processes

MDPI

Article

Towards Surrogate Modeling for Adsorption Processes Using Physics-Informed Neural Networks

Mattia Galanti [1], Mik Janssen [1], Ivo Roghair [1], Jean-Yves Dieulot [2], Pejman Shoeibi Omrani [3], Jurriaan Boon [3] and Martin van Sint Annaland [1,*]

[1] Department of Chemical Engineering and Chemistry, Eindhoven University of Technology, 5612 AZ Eindhoven, The Netherlands; m.galanti@tue.nl (M.G.); i.roghair@tue.nl (I.R.)
[2] CNRS, Centrale Lille, CRIStAL—Centre de Recherche en Informatique Signal et Automatique de Lille, Universite de Lille, 59655 Villeneuve-d'Ascq, France
[3] Netherlands Organisation for Applied Scientific Research (TNO), 2333 BE Leiden, The Netherlands; pejman.shoeibiomrani@tno.nl (P.S.O.); jurriaan.boon@tno.nl (J.B.)
* Correspondence: m.v.sintannaland@tue.nl

Abstract

Physics-informed neural networks (PINNs) have emerged as a promising alternative to purely data-driven neural networks (NNs) for surrogate modeling, particularly in data-scarce scenarios. This study evaluates the performance of hybrid-PINNs against traditional NNs for modeling the adsorption step of a Direct Air Capture (DAC) process. As the complexity of the modeled system increases, larger datasets and longer computational times are required for numerical methods. Therefore, the study aims to develop approaches that minimize data requirements while maintaining accuracy, which is crucial for efficient modeling of complex physical systems. While both AI models can achieve high accuracy with abundant data, the advantages of hybrid-PINNs become more evident as data becomes scarce. In the intermediate and low-data regimes, the physics constraints embedded in hybrid-PINNs significantly improve generalization and predictive accuracy. For extreme low-data conditions, a curriculum learning strategy is implemented, progressively enforcing physics constraints to mitigate underfitting and enhance model stability. Despite these benefits, hybrid-PINNs exhibit a computational cost approximately one order of magnitude higher than traditional NNs as enforcing physics constraints increases training complexity. The results suggest that PINNs hold potential for modeling complex multi-physics problems in DAC and beyond, provided challenges related to gradient balancing and computational efficiency are addressed.

Keywords: PINNs; DAC; surrogate model; adsorption; neural networks; curriculum learning

check for updates

Received: 6 August 2025
Revised: 28 August 2025
Accepted: 2 September 2025
Published: 3 September 2025

Citation: Galanti, M.; Janssen, M.; Roghair, I.; Dieulot, J.-Y.; Shoeibi Omrani, P.; Boon, J.; van Sint Annaland, M. Towards Surrogate Modeling for Adsorption Processes Using Physics-Informed Neural Networks. *Processes* **2025**, *13*, 2824. https://doi.org/10.3390/pr13092824

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Direct Air Capture (DAC) is a promising carbon dioxide removal (CDR) technology that enables extraction of CO_2 directly from ambient air. The captured CO_2 can be concentrated for sequestration or utilized as feedstock to produce valuable products via green hydrogenation [1–4]. Among DAC approaches, Temperature–Vacuum Swing Adsorption (TVSA) using amine-functionalized solid sorbents is particularly attractive due to its high capture efficiency at low CO_2 concentrations [5–7]. However, detailed modeling of TVSA-based DAC processes using classical numerical methods is computationally expensive.

This high computational cost limits their direct integration into optimization frameworks, thus hindering practical DAC process design and optimization [8–10].

To address this challenge, machine-learning-based surrogate models offer a promising alternative by approximating the behavior of computationally expensive numerical simulations with orders-of-magnitude-lower inference times [11,12]. The goal of this study is to assess which surrogate modeling method is best suited to describe the adsorption step of the DAC-TVSA process. This is a crucial first step towards developing a complete surrogate model that can replace expensive numerical simulations in optimization frameworks, enabling more efficient DAC process design and control.

Previous attempts to apply artificial neural networks (ANNs) in DAC focused mainly on solvent and sorbent discovery, screening, and design [13,14]. However, only a few studies have explored the combination of AI with DAC process modeling. Sipocz et al. [15] first proposed a purely data-driven surrogate model using ANN for a complete MEA-based capture plant CO_2. More recently, Leperi et al. [11] proposed a more detailed PSA surrogate model using data-driven ANN, where different multiple ANNs were used to predict state variables in specific bed locations or to predict outlet bed conditions. Young et al. [16] successfully applied a data-driven ANN surrogate model to perform a detailed global sensitivity analysis of the TVSA process for DAC, which would have required years of computations with traditional numerical methods.

To the best of our knowledge, no previous study has integrated physics-informed neural networks (PINNs) with data-driven models to predict the spatio-temporal evolution of internal states in a DAC-TVSA adsorber. Such integration could significantly reduce training data requirements and enhance prediction accuracy and model generalization.

In this work, we compare two machine learning approaches: traditional data-driven neural networks (data-NNs) and hybrid-physics-informed neural networks (hybrid-PINNs). While standard NNs rely purely on labeled data, hybrid-PINNs incorporate governing physics equations as constraints during training alongside labeled data, potentially reducing the reliance on large datasets and improving model generalization in data-scarce conditions. The study systematically evaluates the performance of these models across different data availability regimes, including extremely low-data conditions, where a curriculum learning strategy is employed to improve model convergence.

The findings serve as a foundation for the development of a full surrogate model, which will replace expensive DAC process steps within an optimization framework, making TVSA-based DAC optimization more computationally efficient.

2. Machine Learning Models

We considered different machine learning techniques to assess their potential as surrogate models to approximate the axial concentration and temperature profiles during the adsorption step for fixed-bed Direct Air Capture: neural networks (NNs) and physics-informed neural networks (PINNs).

2.1. Artificial Neural Networks and PINNs

2.1.1. Feed-Forward Neural Networks

The concept of a neural network emerged in 1943 when Warren McCulloch and Walter Pitts proposed a mathematical model of how neurons work [17]. This model provided the foundation for understanding computation through interconnected processing units. In 1958, Frank Rosenblatt extended this idea with the development of the single-layer perceptron [18], which became the simplest form of a neural network (NN) and laid the foundations for modern feed-forward neural networks.

At its core, a neural network is a powerful tool able to approximate complex functions. This remarkable feature is supported by the Universal Approximation Theorem [19], which states that a sufficiently large neural network, with at least one hidden layer and non-linear activation functions, can approximate any continuous function on a closed interval to any desired level of accuracy.

In general, given a neural network (NN) with L layers, the resulting approximated function $F(\boldsymbol{x}, \boldsymbol{\theta})$ can be formalized as the composition of L functions $f^l(x^l, \theta^l)$, such that

$$F(\boldsymbol{x}, \boldsymbol{\theta}) = \left(f^L \circ f^{L-1} \circ \cdots \circ f^1\right)(\boldsymbol{x}), \tag{1}$$

where each function f^l represents a layer of the network, $\boldsymbol{x}$ is the input to the network, and $\boldsymbol{\theta} = \{\theta^1, \theta^2, \ldots, \theta^L\}$ is the set of parameters (weights and biases) of all layers.

Among the various neural network architectures, the feed-forward neural network (FFNN), also called the multi-layer perceptron (MLP), is one of the most well-known and extensively studied types. The name "feed-forward" reflects its structure: information flows in a single direction from the input layer to the output layer without any feedback loops [20]. A FFNN consists of multiple layers of interconnected neurons, where each layer performs a linear transformation of the input (via weights and biases) followed by a nonlinear transformation using an activation function. This process enables the network to capture complex patterns in the data.

2.1.2. Physics-Informed Neural Networks

Physics-informed neural networks (PINNs) are a class of neural networks that incorporate physical laws into their training process. PINNs represent an intersection between numerical methods and data-driven machine learning. In fact, numerical methods have been the go-to method for solving systems of partial differential equations by discretizing space and time, often with finite-element, finite-difference or finite-volume methods. Contrary to the mesh-dependent numerical methods, a PINN aims to learn a continuous, mesh-free, solution to the partial differential equations by training a NN. When the PINN is trained, it should satisfy both data (either experimental or numerically obtained) and the governing physical laws. This is achieved by adding a physics-informed loss term to the standard loss function, which enforces the network to satisfy the governing equations of the system, by embedding partial differential equations into the network's training objective. The most common PINN is a variation on the traditional NN, first introduced by Raissi, M. [21], in 2019 as a novel method to solve forward and inverse problems involving nonlinear partial differential equations. PINNs have been shown to be particularly effective in solving partial differential equations (PDEs) and other physics-based problems, where the underlying equations are known but difficult to solve analytically or numerically [22]. PINNs are also particularly useful for solving problems where the amount of data is scarce but the physical laws of the modeled system are well understood.

The mathematical framework of a PINN builds on the traditional NN but introduces physical constraints through additional terms in the loss function. Following the original formalism of Toscano et al. (2025), given a general differential operator $\mathcal{N}_\sim$ associated with the general boundary condition operator $\mathcal{B}_\sim$, both parametrized by ξ, a general PDE system can be written as

$$\begin{aligned} \mathcal{N}_\xi[\hat{u}](x) - f(x) = 0, \quad x \in \Omega, \\ \mathcal{B}_\xi[\hat{u}](x) - b(x) = 0, \quad x \in \partial\Omega, \end{aligned} \tag{2}$$

where $\hat{u}$ is the solution of the PDE in the domain Ω and its boundary $\partial\Omega$, x refers to the spatial–temporal coordinates, and $f(x)$ and $b(x)$ are source terms. Following what was

illustrated in the previous section, from Equation (1), a neural network can be used to approximate the solution $\hat{u}$ as follows [23]:

$$\hat{u}(x) \approx F(\boldsymbol{x}, \boldsymbol{\theta}) := u(\boldsymbol{x}, \boldsymbol{\theta}), \quad \boldsymbol{x} \in \Omega \cup \partial\Omega \tag{3}$$

The physics contribution is embedded in the network directly in the loss term $\mathcal{L}$. Thanks to automatic differentiation [24], the residuals of Equation (2) can be easily computed during the network training. In fact, given a set of input-training points $\boldsymbol{x_f} = \{x_{f1}, x_{f2}, \ldots, x_{f_{|x_f|}}\} \in \Omega$ and $\boldsymbol{x_b} = \{x_{b1}, x_{b2}, \ldots, x_{b_{|x_b|}}\} \in \partial\Omega$ of sizes $|\boldsymbol{x_f}|$ and $|\boldsymbol{x_b}|$, respectively, the physics loss terms can be defined using the residuals of Equation (2):

$$\begin{aligned} \mathcal{L}_f &= \frac{1}{|\boldsymbol{x_f}|} \sum_{x \in \boldsymbol{x_f}} \left| \mathcal{N}_{\tilde{\xi}}[u](\boldsymbol{x}, \boldsymbol{\theta}) - f(\boldsymbol{x}) \right|^2 \\ \mathcal{L}_b &= \frac{1}{|\boldsymbol{x_b}|} \sum_{x \in \boldsymbol{x_b}} \left| \mathcal{B}_{\tilde{\xi}}[u](\boldsymbol{x}, \boldsymbol{\theta}) - b(\boldsymbol{x}) \right|^2 \end{aligned} \tag{4}$$

where $\boldsymbol{x_f}$ and $\boldsymbol{x_b}$ are the designated sets of points where the residuals of the partial differential equations are evaluated, called "collocation points". The total physics loss is then computed by a direct summing of the two terms or linearly combining them with specified weights. In this way, the PINN is forced to adhere to the governing equations of the system by minimizing the MSE of PDEs residuals, computed using the approximated solution given by the network.

3. Building the PINN-DAC Model

The Direct Air Capture process considered in this work is the adsorption-based technology using solid amine sorbents. A packed bed adsorber designed for DAC is filled with spherical particles with uniform diameter (d_p) that selectively absorb CO_2. Due to the wide availability of data in the literature, the reference sorbent used in this work is LEWATIT® VPOC 1065, supplied by LANXESS GmbH, Cologne, Germany [25]. This sorbent is also considered the benchmark amine-functionalized sorbent for DAC applications due to its commercial availability and its high CO_2 capture capacity at concentrations corresponding to the ppm levels found in ambient air. However, H_2O co-adsorption kinetics and thermodynamics are still not fully understood, and much effort is currently being put into the development of a unified description of the CO_2-H_2O co-adsorption process [5,26,27]. Therefore, the exemplary case study considered in this work will be the non-isothermal dry adsorption of CO_2.

3.1. Problem Statement and Governing Equations

The packed column is fed with a simple gas mixture under ambient conditions, consisting of nitrogen N_2 and CO_2. The particles do not adsorb N_2 and have a finite loading capacity for CO_2. The exothermic nature of the adsorption process raises the column temperature during CO_2 capture, causing spatial and temporal changes in equilibrium concentration. Initially, at time $t < 0$, an inert gas flows through the adsorption column. At time $t = 0$, CO_2 containing the gas mixture is introduced into the column. The following assumptions are made when modeling this system:

- The ideal gas law applies;
- The system is under constant pressure;
- The flow is described by an axially dispersed plug-flow regime;
- Gradients in the radial direction are neglected;
- The densities, specific heat capacities, thermal conductivities, and axial gas dispersion coefficients are temperature-independent and constant throughout the reactor;

- Instantaneous thermal equilibrium is established between the fluid and solid phase;
- The gas velocity is constant and equal to the inlet velocity;
- The adsorption is limited by internal mass transfer described by the Linear Driving Force approximation (LDF).

Under these assumptions, the CO_2 mass balance can be described by the following PDEs:

$$\frac{\partial c}{\partial t} + u_z \frac{\partial c}{\partial z} + \mathcal{D}_{ax} \frac{\partial^2 c}{\partial z^2} + \frac{(1-\varepsilon_B)}{\varepsilon_B} \rho_p \frac{\partial q}{\partial t} = 0 \tag{5}$$

$$\frac{\partial q}{\partial t} - k_{LDF}(q^{eq} - q) = 0 \tag{6}$$

where u_z is the interstitial gas velocity, $\mathcal{D}_{ax}$ is the axial dispersion coefficient, ε_B is the bed porosity, ρ_p is the particle density, and k_{LDF} is the mass transfer coefficient from the Linear Driving Force approximation [28,29]. The main variables are the gas and solid phase concentrations c [mol m^{-3}] and q [mol kg^{-1}], respectively.

The solid equilibrium concentration $q^{eq}_{CO_2}$ is modeled using the non-isothermal Toth isotherm [30], thanks to its strong resemblance to experimental data [26,31,32]:

$$q_{eq}(P,T) = q_m(T) \frac{b(T)\,P}{\left[1 + (b(T)\,P)^{\tau(T)}\right]^{1/\tau(T)}} \tag{7}$$

$$q_m(T) = q_\infty \exp\left[\chi\left(1 - \frac{T}{T_0}\right)\right] \tag{8}$$

$$b(T) = b_0 \exp\left[\frac{-\Delta H_{\text{ads}}}{R_{\text{gas}} T}\left(\frac{T_0}{T} - 1\right)\right] \tag{9}$$

$$\tau(T) = \tau_0 + \beta\left(1 - \frac{T_0}{T}\right) \tag{10}$$

The overall energy balance is added considering a negligible gas and adsorbed-phase heat capacity contribution and dominant heat effects due to adsorption:

$$(1-\varepsilon_b)\rho_p \hat{c}_{ps} \frac{\partial T}{\partial t} + \varepsilon_b u_z c_{tot} \tilde{c}_{pf} \frac{\partial T}{\partial z} = \varepsilon_b \lambda_{ax} \frac{\partial^2 T}{\partial z^2} + (1-\varepsilon_b)\rho_p(-\Delta H_{\text{ads}}) \frac{\partial q}{\partial t} \tag{11}$$

where $\hat{c}_{ps}$ is the heat capacity of the solid, c_{tot} is the total fluid concentration, $\tilde{c}_{pf}$ is the heat capacity of the fluid, λ_{ax} is the effective axial thermal conductivity, and ΔH_{ads} is the adsorption heat. A list of all the parameters used can be found in Table 1.

Table 1. Simulation parameters for the Direct Air Capture system.

Variable	Definition	Value	Unit
L	Length of packed bed	0.1	m
R	Radius of packed bed	0.001	m
u_z	Inlet interstitial gas velocity	0.1	m s^{-1}
ε_B	Packed bed porosity	0.5	-
ρ_p	Particle density [31]	744	kg m^{-3}
$\mathcal{D}_{ax}$	Axial dispersion coefficient	2.23×10^{-4}	m^2 s^{-1}
T_{feed}	Feed temperature	318.15	K
$y_{CO_2,\text{feed}}$	Molar fraction of CO_2 in feed	500	ppm
P_{tot}	Total pressure	101,325	Pa
$\hat{c}_{ps}$	Specific heat capacity particle [31]	1396	J kg^{-1} K^{-1}
λ_{ax}	Effective axial thermal conductivity	0.22	W m^{-1} K^{-1}
ΔH_{ads}	Heat of adsorption [31]	−70.125	kJ mol^{-1}
k_{LDF}	Linear Driving Force coefficient	0.005	s^{-1}
t_{tot}	Total simulation time	20,000	s

3.1.1. Normalization

The governing equations will be expressed in their non-dimensional form to preserve the generality of the system and facilitate comparisons across different scales. The chosen sets of reference variables will scale both the input and output of the network between zero and one to enhance the learning efficiency and convergence of the neural network.

$$\frac{1}{\tau_m}\frac{\partial \varphi}{\partial \tau} + \frac{\partial \varphi}{\partial \xi} = \frac{1}{\mathcal{P}e_m}\frac{\partial^2 \varphi}{\partial \xi^2} - \frac{1}{\alpha_{ads}}\frac{1}{\tau_m}\frac{\partial \psi}{\partial \tau} \tag{12}$$

$$\frac{1}{\tau_m}\frac{\partial \psi}{\partial \tau} = \mathcal{D}a_I(\psi^{eq} - \psi) \tag{13}$$

$$\frac{1}{\tau_m}\frac{\partial \vartheta}{\partial \tau} + \zeta\frac{\partial \vartheta}{\partial \xi} = \frac{1}{\mathcal{P}e_h}\frac{\partial^2 \vartheta}{\partial \xi^2} - \frac{\Theta}{\tau_m}\frac{\partial \psi}{\partial \tau} \tag{14}$$

This normalization minimizes the risk of vanishing or exploding gradients, which are common issues in deep learning models when dealing with variables of vastly different magnitudes [33].

3.1.2. Boundary Conditions

A Danckwerts boundary condition is applied at the left boundary, while a Neumann boundary condition is used at the right boundary:

$$\varphi\big|_{\xi=0} - \frac{1}{\mathcal{P}e_m}\frac{\partial \varphi}{\partial \xi}\big|_{\xi=0} = 1 \tag{15}$$

$$\vartheta\big|_{\xi=0} - \frac{1}{\mathcal{P}e_h}\frac{\partial \vartheta}{\partial \xi}\big|_{\xi=0} = 0 \tag{16}$$

$$\frac{\partial \varphi}{\partial \xi}\big|_{\xi=1} = 0 \tag{17}$$

$$\frac{\partial \vartheta}{\partial \xi}\big|_{\xi=1} = 0 \tag{18}$$

along with the non-dimensional initial conditions:

$$\varphi\big|_{\tau=0} = 0 \tag{19}$$

$$\psi\big|_{\tau=0} = 0 \tag{20}$$

$$\vartheta\big|_{\tau=0} = 0 \tag{21}$$

3.2. Data-NN and Hybrid-PINN

To evaluate the neural network approach for constructing a surrogate model, we tested both a data-driven neural network (data-NN) and a hybrid physics-informed neural network (hybrid-PINN).

The data-NN approach focuses solely on minimizing a data-driven loss function, denoted as $\mathcal{L}_{\text{data}}$, which represents the discrepancy between the solution obtained with the network and the solution obtained by solving Equations (12)–(21) via a finite-differences method (FDM). The FDM simulations used for training and validation were derived from a simplified version of a more comprehensive numerical model, which has been validated against independent experimental data from Young et al. (2021) [26], thereby providing confidence in the physical fidelity of the reference simulations. A summary of this validation is reported in Appendixes A and B.

In contrast, the hybrid-PINN combines data-driven learning with physics-based constraints. In this framework, the total loss function $\mathcal{L}_{\text{total}}$ is a combination of two components: the data-driven loss $\mathcal{L}_{\text{data}}$ and the physics-informed loss $\mathcal{L}_{\text{physics}}$. The total composite loss function can be expressed as

$$\mathcal{L}_{total} = \lambda_d \, \mathcal{L}_{data} + \lambda_p \, \mathcal{L}_{physics} \tag{22}$$

where λ_i are weights factor that balances the contribution of the data and physics loss terms. By tuning these parameters, one can adjust the network's emphasis on fitting the data versus satisfying the physical constraints during training.

The physics loss function, composed of the residuals from the partial differential equations, boundary, and initial condition, is defined as follows:

$$\mathcal{L}_{\text{physics}} = \frac{1}{N_{\text{BC}}} \sum_{j=1}^{N_{\text{BC}}} \left(r_{\text{BC},j}\right)^2 + \frac{1}{N_{\text{IC}}} \sum_{j=1}^{N_{\text{IC}}} \left(r_{\text{IC},j}\right)^2 + \frac{1}{N_{\text{PDE}}} \sum_{j=1}^{N_{\text{PDE}}} \left(r_{\text{PDE},j}\right)^2, \tag{23}$$

where $r_{\text{BC},j}$, $r_{\text{IC},j}$, and $r_{\text{PDE},j}$ represent the residuals associated with the boundary conditions, initial conditions, and PDEs, respectively, evaluated at collocation point j.

In the specific case of the DAC-PINN investigated in this work, the total loss function was composed of 11 individual loss components:

- Three PDE residuals for each field: gaseous CO_2, adsorbed CO_2, and temperature.
- Three initial condition residuals for each field like above.
- Four boundary condition residuals: two spatial boundaries for the gas-phase equation and two for the temperature equation.
- One data loss component: penalizing the deviation from available data points.

The development of a hybrid-PINN approach, rather than a pure PINN approach ($\lambda_d = 0$), is motivated by preliminary tests showing that employing a PINN network exclusively tends to converge to trivial solutions that satisfy the PDE system but do not capture the desired physical behavior, as also found in Wu et al. [34]. Hybrid-PINN modeling provides additional advantages in addressing the challenges posed by standard PINNs, which often struggle to accurately approximate sharp fronts and regions with large gradients [35,36].

The FDM simulation provided a uniform distribution of data in the x–t domain, with resolution provided by time step $\Delta t = 1$ s and grid size $N_x = 1000$. From the data points obtained through the FDM simulation, only a subset was selected for training the models, aiming to balance prediction accuracy and computational cost. Since the optimal balance was not known a priori, a study was conducted to assess the capability of PINNs to enhance prediction accuracy as a function of data availability. The subset was obtained by a uniform random sampling using the NumPy *random* module [37]. Then, the sampled subset was further split into 70% for training and 30% for testing using the scikit-learn *model_selection* module [38]. Model validation was performed exclusively against data from the FDM simulation, which was considered the ground truth.

3.3. Network Construction

The results of the hyperparameters tuning are summarized in Table 2. A brief discussion on the training routine will be presented, while details regarding other hyperparameters will be excluded for brevity.

Table 2. *DAC-PINN* architecture.

Parameter	Value
Activation Function	Tanh
Final Smoothing Function	Sigmoid
Hidden Layers	5
Nodes per Layer	96
Initialization	Xavier
Optimizers	ADAM + L-BFGS in series
Data—Physics Scaling	EMA—Softmax
Individual Physics Loss Scaling	Self-adaptive weighting
PDE collocation points	50,000
BCs collocation points	20,000
ICs collocation points	10,000

3.4. Initial Setup

The neural network architecture consists of a total of five layers: one input layer, three hidden layers, and one output layer. The input layer accepts the spatial and temporal coordinates as input features, while the output layer predicts the fields of gaseous and adsorbed CO_2, along with the corresponding temperature field. Each hidden layer comprises 96 neurons, with *tanh* activation functions to capture the smooth and non-linear features of the PDEs system, while a *sigmoid* activation function is applied to the output layer to ensure that the predicted fields are constrained between 0 and 1, reflecting the physical bounds of the problem. A visual representation of the network architecture can be found in Figure 1.

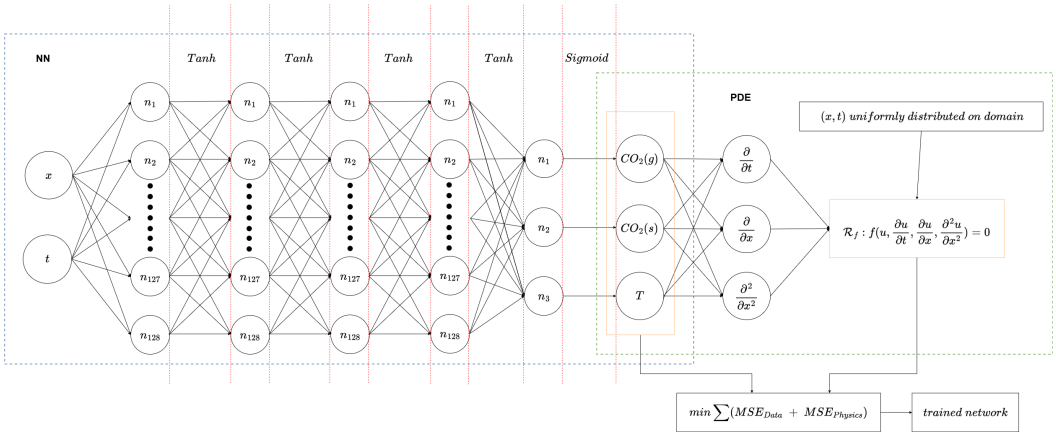


Figure 1. Representation of the hybrid-PINN architecture employed for this work.

The networks were initially trained using the ADAM optimizer with an initial learning rate of 10^{-3}. A learning rate scheduler was employed to reduce the learning rate by a factor of 2 after 50 epochs without improvement in the validation loss. Training was subject to a maximum number of epochs; however, an early stopping criterion was also implemented to terminate training when the validation loss plateaued. Following this initial training phase, the best-performing model (based on validation loss) was further refined using the L-BFGS optimizer with a learning rate of 10^{-1}. This secondary optimization step aimed to locate a better minimum in the loss landscape, leveraging the L-BFGS algorithm's quasi-Newton approach to fine-tune the solution near the local minimum found by ADAM. A visual representation of an exemplary test can be found in Figure 2.

Specifically, the entire work was conducted in Python (version 3.12.9), using PyTorch (version 2.6.0+cu126) [39] as the primary deep learning framework. All simulations and training were performed on a workstation equipped with an NVIDIA GeForce RTX 4060 Ti

GPU (NVIDIA Corporation, Santa Clara, CA, USA) (16 GB memory, CUDA version 12.4, driver version 550.142).

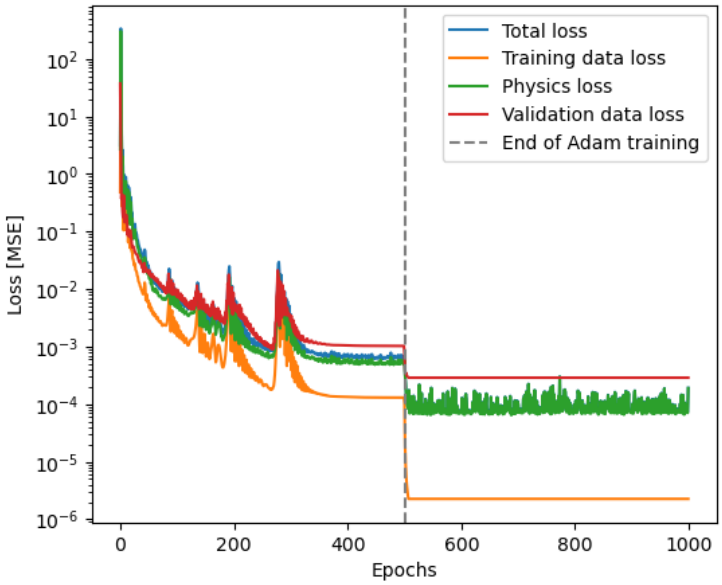


Figure 2. Exemplary training and validation routine where ADAM and L-BFGS are used in series.

3.4.1. Direct Warm-Start and Curriculum Learning

To enhance the training process of the hybrid-PINN, the knowledge acquired from the data-only training phase was exploited through a warm-start initialization. Specifically, this procedure involved initializing the weights and biases of the hybrid-PINN using those obtained from the data-only model. This approach allowed the hybrid-PINN to start from an already partially trained state, accelerating convergence and improving training stability.

However, this method proved ineffective in cases where the amount of data was significantly scarce. Under such conditions, the hybrid-PINN struggled to converge to a stable solution due to the complexity and strong nonlinearity of the problem. To address this limitation, a tailored approach based on curriculum learning was developed and tested.

Curriculum learning involves gradually increasing the complexity of the training process [40,41]. In this work, the curriculum strategy was implemented by progressively introducing the physics-based loss components into the training procedure. Specifically, after the data-only warm-start initialization, the network was first trained using only the initial conditions to anchor the solution at $t = 0$. Next, the governing PDE residuals were introduced sequentially in a logical order reflecting the underlying physics of the system:

1. Gas-phase transport equation: The gas phase serves as the primary carrier of the system's dynamics, driving mass and energy transfer.
2. Adsorbed-phase mass balance equation: Adsorption depends on the gas-phase transport as the solid uptake is a consequence of gas–solid interactions.
3. Energy (temperature) equation: The heat release or uptake is driven by adsorption phenomena, making it the last equation to be introduced.

The ordering was chosen to reflect the physical causality of the system: Gas transport drives adsorption, and adsorption drives the thermal response. While heuristic in nature, this hierarchy provides a logical structure for introducing physics constraints.

Finally, the left and right boundary conditions were enforced to refine the solution and ensure consistency across the domain.

3.4.2. Dynamic Weight Scaling: Self-Adaptation and EMA

Effective and efficient training of PINNs still face many challenges, such as the imbalance in the gradient sizes of the different loss components [42]. This problem occurs when the magnitude of a single loss term in the composite loss function dominates the others. If this happens, the network may only focus on minimizing these larger loss terms, ignoring smaller ones. This can lead to unbalanced training, where some parts of the model do not learn as effectively, making the training process less efficient, especially in the DAC-PINN case where 11 losses need to be balanced (see Section 3.2). A potential solution to this problem is the introduction of dynamic weighting during the PINN training. Dynamic weighting has shown promising results, with several studies on the subject published in recent years [43–45].

During training, we dynamically adjusted the weighting between the data loss and the physics-based loss at each epoch. This was achieved through a combination of Exponential Moving Average (EMA) tracking of the loss values and a Softmax-based normalization scheme. The EMA estimate of the losses was updated as follows:

$$\mathcal{L}_j^{(k)} = \alpha_{EMA}\mathcal{L}_j^{(k)} + (1-\alpha_{EMA})\mathcal{L}_j^{(k-1)} \tag{24}$$

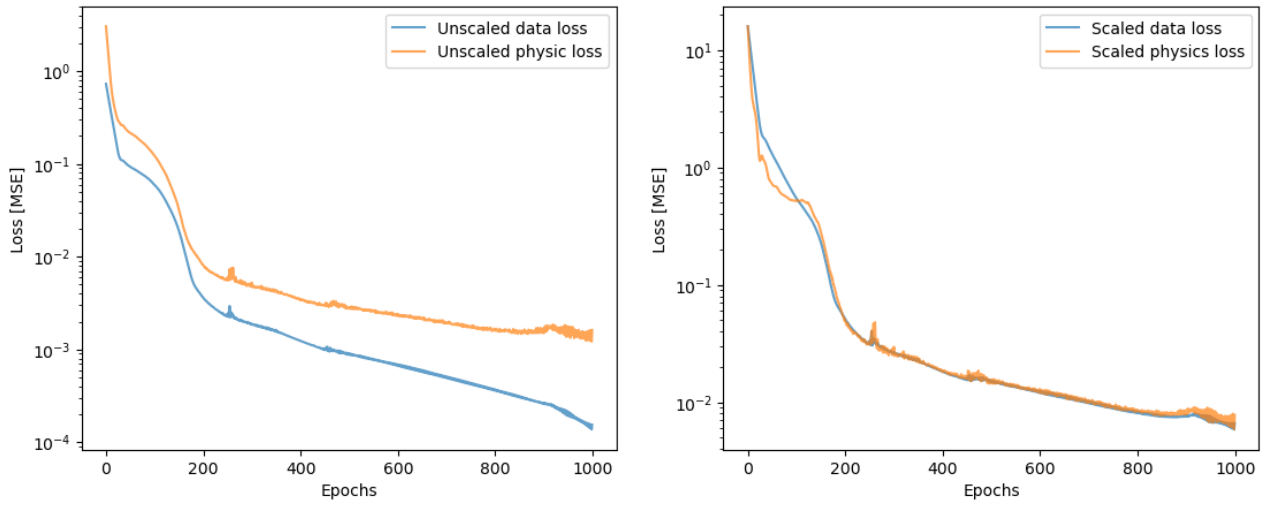
where the index j refers to physics and data loss, while $\alpha_{EMA} \in [0,1]$ is the smoothing parameter controlling the contribution of the current loss to the moving average. The EMA values are normalized relative to their sum to prevent the losses scale imbalance.

$$\tilde{\mathcal{L}}_j^{(k)} = \frac{\mathcal{L}_j^{(k)}}{\mathcal{L}_{\text{data}}^{(k)} + \mathcal{L}_{\text{physics}}^{(k)}} \tag{25}$$

Finally, the weighting coefficients λ_{data} and λ_{phys} are computed using a Softmax transformation with temperature parameter θ:

$$\lambda_j^{(k)} = \frac{\exp\left(\tilde{\mathcal{L}}_j^{(k)}/\theta\right)}{\exp\left(\tilde{\mathcal{L}}_{\text{data}}^{(k)}/\theta\right) + \exp\left(\tilde{\mathcal{L}}_{\text{phys}}^{(k)}/\theta\right)} \tag{26}$$

A visualization of dynamic scaling is shown in Figure 3.



(**a**) (**b**)

Figure 3. Comparison of unscaled and scaled losses used in the back-propagation step. (**a**) Unscaled data loss and physics loss during training. (**b**) Scaled data loss and physics loss during training.

For the 10 individual physics loss contributions, we employed a self-adaptive physics weights scaling strategy to dynamically balance the different loss components during the training process. Specifically, we introduced learnable logarithmic weights $w_i = log(w_i)$, initialized to zero, for each individual loss term (e.g., PDE residuals, boundary conditions, and initial conditions). The final weights are computed as $w_i = exp(log(w_i))$ ensuring positivity and enabling the optimizer to automatically adjust the relative importance of the different terms. The weights are treated as trainable parameters and optimized jointly with the neural network parameters using the ADAM optimizer. Additionally, before applying the weights, each loss component is pre-normalized by a baseline factor to mitigate scale discrepancies across different physical constraints. This approach prevents the dominance of a single loss component and enhances the training stability.

However, applying Dynamic Weight Scaling (DWS) at every epoch during the second-order L-BFGS optimization can interfere with the optimization process. L-BFGS relies on maintaining a history of gradient and step evaluations to approximate the curvature of the loss landscape and guide updates. When the loss weights are dynamically adjusted across epochs, the underlying loss surface is effectively changed, invalidating the gradient history and potentially disrupting the curvature approximation. This can degrade both convergence stability and overall performance.

Nonetheless, empirical tests indicated that properly balancing the loss terms significantly improves model accuracy. Based on these insights, we adopted a hybrid strategy: we applied DWS only once prior to initializing L-BFGS. This pre-scaling step provided a set of static, optimized loss weights, which were then held constant during the L-BFGS training phase. This approach ensured that the loss landscape remained smooth and consistent during L-BFGS while still leveraging the benefits of loss balancing obtained through DWS.

3.4.3. Error Metrics

To assess the predictive performance of our models, we employ the Percentage L1 Relative Error (%L1RE), which provides a normalized measure of the discrepancy between predicted and reference data, enabling error comparisons of the internal profiles across different species.

The Percentage L1 Relative Error (%L1RE) is defined as

$$\%\text{L1RE} = 100 \, \frac{\sum_i \left| u_i^{\text{pred}} - u_i^{\text{true}} \right|}{\sum_i \left| u_i^{\text{true}} \right|}$$

where u_i^{pred} and u_i^{true} denote the predicted and true values at index i, respectively. %L1RE is preferred as it preserves the scale of the error without disproportionately emphasizing large deviations. This is particularly important when dealing with localized phenomena or sparse regions of high error, where alternative metrics could amplify the impact of a few large discrepancies and obscure the overall accuracy.

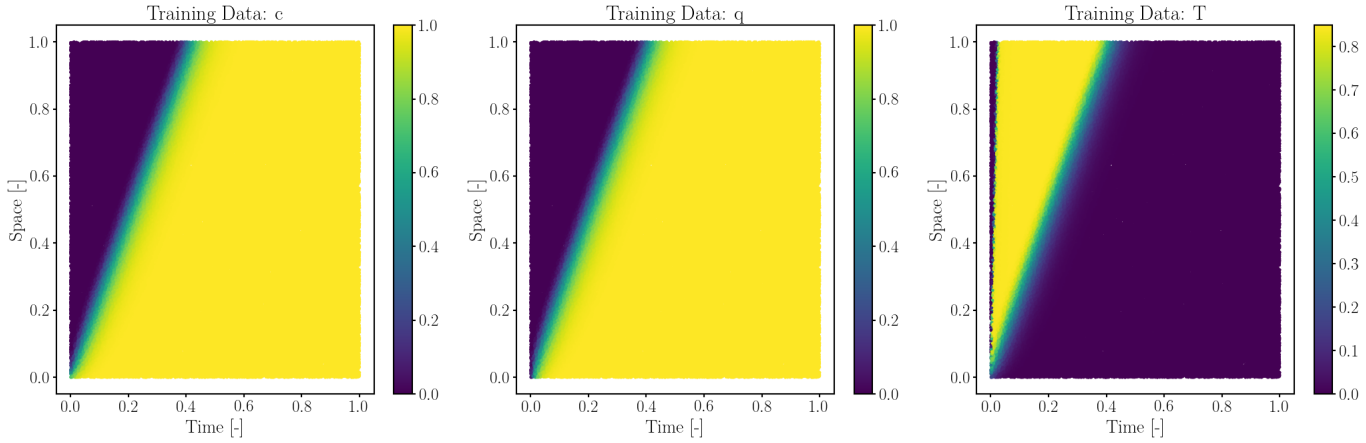
4. Results

One of the primary objectives of using hybrid-PINNs over traditional neural networks is to reduce the reliance on large datasets, which can be costly and time-consuming to obtain. To evaluate this capability, data-NN and hybrid-PINN were evaluated in four different sets of training points (labeled data) constructed from the results of the numerical simulation, given in Table 3. These datasets span multiple orders of magnitude, allowing an evaluation of how the model scales with increasing or decreasing labeled data availability. A representative visual representation can be found in Figure 4.

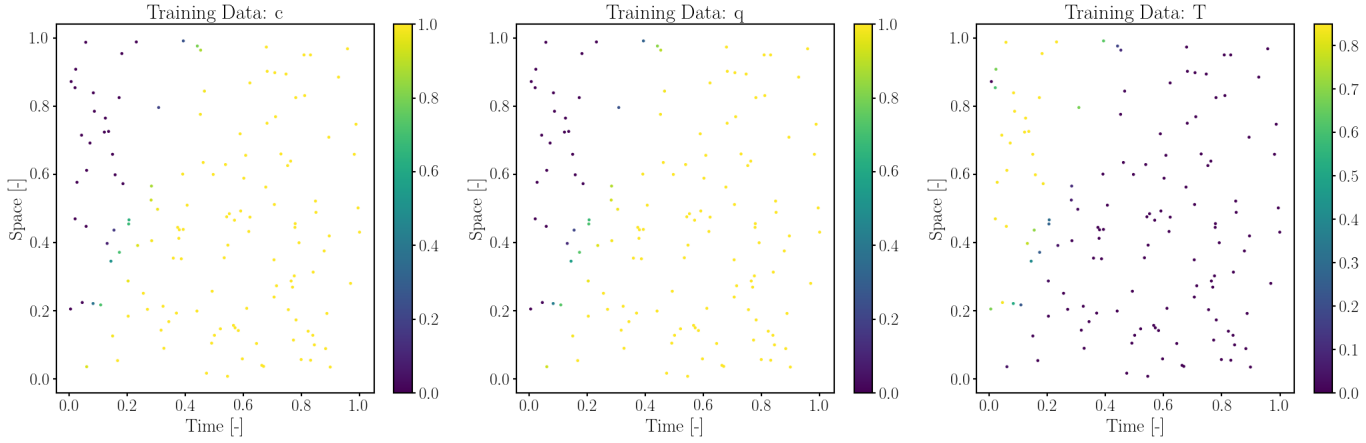
A summary of the main findings across these regimes is shown in Figure 5, where the L1 Relative Error (%L1RE) is plotted as a function of the training dataset size. This analysis highlights the trade-off between purely data-driven neural networks and the hybrid-PINN approach. While both methods achieve high accuracy in the large data regime (1%), the performance of the data-driven model deteriorates as labeled data becomes scarce, particularly in the intermediate and low-data regimes. In contrast, the hybrid-PINN effectively leverages physics constraints to maintain predictive accuracy, demonstrating its robustness even when only a minimal fraction (0.001%) of labeled data is available.

Table 3. Labeled and training/test data overview.

% of Labeled Data	Labeled Data	Training Data	Test Data
1%	200,010	140,007	60,003
0.1%	20,001	14,000	6001
0.01%	2000	1400	600
0.001%	200	140	60



(**a**)



(**b**)

Figure 4. Representative labeled data visualization. The three output variables are sampled randomly inside the spatial–temporal training domain. (**a**) 1% sampled set. (**b**) 0.001% sampled set.

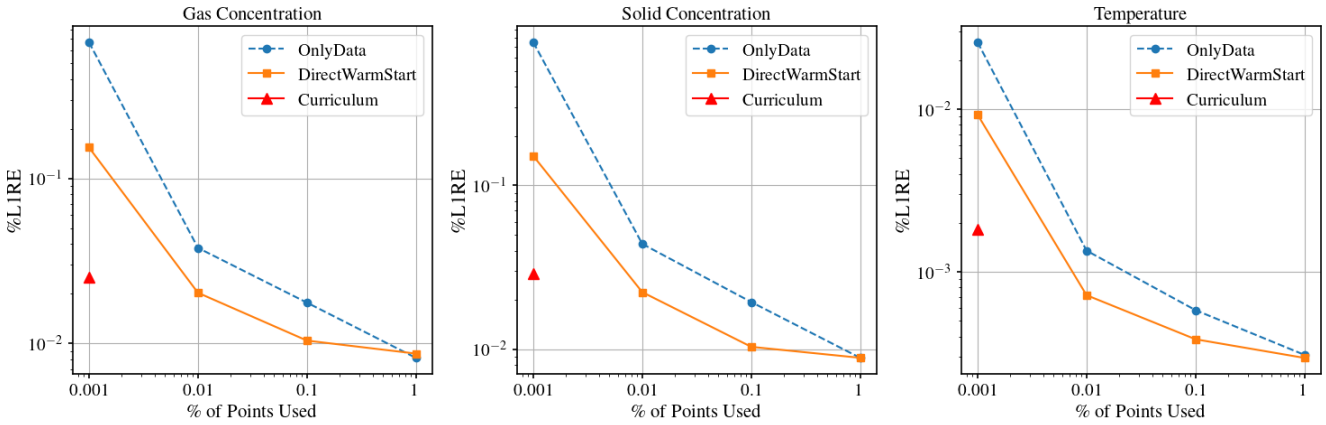


Figure 5. Internal profiles %L1RE as a function of the percentage of training points used for different learning strategies.

4.1. Large and Intermediate Data Regime

In the large data regime (1% of the total training points), both the purely data-driven neural network and the warmstart-hybrid-PINN achieved very low errors, as illustrated in Figure 6. The data-driven model alone, having access to a sufficiently large subset of labeled points, manages to learn the system's behavior with high fidelity. Consequently, the hybrid-PINN approach does not exhibit significant improvements. In scenarios where labeled data is plentiful, the incorporation of physical constraints yields minimal additional benefit as the learning process is already robustly underpinned by the available data.

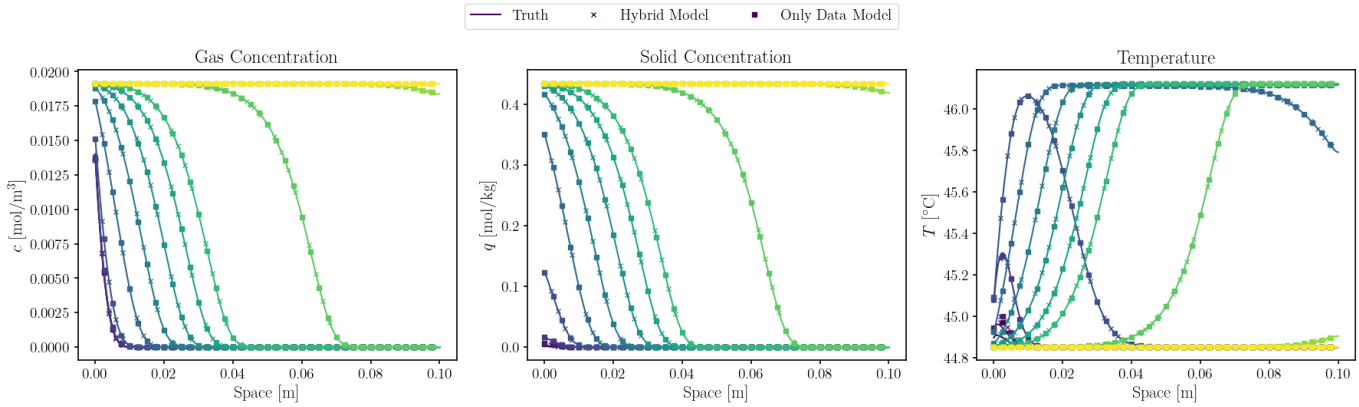
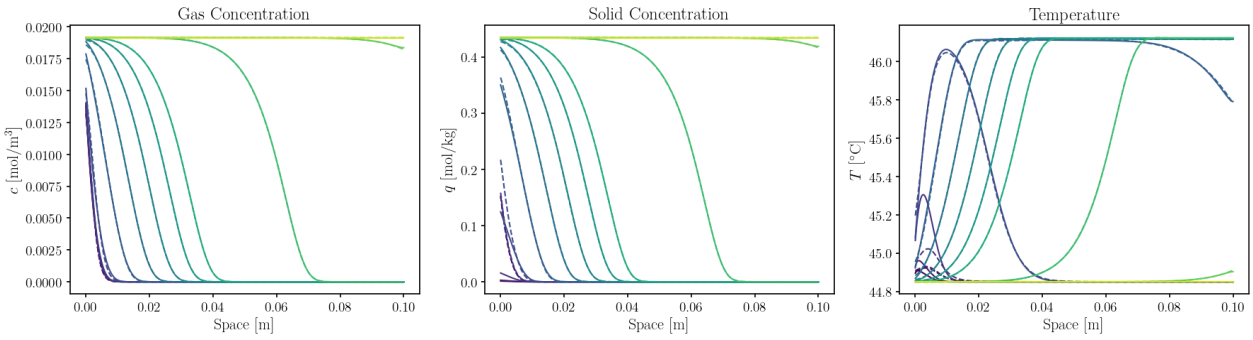
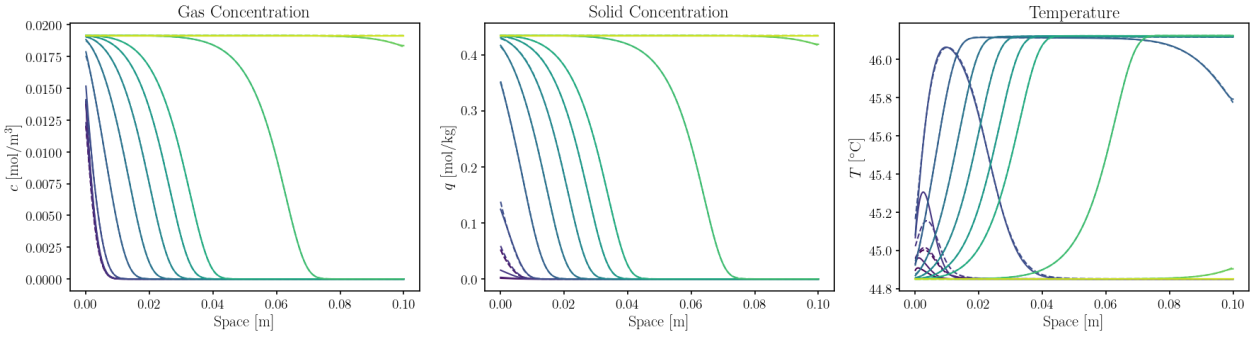


Figure 6. Internal profiles predictions comparison in the large data regime (1% training points). Each colored line represents a different time snapshot of the spatial profiles along the packed bed.

More interesting differences emerge in the intermediate data regime (0.1% and 0.01% of training points). Here, the data-driven model alone begins to lose accuracy, particularly for the temperature and the solid concentration fields. The warmstart-hybrid-PINN, however, makes better use of limited labeled data by leveraging physics constraints to regularize and guide the solution. As demonstrated in Figure 7, improvements are visible, especially in the solid-phase concentration and temperature profiles.



(**a**)



(**b**)

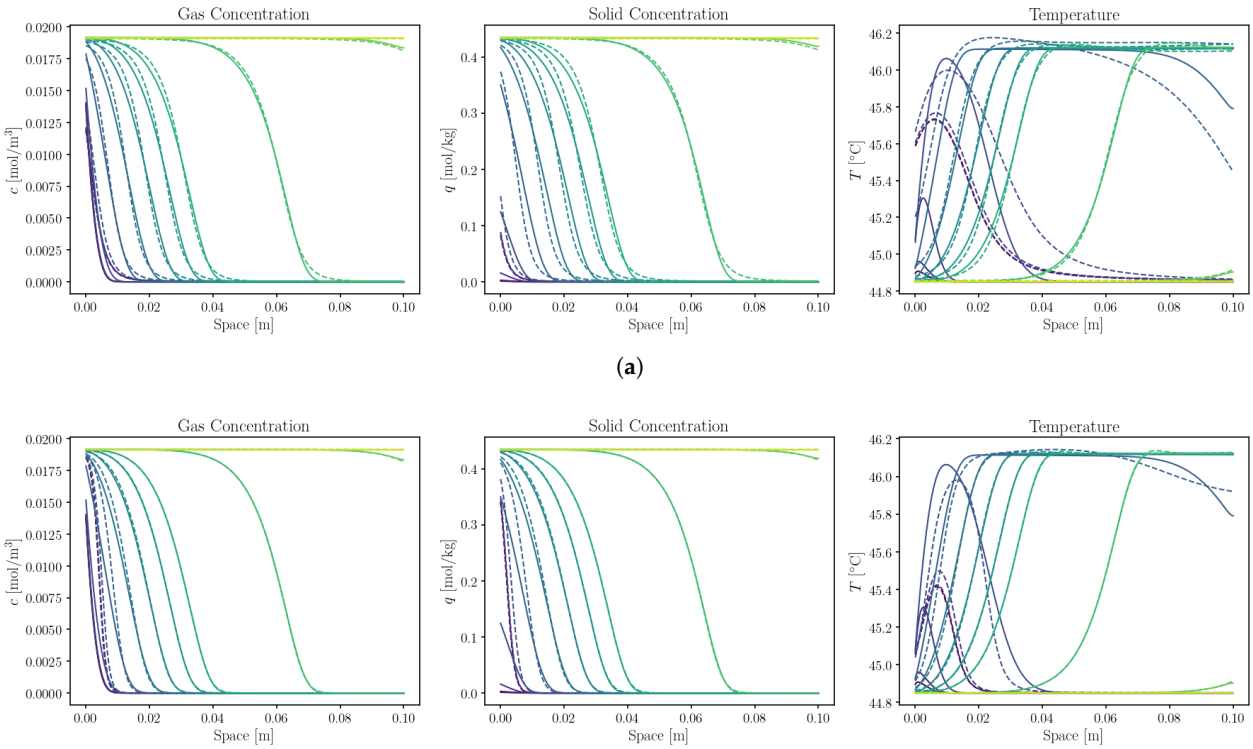
Figure 7. Predicted internal profiles at 0.01% training points. Dashed lines represent the model predictions, while solid lines denote the ground truth. Each colored line represents a different time snapshot of the spatial profiles along the packed bed. (**a**) Data-driven model. (**b**) Warmstart-hybrid-PINN model.

4.2. Low-Data Regime: Curriculum Learning

In the low-data regime (below 0.01% of training points), Figure 5 shows that the warmstart-hybrid-PINN maintains lower L1RE values than the purely data-driven model, mirroring the trend seen at higher data levels. Nevertheless, Figure 8 indicates that at *0.001%* of the training points (roughly 140 samples), neither approach achieves qualitatively acceptable internal profiles. The purely data-driven model (Figure 8a) underfits the ground truth, while the warmstart-hybrid-PINN (Figure 8b) still struggles to resolve the three coupled PDEs since 140 labeled points remain insufficient to anchor such a complex model, even with the benefit of physics-informed constraints.

To address this persistent challenge in the low-data regime, the *Curriculum Learning* strategy was applied to the hybrid-PINN, yielding significantly reduced L1RE and evidently improved internal profiles, as shown in Figure 9b. The curriculum-hybrid-PINN approach sequentially introduces each element of the physics-informed network by first anchoring the solution to the initial conditions and then refining step-by-step the prediction incorporating the full set of governing equations and boundary conditions.

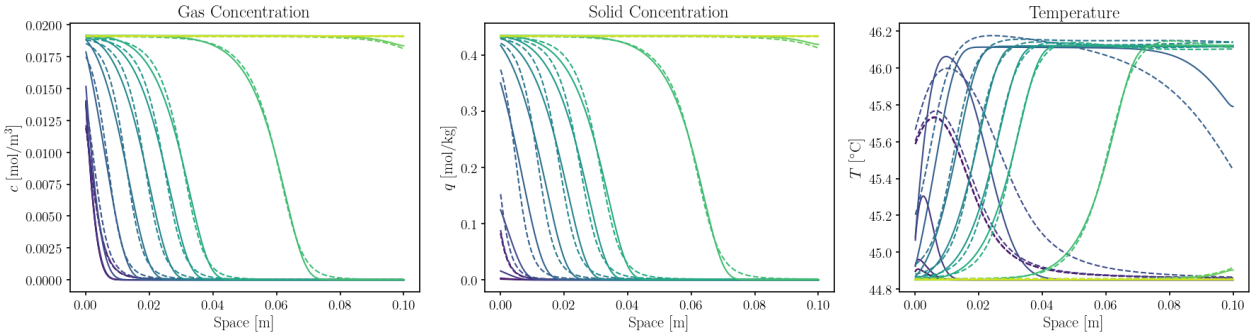
This phased training allows the model to stabilize its representations before tackling the full complexity of the coupled PDEs. Consequently, even at 0.001% training data, the curriculum-hybrid-PINN exhibits near intermediate data regime accuracy, closing the gap in gas-phase and solid-phase predictions, where the data-driven baseline (Figure 9a) struggles with large discrepancies from the ground truth.



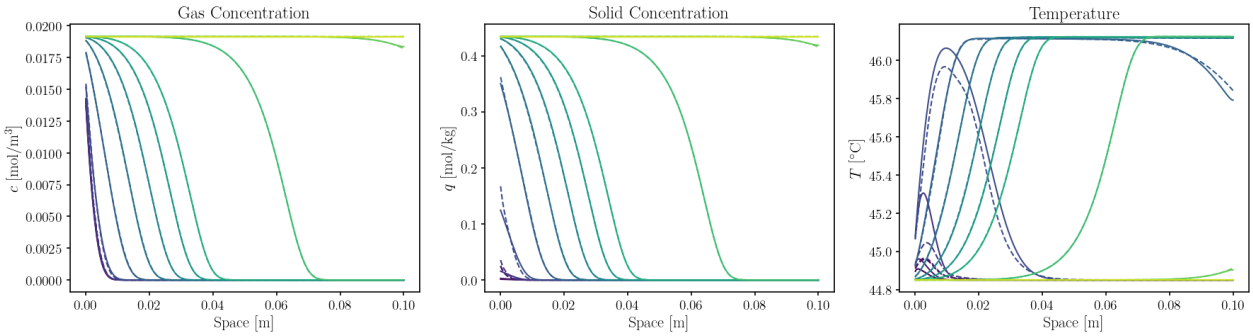
(**a**)

(**b**)

Figure 8. Predicted internal profiles at 0.001% training points. Dashed lines represent the model predictions, while solid lines denote the ground truth. Each colored line represents a different time snapshot of the spatial profiles along the packed bed. (**a**) Data-driven model. (**b**) Warmstart-hybrid-PINN model.



(**a**)



(**b**)

Figure 9. Predicted internal profiles at 0.001% training points. (**a**) shows the data-driven model; (**b**) shows the curriculum-hybrid-PINN model. Dashed lines represent the model prediction, while solid lines refer to the ground truth.

The reduced accuracy observed in the temperature predictions can be attributed primarily to the distinct dynamics of this field compared to the gas and solid concentrations. While c and q evolve as traveling fronts with relatively smooth, flat wave profiles, the temperature field is characterized by a localized peak that propagates through the bed. This peak-shaped transient is more challenging to approximate as it requires the network to capture the nonlinear interplay between adsorption kinetics, heat release, and thermal transport. Unlike concentrations, which are governed mainly by advection–dispersion dynamics and, therefore, retain a more uniform profile, the temperature reflects the balance of multiple competing effects: the strongly exothermic adsorption source term, thermal diffusion, and heat exchange with the reactor walls.

Moreover, the smoother, lower-gradient regions of the temperature wave provide a weaker training signal for the network compared to the sharper concentration fronts, further complicating the learning task.

Despite these challenges, the exothermic peak and the traveling thermal wave are both captured with sufficient fidelity for process-level analysis. Nonetheless, these results suggest that temperature fields may benefit from dedicated network representations or task-specific weighting in the training process in order to better accommodate their distinct spatial–temporal patterns.

5. Discussion

The comparison between DAC-PINN and a conventional neural network (NN) revealed several key insights. While the traditional NN demonstrated strong predictive performance when trained on full-domain simulations, its limitations became apparent when extrapolating to regions of the system with little or no training data coverage. In contrast, the hybrid-PINN approach consistently provided better generalization particularly in data-sparse regions.

However, these accuracy gains come at a computational cost. As shown in Figure 10, training a hybrid-PINN requires an order of magnitude more computational time compared to a traditional NN. The exact increase in training time varies depending on the size of the labeled dataset as the enforcement of physics-informed constraints introduces additional optimization complexity. In particular, the overhead is mainly due to the repeated use of automatic differentiation to compute PDE residuals, as well as their evaluation at a large number of collocation points, both of which increase the cost of each back-propagation step relative to a purely data-driven NN. Among the strategies tested, the curriculum-based training was the most computationally demanding, followed by the hybrid-PINN, with the conventional NN being the most efficient.

Despite this, the added cost is primarily a training-time issue, while inference remains as fast as for a standard NN. This makes the approach practically deployable since, once trained, the model can be used for rapid evaluation in process optimization or control tasks without additional overhead. Furthermore, training cost can be mitigated by techniques such as warm-starting from data-only models or adaptive residual sampling, improving efficiency while retaining the generalization benefits of the hybrid-PINN framework.

Although PINNs have a lot of potential, their current technological state does not yet allow them to compete with traditional high-performance numerical solvers [46]. Despite these advantages, the overall performance of PINNs often falls short compared to well-established numerical methods. Additionally, most studies on PINNs for simulations without an analytical solution depend on numerical results as truth data for training the network, highlighting the scientific community's confidence in these mature and thoroughly tested numerical methods. Moreover, the validation in this work relies solely on finite-difference simulations, without direct comparison to experimental data. While this

approach is sufficient to demonstrate the capability of the proposed framework under controlled conditions and to isolate numerical effects, it does not yet establish predictive accuracy for physical systems. Future work will, therefore, involve experimental benchmarking, which will be essential to fully assess the applicability of the method to real-world DAC processes.

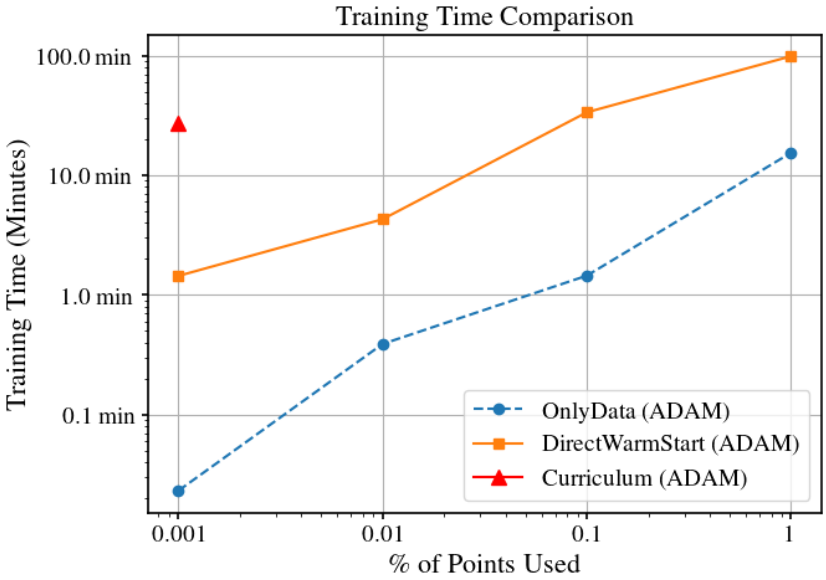


Figure 10. Training time comparison as a function of the percentage of training points used for different learning strategies.

Nevertheless, several advantages exist in using PINNs over traditional numerical solvers. PINNs can approximate a function across the entire domain, offering a meshless solution. This enables accurate solutions at any point within the domain, without the need to extrapolate between cells and faces. Moreover, once trained, *DAC-PINN* can predict concentrations and temperatures at any spatial or temporal point almost instantly, enabling rapid process simulations and optimization. In the context of DAC, such surrogate models can be used to efficiently generate CAPEX–OPEX Pareto fronts in combination with fast sorbent performance evaluations, supporting comprehensive techno-economic evaluations of process configurations [47,48]. This approach allows for potential integration, within defined limits and boundaries, into other simulation softwares as a fast alternative to repeatedly solving systems of equations with numerical methods.

The strong performance of this surrogate model suggests that the development of a digital twin for DAC systems with various input parameters using PINNs is a realistic concept, with a multi-input PINN model being currently under development.

6. Conclusions

Overall, our results confirm that hybrid-PINN offer a robust alternative to purely data-driven neural networks when the labeled dataset is limited. While both approaches perform well in the large-data regime (1%), physics constraints add little beyond what abundant data already provides. However, as labeled data availability diminishes (0.1–0.001%), incorporating physics-informed elements becomes increasingly beneficial: hybrid-PINN retains lower error and capture essential dynamics more faithfully. In the extreme low-data regime (0.001%), a *curriculum learning* strategy was shown to further reduce prediction error by introducing PDE constraints in stages, thereby mitigating model underfitting. Though temperature predictions remain less precise due to small thermal gradients, these discrepancies are minor compared to the overall improvements in gas and solid-phase predictions. Hence, for complex, multi-

physics problems under data scarcity, hybrid-PINNs with a curriculum-based approach provide a significant advantage over purely data-driven models.

The modeling of a DAC system using PINNs represents an innovative approach. While significant progress has been made in this field, much of the knowledge is relatively recent, suggesting that this technology will require time to fully mature. One of the key challenges lies in modeling processes involving domains with rapid gradient changes, a difficulty widely acknowledged within the scientific machine learning community. The construction of a digital twin of a CO_2 adsorption system using PINNs is feasible, but the presence of sharp gradients in the domain requires the use of labeled data points to achieve accurate predictions. Notably, this reliance on labeled data is consistent with other adsorption cases reported in the literature (see Wu et al. 2024), where insufficient data can lead to trivial or non-physical solutions, particularly in regions with sparse information. Furthermore, while constructing *DAC-PINN*, the importance of normalization and gradient balancing became evident in line with literature findings where unbalanced gradients within the physics-incorporated loss are a well-documented challenge [49]. The custom loss-balancing algorithm developed for *DAC-PINN* addressed this problem, allowing the network to converge to a physically representative and accurate solution.

The strong performance of the surrogate model demonstrates the feasibility of developing a digital twin for DAC systems using physics-informed neural networks (PINNs) across a range of input parameters. Advancing in the digital twin development involves incorporating the initial concentration and temperature profiles as input fields to the model, enabling the evaluation full cyclic operations and the computation cyclic steady-state profiles. A multi-input, initial-condition-aware PINN is currently under development to support this goal.

Author Contributions: Conceptualization, M.G.; methodology, M.G. and M.J.; software, M.G. and M.J.; validation, M.G. and M.J.; formal analysis, M.G. and M.J.; investigation, M.J.; data curation, M.J.; writing—original draft preparation, M.G.; writing—review and editing, M.G., I.R., J.-Y.D., J.B., P.S.O. and M.v.S.A.; visualization, M.G.; supervision, I.R. and M.v.S.A.; project administration, M.G.; funding acquisition, I.R. and M.v.S.A. All authors have read and agreed to the published version of the manuscript.

Funding: We thank Shell Global Solutions International BV for financial support.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

PINNs	Physics-Informed Neural Networks
NNs	Neural Networks
DAC	Direct Air Capture
AI	Artificial Intelligence
CDR	Carbon Dioxide Removal
TVSA	Temperature–Vacuum Swing Adsorption
ANN	Artificial Neural Network
MEA	Monoethanolamine
MLP	Multi-Layer Perceptron
FFNN	Feed-Forward Neural Network
PDE	Partial Differential Equation
MSE	Mean Squared Error

FDM	Finite Difference Method
EMA	Exponential Moving Average
L1RE	L1 Relative Error

Appendix A. Model Validation

The finite-difference model (FDM) used in this study is a simplified version of a more detailed adsorption–desorption cycle simulator developed by our group for Direct Air Capture (DAC) with temperature–vacuum swing adsorption (TVSA). To establish the fidelity of the model, we benchmarked it against the experimental results reported by Young et al. [26] for the commercial sorbent Lewatit® VPOC 1065 under representative DAC conditions.

The baseline case is defined by the cycle parameters and adsorber specifications summarized in Table A1. The operating steps include adsorption, vacuum, heating, desorption, cooling, and pressurization, with durations and boundary conditions chosen to match those reported in the reference paper.

Table A1. Baseline case parameters used for model validation.

Bed length L_{bed}	0.01 m
Bed radius R_{int}	0.05 m
Bed porosity ε_{bed}	0.40
Particle radius R_{p}	0.26 mm
Adsorption temperature T_{ads}	288.15 K
Desorption temperature T_{des}	373.15 K
Relative humidity (feed)	0.55
Feed velocity v_{feed}	0.0067 m/s
Vacuum pressure P_{vac}	0.2 bar
Dry CO_2 capacity $m_{CO_2}^{\text{dry}}$	1.87 g

Figures A1 and A2 show the comparison of the model predictions with the experimental outlet profiles of CO_2 molar fraction and outlet temperature, respectively. The model reproduces both the shape and magnitude of the experimental curves across the adsorption, heating, desorption, and cooling steps, including the CO_2 breakthrough behavior and the thermal response of the bed.

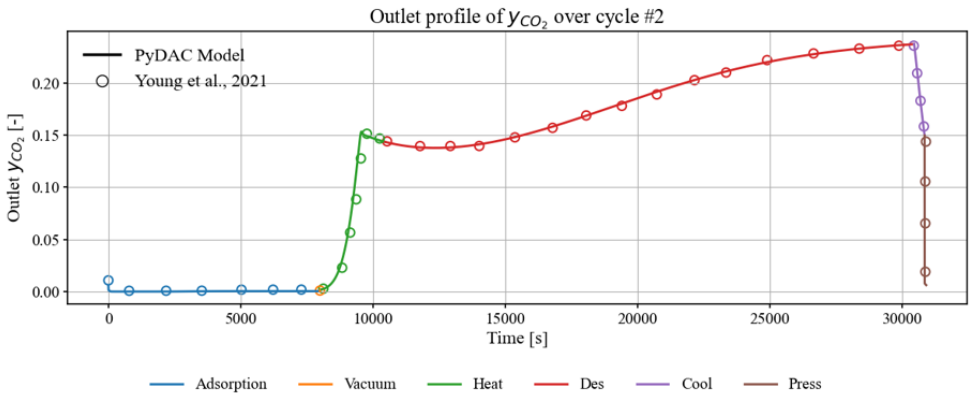


Figure A1. Comparison between model prediction and data from Young et al. (2021) [26] for the outlet CO_2 molar fraction over a full TVSA cycle.

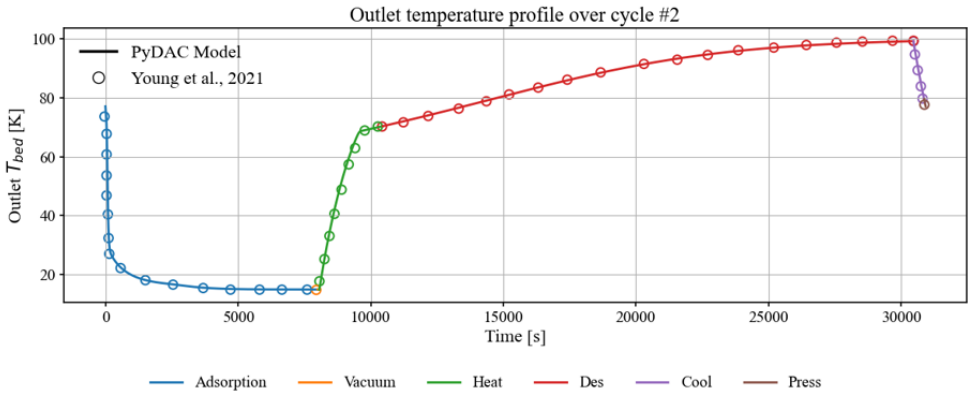


Figure A2. Comparison between model prediction and data from Young et al. (2021) [26] for the outlet temperature profile over a full TVSA cycle.

Appendix B. Feed-Forward Neural Networks

In the feed-forward process, a neuron i with n inputs (state variables) $x_i = \{x_{i1}, x_{i2}, \ldots, x_{in}\}$ computes its output in the two aforementioned steps:

1. Weighted Sum and Bias Addition: Each input x_{ij} is multiplied by a corresponding weight w_{ij}, and a bias term b_i is added. The resulting weighted sum z_i is given by

$$z_i = \sum_{j=1}^{n} w_{ij} x_{ij} + b_i = W_i x_i + b_i$$

2. Nonlinear Activation: The neuron applies an activation function $\sigma(z_i)$ to the weighted sum, such that $f_i(W_i, x_i, b_i) = \sigma(W_i x_i + b_i)$. This introduces non-linearity, which is crucial for modeling complex data relationships.

Some examples of activation functions are as follows:

- **Hyperbolic tangent (Tanh):** Outputs values in the range $(-1, 1)$, making it useful for centered data.

$$f(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$$

- **Sigmoid (logistic function):** Maps values to the range $(0, 1)$, often used for probabilities in binary classification tasks.

$$f(z) = \frac{1}{1 + e^{-z}}$$

- **Rectified Linear Unit (ReLU):** Outputs the input if it is positive, otherwise zero. Widely used for its simplicity and efficiency.

$$f(z) = \begin{cases} z, & \text{if } z > 0 \\ 0, & \text{otherwise} \end{cases}$$

- **Leaky ReLU:** Similar to ReLU but allows a small negative slope for $z < 0$, addressing the "dying ReLU" problem.

$$f(z) = \begin{cases} z, & \text{if } z > 0 \\ 0.01z, & \text{otherwise} \end{cases}$$

- **Softplus:** A smooth approximation of ReLU, useful when differentiability is important.

$$f(z) = \log(1 + e^z)$$

A graphical representation of the activation function is shown in Figure A3.

For a layer consisting of m neurons, the lth layer takes the input vector $\boldsymbol{z^{(l-1)}}$ and computes the output for all neurons in parallel:

$$\boldsymbol{z^{(l)}} = \sigma^{(l)}(W^{(l)}\boldsymbol{z^{(l-1)}} + b^{(l)}) \tag{A1}$$

where $\boldsymbol{z^{(l)}} = \{z_0^{(l)}, \ldots, z_m^{(l)}\}$, $W^{(l)}$ is the matrix containing all the weights of the layer, and $b^{(l)}$ is the bias vector of the layer.

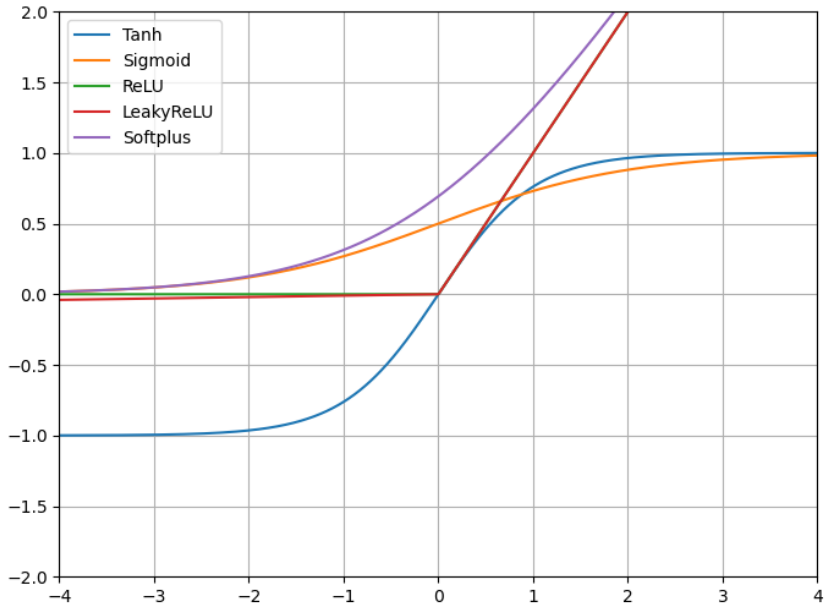


Figure A3. Graphical representation of the used activation functions.

Finally, the output of the activation function is passed to the next layer as the input, where the same process of a linear combination followed by a transformation by an activation function is repeated. This procedure will continue until the final output layer is reached [23]:

$$F(\boldsymbol{x}, \boldsymbol{\theta}) = \sigma\Big(W^{(L)}\sigma\Big(W^{(L-1)} \cdots \sigma\Big(W^{(1)}\boldsymbol{x} + b^{(1)}\Big) \cdots + b^{(L-1)}\Big) + b^{(L)}\Big) \tag{A2}$$

where $\boldsymbol{\theta} = \{W^{(l)}, b^{(l)}\}$ is the trainable parameters of the network.

A neural network is trained using known output ("truth" data), with the aim of optimizing weights and biases in order to minimize the loss $\mathcal{L}$, which represents the difference between the network's predictions and the truth values. The method used to minimize this loss is called back-propagation, which involves two key steps:

- **The forward pass**: The input data is passed through the network layer by layer, until the final layer where the output is calculated (network prediction), which is then compared to the truth values to calculate the loss.
- **The backward pass**: The network calculates how much each weight contributes to the total loss. This is carried out efficiently using the chain rule of calculus [50]. The weights are then updated using an optimization algorithm like gradient descent to decrease the loss of the NN.

The most common loss function for NNs is the Mean Squared Error (MSE), as defined in Equation (A3) [51]:

$$\mathcal{L} = \frac{1}{N}\sum_{j=1}^{N}(y_j - \hat{y}_j)^2 \tag{A3}$$

where y_j is the predicted value from the network, and $\hat{y}_j$ is the true value. Using this loss function in combination with the chain rule from Calculus, the weights in the network can then be updated relative to their contribution to the loss function. The new weights are then calculated according to Equation (A4) and the new biases according to Equation (A5) [52]:

$$w_{new} = w_{old} - \eta \cdot \frac{\partial \mathcal{L}}{\partial w} \tag{A4}$$

$$b_{new} = b_{old} - \eta \cdot \frac{\partial \mathcal{L}}{\partial b} \tag{A5}$$

where η is the learning rate, a small positive number that controls the step size of the updates.

The update of the weights and biases continues until convergence, ideally reaching a point where the loss is minimized.

The downside to this method, and gradient-descent-based optimization methods in general, is that there is no guarantee that at convergence a global minimum is reached. The reason for this is the non-convex nature of the loss landscape, which makes it impossible to prove that the minimum obtained by the iterative updating procedure is indeed the global minimum in the infinite domain [53–55].

References

1. Ozkan, M.; Nayak, S.P.; Ruiz, A.D.; Jiang, W. Current status and pillars of direct air capture technologies. *iScience* **2022**, *25*, 103990. [CrossRef]
2. Marinič, D.; Likozar, B. Direct air capture multiscale modelling: From capture material optimization to process simulations. *J. Clean. Prod.* **2023**, *408*, 137185. [CrossRef]
3. Zeeshan, M.; Kidder, M.K.; Pentzer, E.; Getman, R.B.; Gurkan, B. Direct air capture of CO_2: From insights into the current and emerging approaches to future opportunities. *Front. Sustain.* **2023**, *4*, 1167713. [CrossRef]
4. Erans, M.; Sanz-Pérez, E.S.; Hanak, D.P.; Clulow, Z.; Reiner, D.M.; Mutch, G.A. Direct air capture: Process technology, techno-economic and socio-political challenges. *Energy Environ. Sci.* **2022**, *15*, 1360–1405. [CrossRef]
5. Sabatino, F.; Grimm, A.; Gallucci, F.; van Sint Annaland, M.; Kramer, G.J.; Gazzani, M. A comparative energy and costs assessment and optimization for direct air capture technologies. *Joule* **2021**, *5*, 2047–2076. [CrossRef]
6. Zhu, X.; Xie, W.; Wu, J.; Miao, Y.; Xiang, C.; Chen, C.; Ge, B.; Gan, Z.; Yang, F.; Zhang, M.; et al. Recent advances in direct air capture by adsorption. *Chem. Soc. Rev.* **2022**, *51*, 6574–6651. [CrossRef] [PubMed]
7. Low, M.Y.A.; Barton, L.V.; Pini, R.; Petit, C. Analytical review of the current state of knowledge of adsorption materials and processes for direct air capture. *Chem. Eng. Res. Des.* **2023**, *189*, 745–767. [CrossRef]
8. Ward, A.; Pini, R. Efficient Bayesian Optimization of Industrial-Scale Pressure-Vacuum Swing Adsorption Processes for CO_2 Capture. *Ind. Eng. Chem. Res.* **2022**, *61*, 13650–13668. [CrossRef]
9. Ward, A.; Papathanasiou, M.M.; Pini, R. The impact of design and operational parameters on the optimal performance of direct air capture units using solid sorbents. *Adsorption* **2024**, *30*, 1829–1848. [CrossRef]
10. Chung, W.; Kim, J.; Jung, H.; Lee, J.H. Hybrid modeling of vacuum swing adsorption carbon capture process for rapid process-level evaluation of adsorbents. *Chem. Eng. J.* **2024**, *495*, 153664. [CrossRef]
11. Leperi, K.T.; Yancy-Caballero, D.; Snurr, R.Q.; You, F. 110th Anniversary: Surrogate Models Based on Artificial Neural Networks to Simulate and Optimize Pressure Swing Adsorption Cycles for CO_2 Capture. *Ind. Eng. Chem. Res.* **2019**, *58*, 18241–18252. [CrossRef]
12. Ye, F.; Ma, S.; Tong, L.; Xiao, J.; Bénard, P.; Chahine, R. Artificial neural network based optimization for hydrogen purification performance of pressure swing adsorption. *Int. J. Hydrogen Energy* **2019**, *44*, 5334–5344. [CrossRef]
13. Jerng, S.E.; Park, Y.J.; Li, J. Machine learning for CO_2 capture and conversion: A review. *Energy AI* **2024**, *16*, 100361. [CrossRef]
14. Rahimi, M.; Moosavi, S.M.; Smit, B.; Hatton, T.A. Toward smart carbon capture with machine learning. *Cell Rep. Phys. Sci.* **2021**, *2*, 100396. [CrossRef]

15. Sipöcz, N.; Tobiesen, F.A.; Assadi, M. The use of Artificial Neural Network models for CO_2 capture plants. *Appl. Energy* **2011**, *88*, 2368–2376. [CrossRef]
16. Young, J.; Mcilwaine, F.; Smit, B.; Garcia, S.; van der Spek, M. Process-informed adsorbent design guidelines for direct air capture. *Chem. Eng. J.* **2023**, *456*, 141035. [CrossRef]
17. McCulloch, W.S.; Pitts, W. A logical calculus of the ideas immanent in nervous activity. *Bull. Math. Biophys.* **1943**, *5*, 115–133. [CrossRef]
18. Rosenblatt, F. The Perceptron: A Probabilistic Model For Information And Storage In The Brain. *Psychol. Rev.* **1958**, *65*, 19–27. [CrossRef] [PubMed]
19. Hornik, K.; Stinchcombe, M.; White, H. Multilayer feedforward networks are universal approximators. *Neural Netw.* **1989**, *2*, 359–366. [CrossRef]
20. Bebis, G.; Georgiopoulos, M. Feed-forward neural networks. *IEEE Potentials* **1994**, *13*, 27–31. [CrossRef]
21. Raissi, M.; Perdikaris, P.; Karniadakis, G.E. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* **2019**, *378*, 686–707. [CrossRef]
22. Cuomo, S.; Di Cola, V.S.; 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*, 88. [CrossRef]
23. Toscano, J.D.; Oommen, V.; Varghese, A.J.; Zou, Z.; Ahmadi Daryakenari, N.; Wu, C.; Karniadakis, G.E. From PINNs to PIKANs: Recent advances in physics-informed machine learning. *Mach. Learn. Comput. Sci. Eng.* **2025**, *1*, 15. [CrossRef]
24. Güneş Baydin, A.; Pearlmutter, B.A.; Andreyevich Radul, A.; Mark Siskind, J. Automatic differentiation in machine learning: A survey. *J. Mach. Learn. Res.* **2015**, *18*, 5595–5637.
25. LEWATIT® VP OC 1065. Available online: https://lanxess.com/en-us/products-andbrands/products/l/lewatit--vp-oc-1065 (accessed on 25 July 2024).
26. Young, J.; García-Díez, E.; Garcia, S.; Van Der Spek, M. The impact of binary water–CO_2 isotherm models on the optimal performance of sorbent-based direct air capture processes. *Energy Environ. Sci.* **2021**, *14*, 5377–5394. [CrossRef]
27. Shan, M.; Jin, Y.; Chimani, F.M.; Bhandari, A.A.; Wallmüller, A.; Schöny, G.; Müller, S.; Fuchs, J. Evaluation of CO_2/H_2O Co-Adsorption Models for the Anion Exchange Resin Lewatit VPOC 1065 under Direct Air Capture Conditions Using a Novel Lab Setup. *Separations* **2024**, *11*, 160. [CrossRef]
28. Glueckauf, E. *Formula for Diffusion into Spheres and Their Applications to Chromatography*; Technical Report; Atomic Energy Research Establishment: Harwell, UK, 1955.
29. Sircar, S.; Hufton, J.R. Why does the linear driving force model for adsorption kinetics work? *Adsorption* **2000**, *6*, 137–147. [CrossRef]
30. Duong, D.D. *Adsorption Analysis: Equilibria and Kinetics*; Imperial College Press: London, UK, 1998; p. 892.
31. Low, M.Y.A.; Danaci, D.; Azzan, H.; Woodward, R.T.; Petit, C. Measurement of Physicochemical Properties and CO_2, N_2, Ar, O_2, and H_2O Unary Adsorption Isotherms of Purolite A110 and Lewatit VP OC 1065 for Application in Direct Air Capture. *J. Chem. Eng. Data* **2023**, *68*, 3499–3511. [CrossRef]
32. Shi, W.; Chen, S.; Xie, R.; Gonzalez-Diaz, A.; Zhang, X.; Wang, T.; Jiang, L. Bi-disperse adsorption model of the performance of amine adsorbents for direct air capture. *Chem. Eng. J.* **2024**, *496*, 154090. [CrossRef]
33. Yu, J.; Spiliopoulos, K. Normalization effects on deep neural networks. *Found. Data Sci.* **2022**, *5*, 389–465. [CrossRef]
34. Wu, Z.; Chen, Y.; Zhang, B.; Ren, J.; Chen, Q.; Wang, H.; He, C. Pressure swing adsorption process modeling using physics-informed machine learning with transfer learning and labeled data. *Green Chem. Eng.* **2024**, *6*, 233–248. [CrossRef]
35. Krishnapriyan, A.S.; Gholami, A.; Zhe, S.; Kirby, R.M.; Mahoney, M.W. Characterizing possible failure modes in physics-informed neural networks. *Adv. Neural Inf. Process. Syst.* **2021**, *32*, 26548–26560.
36. De Florio, M.; Schiassi, E.; Calabrò, F.; Furfaro, R. Physics-Informed Neural Networks for 2nd order ODEs with sharp gradients. *J. Comput. Appl. Math.* **2024**, *436*, 115396. [CrossRef]
37. Harris, C.R.; Millman, K.J.; van der Walt, S.J.; Gommers, R.; Virtanen, P.; Cournapeau, D.; Wieser, E.; Taylor, J.; Berg, S.; Smith, N.J.; et al. Array programming with NumPy. *Nature* **2020**, *585*, 357–362. [CrossRef]
38. Pedregosa, F.; Michel, V.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Vanderplas, J.; Cournapeau, D.; Pedregosa, F.; Varoquaux, G.; et al. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
39. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In Proceedings of the 33rd International Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019.
40. Monaco, S.; Apiletti, D. Training physics-informed neural networks: One learning to rule them all? *Results Eng.* **2023**, *18*, 101023. [CrossRef]
41. Bekele, Y.W. Physics-informed neural networks with curriculum training for poroelastic flow and deformation processes. *arXiv* **2024**, arXiv:2404.13909. [CrossRef]

42. Wang, S.; Teng, Y.; Perdikaris, P. Understanding and mitigating gradient pathologies in physics-informed neural networks. *SIAM J. Sci. Comput.* **2020**, *43*, 3055–3081. [CrossRef]
43. Li, S.; Feng, X. Dynamic Weight Strategy of Physics-Informed Neural Networks for the 2D Navier–Stokes Equations. *Entropy* **2022**, *24*, 1254. [CrossRef] [PubMed]
44. Deguchi, S.; Asai, M. Dynamic & norm-based weights to normalize imbalance in back-propagated gradients of physics-informed neural networks. *J. Phys. Commun.* **2023**, *7*, 075005. [CrossRef]
45. Wang, J.; Xiao, X.; Feng, X.; Xu, H. An improved physics-informed neural network with adaptive weighting and mixed differentiation for solving the incompressible Navier–Stokes equations. *Nonlinear Dyn.* **2024**, *112*, 16113–16134. [CrossRef]
46. Markidis, S. The Old and the New: Can Physics-Informed Deep-Learning Replace Traditional Linear Solvers? *Front. Big Data* **2021**, *4*, 669097. [CrossRef]
47. Chisăliță, D.A.; Boon, J.; Lücking, L. Adsorbent shaping as enabler for intensified pressure swing adsorption (PSA): A critical review. *Sep. Purif. Technol.* **2025**, *353*, 128466. [CrossRef]
48. Krishnamurthy, S.; Boon, J.; Grande, C.; Lind, A.; Blom, R.; de Boer, R.; Willemsen, H.; de Scheemaker, G. Screening Supported Amine Sorbents in the Context of Post-combustion Carbon Capture by Vacuum Swing Adsorption. *Chem. Ing. Tech.* **2021**, *93*, 929–940. [CrossRef]
49. Rathore, P.; Lei, W.; Frangella, Z.; Lu, L.; Udell, M. Challenges in Training PINNs: A Loss Landscape Perspective. *Proc. Mach. Learn. Res.* **2024**, *235*, 42159–42191.
50. Rumelhart, D.E.; Hinton, G.E.; Williams, R.J. Learning representations by back-propagating errors. *Nature* **1986**, *323*, 533–536. [CrossRef]
51. Terven, J.; Cordova-Esparza, D.M.; Ramirez-Pedraza, A.; Chavez-Urbiola, E.A.; Romero-Gonzalez, J.A. Loss Functions and Metrics in Deep Learning. *arXiv* **2023**, arXiv:2307.02694. [CrossRef]
52. Zhang, J. Gradient Descent based Optimization Algorithms for Deep Learning Models Training. *arXiv* **2019**, arXiv:1903.03614. [CrossRef]
53. Jentzen, A.; Riekert, A. On the existence of global minima and convergence analyses for gradient descent methods in the training of deep neural networks. *J. Mach. Learn.* **2021**, *1*, 141–246. [CrossRef]
54. Yun, C.; Sra, S.; Jadbabaie, A. Global optimality conditions for deep neural networks. In Proceedings of the 6th International Conference on Learning Representations (ICLR 2018), Vancouver, BC, Canada, 30 April–3 May 2018; Conference Track Proceedings.
55. Chatterjee, S. Convergence of gradient descent for deep neural networks. *arXiv* **2022**, arXiv:2203.16462. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.