized mini-batches has a strongly positive effect on the stability of the training process. Finally, the training duration is chosen such that the loss function reaches a desired level of convergence.

Varying the model parameters in the gas channel model and the catalyst model affects not only the loss value but also the training time, which must be considered when selecting the respective parameters. For the modeling of the gas channel, increasing the number of hidden layers from 8 to 11 results in an additional training time $t_{\text{train}}$ of about 35 min without improving the model accuracy.

The corresponding optimal hyperparameter values are listed in Table 4. The corresponding model behavior with respect to the loss value after the end of training is indicated by the red star in Figs. 8(a) and (b). A comparison of the hyperparameters of the two models reveals that, due to the lower physical complexity of the catalyst layer, considerably less training effort is required, and the PINN architecture can be much smaller while still accurately capturing and learning the underlying physical problem. This deliberate reduction in network depth and width off between model accuracy and computational cost: the simpler physical formulation of the catalyst layer allows for a compact architecture without loss of predictive accuracy, whereas the higher-dimensional and strongly coupled transport problem in the gas channel necessitates deeper and wider networks to achieve comparable accuracy.

In addition, careful coordination of the use of the two optimizers is required. While the Adam optimizer is highly parallelizable due to the distribution of individual batches on the GPU, the L-BFGS optimizer can significantly increase the training time if not weighted optimally, since it relies solely on the CPU for training.

It is important to note that both the PINN and the ANN were trained for the same number of epochs and iterations. In practice, fewer epochs and iterations would likely suffice for the ANN, since its loss function only considers the discrepancy with the reference data and does not include multiple coupled equations as in the PINN. Furthermore, each iteration of the ANN is computationally faster, resulting in total training times that are approximately three times shorter during the Adam phase and 4.5 times faster during L-BFGS fine-tuning.

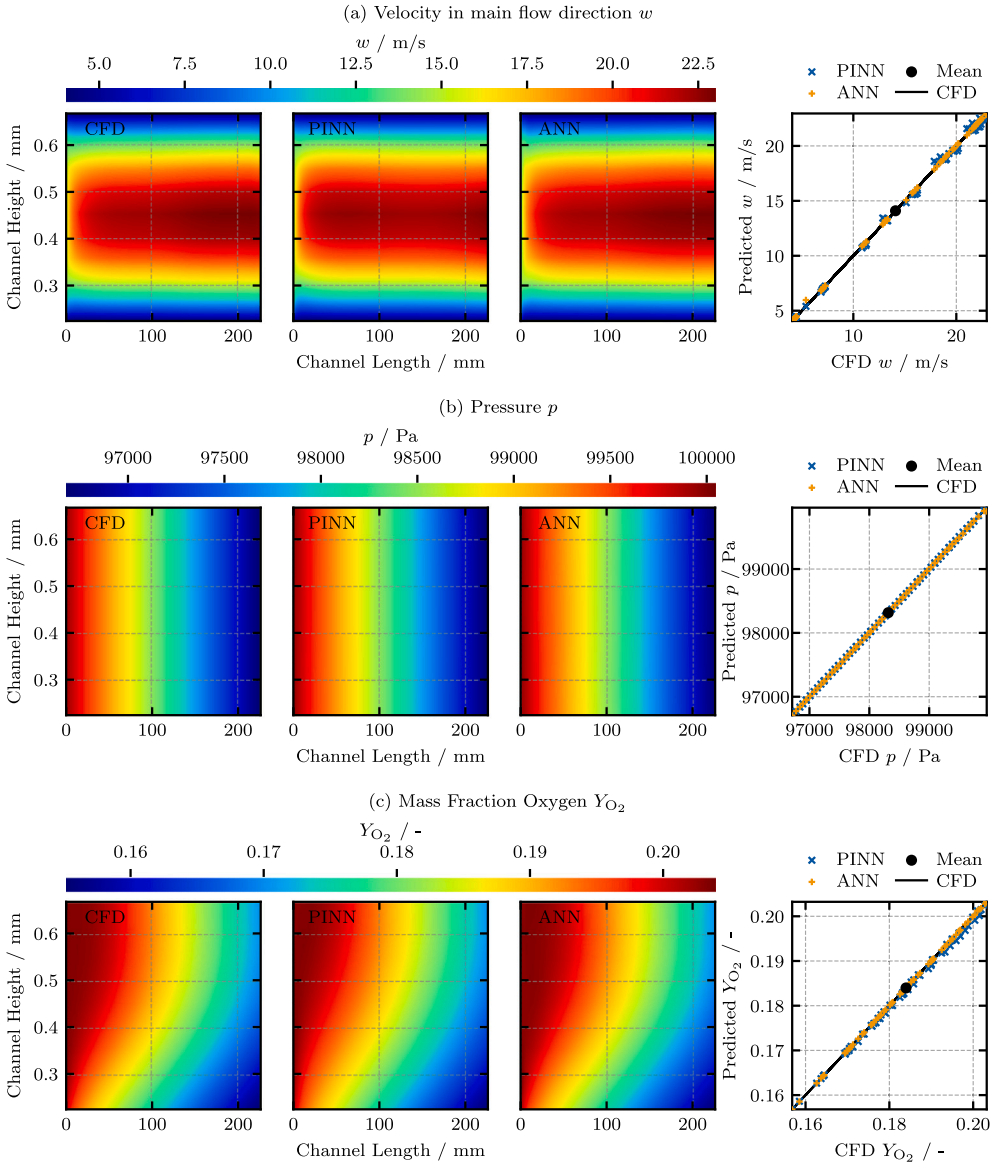



**Fig. 9.** Ground truth (CFD data) and PINN predictions for the (a) main flow velocity $w$, (b) pressure $p$, and (c) oxygen mass fraction $Y_{O_2}$ represented by a color mesh and a parity plot inside the gas channel. Note that for the parity plot every 75th data point is shown to enhance the visibility.

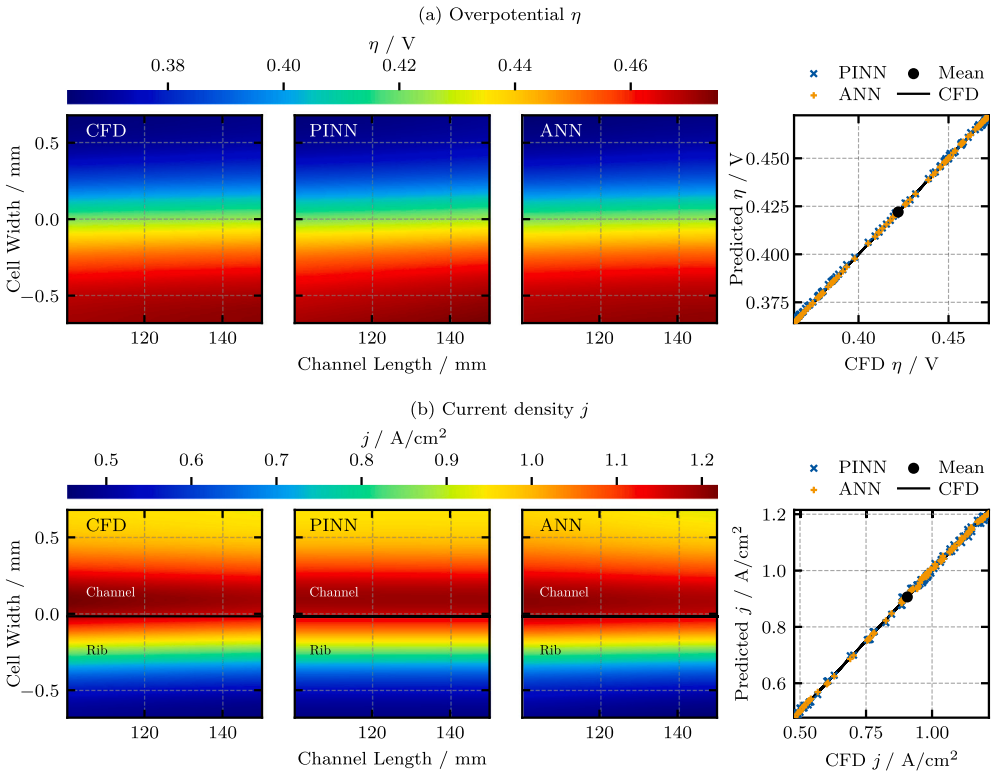



**Fig. 10.** Ground truth (CFD data) and PINN predictions for the (a) overpotential $\eta$ and (b) current density $j$ represented by a color mesh and a parity plot inside the gas channel. Note that for the parity plot every 75th data point is shown to enhance the visibility.

### 3.2. Reproductive power of the PINN and ANN

In a first step, the PINN is trained on CFD data of a single channel width to demonstrate its learning capabilities.

Fig. 9 shows the main flow velocity $w$, pressure $p$, and oxygen mass fraction $Y_{O_2}$ for the CFD reference data, as well as the predictions from the PINN and a conventional ANN. Parity plots for all three quantities are also shown, displaying every 75th data point to improve visibility. The training data corresponds to an operating point with a constant potential of 0.6 V applied at the end plates of the Fuel Cell (FC).

Both models were initially trained for 12,000 epochs using the Adam optimizer with a batch size of 1000. Subsequently, they were fine-tuned using the L-BFGS optimizer for 6090 iterations. Overall, all three quantities are reproduced accurately by both models.

Along the channel length, the characteristic velocity profile in the main flow direction develops, with the velocity approaching zero at the boundaries due to the no-slip condition. A velocity gradient is also observed in the direction of the GDL, as diffusive species transport dominates in this porous medium. In addition, a nearly linear pressure drop of approximately $\Delta p = 30\,\text{mbar}$ is identified along the channel. Furthermore, the distribution of the oxygen mass fraction is shown exemplarily. At the channel inlet, the oxygen mass fraction is about $Y_{O_2} = 0.21$, decreasing to approximately $Y_{O_2} = 0.15$ at the channel outlet near the GDL. A pronounced gradient of the mass fraction in the channel height direction can also be observed. Both effects can be attributed to oxygen consumption at the catalyst layer and are well reproduced by both the PINN and the ANN.

Similarly, Fig. 10 presents the results of the catalyst layer modeling, shown in terms of the overpotential $\eta$ and the current density $j$. The same operating point as for the gas channel modeling is used here, corresponding to a constant potential of 0.6 V applied at the end plates of the fuel cell. For this purpose, the corresponding equations Eqs. (32) and (33) are included as part of the loss function.

**Table 5**
Training time $t_{\text{train}}$ comparison of PINN and ANN for a single channel width on a NVIDIA RTX 5090 (Adam Optimizer) and an Intel i9-14900K (L-BFGS Optimizer). Minor differences in training time are attributed to optimizer convergence behavior.

| | PINN | ANN |
|---|---|---|
| Gas channel | 717.60 s | 185.85 s |
| Catalyst layer | 1.72 s | 4.96 s |

The predicted overpotential (cf. Fig. 10a) exhibits a strong correlation with the previously discussed distribution of the electrical resistance in Fig. 3(a). However, owing to the simplification along the main flow direction, no pronounced gradient occurs in this case. A slight gradient remains visible in the $z$-direction for the overpotential, whereas the current density reveals an even less pronounced variation.

In Fig. 10(b) a significantly higher current density can be observed beneath the channel. This is due to the considerably better supply of reactants to the catalyst layer in this region. Since the transport of reactants through the porous layers is predominantly diffusive, the regions beneath the ribs are much more difficult to supply with reactants compared to those beneath the channels, owing to the longer diffusion paths. Toussaint et al. [38]

Overall, both the PINN and the ANN are highly capable of reproducing the mechanisms within the catalyst layer, as can be quantitatively inferred from the parity plots in Fig. 10.

Even though both the PINN and the ANN achieve comparable reproductive accuracy, the training time of the PINN is significantly higher, as shown in Table 5. The increased training time of the PINN primarily arises from additional computations, namely the calculation of derivatives through automatic differentiation.

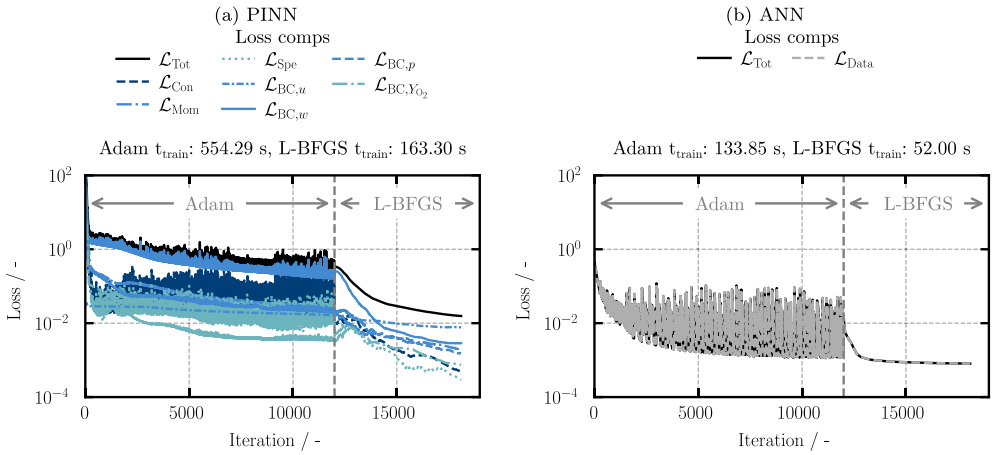



**Fig. 11.** Comparison of training time and loss curve of different loss shares for reproduction of gas channel.

**Table 6**
Training and evaluation channel widths used for assessing the generalization behavior of the ANN and PINN models. Channel widths marked as *train* were used for model training, while all others represent unseen cases evaluated within the training envelope.

| Case | Channel width $b$ [mm] | | | | | | |
|---|---|---|---|---|---|---|---|
| | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 1.0 | 1.1 |
| Two training points | train | | | | | | train |
| Three training points | train | | | train | | | train |

While the PINN requires additional computations, it offers a decisive advantage over the ANN: the gas channel can be modeled using collocation points, without relying on training data from CFD simulations. For that purpose, analytical boundary conditions, such as the no-slip condition, need to be integrated. The ANN, however, inherently requires CFD data for its direct comparison in the loss function.

Comparing the losses of both the PINN and ANN as a function of the iterations in Fig. 11 shows instabilities in both models. This can be seen by the fluctuation of the total loss $\mathcal{L}_{Tot}$ during the optimization with the Adam optimizer. They are more pronounced in the ANN. The subsequent optimization with L-BFGS shows no instability. For the ANN model, this is a clear sign that the learning rate is in fact too high. However, for the PINN model this might be attributable to the PINN trying to balance different physical and boundary losses to achieve consistency among the coupled equations.

### 3.3. Predictive power of the PINN and ANN

In the next step, the PINN and the ANN are trained on CFD data of multiple channel widths. Since the channel width strongly affects fuel cell performance [38], this case exemplarily highlights the potential of comparing both models with respect to their ability to generalize beyond the trained data. Hence, the overarching goal is to incorporate NNs into the design process by first training them on some channel widths and profiting from their negligible prediction time for unseen channel widths afterwards. Thus, an overall reduction in computational costs and/or time may be achievable.

For the present investigation, both the PINN and the ANN models of the catalyst layer are employed to predict the overall FC performance in potentiostatic operation at 0.6 V under variation of the channel width.

All evaluated channel widths lie within the training envelope of the respective training widths (see Table 6). The reported results therefore assess interpolation and generalization within the training envelope rather than extrapolation beyond the trained parameter range.

Two cases are distinguished. In both cases, the ANN is trained using exactly the same channel widths as the PINN, ensuring a fair comparison of their generalization behavior under identical training conditions. In Fig. 12, the models are trained for channel widths of $b_{train} = 0.4$ mm and $b_{train} = 1.1$ mm. Subsequently, the prediction is extended to all channel widths within this interval in order to assess the interpolation and generalization performance for unseen cases within the training envelope. The figures display the validated CFD reference data (black) together with the model predictions, where the PINN is shown in blue and the ANN in orange, exemplarily for the overpotential $\eta$ (Fig. 12a) and the current density $j_{VC}$ (Fig. 12b).[1] For the unseen channel widths, a scattering of the model predictions can be observed as shaded areas in the colors of their respective models. For each model, 20 independent training runs with different random initializations were performed. The shaded regions therefore represent the envelope defined by the minimum and maximum predicted values across these 20 realizations for each channel width independently. All RMSE values are evaluated after de-normalization of the predicted quantities and are computed over all investigated channel widths shown in the respective figure. The shaded areas were determined by first combining the predictions of all models for each channel width. Then, the minimum and maximum values for each channel width are determined. Lastly, the shaded area is added between the two boundaries that connect the minimum and maximum values for all channel widths, respectively. Thus, the shaded area only illustrates what value range can be expected for each prediction as a function of the channel width. For the case with two training channel widths, both models are able to capture the overall trend of the fuel cell performance. However, differences in the generalization behavior become apparent: while the ANN achieves a slightly lower error for the overpotential, it exhibits larger deviations and a wider prediction range for the current density compared to the PINN. This indicates a less consistent generalization across different physical output quantities when only limited training data are available.

The observed scatter in the prediction results can be attributed to the initialization of the network weights. In this work, the weights are initialized using the Glorot scheme, which samples from a uniform probability distribution. Consequently, each model starts from a different set of initial weights. During training, these different starting points lead the models to converge to distinct local minima, as the optimization is intentionally stopped before the global minimum is reached. While convergence toward the global minimum would eventually yield identical

[1] The current density $j_{VC}$ is normalized to the area of the anodic voltage collector with $j_{VC} = j \cdot A_{geom}/A_{VC}$ with $A_{VC} = 1.5778 \cdot 10^{-4}$ m$^2$.

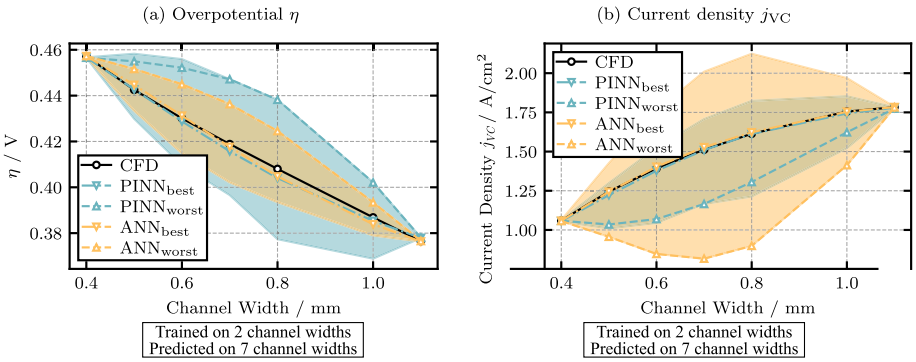



**Fig. 12.** Prediction of fuel cell performance at 0.6 V for varying channel widths $b_{train} \in [0.4, 1.1]$ mm using a PINN (blue) and an ANN (orange), both trained only at $b_{train} = 0.4$ mm and $b_{train} = 1.1$ mm. Shown are the overpotential (a) $\eta$ (RMSE ANN = 2.155E-3; RMSE PINN = 2.004E-3) and (b) the current density $j_{VC}$ (RMSE ANN = 9.29E-3; RMSE PINN = 11.40E-3) compared to validated CFD reference data (black). Shaded areas indicate the prediction range between the best and worst performing case of each model for unseen channel widths.

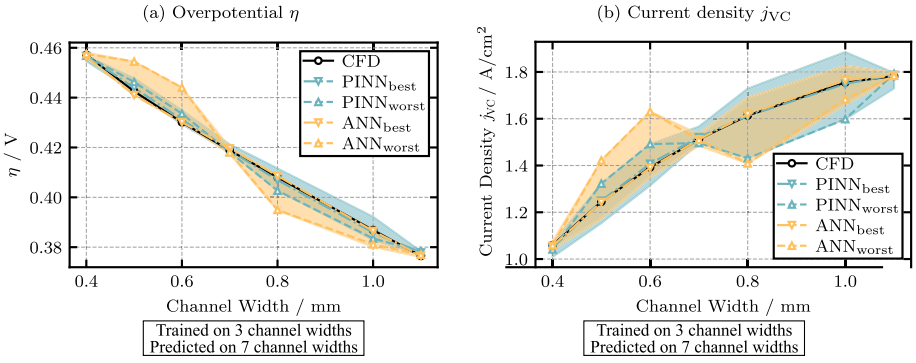



**Fig. 13.** Prediction of fuel cell performance at 0.6 V for varying channel widths $b_{train} \in [0.4, 1.1]$ mm using a PINN (blue) and an ANN (orange), both trained only at $b_{train} = 0.4$ mm, $b_{train} = 0.7$ mm and $b_{train} = 1.1$ mm. Shown are the (a) overpotential $\eta$ (RMSE ANN = 1.15E-3; RMSE PINN = 0.97E-3) and the (b) current density $j_{VC}$ (RMSE ANN = 8.63E-3; RMSE PINN = 6.59E-3) compared to validated CFD reference data (black). Shaded areas indicate the prediction range between the best and worst performing case of each model for unseen channel widths.

results across all models, this regime would lie in the range of overfitting and is therefore avoided in the present study.

Overall, the best-performing models show a high prediction accuracy for the respective optimal case. In addition, the prediction intervals reveal that the scatter for the PINN model is narrower in the current density prediction compared to the ANN.

Analogously, the case shown in Fig. 13 is examined for three training points, including an additional intermediate channel width of $b_{train} = 0.7$ mm. This results in an improved prediction accuracy, with the scatter of the predictions becoming significantly narrower. When three channel widths are used for training, both the ANN and the PINN show a similarly high prediction accuracy and a strongly reduced scatter. This indicates that, given sufficient training data, both approaches exhibit comparable generalization performance within the investigated parameter range.

Overall, these results show that both ANN and PINN models can generalize well when trained on a sufficiently representative set of channel widths. However, when the amount of training data is significantly reduced, the PINN demonstrates a more robust and consistent generalization behavior across multiple physical quantities, highlighting its advantage in data-scarce scenarios.

Using three training channel widths was found to provide a favorable balance between prediction accuracy and computational cost, yielding narrow prediction intervals while still requiring only a small number of CFD simulations compared to a full parametric study.

Fig. 14 provides a comparative assessment of the computational effort required for the exemplary design study of channel width using PINNs and ANNs. It should be noted that Fig. 14 reports the cumulative computational effort for the complete channel width study. The evaluation was performed on an Intel i9-14900K CPU combined with an Nvidia RTX 5090 GPU. For the selected range of channel widths between 0.4 mm and 1.1 mm, the CFD simulations require the calculation of seven supporting points to determine the current density, resulting in a total computational time of 222.6 min. When using an ANN, CFD data for three representative points are required for training, and the resulting computation time is reduced by 52.9 %. In contrast, PINNs do not rely on precomputed CFD data but are trained using collocation points based on the underlying physics, further reducing the computational effort by 65.8 % compared to the ANN. Overall, the application of PINNs achieves a total reduction of 83.9% in computational time relative to the full CFD calculations. This demonstrates the high potential of AI, and PINNs in particular, to significantly reduce computational effort, as exemplarily shown for channel width. The approach is naturally transferable to other geometric parameters of a fuel cell channel, indicating substantial potential for reducing computational time across the entire channel design process.

## 4. Limitations of the proposed PINN framework

While the proposed PINN framework demonstrates strong potential for accelerating PEMFC channel design, several limitations must be acknowledged. First, the present implementation is restricted to a two-dimensional representation of the gas channel. Although suitable for evaluating feasibility and generalization behavior, full three-dimensional modeling will be required to capture secondary flow structures, corner effects, and realistic channel cross-sectional features.

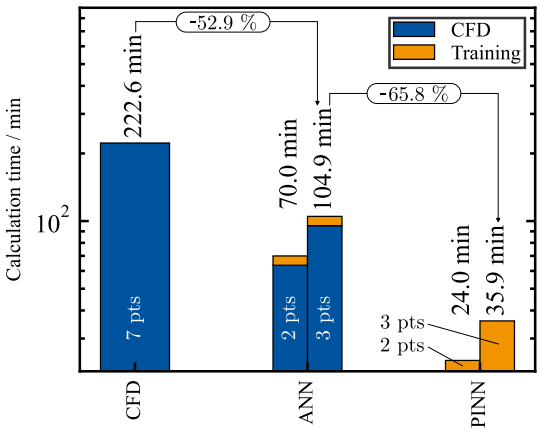



**Fig. 14.** Comparison of calculation duration for CFD, PINN and ANN for full channel with variation.

Second, a degree of prediction scatter was observed when applying the PINN to unseen channel widths. This variability results from random initialization of network weights, which leads individual training runs to converge to different local minima. More robust prediction strategies such as ensemble averaging, weight-space sampling, or deterministic initialization schemes may mitigate this effect in future work.

Third, the PINN performance depends on manually selected weighting factors in the loss function. Since different physical residuals may differ in magnitude by several orders, inappropriate weighting can introduce training imbalance or slow convergence. Adaptive loss-balancing strategies—such as gradient normalization, uncertainty-based weighting, or inverse-Dirichlet methods—represent a promising direction for increasing training stability and model robustness.

Finally, while the present work focuses on gas-phase transport, further extensions are required to represent liquid water, two-phase flow, and electrochemical transport coupling. Incorporating these mechanisms will increase numerical stiffness and computational cost, potentially requiring domain decomposition approaches such as XPINNs or multi-fidelity hybrid training.

## 5. Conclusion

In this work, the potential of PINNs for supporting the design process of fuel cell geometries, with a particular focus on channel dimensioning, has been investigated. A PINN framework was developed to model channel flow as a multicomponent mixture with special emphasis on oxygen consumption. For the reproduction of CFD data, PINNs were compared to conventional ANNs. Although the training effort for PINNs is higher, they offer a decisive advantage, as they do not require CFD-based training data and can therefore be applied in cases where such data is not readily available. Furthermore, a representative design application was demonstrated by varying the channel width, a key geometric design parameter, and analyzing its impact on the current density distribution. The results show that, with three training support points, PINNs achieve accurate predictions and exhibit strong potential as a powerful tool for fuel cell channel design and optimization.

This study answers the central research question formulated in the introduction: *Can a physics-informed neural network accurately predict spatially resolved flow and species fields in PEMFC gas channels for geometries not included during training, and can it do so with substantially reduced computational cost compared to CFD?* The findings confirm that this is indeed the case. The developed PINN generalizes reliably across different channel widths, enabling accurate predictions for unseen geometries. Furthermore, the achieved computational savings of approximately 83.9% compared to full CFD simulations directly address the second part of the research question—demonstrating that the method significantly accelerates fuel-cell design iterations.

Compared to previous PEMFC surrogate modeling approaches based on ANNs, ANFIS models, or PINN studies in other electrochemical domains, the present work provides three key advances. First, unlike conventional data-driven surrogates that require extensive CFD or experimental labels for each geometry, the proposed PINN embeds the governing transport equations directly, enabling accurate predictions without labelled training data. Second, in contrast to existing PINN applications that focus on simplified or fixed geometries, our model demonstrates generalization across different channel widths, addressing a limitation repeatedly highlighted in prior work. Third, the achieved computational savings of approximately 83.9% relative to full CFD simulations indicate that the PINN surrogate is well suited for iterative bipolar-plate and channel design, where previous methods remain too slow or insufficiently flexible. These improvements position the proposed framework as a significant step beyond the state of the art in PEMFC channel-flow modeling.

Overall, the results demonstrate that the proposed PINN framework provides (i) accurate reproduction of gas-channel and catalyst-layer physics, (ii) reliable prediction capabilities across unseen geometries, (iii) strong robustness with reduced prediction scatter compared to ANNs, and (iv) substantial acceleration of the design workflow. Through this aggregation of results, it becomes clear that PINNs offer a unique combination of physical consistency, generalization capability, and computational efficiency that conventional data-driven methods cannot achieve.

To further mature this approach toward series-ready application in industrial fuel-cell development, several research directions are proposed. First, the presented PINN should be embedded into a broader design framework capable of evaluating multiple channel geometries in rapid succession, thereby enabling full geometry screening at significantly reduced computational cost. Second, the PINN architecture can be extended to form a modular surrogate model for entire flow-field design families, allowing the evaluation of complex variations in channel patterns and rib–channel arrangements. Third, incorporating liquid-water transport into the PINN formulation represents a highly impactful extension, as two-phase CFD simulations are extremely time-consuming. A physics-informed surrogate for liquid-water formation, transport, and removal would offer dramatic additional computational savings. Fourth, future studies may integrate PINNs with multi-objective optimization loops to derive application-specific channel architectures tailored to automotive, heavy-duty, or stationary PEMFC systems.

By addressing these aspects, future developments can leverage the strengths of PINNs even further and contribute to the establishment of fast, flexible, and physically accurate surrogate models for next-generation PEMFC flow-field design.

## CRediT authorship contribution statement

**Julian Nicolas Toussaint:** Writing – original draft, Project administration, Methodology, Investigation, Formal analysis, Conceptualization. **Sebastian Pieper:** Writing – original draft, Software, Methodology, Investigation, Formal analysis. **Max Paul Mally:** Writing – review & editing, Resources, Methodology, Conceptualization. **Stefan Pischinger:** Writing – review & editing, Supervision, Funding acquisition.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgments

This work was conducted within the publicly funded *BiFoilStack* project, supported by the German Federal Ministry for Economic Affairs and Energy (BMWi) under grant number 03EN5018A–D. The authors

gratefully acknowledge the computing time provided to them at the NHR Center NHR4CES at RWTH Aachen University (project number p0024400). This is funded by the Federal Ministry of Education and Research, and the state governments participating on the basis of the resolutions of the GWK for national high performance computing at universities (www.nhr-verein.de/unsere-partner).

## Appendix A

### A.1. Calculation of partial pressure $p_{O_2}$

$$p_{O_2} = p\, X_{O_2}. \tag{A.1}$$

To calculate the molar fraction of oxygen $X_{O_2}$ from its mass fraction $Y_{O_2}$, the molar mass of oxygen $M_{O_2}$ and the other gaseous species are needed to compute the molar mass of the mixture $M$:

$$X_{O_2} = \frac{M}{M_{O_2}} Y_{O_2} \tag{A.2}$$

$$M = \left( \frac{Y_{H_2}}{M_{H_2}} + \frac{Y_{O_2}}{M_{O_2}} + \frac{Y_{H_2O}}{M_{H_2O}} + \frac{Y_{N_2}}{M_{N_2}} \right)^{-1} \tag{A.3}$$

### A.2. Data tables

See Table A.8.

**Table A.7**
Coefficients for the resistance polynomial in Eq. (14) for the different channel widths $b$.

| Coefficient | Channel width $b$ / mm | | |
|---|---|---|---|
| | 0.4 | 0.5 | 0.6 |
| $a_5$ | $2.76898093 \cdot 10^{15}$ | $+1.68754465 \cdot 10^{15}$ | $+1.21415834 \cdot 10^{15}$ |
| $a_4$ | $-1.51495140 \cdot 10^{12}$ | $-1.12139495 \cdot 10^{12}$ | $-9.86853937 \cdot 10^{11}$ |
| $a_3$ | $-1.31711777 \cdot 10^{9}$ | $-8.80116928 \cdot 10^{8}$ | $-7.44716147 \cdot 10^{8}$ |
| $a_2$ | $+1.69314798 \cdot 10^{6}$ | $+1.22307006 \cdot 10^{6}$ | $+1.00250420 \cdot 10^{6}$ |
| $a_1$ | $-9.85930864 \cdot 10^{2}$ | $-5.38962396 \cdot 10^{2}$ | $-2.48220767 \cdot 10^{2}$ |
| $a_0$ | $+5.82312024 \cdot 10^{-1}$ | $+4.44783212 \cdot 10^{-1}$ | $+3.71842443 \cdot 10^{-1}$ |
| Coefficient | Channel width $b$ / mm | | |
| | 0.687388 | 0.7 | 0.8 |
| $a_5$ | $+8.64744578 \cdot 10^{14}$ | $+1.11164587 \cdot 10^{15}$ | $+1.14366483 \cdot 10^{15}$ |
| $a_4$ | $-1.06028635 \cdot 10^{12}$ | $-8.86318163 \cdot 10^{11}$ | $-7.37359582 \cdot 10^{11}$ |
| $a_3$ | $-5.68186036 \cdot 10^{8}$ | $-8.05160160 \cdot 10^{8}$ | $-9.30252751 \cdot 10^{8}$ |
| $a_2$ | $+9.52843858 \cdot 10^{5}$ | $+8.43310367 \cdot 10^{5}$ | $+6.83902270 \cdot 10^{5}$ |
| $a_1$ | $-1.14282870 \cdot 10^{2}$ | $-1.93119690 \cdot 10^{1}$ | $+1.70571364 \cdot 10^{2}$ |
| $a_0$ | $+2.70007141 \cdot 10^{-1}$ | $+3.35756103 \cdot 10^{-1}$ | $+3.23970437 \cdot 10^{-1}$ |
| Coefficient | Channel width $b$ / mm | | |
| | 1.0 | 1.1 | |
| $a_5$ | $+8.92347961 \cdot 10^{14}$ | $+6.44801872 \cdot 10^{14}$ | |
| $a_4$ | $-3.49723869 \cdot 10^{11}$ | $-2.13204872 \cdot 10^{11}$ | |
| $a_3$ | $-9.30274269 \cdot 10^{8}$ | $-8.05319343 \cdot 10^{8}$ | |
| $a_2$ | $+3.51912309 \cdot 10^{5}$ | $+2.25704794 \cdot 10^{5}$ | |
| $a_1$ | $+4.25263459 \cdot 10^{2}$ | $+4.99855110 \cdot 10^{2}$ | |
| $a_0$ | $+3.50423345 \cdot 10^{-1}$ | $+3.79098561 \cdot 10^{-1}$ | |

**Table A.8**
Parameter values.

| Parameter | Value | Source |
|---|---|---|
| $E_C$ | 72.4 kJ/mol | [63,64] |
| $j_0^{ref}$ | 6.97872 A/cm$^2$ | |
| $\alpha_a$ | 0.62 | |
| $A_{geom}$ | $3.1114 \cdot 10^{-4}$ m$^2$ | |
| $\bar{\rho}$ | 1.793 kg/m$^3$ | |
| $\bar{\mu}$ | $1.759 \cdot 10^{-5}$ Pa s | |
| $\bar{D}_{O_2}$ | $9.809 \cdot 10^{-6}$ m$^2$/s | |

## Data availability

The developed PINN and ANN models are available via open-access [62].

## References

[1] European Commission. Regulation (EU) 2019/631 of the european parliament and of the council of 17 April 2019 setting co2 emission performance standards for new passenger cars and for new light commercial vehicles, and repealing regulations (EC) no 443/2009 and (EU) no 510/2011; 2025. https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A02019R0631-20250709.
[2] Council of the European Union. Heavy-duty vehicles: council signs off on stricter co2 emission standards; 2024. https://www.consilium.europa.eu/en/press/press-releases/2024/05/13/heavy-duty-vehicles-council-signs-off-on-stricter-co2-emission-standards/
[3] Cullen DA, Neyerlin KC, Ahluwalia RK, Mukundan R, More KL, Borup RL, Weber AZ, Myers DJ, Kusoglu A. New roads and challenges for fuel cells in heavy-duty transportation. Nature Energy 2021;6(5):462–74. https://doi.org/10.1038/s41560-021-00775-z
[4] Emran A, Mertes S, Garg S, Wick M, Walters M, Sharma V. Fuel cell electric vehicles - an optimal solution for future electrification of heavy commercial vehicles in India. In: 2021 IEEE transportation electrification conference (ITEC-india). IEEE; 2021. p. 1–8. https://doi.org/10.1109/ITEC-India53713.2021.9932475
[5] Toussaint JN, Mertes S, Schmidt M, Sterlepper S, Walters M, Pischinger S. Development of a cloud-based toolchain for data science in fuel cell vehicles. J Power Sources 2025;648:237276. https://doi.org/10.1016/j.jpowsour.2025.237276
[6] O'Hayre RP, Cha S-W, Colella WG, Prinz FB. Fuel cell fundamentals. 3rd ed. Hoboken, New Jersey: Wiley; 2016.
[7] Klell M, Eichlseder H, Trattner A. Wasserstoff in DER fahrzeugtechnik. Wiesbaden: Springer Fachmedien Wiesbaden; 2018. https://doi.org/10.1007/978-3-658-20447-1
[8] Kurzweil P. Brennstoffzellentechnik. Wiesbaden: Springer Fachmedien Wiesbaden; 2016. https://doi.org/10.1007/978-3-658-14935-2
[9] Wang J, Wang H, Fan Y. Techno-economic challenges of fuel cell commercialization. Engineering 2018;4(3):352–60. https://doi.org/10.1016/j.eng.2018.05.007
[10] Brooks AN, Hughes TJR. Streamline upwind/petrov-galerkin formulations for convection dominated flows with particular emphasis on the incompressible navier-stokes equations. Comput Methods Appl Mech Eng 1982;32(1–3):199–259. https://doi.org/10.1016/0045-7825(82)90071-8
[11] Karniadakis G, Sherwin S. Spectral/hp element methods for computational fluid dynamics. Oxford University Press; 2005. https://doi.org/10.1093/acprof:oso/9780198528692.001.0001
[12] Ferziger JH, Perić M, Street RL. Computational methods for fluid dynamics. 4th ed. Cham: Springer; 2020.
[13] Haykin SS. Neural networks: a comprehensive foundation. 2nd ed. Upper Saddle River, NJ: Prentice-Hall; 1999.
[14] Raissi M, Perdikaris P, Karniadakis GE. Physics informed deep learning (part I): Data-driven solutions of nonlinear partial differential equations. arXiv:1711.10561. 2017. https://arxiv.org/pdf/1711.10561
[15] Raissi M, Perdikaris P, Karniadakis GE. Physics informed deep learning (part II): Data-driven discovery of nonlinear partial differential equations. arXiv:1711.10566. 2017. http://arxiv.org/pdf/1711.10566v1
[16] Raissi M. Deep hidden physics models: Deep learning of nonlinear partial differential equations. arXiv:1801.06637. 2018. http://arxiv.org/pdf/1801.06637v1
[17] Raissi M, Perdikaris P, Karniadakis GE. Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J Comput Phys 2019;378:686–707. https://doi.org/10.1016/j.jcp.2018.10.045
[18] Zerrougui I, Li Z, Hissel D. Physics-informed neural network for modeling and predicting temperature fluctuations in proton exchange membrane electrolysis. Energy and AI 2025;20:100474. https://doi.org/10.1016/j.egyai.2025.100474
[19] Chen Y-C, Karageorgiou C, Eller J, Schmidt TJ, Büchi FN. Determination of the porosity and its heterogeneity of fuel cell microporous layers by x-ray tomographic microscopy. J Power Sources 2022;539:231612. https://doi.org/10.1016/j.jpowsour.2022.231612
[20] Chen H, Kätelhön E, Compton RG. Predicting voltammetry using physics-informed neural networks. J Phys Chem Lett 2022;13(2):536–43. https://doi.org/10.1021/acs.jpclett.1c04054
[21] Ko T, Kim D, Park J, Lee SH. Physics-informed neural network for long-term prognostics of proton exchange membrane fuel cells. Appl Energy 2025;382:125318. https://doi.org/10.1016/j.apenergy.2025.125318
[22] Rezazadeh S, Mehrabi M, Pashaee T, Mirzaee I. Using adaptive neuro-fuzzy inference system (anfis) for proton exchange membrane fuel cell (pemfc) performance modeling. J Mech Sci Technol 2012;26(11):3701–9. https://doi.org/10.1007/s12206-012-0844-2
[23] Wilberforce T, Olabi AG. Performance prediction of proton exchange membrane fuel cells (pemfc) using adaptive neuro inference system (anfis). Sustainability 2020;12(12):4952. https://doi.org/10.3390/su12124952
[24] Biswas M, Wilberforce T, Biswas MA. Prediction of transient hydrogen flow of proton exchange membrane electrolyzer using artificial neural network. Hydrogen 2023;4(3):542–55. https://doi.org/10.3390/hydrogen4030035

[25] Wilberforce T, Biswas M, Omran A. Power and voltage modelling of a proton-exchange membrane fuel cell using artificial neural networks. Energies 2022;15(15):5587. https://doi.org/10.3390/en15155587
[26] Legala A, Zhao J, Li X. Machine learning modeling for proton exchange membrane fuel cell performance. Energy and AI 2022;10:100183. https://doi.org/10.1016/j.egyai.2022.100183
[27] Wilberforce T, Olabi AG. Proton exchange membrane fuel cell performance prediction using artificial neural network. Int J Hydrogen Energy 2021;46(8):6037–50. https://doi.org/10.1016/j.ijhydene.2020.07.263
[28] Saengrung A, Abtahi A, Zilouchian A. Neural network model for a commercial PEM fuel cell system. J Power Sources 2007;172(2):749–59. https://doi.org/10.1016/j.jpowsour.2007.05.039
[29] Zheng L, Hou Y, Zhang T, Pan X. Performance prediction of fuel cells using long short–term memory recurrent neural network. Int J Energy Res 2021;45(6):9141–61. https://doi.org/10.1002/er.6443
[30] Tang Y, Huang X, Li Y, Ma H, Zhang K, Song K. Degradation prediction of proton exchange membrane fuel cell based on multi-head attention neural network and transformer model. Energies 2025;18(12):3177. https://doi.org/10.3390/en18123177
[31] Pan H, Zou Y, Sun X, Fu J. An enhanced neural network model for predicting the remaining useful life of proton exchange membrane fuel cells. Scientific reports 2025;15(1):37320. https://doi.org/10.1038/s41598-025-21247-4
[32] Jin X, Cai S, Li H, Karniadakis GE. Nsfnets (navier-stokes flow nets): physics-informed neural networks for the incompressible navier-stokes equations. J Comput Phys 2021;426:109951. https://doi.org/10.1016/j.jcp.2020.109951
[33] Cai K, Wang J. Physics-informed neural networks for solving incompressible navier–stokes equations in wind engineering. Phys Fluids 2024;36(12). https://doi.org/10.1063/5.0244094
[34] He Q, Tartakovsky AM. Physics-informed neural network method for forward and backward advection-dispersion equations. arXiv:2012.11658. 2020. https://doi.org/10.48550/ARXIV.2012.11658
[35] Frerichs-Mihov D, Henning L, John V. On loss functionals for physics-informed neural networks for steady-state convection-dominated convection-diffusion problems. Commun Appl Math Comput 2024. https://doi.org/10.1007/s42967-024-00433-7
[36] Yang Z, Xu Y, Jing J, Fu X, Wang B, Ren H, Zhang M, Sun T. Investigation of physics-informed neural networks to reconstruct a flow field with high resolution. J Mar Sci Eng 2023;11(11):2045. https://doi.org/10.3390/jmse11112045
[37] Delpisheh M, Ebrahimpour B, Fattahi A, Siavashi M, Mir H, Mashhadimoslem H, Abdol MA, Ghorbani M, Shokri J, Niblett D, Khosravi K, Rahimi S, Alirahmi SM, Yu H, Elkamel A, Niasar V, Mamlouk M. Leveraging machine learning in porous media. J Mater Chem A 2024;12(32):20717–82. https://doi.org/10.1039/D4TA00251B
[38] Toussaint JN, Mally MP, Mertes S, Pischinger S. Cfd-based optimization toolchain for channel geometry in PEM fuel cells with novel bpp material. Int J Hydrogen Energy 2026;203:153101. https://doi.org/10.1016/j.ijhydene.2025.153101
[39] Xu S, Yan C, Sun Z, Huang R, Guo D, Yang G. On the preprocessing of physics-informed neural networks: How to better utilize data in fluid mechanics. arXiv:2403.19923. 2024. https://doi.org/10.48550/arXiv.2403.19923
[40] Bird RB, Stewart WI. Transport phenomena. 2nd ed. New York: J. Wiley; 2007.
[41] Mench MM. Fuel cell engines. Hoboken N.J.: John Wiley & Sons; 2008.
[42] Töpler J, Lehmann J. Wasserstoff und brennstoffzelle. Berlin, Heidelberg: Springer Berlin Heidelberg; 2017. https://doi.org/10.1007/978-3-662-53360-4
[43] Wu H, Berg P, Li X. Non-isothermal transient modeling of water transport in PEM fuel cells. J Power Sources 2007;165(1):232–43. https://doi.org/10.1016/j.jpowsour.2006.11.061
[44] Bard AJ, Faulkner LR, White HS. Electrochemical methods: fundamentals and applications. 3rd ed. Hoboken, NJ, USA and Chichester, West Sussex, UK: Wiley; 2022.
[45] Barbir F. PEM fuel cells: theory and practice, Sustainable World. 1st ed. Burlington: Elsevier Science & Technology; 2005. https://ebookcentral.proquest.com/lib/kxp/detail.action?docID=269935
[46] Gasteiger HA, Gu W, Makharia R, Mathias MF. Tutorial: catalyst utilization and mass transfer limitations in the polymer electrolyte fuel cell. In: The 2003 electrochemical society meeting. 2003, https://link.springer.com/chapter/10.1007/1-84628-207-1_2.
[47] Vikram A, Chowdhury PR, Phillips RK, Hoorfar M. Measurement of effective bulk and contact resistance of gas diffusion layer under inhomogeneous compression – Part I: electrical Conductivity. J Power Sources 2016;320:274–85. https://doi.org/10.1016/j.jpowsour.2016.04.110
[48] Gnanasambandam R, Shen B, Chung J, Yue X, Zhenyu K. Self-scalable tanh (stan): Faster convergence and better generalization in physics-informed neural networks. arXiv:2204.12589. 2022. https://doi.org/10.48550/arXiv.2204.12589
[49] Glorot X, Bengio Y. Understanding the difficulty of training deep feedforward neural networks. In: Teh YW, Titterington M, editors. Proceedings of the thirteenth international conference on artificial intelligence and statistics, vol. 9 of proceedings of machine learning research. Chia Laguna Resort, Sardinia, Italy: PMLR; 2010. p. 249–56. https://proceedings.mlr.press/v9/glorot10a.html
[50] Jagtap AD, Karniadakis GE. Extended physics-informed neural networks (xpinns): a generalized space-time domain decomposition based deep learning framework for nonlinear partial differential equations. Commun Comput Phys 2020;28(5):2002–41. https://doi.org/10.4208/cicp.OA-2020-0164
[51] Karniadakis GE, Kevrekidis IG, Lu L, Perdikaris P, Wang S, Yang L. Physics-informed machine learning. Nat Rev Phys 2021;3(6):422–40. https://doi.org/10.1038/s42254-021-00314-5
[52] Xu S, Yan C, Zhang G, Sun Z, Huang R, Ju S, Guo D, Yang G. Spatiotemporal parallel physics-informed neural networks: a framework to solve inverse problems in fluid mechanics. Phys Fluids 2023;35(6). https://doi.org/10.1063/5.0155087
[53] Rohrhofer FM, Posch S, Gößnitzer C, Geiger BC. Data VS. Physics: the apparent pareto front of physics-informed neural networks. IEEE Access 2023;11:86252–61. https://doi.org/10.1109/ACCESS.2023.3302892
[54] Maddu S, Sturm D, Müller CL, Sbalzarini IF. Inverse-dirichlet weighting enables reliable training of physics informed neural networks. arXiv:2107.00940. 2021. https://doi.org/10.48550/arXiv.2107.00940
[55] Cuomo S, Di Cola VS, Giampaolo F, Rozza G, Raissi M, Piccialli F. Scientific machine learning through physics–informed neural networks: where we are and what's next. J Sci Comput 2022;92(3). https://doi.org/10.1007/s10915-022-01939-z
[56] Eivazi H, Tahani M, Schlatter P, Vinuesa R. Physics-informed neural networks for solving reynolds-averaged navier–stokes equations. Phys Fluids 2022;34(7). https://doi.org/10.1063/5.0095270
[57] Zhou W, Xu YF. Data-guided physics-informed neural networks for solving inverse problems in partial differential equations. arXiv:2407.10836. 2024. https://doi.org/10.48550/arXiv.2407.10836
[58] Kingma DP, Ba J. Adam: A method for stochastic optimization. arXiv:1412.6980. 2014. http://arxiv.org/pdf/1412.6980v9
[59] Liu DC, Nocedal J. On the limited memory BFGS method for large scale optimization. Math Programming 1989;45(1–3):503–28. https://doi.org/10.1007/BF01589116
[60] Zhou C, Chen J, Yang Z, Png CE. Dual-balancing for physics-informed neural networks. arXiv:2505.11117. 2025. https://doi.org/10.48550/arXiv.2505.11117
[61] Wang S, Teng Y, Perdikaris P. Understanding and mitigating gradient pathologies in physics-informed neural networks. arXiv:2001.04536. 2020. https://doi.org/10.48550/arXiv.2001.04536
[62] Toussaint JN, Pieper S, Mally MP, Pischinger S. Python model PINN fuel cell v1.0.0. 2025. https://doi.org/10.5281/zenodo.17131101
[63] Das PK, Jiao K, Wang Y, Barbir F, Li X, Eds Fuel cells for transportation: fundamental principles and applications, Woodhead Publishing series in energy. Cambridge, MA and Kidlington, OX: Woodhead Publishing; 2023.
[64] Birgersson E, Noponen M, Vynnycky M. Analysis of a two-phase non-isothermal model for a pefc. J Electrochem Soc 2005;152(5):A1021. https://doi.org/10.1149/1.1877992