Please cite the following article as:

Cirne, M., Menke, H., Abdellatif, A., Maes, J., Doster, F., & Elsheikh, A. (2026). A Deep-Learning Iterative Stacked Approach for Prediction of Reactive Dissolution in Porous Media. *InterPore Journal*. Advance online publication. https://doi.org/10.69631/j1re7z03

Received on: 15 Dec. 2025
Accepted on: 1 Apr. 2026
Published online (Accepted version): 20 Apr. 2026

A Deep-Learning Iterative Stacked Approach for Prediction of Reactive Dissolution in Porous Media

Marcos Cirne[1], Hannah P. Menke[1], Alhasan Abdellatif[1], Julien Maes[1], Florian Doster[1], Ahmed H. Elsheikh[1]

[1]School of Energy, Geoscience, Infrastructure and Society (EGIS), Heriot-Watt University
[1]EH14 4AS, Edinburgh, United Kingdom

Key Points:

- Data-driven deep-learning approach for predicting multiple future states of reactive dissolution in porous media using an iterative strategy;
- Proposal of a multi-level network stacking pipeline, where each level is trained to minimize the errors produced by its predecessor;
- Comparative analysis among several deep learning algorithms, which are trained on an ensemble of numerical simulation models, encompassing various pore structures and fluid trajectories.

Corresponding author: Marcos Cirne, M.Cirne@ncl.ac.uk

Corresponding author: Ahmed Elsheikh, A.Elsheikh@hw.ac.uk

Abstract

Simulating reactive dissolution of solid minerals in porous media has many subsurface applications, including carbon capture and storage (CCS), geothermal systems and oil & gas recovery. As traditional direct numerical simulators are computationally expensive, it is of paramount importance to develop faster and more efficient alternatives. Deep-learning-based solutions, most of them built upon convolutional neural networks (CNNs), have been recently designed to tackle this problem. However, these solutions were limited to approximating one field over the domain (e.g. velocity field), not accounting for the coupled evolution of multiple interacting fields, including concentration, porosity and flow rates. In this manuscript, we present a novel deep learning approach that incorporates both temporal and spatial information to predict the future states of the dissolution process at a fixed time-step horizon, given a sequence of input states. The overall performance, in terms of speed and prediction accuracy, is demonstrated on a numerical simulation dataset, comparing its prediction results against state-of-the-art approaches, also achieving a speedup around 10^4 over traditional numerical simulators.

1 Introduction

Numerical solvers have been extensively used to simulate and understand the effects of reactive dissolution of solid minerals in subsurface porous media, in diverse applications such as CO_2 sequestration (Wang et al., 2023), hydrogen storage (Heinemann et al., 2021), enhanced oil recovery (Esfe & Esfandeh, 2020), radioactive waste disposal (Liang et al., 2021) and geothermal systems (Salimzadeh & Nick, 2019). Due to the intrinsic complexity of these processes, which are governed by a set of highly non-linear partial differential equations (PDEs), it is computationally expensive to simulate (Khebzegga et al., 2020). Recently, deep learning (DL) algorithms have become a prominent tool for speeding up the modelling process, while at the same time generating highly accurate simulations of subsurface fluid dynamics (Zhu et al., 2022; Garnier et al., 2021; Da Wang et al., 2021; Kochkov et al., 2021).

Typically, DL algorithms for subsurface applications rely on data-driven approaches that require a significant number of examples so that these algorithms can properly learn the underlying physics of the phenomena to be studied. CNN-based methods (Alqahtani et al., 2018; Graczyk & Matyka, 2020; A. Li et al., 2020; Santos et al., 2020; Tang et al., 2021), which employ a series of spatial convolutions to extract meaningful features from image-like data, have been widely adopted to predict specific properties of porous media (such as porosity, permeability and fluid flow). Alternatively, physics-informed neural networks (PINNs) (Yan et al., 2022; Du et al., 2023; He et al., 2020) have been adopted to embed the physical laws that govern a given dataset as a prior information into deep neural networks (DNNs).

When it comes to the task of forecasting the future evolution of nonlinear dynamic systems, both categories of neural networks naturally struggle to yield reasonable predictions. Several approaches attempt to combine those types of networks with time-dependent units, including recurrent neural networks (RNNs) (Mohajerin & Waslander, 2019; MS & Menon, 2021), convolutional long short-term networks (ConvLSTMs) (Cheng et al., 2023; Feng et al., 2024) and gated recurrent units (GRU) (Ding et al., 2022; Al-Shabandar et al., 2021). A more recent approach named recurrent neural operator (RNO) (Karimi & Bhattacharya, 2024) was used to map functions rather than discrete data points (thus reducing computational costs) for prediction of reactive flow, but this approach is only applied within large-scale domains. Other have adopted a hybrid approach, such as Reichstein et al. (2019), who leverages physical process models along with data-driven machine-learning algorithms. The forecasted lead time is bound to the studied phenomenon, and may vary from milliseconds to hundreds of years. However, these methods can only be trained to predict a fixed (limited) amount of future steps, regardless of the time unit.

In order to yield predictions from an initial state for long-term horizons, those dynamic systems employ an iterative strategy in which the output derived from a prediction is used to comprise the input for the subsequent prediction and so on. However, this approach suffers from error propagation with each future step, as the input distributions are more likely to shift away from the distribution under which those systems were trained (Koesdwiady et al., 2018). Although there are alternatives to avoid the pitfalls of iterative strategies, they are still highly vulnerable to performance degradation due to the intrinsic uncertainties in forecasting further time steps. A possible heuristic to minimize the prediction errors is by stacking multiple networks and perform an iterative process to minimize the overall residual error of the predictions (Kani & Elsheikh, 2017). In other words, at each level of the stack, the corresponding network tries to improve the results achieved in the previous level.

This paper presents an approach to predict the dynamical evolution of reactive dissolution in porous media. Given an ensemble of numerical pore-scale simulations containing different trajectories for the dissolution process, we train a DL algorithm in a data-driven and supervised way, accounting for spatial and temporal features, as a surrogate model to forecast the future dissolution states. From a sequence of input states, the algorithm is first trained to predict a fixed amount of output states. To assess the quality of the dissolution forecasting, an iterative stacked strategy is adopted and the outputs are evaluated against a ground truth by means of similarity and error metrics. Moreover, a multi-level stacking approach will be investigated as an attempt to reduce the error accumulation.

The novel contributions of this paper are twofold:

- We develop an iterative stacked framework that performs simultaneous prediction of multiple fields and can be applied to both single-step and multi-step scenarios, while yielding accurate results for reactive dissolution simulations that are orders of magnitude faster than traditional numerical solvers;
-
- We conduct a comparative analysis among different DL architectures in terms of speed and prediction accuracy.

The remainder of this manuscript is organized as follows: Section 2 defines the problem addressed by our work, including the principles of reactive dissolution, forecasting strategies and deep learning algorithms; Section 3 details our proposed methodology for multi-step prediction of reactive dissolution, as well as the DL methods used in our experiments; Section 4 describes our dataset and the preprocessing steps prior to training the DL algorithms; Section 5 discusses the produced results; finally, we conclude the paper in Section 6.

2 Preliminaries

2.1 Problem Statement

Let $\mathcal{S} = \{X_1, X_2, ..., X_N\}$, where $X_i \in \mathbb{R}^{C \times H \times W}$, be a numerical simulation of a reactive dissolution process composed of N consecutive time steps of state maps of size $H \times W$ for each of the C physical properties. The governing equations for pore-scale reactive dissolution follow the formulation of Maes et al. (2022). Also, let m be the size of an input sequence of consecutive states and t a starting point from which we want to predict the next n subsequent states. The problem of forecasting consists of taking an input sequence $\mathcal{X} = X_{t-m+1:t} = \{X_{t-m+1}, X_{t-m+2}, ..., X_{t-1}, X_t\}$ and predict an output sequence $\mathcal{Y} = \hat{X}_{t+1:t+n}$ based on the input $\mathcal{X}$. In other words, we want to train a deep learning model $\mathcal{F}$ with learnable parameters Θ which learns a mapping $\mathcal{F}_\Theta : \mathcal{X} \mapsto \mathcal{Y}$, where $\mathcal{Y}$ represent the ground truth states, trying to minimize the error between the set of predicted states $\hat{\mathcal{Y}} = \mathcal{F}_\Theta(\mathcal{X})$ and the ground truth states $\mathcal{Y}$ according to a loss function $\mathcal{L}$. Therefore, the optimal set of parameters Θ^* for $\mathcal{F}$ is stated as in Equation 1:

$$\Theta^* = \arg\min_{\Theta} \ \mathcal{L}(\hat{\mathcal{Y}}, \mathcal{Y}) \tag{1}$$

We refer as **multi-step forecasting** the cases where $n > 1$, and as **single-step forecasting** when $n = 1$.

2.2 Forecasting Strategies

An in-depth analysis of possible forecasting strategies for multivariate time-series data can be found in Lim and Zohren (2021). Among the existing strategies, there is **direct forecasting**, in which a DL model is trained to forecast each of the n future steps (Makridakis et al., 2018), but it does not consider the relationships among the predicted states $\hat{X}_{t+i}$, also being limited to a maximum forecast horizon n. Another strategy is known as **iterative forecasting** (also called **recursive forecasting**), in which a trained model outputs one step ahead, then uses this output to comprise the inputs for the prediction of the next step, which yields the output for the subsequent step, and so on. However, this strategy is highly prone to error accumulation, especially for longer horizons in which all inputs are forecasted values rather than actual observations (Taieb et al., 2012).

To circumvent those issues, there are alternate strategies that predict multiple steps at the same time. The most widely used is known as **Multiple-Input Multiple-Output (MIMO)**, which preserves the dependency between the forecasted values, also reducing the error accumulation problem from the iterative strategy (up to time step $t + n$).

Besides being applied in time-series forecasting, this strategy has been widely used for spatiotemporal forecasting tasks, such as video prediction (Gao et al., 2022; Oprea et al., 2020) and earth science forecasting (Xu et al., 2021; Nguyen et al., 2023). Even though the forecasting horizon is limited by the model, one can perform recursive steps to yield predictions for longer time steps.

2.3 Deep Learning Methods

2.3.1 Encoder-Decoder ConvLSTMs

Convolutional Long Short-Term Memory Networks (ConvLSTMs) (Shi et al., 2015) are a class of neural networks designed to capture spatiotemporal dependencies in a time-evolving sequence, combining the strengths of CNNs and LSTMs into a single method. Given an input sequence $\mathcal{X}_t$, as well as the hidden state $\mathcal{H}_{t-1}$ and the cell state $\mathcal{C}_{t-1}$ from the previous time step, the gate mechanisms contained in a ConvLSTM cell – represented by i_t (input gate), f_t (forget gate), and o_t (output gate) – control the amount of information that is going to be propagated (or forgotten) for the next states. The process of computing the next hidden and cell states from an input sequence $\mathcal{X}$ can be mathematically described as shown in Equation 2, where σ is the sigmoid activation function, $*$ represents the convolution operations, $\circ$ stands for the Hadamard product (element-wise product), W and b are weights and biases for each of the gates and the cell state.

$$\begin{aligned}
i_t &= \sigma\left(W_{xi} * \mathcal{X}_t + W_{hi} * \mathcal{H}_{t-1} + W_{ci} \circ \mathcal{C}_{t-1} + b_i\right) \\
f_t &= \sigma\left(W_{xf} * \mathcal{X}_t + W_{hf} * \mathcal{H}_{t-1} + W_{cf} \circ \mathcal{C}_{t-1} + b_f\right) \\
\mathcal{C}_t &= f_t \circ \mathcal{C}_{t-1} + i_t \circ \tanh\left(W_{xc} * \mathcal{X}_t + W_{hc} * \mathcal{H}_{t-1} + b_c\right) \\
o_t &= \sigma\left(W_{xo} * \mathcal{X}_t + W_{ho} * \mathcal{H}_{t-1} + W_{co} \circ \mathcal{C}_t + b_o\right) \\
\mathcal{H}_t &= o_t \circ \tanh(\mathcal{C}_t)
\end{aligned} \tag{2}$$

To perform multi-step prediction with ConvLSTMs, a sequence of ConvLSTM cells is structured as an encoder-decoder architecture (also known as **Seq2Seq** model), where

both encoder and decoder blocks contain the same number of ConvLSTM cells. This type of architecture is inspired on sequence-to-sequence (Seq2Seq) models for natural language processing and time-series tasks (Bahdanau, 2014; Sutskever et al., 2014). An example is depicted in Figure 1, with two ConvLSTM cells in each block. After an input sequence $\mathcal{X}$ is processed by the encoder block, the final hidden state from the last ConvLSTM cell forms a compact representation of $\mathcal{X}$ (latent representation), which is fed to the decoder block to produce the hidden states for each output time step. Finally, a 3D convolutional layer receives the output from the decoder block and produces the predictions of all physical properties for each desired time step. More details about the encoding and decoding processes can be found in (Kakka, 2022).

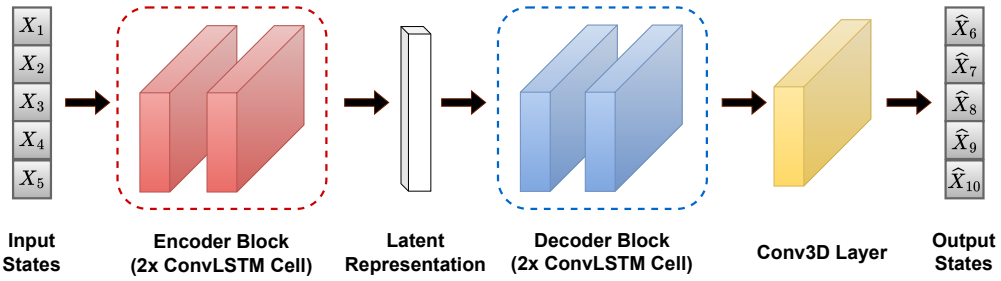


Figure 1: Encoder-Decoder structure of ConvLSTM cells for multi-step prediction.

2.3.2 U-Shaped Fourier Neural Operator

Proposed by Wen et al. (2022), the U-Shaped Fourier Neural Operator (U-FNO) is an extension of the Fourier neural operator (FNO) (Z. Li et al., 2020), designed to solve PDEs across diverse problems involving computational fluid dynamics. FNOs are known to be resolution-invariant, meaning that they can be trained on a lower resolution and evaluated on higher resolution, and yield superior performance against conventional CNNs by operating directly on the Fourier space (frequency domain), replacing convolution operations by pointwise multiplications, which are much faster and efficient.

Originally, the core idea of FNOs is to learn mappings of functions (also known as operators) defined in a continuous domain, rather than operating on discretized functions on 2-D grids, like images. Still, FNOs are able to operate on image-like data (as used in our work), despite not leveraging the full potential of the process of operator-learning.

To increase the overall performance in multiphase flow problems, the aforementioned authors introduced the U-Fourier layer as an upgrade of the original FNO architecture, combining the advantages of both CNN- and FNO-based models, increasing both training and test accuracies. On the other hand, this improvement on the accuracies comes in expense of the flexibility of training and testing at different resolutions. Moreover, U-FNO's usually take a longer time to be trained than traditional FNO's.

2.3.3 Temporal Attention Unit

Developed for video prediction tasks, Temporal Attention Units (TAUs) (Tan et al., 2023) leverage the ability to capture time evolution in image sequences by introducing parallelizable attention mechanisms that eliminate the need of recurrent-based units (such as RNNs and ConvLSTMs), speeding up the training process. In turn, the spatial modules are represented by simple 2D convolutions. Moreover, not only TAUs account for intra-frame differences through the mean squared error loss, but also for inter-frame variations

by embedding a differential divergence regularization term. The resulting loss function is expressed in Equation 3:

$$\mathcal{L}(\hat{\mathcal{Y}}, \mathcal{Y}) = \sum ||\hat{\mathcal{Y}} - \mathcal{Y}||^2 + \alpha \mathcal{L}_{reg}(\hat{\mathcal{Y}}, \mathcal{Y}) \tag{3}$$

where $\mathcal{L}_{reg}$ represents the Kullback-Leibler divergence between the probability distributions of the inter-frame differences from $\hat{\mathcal{Y}}$ and $\mathcal{Y}$, and α is a weight term defined empirically.

3 Methodology

Without loss of generality and assuming $m = 5$ input steps and $n = 5$ output steps, we first train a base network by making it receive a sequence of 5 "perfect" input states and predict the subsequent 5 states (i.e., without successive iterations), as done in a traditional MIMO approach. After the training process, to evaluate the full evolution of the dissolution process on a given simulation $\mathcal{S}$ containing N total steps, we conduct successive iterative predictions, as illustrated in Figure 2, by taking the first 5 input states from $\mathcal{S}$, yielding outputs at time steps 6-10. In turn, this output is fed as an input to the same base network to produce the states at time steps 11-15, and so on.

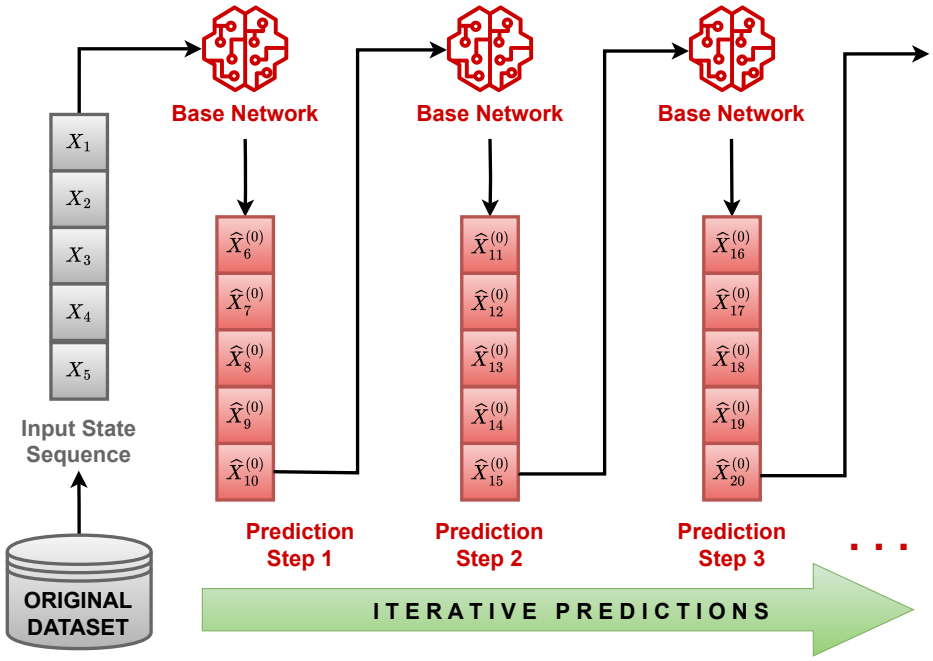


Figure 2: Flowchart of our iterative multi-step approach with network stacking for full evaluation of a reactive dissolution process on simulation data.

To improve the initial solution produced by the base network, we propose a multi-level stacking of trained networks. In this case, the same neural network architecture is used to train each level, with the same sizes for both input and output tensors as the base network, as well as the loss function. Considering the base network as Level 0, trained with a dataset $\mathcal{D} = \{\mathcal{S}_1, \mathcal{S}_2, ..., \mathcal{S}_{ts}\}$, where ts is the number of training samples, we generate a new dataset comprised of all possible output sequences $\hat{\mathcal{Y}}$ computed by the base network for each $\mathcal{S}_i \in \mathcal{D}$, representing initial approximations to the true states $\mathcal{Y}$. The Level 1 network

is then trained to receive, as input, one of those responses produced by the base network, and outputs a new approximation (correction) to the ground truth within the same time-step interval, hypothesizing that it will yield less errors than the initial solution. Following this rationale, we can continue the stacking process by training a Level 2 network with the outputs produced by Level 1, and so on until no further improvement is observed. During this process, only one level can be trained at a time.

Conceptually, this multi-level stacking strategy should be interpreted as a sequential residual-correction procedure rather than as a single deeper end-to-end network. The base network first produces an approximate prediction over the target horizon, and each subsequent level is trained on the prediction distribution generated by the previous level in order to learn a correction over the same interval. In this sense, the later levels are exposed to structured residual errors that arise from the preceding predictor, which differs from simply increasing the depth or width of a single jointly optimized architecture. The usefulness of this strategy is expected to depend on the existence of a learnable residual structure in the errors of the previous level. Conversely, if the base model already captures the dynamics well, or if the residuals become dominated by noise and distribution shift, additional correction levels may provide limited gains or even degrade performance.

In general, let L be the number of levels of correction to be applied on the base solution. In the present study, we evaluate corrections up to $L = 3$ as a practical upper bound to examine whether improvements persist beyond the first correction while keeping the overall training cost manageable; however, this value should be regarded as a tunable design choice rather than an optimal or universal setting. Increasing the number of correction levels may also increase the risk of overfitting, particularly when the residual errors become less structured at higher levels. After the base and all of the L correction networks are trained, the iterative multi-step prediction process can be conducted as illustrated in Figure 3 with $L = 3$. Here, the first 5 input states of a simulation sample are fed into the base network, producing an initial approximation for the subsequent 5 states. Then, this output is fed to the Level 1 network which produces the first level of correction for the initial approximation. Later, this output serves as input to Level 2 network, whose output is forwarded as an input to Level 3 network, which yields the final prediction for time steps 6-10. For the next prediction step, the output from the last network of the stack is fed to the base network to produce an initial approximation for steps 11-15, which is corrected by the subsequent levels in the stack. The whole process is repeated until the simulation end time.

Although the methodology was illustrated for $m = n = 5$, it can be easily adapted to receive any m input states and predict n output states. For greater values of m and n, the overall performance of the framework tends to drop due to larger error accumulations. Conversely, lower values of those variables yield more accurate predictions, despite leading to a longer training process and total inference time to predict all time steps.

4 Reactive Dissolution Dataset and Preprocessing

Our reactive dissolution dataset (Cirne et al., 2025) consists of 32 numerical simulations generated using `GeoChemFoam` (Maes et al., 2022), an open-source micro-continuum solver for pore-scale reactive transport. Each simulation follows the *Benchmark 2* configuration presented by Maes et al. (2022), which models calcite dissolution in a quasi two-dimensional micromodel. The computational domain consists of a polydisperse disk pack with an initial porosity of approximately $\phi_0 = 0.45$ and an initial permeability of $K_0 = 5.6 \times 10^{-10}\,\mathrm{m}^2$. Simulations are performed on a uniform Cartesian mesh with a spatial resolution of $\Delta x, y = 25\,\mu\mathrm{m}$ and a $\Delta z = 150\,\mu\mathrm{m}$.

The flow field is incompressible and governed by the Darcy-Brinkman-Stokes equations under isothermal conditions, while the transport of a single reactive species obeys an advection–diffusion equation with a surface reaction rate $R = k_c\, c$ applied at the fluid–

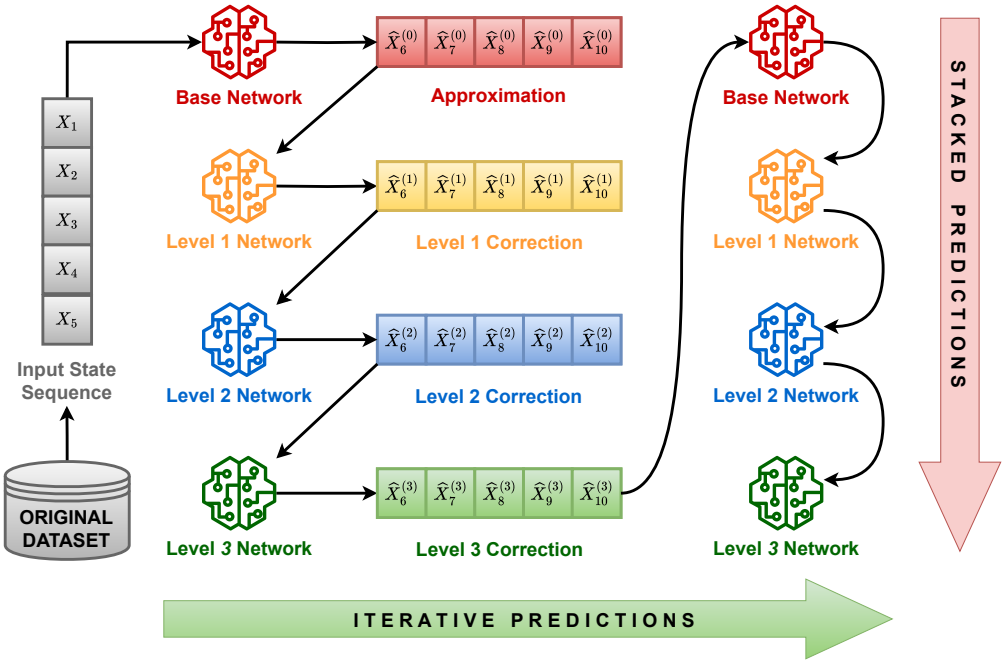


Figure 3: Flowchart of our iterative multi-step approach with network stacking for predicting reactive dissolution.

solid interface. A representative parameter set corresponding to the *conical wormhole regime* is as follows: kinematic viscosity $\nu = 1 \times 10^{-6}\,\mathrm{m}^2/\mathrm{s}$, diffusion coefficient $D = 1 \times 10^{-9}\,\mathrm{m}^2/\mathrm{s}$, inlet concentration $c_i = 0.01\,\mathrm{kmol}/\mathrm{m}^3$, stoichiometric coefficient $\zeta = 1$, calcite density $\rho_s = 2710\,\mathrm{kg}/\mathrm{m}^3$, molar mass $M_{ws} = 100\,\mathrm{kg}/\mathrm{kmol}$, Kozeny–Carman constant $k_f = 2 \times 10^{11}\,\mathrm{m}^2$, a volumetric flow rate $Q = 5.2174 \times 10^{-12}\,\mathrm{m}^3/\mathrm{s}$, and reaction rate constant $k_c = 8.1632 \times 10^{-6}\,\mathrm{m}/\mathrm{s}$. These conditions correspond to Péclet and kinetic numbers of $\mathrm{Pe} = 1$ and $\mathrm{Ki} = 1$, respectively, representing the transition between compact and wormhole dissolution regimes. Boundary conditions follow Maes et al. (2022): a constant inflow rate at the inlet, fixed outlet pressure, and no-slip/no-flow boundaries on the top and bottom walls.

The 2D simulation results are stored at 100 equally spaced time steps, resulting in time-evolving dissolution state maps of size 260×260. For the ML algorithms tested in this work, all maps were cropped to 256×256 by removing the first and last two rows and columns from each state map. Each of these simulations conveys a particular rock sample with its own pore structure, as well as a distinct fluid trajectory over the course of the dissolution process. An example is illustrated in Figure 4. These maps encompass four different input properties: $\mathbf{C}$, the concentration of the acidic solution used in the dissolution; **eps**, a function indicating the volume fraction of the pore space occupied by pore in each voxel, which is proportional to the amount of fluid contained in each voxel; $\mathbf{U}_x$, the magnitude and direction of flow of the acidic solution in the horizontal axis of a 2-D Cartesian plane; and similarly $\mathbf{U}_y$, for the vertical axis. The average time to produce each simulation was approximately 3 hours using 24 CPUs of 3 GHz each. All simulations employ the same flow and reactive transport conditions, where both Péclet and kinetic numbers are equal to 1.

We note that, although each simulation provides a large number of spatial samples over time, the present dataset contains only 32 distinct pore geometries generated within a single benchmark family. Therefore, this dataset should be interpreted as a controlled test bed for

methodological evaluation rather than as an exhaustive representation of the broader class of pore structures encountered in reactive porous media. In our train/validation split, complete simulations (rather than individual time steps) were separated, so the validation set consists of unseen pore geometries that were not used during training. As a result, the reported performance reflects generalization to new realizations within the same benchmark configuration, but does not yet establish robustness across substantially different pore-structure classes, reaction regimes, or boundary conditions. Extending the dataset to cover broader geometric descriptors and multiple flow-reactive regimes remains an important direction for future work.

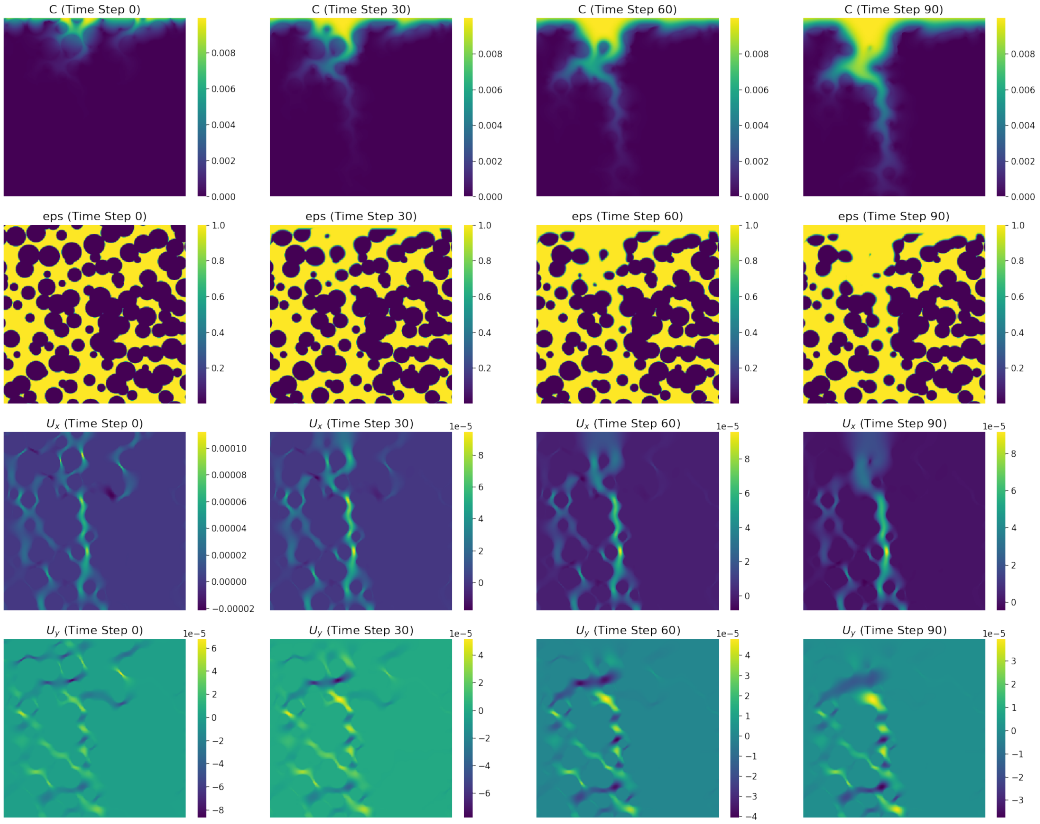


Figure 4: Input features from one dataset sample representing the dissolution states for time steps 0, 30, 60 and 90 (from left to right).

In essence, GeoChemFoam solves the reactive transport in porous media in a quasi-static regime, as described by Maes et al. (2022). Given a porosity map **eps** at a time step t, it first calculates the steady-state velocity (i.e., it produces $\mathbf{U}_x$ and $\mathbf{U}_y$ at time step t). Then, it calculates a steady-state concentration **C** at time step t, which includes the contributions of injection, diffusion, and reactive dissolution at the fluid-solid interface. From these calculations, it estimates a local reaction rate in each computational fluid and the new **eps** field for the time step $t+1$, which in turn is used to calculate the flow once again to produce a new **eps** map at time step $t+2$, and so on.

Aside from the original features, three extra input features were engineered as an attempt to improve the performance of all DL models: 1) **Magnitude of velocity**: defined as $U = \sqrt{U_x^2 + U_y^2}$; 2) **C Scaled**, a log-transformation over the concentration values; and 3) **Combined Filter**, a binary mask based on C and *eps* constraints which shows the portions

of the grains in a porous matrix that are being dissolved at a particular time step t. Given a position (i, j) in a map, the combined filter is calculated according to Equation 4:

$$\text{Filter}(i,j) = \begin{cases} 1 & \text{if } C(i,j) \geq 10^{-4} \text{ and } 0.01 \leq eps(i,j) \leq 0.99, \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

5 Performance Evaluation of Deep Learning Models

In this section, we will discuss the quantitative and qualitative results of our proposed method on all algorithms described in Section 2.3. The source code is available at `https://github.com/ai4netzero/ReactiveDissolution`.

5.1 Training and Evaluation Settings

From the 32 simulation samples in our dataset, 24 were randomly selected as the training set and the remaining 8 as the validation/test set. Before the DL models are trained, the entire data is normalized with respect to the mean and standard deviation values of the training set. All input features, except for C Scaled and the Combined Filter, have their original values subtracted by their respective means and the results are each divided by their respective standard deviations. On the other hand, a min-max normalization is applied to the output values so that they all belong within the $[0, 1]$ range, making it suitable to adopt the sigmoid activation function for all DL models.

Concerning our proposed multi-level stacking of neural networks method, for each DL algorithm described in Section 2.3, we perform corrections up to Level $L = 3$. The models were trained on a NVIDIA Titan RTX with 24 GB of memory for a total of 100 epochs (along with a patience rate of 20 epochs for early stopping), using a batch size of 4, a learning rate of 0.0005 and the Adam optimizer (Kingma, 2014) with moment values $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The mean squared error was adopted as the loss function for ConvLSTM, U-FNO and the intra-frame difference term of TAU. Regarding the latter, the α constant for the regularization term (inter-frame difference) was fixed at 0.1.

For the quantitative evaluation of the iterative predictions of each output property, we calculate the Pearson correlation coefficient (PCC) to assess the similarity between the predicted state $\hat{Y} \in \mathbb{R}^{H \times W}$ and the ground truth state $Y \in \mathbb{R}^{H \times W}$ at each time step, as stated in Equation 5:

$$\text{PCC}(\hat{Y}, Y) = \frac{\text{cov}(\hat{Y}, Y)}{\sigma_{\hat{Y}} \cdot \sigma_Y} \tag{5}$$

where $\text{cov}(\hat{Y}, Y)$ is the covariance between $\hat{Y}$ and Y and σ represents the standard deviation of a state map.

It is important to emphasize that the present inference framework is purely data-driven and does not explicitly enforce physical constraints or conservation laws during prediction. Accordingly, the long-horizon results reported here should be interpreted as surrogate predictions within the range of conditions represented by the training data, rather than as guaranteed physically constrained forecasts under arbitrary reactive transport settings. Incorporating physics-aware constraints into the training objective or architecture is a promising avenue for improving robustness in future work.

5.2 Model Statistics

Table 1 shows some statistics about the average training epoch times, numbers of trainable parameters, final model sizes and average forward times (i.e., the elapsed time

for processing input data through a network to produce the next 5 time steps) for each DL model. Although TAU has the highest numbers of trainable parameters and a relatively larger model size, it still achieved the lowest training and forward times, due to its recurrence-free structure.

DL Model	Epoch Train Time (min)	Total Train Time (h)	Trainable Parameters	Model Size (MB)	Forward Time (ms)
ConvLSTM	6.44	10.73	**4 M**	**13**	60
U-FNO	10.98	18.30	6 M	25	79
TAU	**4.22**	**7.03**	11 M	135	**33**

Table 1: Performance statistics for all tested models. Best results for each category are highlighted in bold.

Despite being the most lightweight model, ConvLSTM has a 50% slower training time per epoch than TAU, also taking nearly twice as long for the forward operation. Finally, U-FNO achieved the slowest training and forward times, but the resulting model after the training process is about five times smaller than TAU and almost two times bigger compared to ConvLSTM.

Analyzing the total runtime of each algorithm to produce a 100-step simulation, TAU takes an average time of 0.66 seconds, followed by ConvLSTM (1.2 seconds) and U-FNO (1.58 seconds). These represent a speedup with an order of magnitude between 10^3 and 10^4 when compared to the average time taken by GeoChemFoam (3 hours).

5.3 Iterative Prediction and Model Stacking Analysis

Figure 5 shows the average correlations of the iterative predictions on all training samples for all algorithms and correction levels. For all properties, it can be observed that both TAU and U-FNO are more robust to error accumulation, with the latter achieving slightly higher correlations at late time steps. Considering the effects of model stacking, ConvLSTM achieved its best results at Level 2 correction (except for U_x prediction). Still, it was not capable of performing better than the base networks (Level 0) of the other two algorithms. TAU showed no significant improvement when applying multi-level stacking for all cases, as evidenced by the fact that the correlations do not systematically increase over the correction levels. On the other hand, U-FNO showed a more consistent evolution from the base network to Level 3 in all scenarios, as the average correlations at the late time steps typically increase after each correction.

Figure 6 illustrates the results for the validation set. The plots show that TAU achieved the best iterative predictions for all cases (except for U_x prediction). Apart from ConvLSTM, the correlations had a reasonable drop when compared to those from the training data. However, this drop is much larger for the U-FNO, suggesting that this network is much more prone to over-fitting than the others.

Regarding the performance evolution along the correction levels, ConvLSTM still yielded its best results on Level 2 network, except for U_x prediction, in which Level 3 was more robust to error accumulation. However, it achieved the lowest correlations in almost all scenarios, being only superior to U-FNO in the prediction of U_y. The validation curves for U-FNO, unlike the results for the training set, did not show a consistent evolution over the levels. For instance, in the prediction of C, the Level 1 network achieved the best results until time step 80, where it was surpassed by Level 3. Moreover, Level 1 prevailed as the best correction network during all time steps of *eps* prediction. Finally, the Level 1 network from TAU achieved the highest correlations for predictions of C and *eps*, as well as for

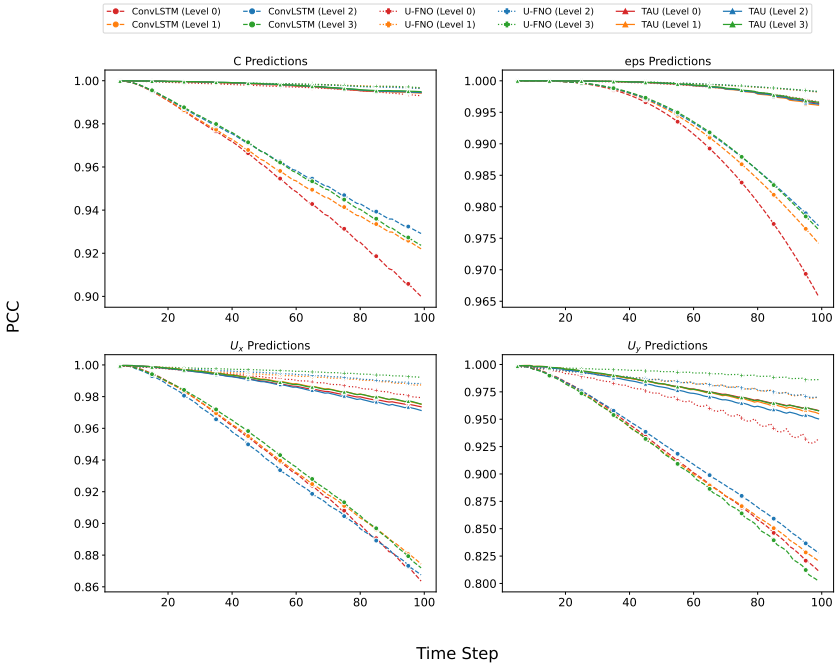


Figure 5: Average correlation scores of all samples from the training set for the iterative predictions produced by each algorithm.

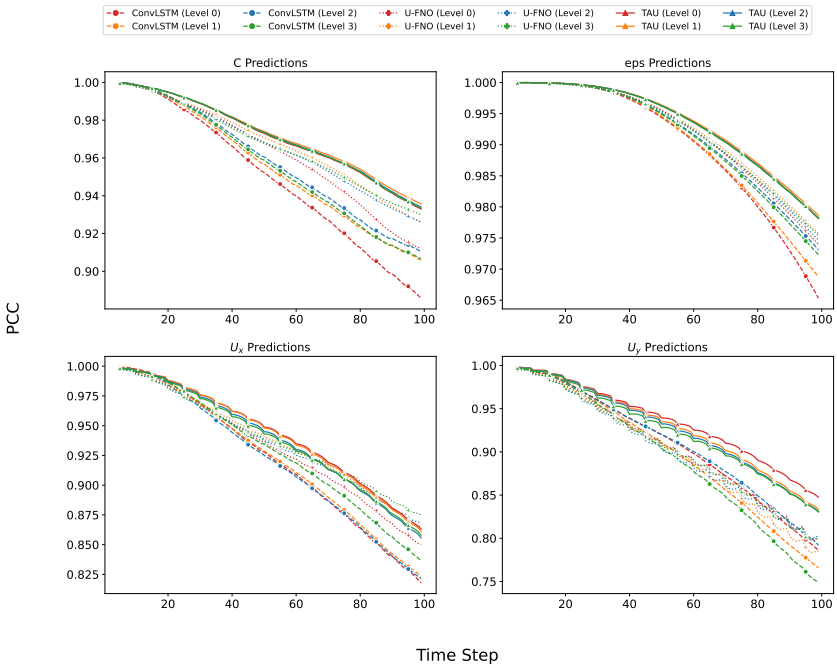


Figure 6: Average correlation scores of all samples from the validation set for the iterative predictions produced by each algorithm.

U_x predictions until time step 50. Despite its better performance against ConvLSTM and U-FNO, the multi-level stacking strategy did not cause a significant improvement over the TAU Level 0 network. As the algorithms were trained using the MSE loss function, a more carefully designed loss function, tailored to capture reactive transport properties, could potentially lead to more systematic improvements, especially in achieving progressively better results at higher correction levels.

More broadly, the results indicate that the effectiveness of the multi-level stacking strategy is architecture-dependent in the present study. For ConvLSTM, the base predictor appears to leave residual errors that remain structured enough to be partially corrected by the subsequent levels, which explains the more consistent gains observed across the correction stages. For TAU, the Level 0 predictor already performs strongly, leaving less room for systematic improvement; in that case, the remaining residuals may be too small or too irregular for later correction networks to provide noticeable benefits. In contrast, the validation behavior of U-FNO is less consistent, suggesting that its correction levels may be more sensitive to distribution shift and overfitting. Therefore, the proposed stacking framework should not be interpreted as uniformly beneficial across all model classes, but rather as a correction mechanism whose utility depends on the residual-error structure left by the base architecture and on the chosen training objective.

To ratify the robustness of our method , we also evaluated our regression results with a complementary metric (MSE), as described in Appendix A. Typically, MSE and PCC capture different aspects of performance in regression tasks. The former is mostly associated with numerical accuracy, which is important in forecasting tasks, while the latter analyzes directions and trends of predictions, regardless of scale. This is particularly useful in our case, as we are predicting four different property fields.

5.4 Relevance of Engineered Features

Figure 7 illustrates the effects of including the three engineered features described in Section 4 along with the four original input properties. To provide a clear visualization of these effects, we will only discuss the results with respect to Level 0 networks. From the plots, we can notice that for both ConvLSTM and U-FNO, there is a performance drop when transitioning from 7 to 4 input features, except for U_x predictions for ConvLSTM. Moreover, U-FNO with 4 input features achieved better results than ConvLSTM with 7 input features.

Conversely, TAU yielded slightly better performances without the engineered features, also achieving superior results than the other two methods. However, a further analysis with respect to its internal parameters (including the α term of the loss function) must be conducted to confirm whether the engineered features are actually relevant. Nevertheless, the plots demonstrate TAU's capability to understand the evolution of the reactive dissolution process with a smaller set of input features.

5.5 Qualitative Analysis

In this section, we will conduct a qualitative analysis on a sample from the validation set, comprising a particular pore geometry. Herein, we will consider the predictions at the last time step (100) after several iterative steps are performed for each algorithm. To simplify our analysis, we only show the predictions for C and *eps* fields. Figure 8 shows the ground truth maps for those two properties at time steps 0 and 100.

The predictions of C for all algorithms and all network levels are displayed in Figure 9, along with their respective difference maps to the ground truth ($\hat{Y} - Y$). We can notice a clear evolution on the ConvLSTM results, as the results are Level 0 contain too much noise, which is mostly mitigated over the subsequent levels. However, there is a slight decrease in the PCC score from Level 2 to Level 3. Regarding U-FNO, the best results were achieved

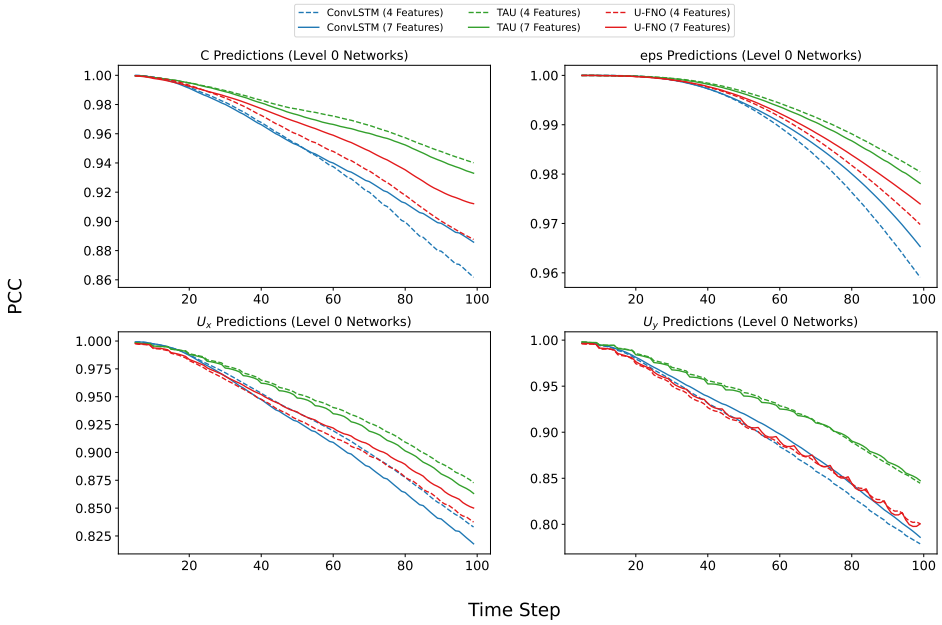


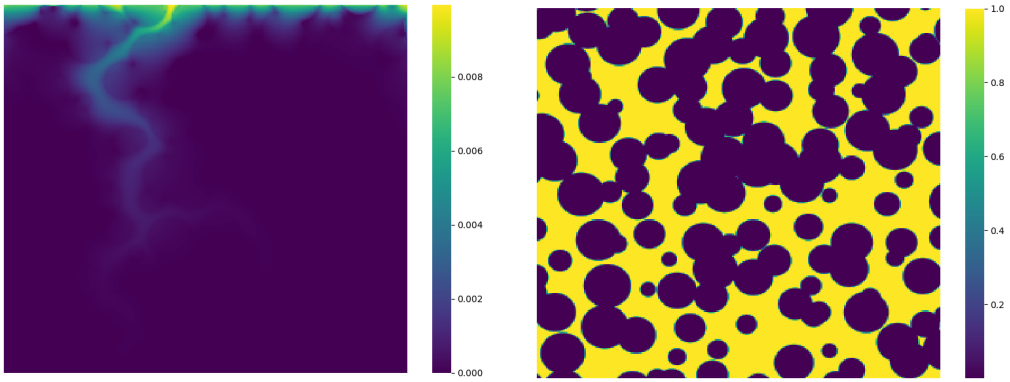
Figure 7: Average correlations for all models in the validation set and each output property, comparing the performances of all Level 0 networks with and without the engineered features, respectively represented by solid and dashed lines.

at Level 0, which yielded a PCC of 0.004 lower than ConvLSTM Level 2. Moreover, all levels produced similar shapes of dissolution channel, only missing the right branching at the bottom-center part of the map. Finally, although not having benefitted from the multi-level stacking approach, TAU achieved the highest scores for all levels, where even its Level 0 network performed better than any other level from both ConvLSTM and U-FNO. Looking at the difference maps, we can also notice a smaller range of errors compared to the other two algorithms. With respect to the dissolution shape, it was able to capture some of the right branching, but not as much as ConvLSTM did.

Figure 10 shows the predictions for *eps* at time step 100. In general, all networks managed to yield nearly perfect predictions when compared to the ground truth. The PCC scores for ConvLSTM strictly increase over all levels. At Level 0, we can notice a prevalence of overpredictions all over the map, which is mitigated at the subsequent levels. On the other hand, neither U-FNO nor TAU showed significant differences from one level to another, with the latter achieving the best PCC scores at each level, having only tied with ConvLSTM at Level 3.

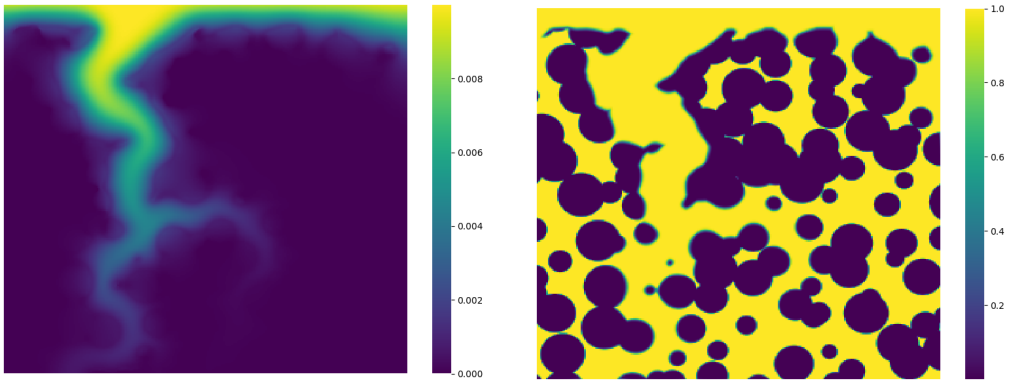
5.6 Estimation of Bulk Properties

To convey a better understanding of the overall dissolution dynamics in our case study, we analyze the evolution of two of the bulk properties used for modelling reactive dissolution at the field-scale: porosity and permeability. Such an analysis is intended to check the physics-consistency of our trained models, rather than an attempt to directly learn from those bulk properties. For each sample from our validation set, we selected 10 *eps* maps, starting at time step 5, with an offset of 10 time steps between each pair of consecutive maps, ending at time step 95. We repeated this process for the *eps* maps predicted by each of the DL algorithms at the aforementioned time steps. The porosity and permeability



(a) True C (Time Step 0)

(b) True *eps* (Time Step 0)



(c) True C (Time Step 100)

(d) True *eps* (Time Step 100)

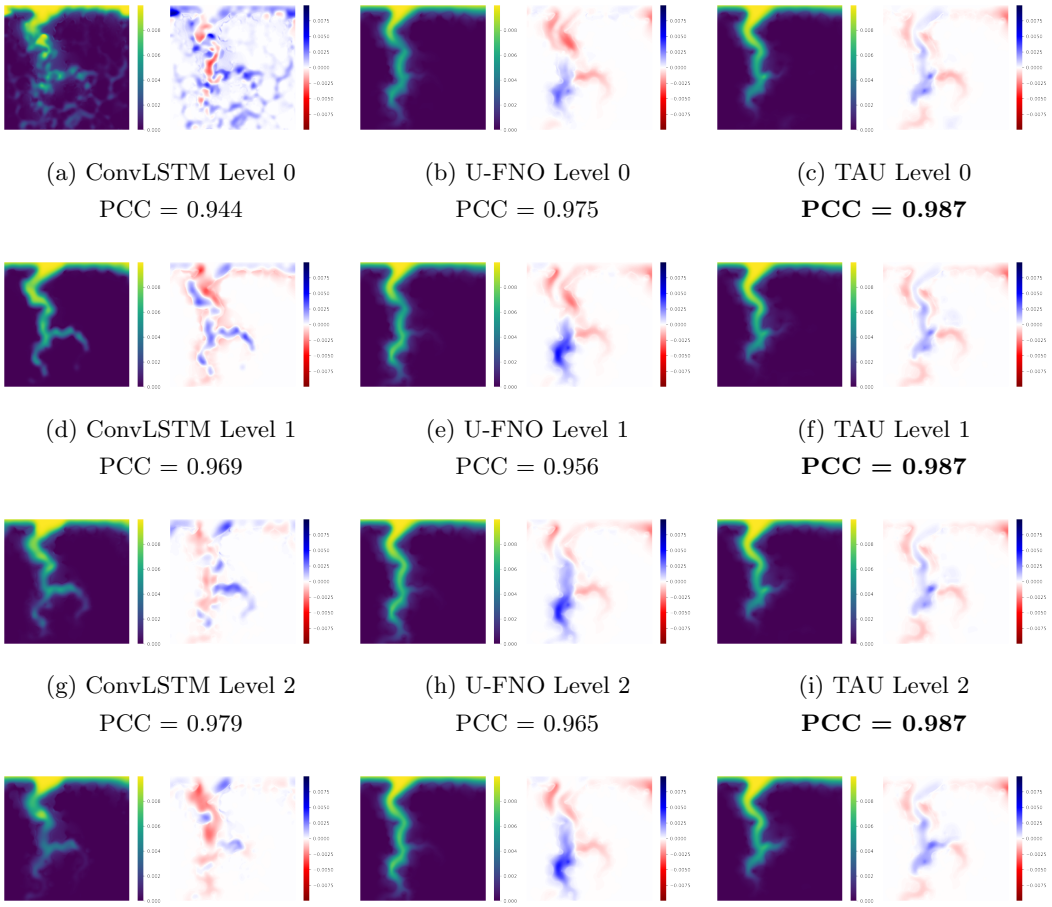
Figure 8: Ground truth maps at time steps 0 and 100 for C and *eps*.

values for all maps are then calculated using GeoChemFoam, and the results achieved by each algorithm are compared against the ones obtained from the original data.

For all cases discussed in this section, we will show how the errors of porosity and permeability for each algorithm evolve due to the dissolution process in terms of: 1) their respective average values; and 2) the RMSE error versus the values from the ground truth (referred as "Original Data" in the subsequent plots).

5.6.1 Porosity Estimation

Figure 11 shows the average error in porosity considering all models from the training set. To enhance the overall visualization of the average error curves, we will only display the values from time step 50 onwards. Looking at the average evolution over the sampled time steps (Figure 11a), we observe that the curves from U-FNO and TAU are the closest from the original porosity values, especially at late time steps, where the error tends to be higher due to the error accumulation problem of the iterative approach. Conversely, the curves from ConvLSTM are farther away from the ground truth, even though it shows some improvement over the network levels. In the RMSE curves (Figure 11b), we observe that TAU Level 3 achieved the lowest errors on time steps 25 and 35, but ends up with a higher error than its Level 0 counterpart and all U-FNO variants (except U-FNO Level 0).



(a) ConvLSTM Level 0
PCC = 0.944

(b) U-FNO Level 0
PCC = 0.975

(c) TAU Level 0
PCC = 0.987

(d) ConvLSTM Level 1
PCC = 0.969

(e) U-FNO Level 1
PCC = 0.956

(f) TAU Level 1
PCC = 0.987

(g) ConvLSTM Level 2
PCC = 0.979

(h) U-FNO Level 2
PCC = 0.965

(i) TAU Level 2
PCC = 0.987

(j) ConvLSTM Level 3
PCC = 0.976

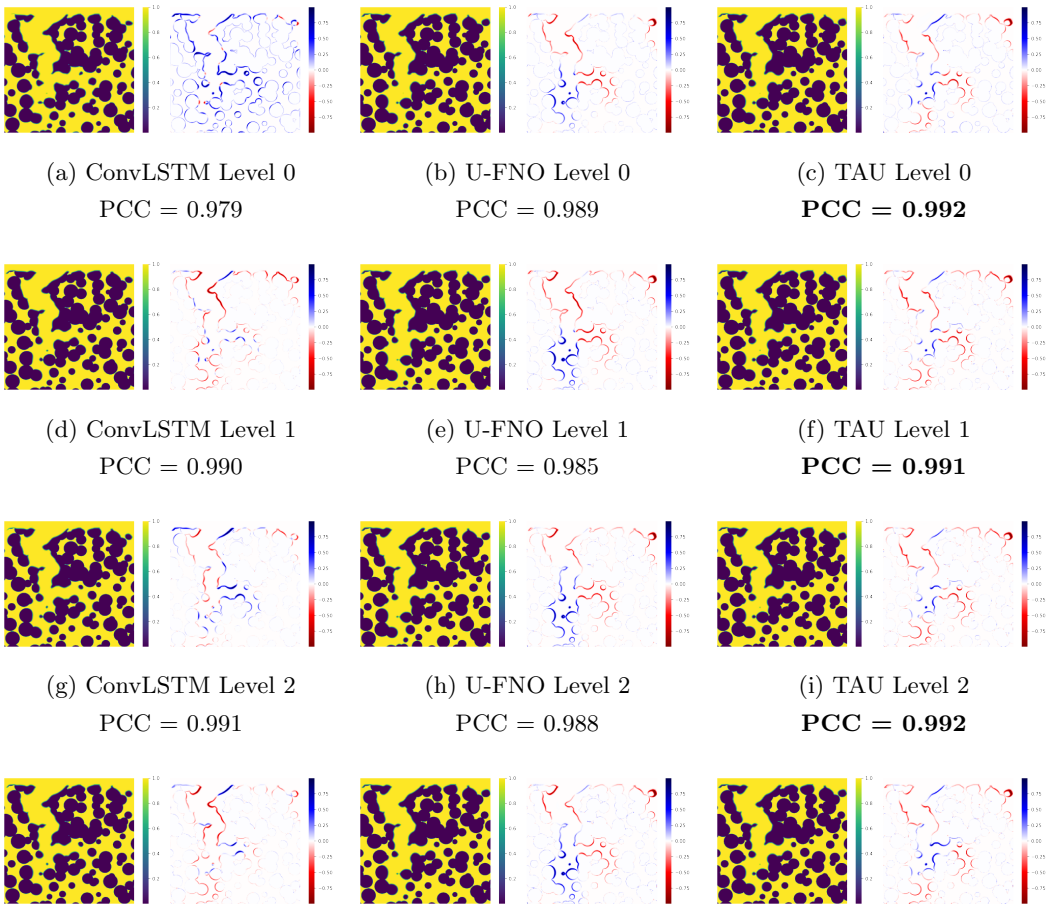
(k) U-FNO Level 3
PCC = 0.961

(l) TAU Level 3
PCC = 0.986

Figure 9: Predictions of C at time step 100 for a case from the validation set, along with their respective PCC scores and difference maps to ground truth (Figure 8c). Best results for each network level are in bold.

Both ConvLSTM and U-FNO showed a more consistent evolution over the network levels, whereas the U-FNO Level 3 having lowest errors at the end of the dissolution.

For the validation set, similar results as the training set are achieved for the average error plots (Figure 12a). Here, TAU Level 3 was the closest curve to the ground truth at the end of the dissolution, although the curves from its remaining levels and the ones from U-FNO were slightly farther away, but still constantly lying inside the error interval. Concerning the RMSE plots (Figure 12b), we can also notice an error reduction on higher network levels for all algorithms, especially after time step 45. This reduction is more noticeable for ConvLSTM, whose Level 2 network achieved significantly lower results than its Level 0 counterpart, at the same time it yielded the lowest errors between time steps 25 and 55. TAU produced very close results among all levels, where Level 2 achieved the lowest error rates at late time steps, but still being slightly worse than ConvLSTM Level 2. Last, U-FNO showed the highest errors at early time steps (5 to 25), achieving the lowest error rates for late time steps (65 to 95).



(a) ConvLSTM Level 0
PCC = 0.979

(b) U-FNO Level 0
PCC = 0.989

(c) TAU Level 0
PCC = 0.992

(d) ConvLSTM Level 1
PCC = 0.990

(e) U-FNO Level 1
PCC = 0.985

(f) TAU Level 1
PCC = 0.991

(g) ConvLSTM Level 2
PCC = 0.991

(h) U-FNO Level 2
PCC = 0.988

(i) TAU Level 2
PCC = 0.992

(j) ConvLSTM Level 3
PCC = 0.992

(k) U-FNO Level 3
PCC = 0.987

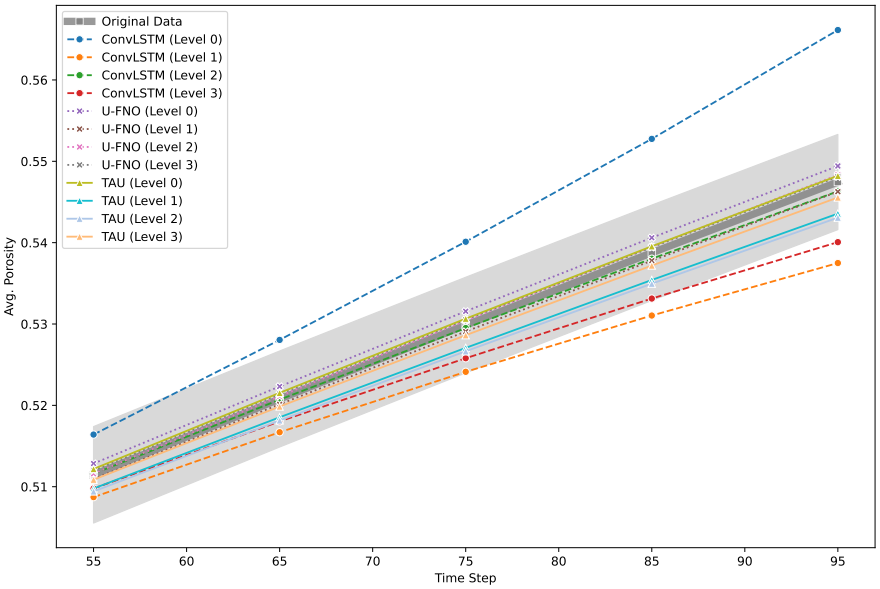
(l) TAU Level 3
PCC = 0.992

Figure 10: Predictions of *eps* at time step 100 for a case from the validation set, along with their respective PCC scores and difference maps to ground truth (Figure 8d). Best results for each network level are in bold.

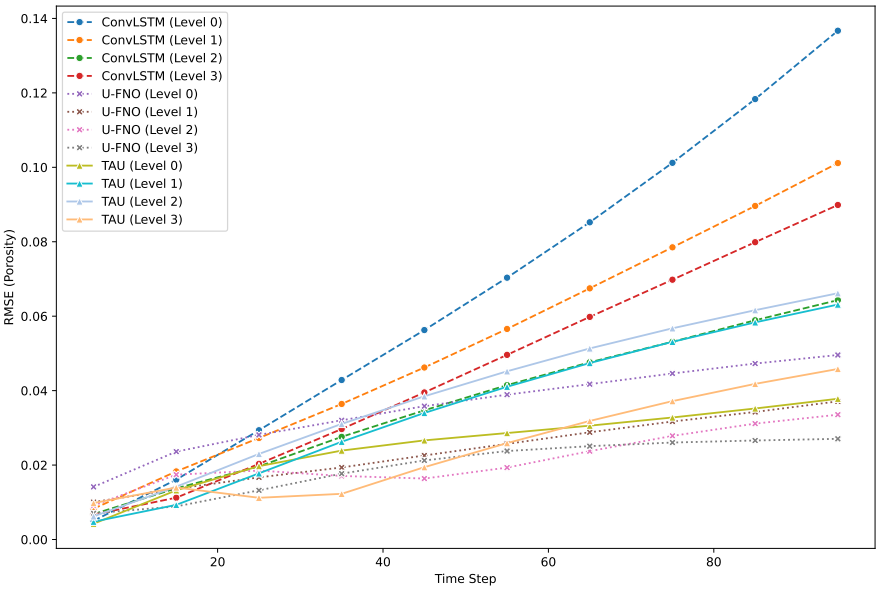
5.6.2 Permeability Estimation

Figure 13 showcases the results for the permeability estimation on the training set. From the curves in both plots, we can observe that the multi-level stacking was not enough to produce a consistent evolution over the levels for all algorithms. This is corroborated by the fact that the closest curves to the ground truth (Figure 13a) were obtained from TAU Level 0 and U-FNO Level 0. Moreover, when analyzing the RMSE curves for each algorithm (Figure 13b), ConvLSTM Level 3 yielded the worst results during all the dissolution steps. Regarding the TAU curves, the Level 3 network achieved the lowest error rates until the last time step, where the Level 0 network was slightly better. For the U-FNO, the Level 3 network was the best among all networks between time steps 25 and 65, being later surpassed by its Level 1 counterpart.

With respect to the estimation on the validation set, the evolution curves from Figure 14a show that the U-FNO curves were the closest ones to the ground truth. Unlike the training set, the TAU curves were farther away from the ground truth, where the Level 0

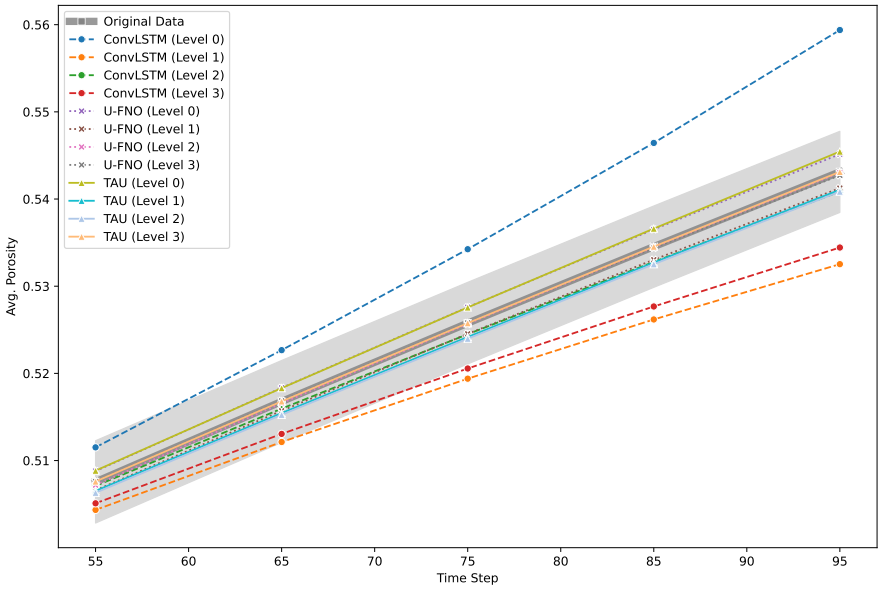


(a) Absolute Porosity Evolution (Training Set)

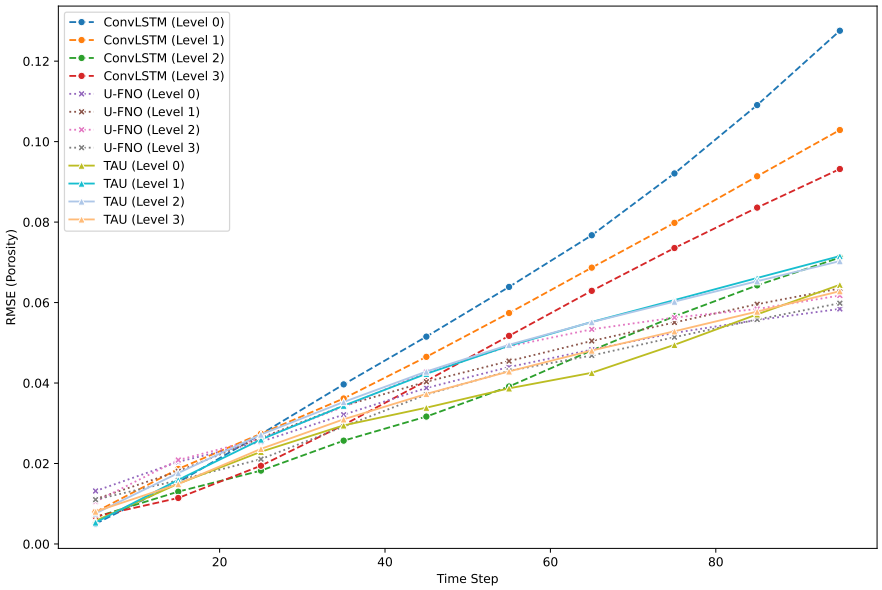


(b) Porosity Error Evolution vs. Original Data (Training Set)

Figure 11: Porosity error analysis, considering the averages for each time step across all samples from the training set. (a) absolute porosity evolutions vs. ground truth (with error margin bounds in grey); (b) RMSE evolutions.

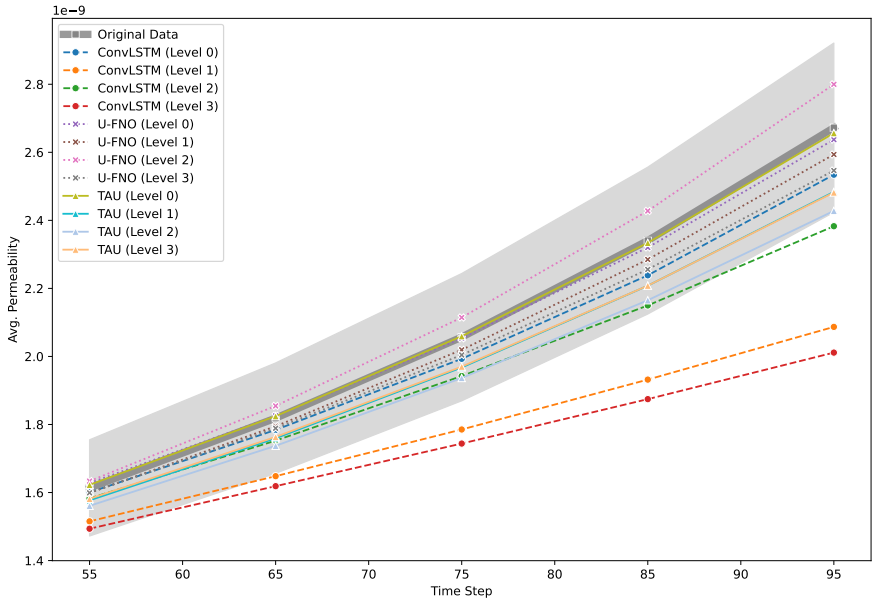


(a) Absolute Porosity Evolution (Validation Set)

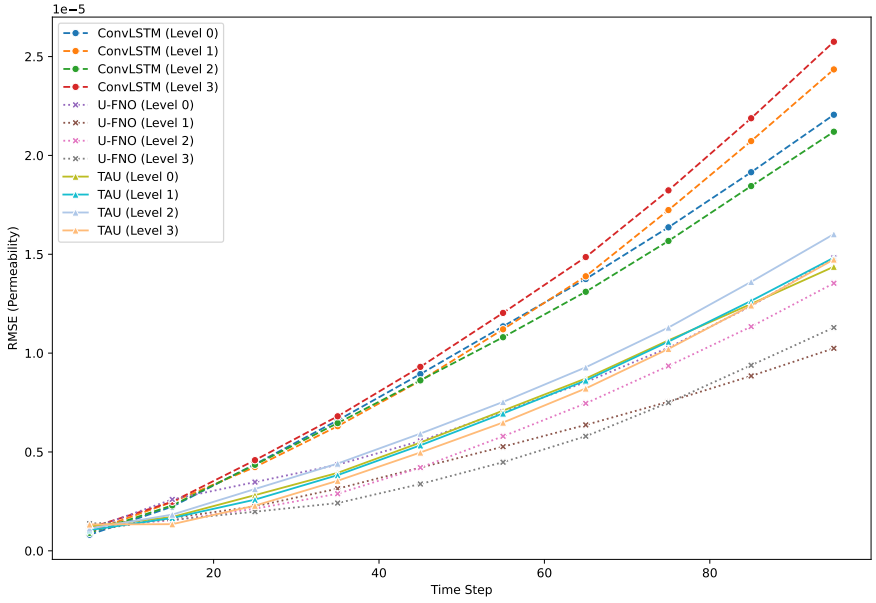


(b) Porosity Error Evolution vs. Original Data (Validation Set)

Figure 12: Porosity error analysis, considering the averages for each time step across all samples from the training validation set. (a) absolute porosity evolutions vs. ground truth (with error margin bounds in grey); (b) RMSE evolutions.

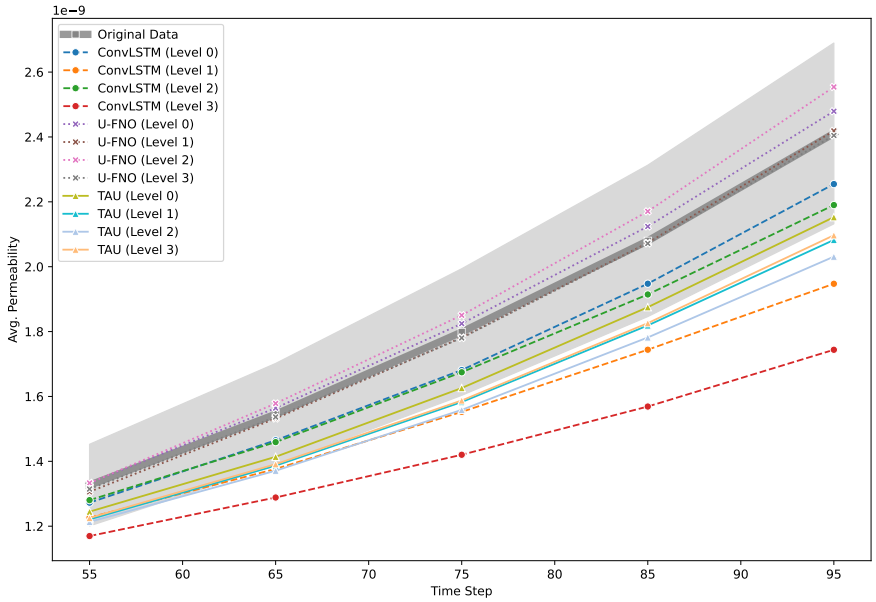


(a) Absolute Permeability Evolution (Training Set)

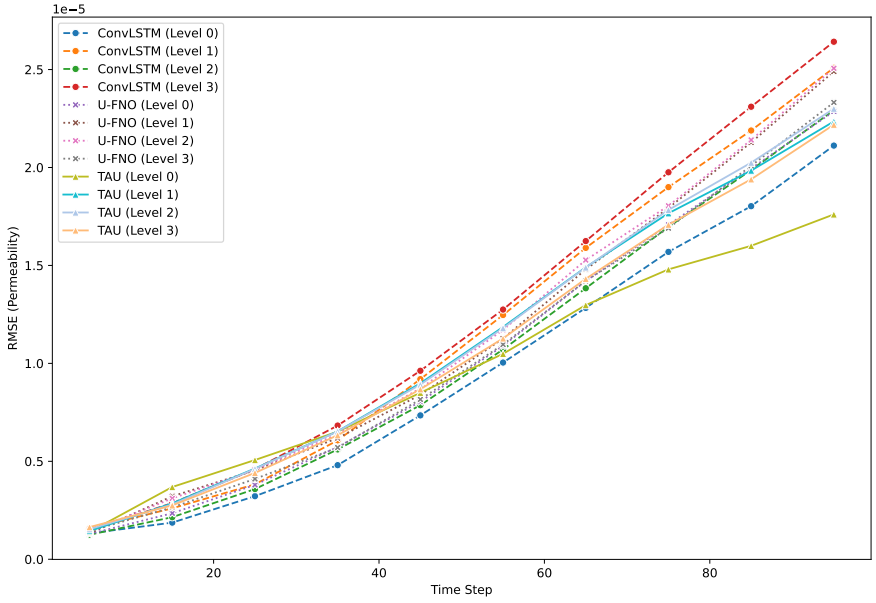


(b) Permeability Error Evolution vs. Original Data (Training Set)

Figure 13: Permeability error analysis, considering the averages for each time step across all samples from the training set. (a) absolute permeability evolutions vs. ground truth (with error margin bounds in grey); (b) RMSE evolutions.



(a) Average Permeability Evolution (Validation Set)



(b) Permeability Error Evolution vs. Original Data (Validation Set)

Figure 14: Permeability error analysis, considering the averages for each time step across all samples from the validation set. (a) absolute permeability evolutions vs. ground truth (with error margin bounds in grey); (b) RMSE evolutions.

curve was the only one to remain inside the error margin bounds during all the dissolution. About ConvLSTM, the best results were achieved by its Level 0 network.

The discrepancy between the high voxel-wise agreement observed for the *eps* maps and the less systematic behavior of the derived permeability estimates is physically plausible. Although the predicted *eps* fields often achieve very high similarity and correlation scores, permeability is a highly non-linear and connectivity-sensitive bulk property that can be strongly affected by localized errors in narrow flow pathways, channel connectivity, and hydraulically important throat regions. Consequently, small spatial discrepancies in critical regions may lead to non-monotonic changes in permeability, even when the overall voxel-wise agreement remains high. Since the networks in this work were not trained with an explicit objective on bulk properties, improvements in local field prediction do not necessarily translate into monotonic improvements in macroscopic flow estimates across stacking levels.

Considering the RMSE curves from Figure 14b, the lowest errors were achieved by ConvLSTM Level 0 (time steps 15 to 45) and TAU Level 0 (time steps 55 to 95). Furthermore, ConvLSTM Level 3 produced the highest errors between time steps 45 to 95. For the same interval, U-FNO Level 3 achieved the lowest results among all its levels, although still being slightly worse than TAU Level 0.

6 Conclusions

This paper presented a data-driven method that leverages deep learning to predict the temporal evolution of reactive dissolution in porous media within a controlled benchmark setting. An iterative stacked MIMO approach was adopted to produce the future states of a dissolution process, starting with an initial set of "perfect" inputs, generating an output which is then used as input to predict the subsequent states, and so on. To mitigate the overall errors of the predictions, a multi-level stacking approach was proposed, where each level is trained to correct the errors produced by the previous level network. Three different algorithms (ConvLSTM, U-FNO and TAU) were tested in a dataset comprised of 32 numerical simulation models.

Although error accumulation in recursive strategies for prediction of time-series data is still an open problem, all algorithms showed high correlation scores, especially regarding the predictions of C and *eps*, even at late time steps. Moreover, the multi-stacking approach was successful at improving the results from the base model (Level 0) of each algorithm for the majority of the cases. On the other hand, TAU did not benefit so much from this pipeline, which emphasizes the need for further investigation on the weight of the regularization term of its loss function. Even so, its Level 0 network was capable of achieving higher correlations than the other algorithms (and their corrections) for all predicted properties (except U_x), at the same time it achieved faster training and forward times.

Despite the high correlation scores for *eps* prediction, there is still some improvement possible in estimating bulk properties from the predicted *eps* maps, which would bridge the gap between pore-scale interactions and macroscopic flow and transport behaviors, and provide a broader understanding of such phenomena in a porous medium. This is particularly true when assessing the error rates for porosity and permeability estimation, which were expected to decrease over each network level, following similar patterns to the *eps* correlation plots. However, as none of the networks was calibrated to be aware of the overall bulk properties at a given time step, the evolution of those error rates ended up showing an uncorrelated pattern to the *eps* predictions. Nevertheless, all algorithms managed to produce low-magnitude error rates for porosity and permeability, and similar evolutions to their respective ground truths, considering an "average pore geometry" across all samples from our dataset. Hence, within the benchmark setting considered here, our method shows strong potential as a fast surrogate to complement traditional numerical solvers, especially when taking into account the reported speedup and lowered computational expense.

Future directions for this work include: 1) analysis of gradient accumulation to improve iterative predictions at late time steps; 2) in-depth study of bulk property estimation; 3) application of the proposed method in larger-scale domains and in 3-dimensional simulations, which only requires a change of the base architecture, preserving the original idea of multi-level stacking.

Appendix A Mean Squared Error (MSE) Scores for Iterative Prediction

To quantify the error magnitudes of the iterative predictions, we also conducted an analysis of the evolution of MSE scores. Figures A1 and A2 show the MSE scores of the iterative predictions on the training and validation sets, respectively. Compared to the results discussed in Section 5.3, the ranking of all methods on both scenarios remained the same as the ones produced by the PCC metric. These results indicate that TAU not only has a higher linear relationship to the ground truth, but also yields the smallest errors on its predictions among the tested algorithms.

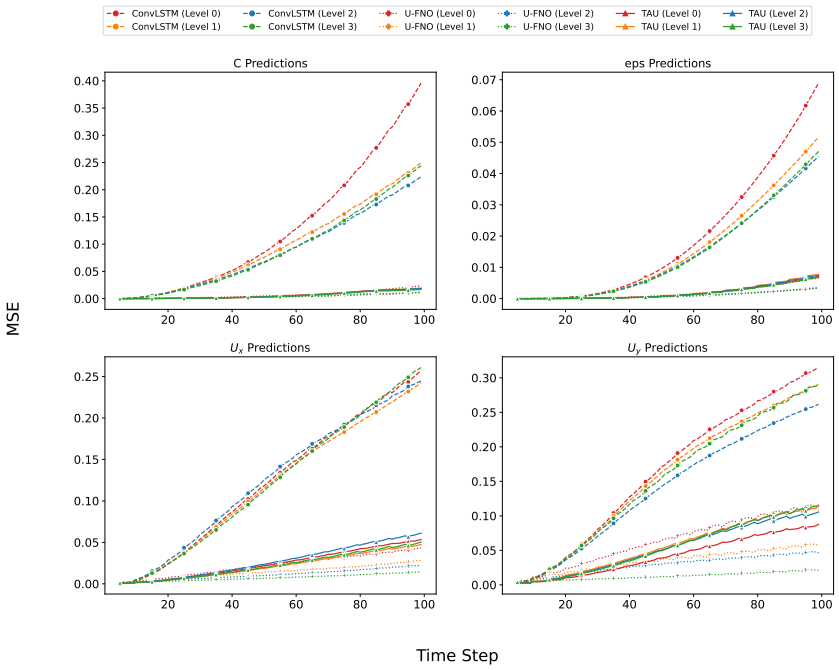


Figure A1: Average MSE scores of all samples from the training set for the iterative predictions produced by each algorithm.

Open Research Section

Data Availability Statement: The source code used to reproduce all results for our iterative stacked method, including pre-trained models for all ML algorithms described in this work, can be found at `https://github.com/ai4netzero/ReactiveDissolution`. The supporting dataset for reactive dissolution is publicly available at `https://zenodo.org/records/14974428` under the Creative Commons Attribution International 4.0 license (Cirne et al., 2025).

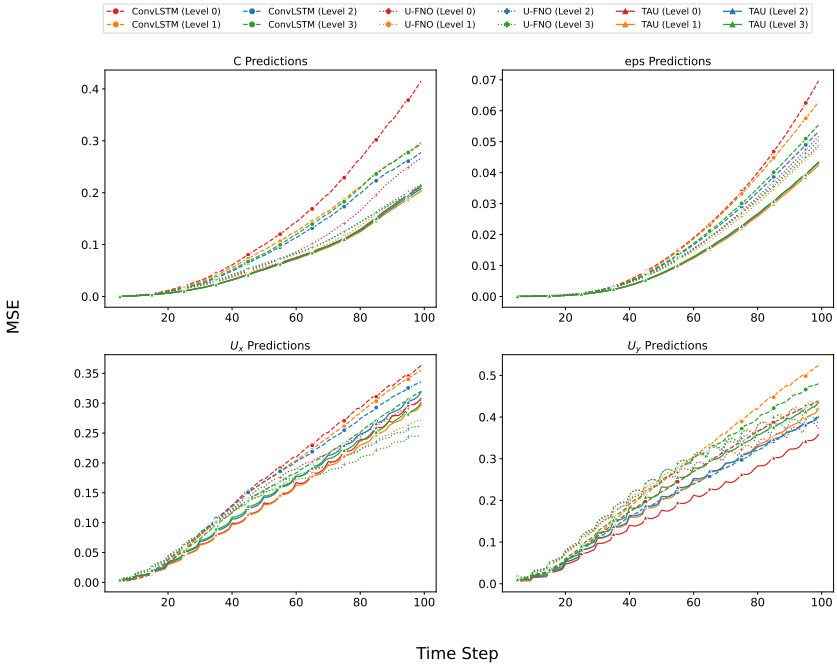


Figure A2: Average MSE scores of all samples from the validation set for the iterative predictions produced by each algorithm.

Software Availability Statement: The experiments on porosity and permeability estimation were run on version 5.1 of GeoChemFoam, available at `https://github.com/GeoChemFoam` under the GNU General Public License (GPL-3.0) (Maes et al., 2022).

Acknowledgments

This work is funded by the Engineering and Physical Sciences Research Council's ECO-AI Project grant (reference number EP/Y006143/1), with additional financial support from the PETRONAS Centre of Excellence in Subsurface Engineering and Energy Transition (PACESET).

References

Alqahtani, N., Armstrong, R. T., & Mostaghimi, P. (2018). Deep learning convolutional neural networks to predict porous media properties. In *SPE Asia Pacific Oil and Gas Conference and Exhibition* (p. D012S032R010).

Al-Shabandar, R., Jaddoa, A., Liatsis, P., & Hussain, A. J. (2021). A deep gated recurrent neural network for petroleum production forecasting. *Machine Learning with Applications*, *3*, 100013.

Bahdanau, D. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Cheng, M., Fang, F., Navon, I. M., & Pain, C. (2023). Ensemble Kalman filter for GAN-ConvLSTM based long lead-time forecasting. *Journal of Computational Science*, *69*, 102024.

Cirne, M., Menke, H., Abdellatif, A., Maes, J., Doster, F., & Elsheikh, A. H. (2025). *Reactive dissolution in porous media [Dataset].* Zenodo. Retrieved from `https://doi.org/10.5281/zenodo.14974428` doi: 10.5281/zenodo.14974428

Da Wang, Y., Blunt, M. J., Armstrong, R. T., & Mostaghimi, P. (2021). Deep learning in pore scale imaging and modeling. *Earth-Science Reviews*, *215*, 103555.

Ding, C., Zhou, Y., Pu, G., & Zhang, H. (2022). Low carbon economic dispatch of power system at multiple time scales considering GRU wind power forecasting and integrated carbon capture. *Frontiers in Energy Research*, *10*, 953883.

Du, H., Zhao, Z., Cheng, H., Yan, J., & He, Q. (2023). Modeling density-driven flow in porous media by physics-informed neural networks for CO2 sequestration. *Computers and Geotechnics*, *159*, 105433.

Esfe, M. H., & Esfandeh, S. (2020). 3D numerical simulation of the enhanced oil recovery process using nanoscale colloidal solution flooding. *Journal of Molecular Liquids*, *301*, 112094.

Feng, Z., Tariq, Z., Shen, X., Yan, B., Tang, X., & Zhang, F. (2024). An encoder-decoder ConvLSTM surrogate model for simulating geological CO2 sequestration with dynamic well controls. *Gas Science and Engineering*, *125*, 205314.

Gao, Z., Tan, C., Wu, L., & Li, S. Z. (2022). SimVP: Simpler yet better video prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 3170–3180).

Garnier, P., Viquerat, J., Rabault, J., Larcher, A., Kuhnle, A., & Hachem, E. (2021). A review on deep reinforcement learning for fluid mechanics. *Computers & Fluids*, *225*, 104973.

Graczyk, K. M., & Matyka, M. (2020). Predicting porosity, permeability, and tortuosity of porous media from images by deep learning. *Scientific Reports*, *10*(1), 21488.

He, Q., Barajas-Solano, D., Tartakovsky, G., & Tartakovsky, A. M. (2020). Physics-informed neural networks for multiphysics data assimilation with application to subsurface transport. *Advances in Water Resources*, *141*, 103610.

Heinemann, N., Alcalde, J., Miocic, J. M., Hangx, S. J. T., Kallmeyer, J., Ostertag-Henning, C., ... Rudloff, A. (2021). Enabling large-scale hydrogen storage in porous media–the scientific challenges. *Energy & Environmental Science*, *14*(2), 853–864.

Kakka, P. R. (2022). Sequence to sequence AE-ConvLSTM network for modelling the dynamics of PDE systems. *arXiv preprint arXiv:2208.07315*.

Kani, J. N., & Elsheikh, A. H. (2017). DR-RNN: A deep residual recurrent neural network for model reduction. *arXiv preprint arXiv:1709.00939*.

Karimi, M., & Bhattacharya, K. (2024). A learning-based multiscale model for reactive flow in porous media. *Water Resources Research*, *60*(9).

Khebzegga, O., Iranshahr, A., & Tchelepi, H. (2020). Continuous relative permeability model for compositional simulation. *Transport in Porous Media*, *134*, 139–172.

Kingma, D. P. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kochkov, D., Smith, J. A., Alieva, A., Wang, Q., Brenner, M. P., & Hoyer, S. (2021). Machine learning–accelerated computational fluid dynamics. *Proceedings of the National Academy of Sciences*, *118*(21), e2101784118.

Koesdwiady, A., El Khatib, A., & Karray, F. (2018). Methods to improve multi-step time series prediction. In *2018 International Joint Conference on Neural Networks (IJCNN)* (pp. 1–8).

Li, A., Chen, R., Farimani, A. B., & Zhang, Y. J. (2020). Reaction diffusion system prediction based on convolutional neural network. *Scientific Reports*, *10*(1), 3894.

Li, Z., Kovachki, N., Azizzadenesheli, K., Liu, B., Bhattacharya, K., Stuart, A., & Anandkumar, A. (2020). Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*.

Liang, S.-Y., Lin, W.-S., Chen, C.-P., Liu, C.-W., & Fan, C. (2021). A review of geochemical modeling for the performance assessment of radioactive waste disposal in a subsurface system. *Applied Sciences*, *11*(13), 5879.

Lim, B., & Zohren, S. (2021). Time-series forecasting with deep learning: a survey. *Philosophical Transactions of the Royal Society A*, *379*(2194), 20200209.

Maes, J., Soulaine, C., & Menke, H. P. (2022). Improved volume-of-solid formulations

for micro-continuum simulation of mineral dissolution at the pore-scale. *Frontiers in Earth Science*, *10*, 917931.

Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2018). Statistical and machine learning forecasting methods: Concerns and ways forward. *PloS One*, *13*(3), e0194889.

Mohajerin, N., & Waslander, S. L. (2019). Multistep prediction of dynamic systems with recurrent neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, *30*(11), 3370–3383.

MS, M., Vishnu Mohan, & Menon, V. (2021). Measuring viscosity of fluids: a deep learning approach using a CNN-RNN architecture. In *Proceedings of the first international conference on ai-ml systems* (pp. 1–5).

Nguyen, T., Shah, R., Bansal, H., Arcomano, T., Madireddy, S., Maulik, R., ... Grover, A. (2023). Scaling transformer neural networks for skillful and reliable medium-range weather forecasting. *arXiv preprint arXiv:2312.03876*.

Oprea, S., Martinez-Gonzalez, P., Garcia-Garcia, A., Castro-Vargas, J. A., Orts-Escolano, S., Garcia-Rodriguez, J., & Argyros, A. (2020). A review on deep learning techniques for video prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *44*(6), 2806–2826.

Reichstein, M., Camps-Valls, G., Stevens, B., Jung, M., Denzler, J., Carvalhais, N., & Prabhat. (2019). Deep learning and process understanding for data-driven Earth system science. *Nature*, *566*(7743), 195–204.

Salimzadeh, S., & Nick, H. (2019). A coupled model for reactive flow through deformable fractures in enhanced geothermal systems. *Geothermics*, *81*, 88–100.

Santos, J. E., Xu, D., Jo, H., Landry, C. J., Prodanović, M., & Pyrcz, M. J. (2020). PoreFlow-Net: A 3D convolutional neural network to predict fluid flow through porous media. *Advances in Water Resources*, *138*, 103539.

Shi, X., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W.-K., & Woo, W.-c. (2015). Convolutional LSTM network: A machine learning approach for precipitation nowcasting. *Advances in Neural Information Processing Systems*, *28*.

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to Sequence Learning with Neural Networks. *arXiv preprint arXiv:1409.3215*.

Taieb, S. B., Bontempi, G., Atiya, A. F., & Sorjamaa, A. (2012). A review and comparison of strategies for multi-step ahead time series forecasting based on the NN5 forecasting competition. *Expert Systems with Applications*, *39*(8), 7067–7083.

Tan, C., Gao, Z., Wu, L., Xu, Y., Xia, J., Li, S., & Li, S. Z. (2023). Temporal attention unit: Towards efficient spatiotemporal predictive learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 18770–18782).

Tang, M., Liu, Y., & Durlofsky, L. J. (2021). Deep-learning-based surrogate flow modeling and geological parameterization for data assimilation in 3D subsurface flow. *Computer Methods in Applied Mechanics and Engineering*, *376*, 113636.

Wang, W., Xie, Q., An, S., Bakhshian, S., Kang, Q., Wang, H., ... Yuan, B. (2023). Pore-scale simulation of multiphase flow and reactive transport processes involved in geologic carbon sequestration. *Earth-Science Reviews*, 104602.

Wen, G., Li, Z., Azizzadenesheli, K., Anandkumar, A., & Benson, S. M. (2022). U-fno—an enhanced fourier neural operator-based deep-learning model for multiphase flow. *Advances in Water Resources*, *163*, 104180.

Xu, L., Chen, N., Chen, Z., Zhang, C., & Yu, H. (2021). Spatiotemporal forecasting in earth system science: Methods, uncertainties, predictability and future directions. *Earth-Science Reviews*, *222*, 103828.

Yan, B., Harp, D. R., Chen, B., & Pawar, R. (2022). A physics-constrained deep learning model for simulating multiphase flow in 3D heterogeneous porous media. *Fuel*, *313*, 122693.

Zhu, L.-T., Chen, X.-Z., Ouyang, B., Yan, W.-C., Lei, H., Chen, Z., & Luo, Z.-H. (2022). Review of machine learning for hydrodynamics, transport, and reactions in multiphase flows and reactors. *Industrial & Engineering Chemistry Research*, *61*(28), 9901–9949.