Modeling Complex Fluid Flow in Porous Media Using Graph Neural Networks

Sathujoda, S.; Veeger, L.; Sheth, S.; Jönsthövel, T.; Yorke-Smith, N.

DOI
10.3997/2214-4609.202437021

Publication date
2024

Document Version
Final published version

Published in
European Conference on the Mathematics of Geological Reservoirs, ECMOR 2024

ECMOR 2024
European Conference on the Mathematics of Geological Reservoirs
EAGE

Modeling Complex Fluid Flow in Porous Media Using Graph Neural Networks

S. Sathujoda[1], L. Veeger[1,2], S. Sheth[1], T. Jönsthövel[1], N. Yorke-Smith[2]
[1] SLB; [2] Delft University of Technology

Summary

Carbon capture and sequestration initiatives make new demands on modern reservoir simulators. To find optimal locations and volumes of CO2 to inject into a subsurface to maximize CO2 storage, we must simulate a large ensemble of injection cases. One possible solution to the computational complexity of this task is to employ machine-learning models which, after a one-off overhead cost of training, can infer and predict future states of a reservoir several orders of magnitude faster than traditional methods.

Most previous work in the literature has primarily focused on either convolution-based methods or, more recently, neural operator-based methods, to predict the evolution of state variables. These architectures have shown promise in predicting on structured reservoir grids but lack the capability to extend the same level of accuracy to unstructured grids. Graph neural networks (GNNs) overcome this bottleneck by incorporating inductive biases arising from local message-passing mechanisms, facilitating convolution operations over complex graphs and meshes.

In this work, we present a novel autoregressive GNN autoencoder to predict time-varying state variables for an ensemble of CO2 injection cases. We implement a graph convolution network for the message-passing protocol and incorporate physics-informed edge weights between cell connections to guide flow. An exhaustive set of node features are used to train the model on the hyperbolic evolution of phase saturations while preserving the ellipticity in the pressure.

We test the performance of the GNN model for (1) its ability to predict state variables for varying injection rates of CO2, (2) for the post-injection phase, and (3) under different unseen geological configurations. Training and testing are performed by constructing ensembles of 2D, 3D, and real field cases that best represent these scenarios. For the 2D regular grid case, we observe that the model can capture pressure and saturation values accurately, even for highly varying injection rates and with only a limited amount of data. This performance is maintained in the post-injection phase. A key advantage of GNNs is that they show a distinct ability for transfer learning on ensembles of unseen geological configurations. We observe that the model can predict the shape and intensity of wavefronts of certain cases with no prior exposure to the specific static properties during training. Similar results are produced for 3D grids and real field cases.

Introduction

In line with global efforts to push for net zero emissions by 2050, the Intergovernmental Panel on Climate Change (IPCC) has set out strategies in its 1.5° report to reduce emissions drastically over the next 25 years. Carbon Capture and Storage (CCS) as a technology for carbon dioxide removal is expected to play a vital role in these strategies to slow down the rate of climate change. Many challenges still remain in this area however, for it to scale to the level required in order for it to have an impact on a global context.

One major challenge in the early stages of the CCS modelling workflow is that of fast numerical simulation of subsurface fluids. In recent years, traditional reservoir simulators have been advanced to more accurately predict the state of a subsurface reservoir under various injection schemes of CO_2 but still remain computationally very expensive. For tasks such as optimal well placement, optimal injection schemes, and uncertainty quantification, a large ensemble of cases are required to be simulated to solve inverse problems within a given confidence interval. This becomes increasingly time consuming when dealing with uncertainty in the underlying geological structure of the reservoir itself as we are then also required to simulate an ensemble of different simulation grids for each one of these tasks.

With the rise of machine learning capabilities in recent years, owed predominantly to the increased sophistication of modern GPUs, training large neural networks has become more feasible. This has in turn made it possible for the area of reservoir proxy modelling using machine learning to gain increased traction with many works, described in the related works section, achieving on the order of $1000\times$ speed up in solution approximation compared to traditional solvers. Significant challenges still persist however, as many of the previous models are still applicable to only regular grids as opposed to unstructured grids used in modelling reservoir, and they don't scale well when grid sizes increase to the order of 10^6 cells. Furthermore, these machine learning models do not extrapolate well to geological structures and static field properties that they have previously not seen in the training dataset, which requires the simulator to be run a very large number of times to produce a varied enough training set.

In this work, we propose to tackle some of these problems by proposing an auto-regressive graph neural network approach which learns local dynamics of subsurface fluids to speed up solution approximations of a high-fidelity reservoir simulator for previously unseen static fields during training. We show that our approach is able to accurately capture the evolution of dynamic variables for various different unseen injection schemes of CO_2 and is also able to produce redirected flow fronts when it is tested on unseen permeability and porosity configurations of the reservoir in 2D. We extend this study to a 3D case and show that the model performs similarly when learning local dynamics in an additional direction.

Related Work

First significant work in the area of machine learning applied to reservoir proxy modelling was carried out using convolution-based methods inspired by the embed-to-control framework Watter et al. (2015), a model to solve optimal control problems by approximating global non-linear dynamics to be locally linear. This was applied to 2D reservoir simulation in Jin et al. (2020) and for well output prediction in Coutinho et al. (2021). The E2CO framework was extended to 3D using 3D convolution blocks in Atadeger et al. (2022) and the whole training process was sped up using a localized learning method in Sathujoda and Sheth (2023).

Despite methods to speed up training for convolution-based methods, they still remain impractical to scale up to large grid sizes. Moreover, new methods for PDE approximation have since been developed which take into account underlying grids and numerical methods, such as Fourier Neural Operators Li et al. (2021) and Graph Convolution Networks Veličković et al. (2018). These methods have been applied to the specific task of reservoir simulation in U-FNO Wen et al. (2022) and Jiang and Guo (2023). Out of these works, the approach using Graph Neural Networks is the most promising as the method was designed specifically for the case of unstructured grids and information propagation on graphs. Little work has been done in this direction however or on the ability of GNNs to predict on grid structures

beyond which they are trained on.

Background

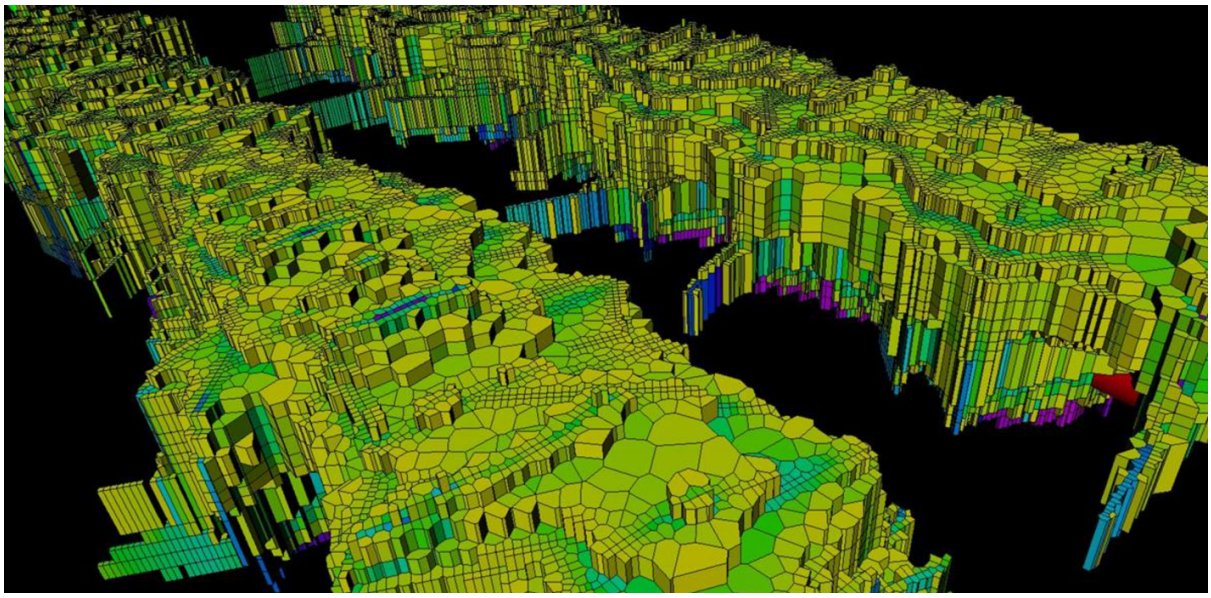

Figure 1 *Subsurface reservoir model constructed using an unstructured grid.*

Subsurface Reservoir Simulation

The governing equation of subsurface flow, derived from the law of mass-conservation and Darcy's law, relates the fluid flow and flow potential gradients of a multi-phase porous-medium as given below

$$\nabla \cdot \left[\alpha \mathbf{k} \frac{k_{r,m}(S_m)}{\mu_m B_m} (\nabla p_m - \gamma_m \nabla z) \right] - \beta \frac{\partial}{\partial t} \left(\frac{\phi S_m}{B_m} \right) + \sum_w \frac{q^w_{sc,m}}{V_b} = 0$$

where $\mathbf{k}$ denotes the permeability tensor, k_r the relative permeability, μ the viscosity, B the formation volume factor, p the pressure, S the saturation, ϕ the porosity, t the time, γ the specific weight, q the source/sink terms, z the depth and V_b the bulk volume. The subscript sc represents standard conditions, m the medium, and α and β are unit field constants.

These equations are discretized on a 3D grid, such as in figure 1, and solved over the space and time domain using numerical methods such as Newton's method. The discretization scheme used to model the subsurface reservoir is one of the main challenges when dealing with machine learning proxies for numerical simulators. Most conventional methods assume a structured rectangular grid but in reservoir modelling, certain geological scenarios such as pinch points lead to a necessity to construct complex non-structured grids. This is one of the main reasons Graph Neural Networks are favourable for the task of proxy modelling.

Graph Neural Networks

Graph Neural Networks (GNNs) are a class of machine learning models designed to incorporate an inductive bias towards graph-structured data. Early ideas behind GNNs were first introduced in Gori et al. (2005) and later developed in Scarselli et al. (2009), before the rapid growth in recent years. Generally, GNNs follow a message-passing paradigm which involves aggregating information from neighbouring nodes and transforming that information with a parameterized function (such as a multi-layer perceptron) to create an updated representation of the graph. This message-passing and updating process is carried out multiple times to widen the range of influence of one node on another in the graph.

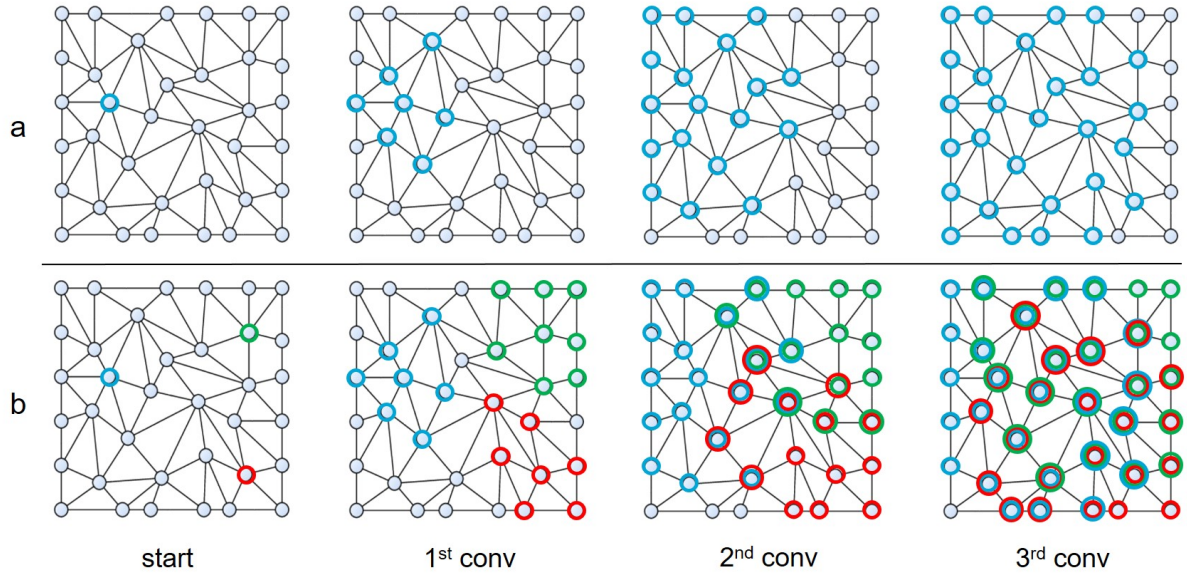

Figure 2 *Information spread of a) a single node b) three nodes during over multiple graph convolutions. Message. a) After 3 convolutions, almost all the nodes have passed information to the starting node and vice versa. b) Information mix increases as more graph convolutions are performed, in practice happening for every node in the graph (only 3 nodes shown for simplification).*

Formally, let $\mathscr{G} = (V, E)$ be a graph with nodes $u \in V$ and edges $e_{uv} \in E$ which represent connections between nodes u and v. Then $\mathscr{N}_u = \{v : e_{uv} \in E\}$ is defined to be the neighbourhood of node u. Additionally, let $\mathbf{x_u} \in \mathbb{R}^n$ represent features associated with node u and $\mathbf{e_{uv}} \in \mathbb{R}^m$ the edge features associated with edge e_{uv}. Then we define a Message-Passing Neural Network (MPNN) layer as follows Bronstein et al. (2021):

$$\mathbf{h}_u = \phi\left(\mathbf{x}_u, \bigoplus_{v \in \mathscr{N}_u} \psi(\mathbf{x}_u, \mathbf{x}_v, \mathbf{e}_{uv})\right) \tag{1}$$

where ϕ and ψ and differentiable functions (such as neural networks) and $\bigoplus$ is a permutation invariant aggregation operator which can take an arbitrary number of inputs, such as element-wise sum or mean. Here, $\mathbf{h}_u$ is the updated representation of the features at u. Graph Neural Networks are defined as l MPNN layers applied sequentially to the initial features of the nodes.

Some major GNN architectures that have arisen in recent times include the Graph Convolution Network (GCN) Kipf and Welling (2017), which proposes a localised graph convolution operation that convolves only over direct neighbours of a node normalised by node degrees, and the Graph Attention Network (GAT) Veličković et al. (2018), which incorporates an attention mechanism to dynamically learn edge weights for the message-passing layer. A Graph Convolution Network is defined by the MPNN layer below,

$$\mathbf{h}_u^{(l+1)} = \sigma\left(\mathbf{W}^{(l)} \sum_{v \in \mathscr{N}_u} \frac{\mathbf{h}_v^{(l)}}{\sqrt{|\mathscr{N}_u| \cdot |\mathscr{N}_v|}}\right) \tag{2}$$

where σ is an activation function (such as ReLU) and $\mathbf{W}^{(l)}$ are the weights corresponding to the (shallow) neural network at the l^{th} layer. This can be rewritten in matrix form as following

$$\mathbf{H} = \sigma\left(\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{X}\Theta\right) \tag{3}$$

where $\tilde{\mathbf{A}}$ is the adjacency matrix, $\tilde{\mathbf{D}}$ is the degree matrix, $\mathbf{X}$ are the node features and Θ are the neural network parameters.

One of the key advantages of GCNs over other variants of GNNs, such as GATs, is that its message-passing layer is computationally lightweight and training times scale significantly better to larger graphs. GCNs still suffer from the problem of over-smoothing Li et al. (2018) however, the phenomenon where the representations of different graph nodes become indistinguishable as the number of message-passing layers increase. To overcome this bottleneck, residual connections and identity mappings were introduced between the layers to maintain a higher signal-to-noise ratio. This architecture, name the Graph Convolution Network II (GCN2) architecture, is given below

$$\mathbf{h}_u^{(l+1)} = \sigma\left(\left((1-\alpha^{(l)})\sum_{v\in\mathcal{N}_u\cup\{u\}}\frac{\mathbf{h}_v^{(l)}}{\sqrt{|\mathcal{N}_u|\cdot|\mathcal{N}_v|}} + \alpha^{(l)}\mathbf{h}_u^{(0)}\right)\left((1-\beta^{(l)})\mathbf{I}+\beta^{(l)}\mathbf{W}^{(l)}\right)\right) \tag{4}$$

where α is the residual strength and β the identity strength.

Methodology

Problem Setting

Let $G=(V,E)$ be a graph representing the structure of a subsurface reservoir, where V is the set of nodes signifying the cells of the grid discretization and E the set of edges between connected cells. We assume the graph structure is static. Let each node $u\in V$ have node features $\mathbf{x}_u^t\in\mathbb{R}^d$, where $t\in\mathbb{N}$ represents the time step to which the features pertain and d the number of static and dynamic properties of simulation. For this work, we consider the static and dynamic properties given in table 1 Let $\mathbf{y}_u^t\in\mathbb{R}^m$ represent the dynamic properties of concern, namely pressure p and saturation s, at the time step $t+1$. We aim to develop a model $\mathscr{M}$ such that $\mathscr{M}:(\mathbf{X}^t,G)\to\mathbf{Y}^t$ and $|\mathscr{M}(\mathbf{x}_u^t,G)-\mathbf{y}_u^t<\tilde{\varepsilon}|$, $\forall u\in V$ and some small $\tilde{\varepsilon}\in\mathbb{R}^m$. Here $\mathbf{X}^t=\{\mathbf{x}_u^t:u\in V\}$ and $\mathbf{Y}^t=\{\mathbf{y}_u^t:u\in V\}$.

Property	Type	Description
p	Dynamic	Pressure
s	Dynamic	Saturation
k_x	Static	Permeability in X
k_y	Static	Permeability in Y
k_z	Static	Permeability in Z
V_p	Static	Pore Volume
d_k	Static	Depth

Table 1 *Simulation properties*

Model

Let $\varepsilon:\mathbb{R}^{n\times d}\to\mathbb{R}^{n\times h}$ be an 'encoder' function parameterized by a multi-layer perceptron of l_e layers, each with n_i neurons in the ith layer transformed by a LeakyReLU, where $n=|V|$ and d is the dimensionality of the data. Similarly, let $\delta:\mathbb{R}^{n\times k}\to\mathbb{R}^{n\times m}$ be a 'decoder' function parameterized by a multi-layer perceptron of l_d layers, each with m_j neurons in the jth layer transformed by a LeakyReLU. Finally, let $\gamma:\mathbb{R}^{n\times h}\to\mathbb{R}^{n\times k}$ be a 'processor' function which is parameterized by l custom Graph Convolution II Layers. The total model is hence $\mathscr{M}\sim\delta\circ\gamma\circ\varepsilon$, where $\mathscr{M}:\mathbb{R}^{n\times d}\to\mathbb{R}^{n\times m}$.

Here, we modify the GCN II layers to operator only with residual connections and apply a heuristic edge weight of the harmonic mean of the permeabilities of the two nodes being connected by the edge. The new MPNN layer is given below.

$$\mathbf{h}_u^{(l+1)} = \sigma\left(\alpha\mathbf{h}_u^{(0)} + (1-\alpha)\mathbf{W}^{(l)}\sum_{v\in\mathcal{N}_u\cup\{u\}}\frac{2k_uk_v}{k_u+k_v}\mathbf{h}_v^{(l)}\right) \tag{5}$$

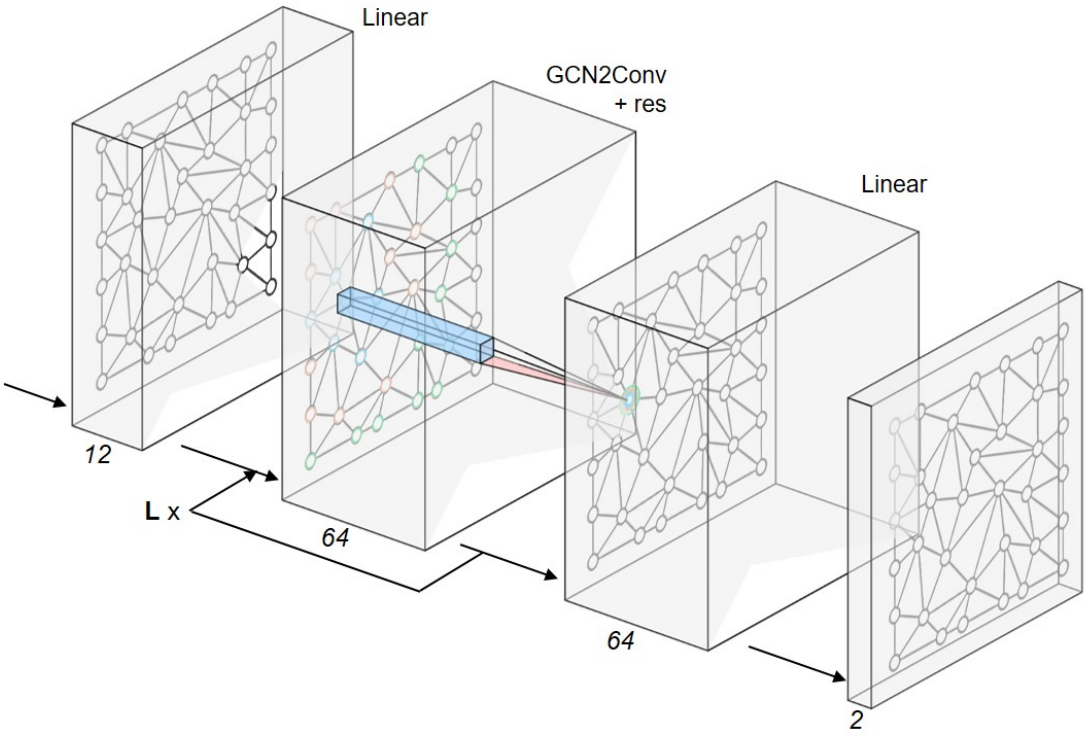

Figure 3 *Representation of model architecture and forward pass.*

Here k_u and k_v are the permeabilities of the two nodes in the spacial direction of the connected edge.

Training procedure

The model is trained to autoregressively predict the next time step given the previous solution but during training we perform a roll-out of predictions for k steps into the future and aggregate the losses over these timesteps to control accumulating errors. The loss function we use is the mean square error loss aggregated over the roll out period.

$$\mathscr{L} = \frac{1}{k}\sum_{i=t}^{t+k}(\hat{\mathbf{y}}_i - \mathbf{y}_i)^2 \tag{6}$$

Hyperparameter tuning for the model was performed using Raytune. The best parameters we tested for during training are given in table 2.

Experiments

In the experiments, we are concerned with three main capabilities of the model in 2D: its ability to autoregressively predict future states of the reservoir for variable injection rates, variable shale compositions and unseen permeability fields. For 3D tests, we assume constant injection rates and test directly for the models ability to predict future states for unseen permeabilities.

Parameter	Value
Learning rate	0.0005
Batch size	8
Res. α	0.15
LReLU	0.1
GCN Layers	6
Mult. k	3
Hidden dim.	128

Table 2 *Best hyperparameters from raytune.*

Datasets

For the experiments, we consider 2D and 3D simulation results from a high-fidelity simulator. The 2D case consists of a 37×50 rectangular grid simulated over 1000 years, $\sim$12 years of 143 monthly timesteps and the remainder being post-injection steps, with one well at the grid cell $(31,0)$. The post-injection phase timesteps increase progressively from 1, 10, 100 to eventually 500 years to model long-term containment behaviour. Two ensembles are created for the 2D case, namely variable injection rates and variable shale compositions. Static properties for a specific case of the 2D dataset is given in figure 4. The 3D dataset consists of a $40 \times 80 \times 10$ grid with 100 different permeability and porosity fields generated using Perlin Noise to simulate channels and other geological structures. We simulate water injection for this case to theoretically test the model on a simple workflow. Permeability values below a threshold are zeroed out to give complexity to the reservoir so as to not just test on a rectangular grid. A high-fidelity simulator is run for each of these 100 cases for 10 monthly timesteps at constant injection rates with 2 injector and 2 producer wells placed at $(39,20)$, $(69,7)$ and $(11,19)$, $(62,30)$ respectively, where the wells are vertically connected through all z layers.

2D Auto-regressive and variable injection capabilities

In the first experiment, we test for the GNN model's auto-regressive capabilities and its ability to capture varying injection rates. The variable injection ensemble consists of different constant inject rates sampled from a uniform distribution. To aid with this, we also pass into the model the change in pressure and saturation values from the previous timestep (accumulation speed terms), to give information on flow rates. The results from this experiment are given in figures 5 and 6. From the results, we see that the model is able to auto-regressively predict future states of the system accurately, even for varying injection rates. This injective expressivity is largely attributed to the introduction of the accumulation speed features. Without them, models initially tended to converge to a more or less fixed propagation speed. In those models, a node likely struggled to spread its state rate of change to neighbouring nodes during message-passing, given that the injection rate information is not passed as a feature. Message-passing of the accumulation features enables the nodes to pass around their state's rate of change, providing all the grid nodes with a sense of how quickly CO2 is entering and saturating the system.

Variable Shale ensemble

Along with good injective expressivity, a proper reservoir simulation model should be capable of handling geological variability. The variable shale ensemble consists of a distribution of regions with low permeability and high porosity to represent shale characteristics in the geology. The results presented in figures 7 and 8 show that the model is able to accurately predict dynamic variables for varying geological cases and that the values of magnitude within the flow front align closely with the true values. The model limits the CO2 flow through the shale layers for the cases with lower porosity/permeability, and builds up more pressure in the areas with high saturation under the shale layers. In some cases the model tends to 'lag' at the start of the injection and builds up some under-expressive error. The model then corrects for this and produces a stable flow. The error generally increases gradually during the stable flow, as some parts of the saturation front become slightly over- or under-expressed when predicting

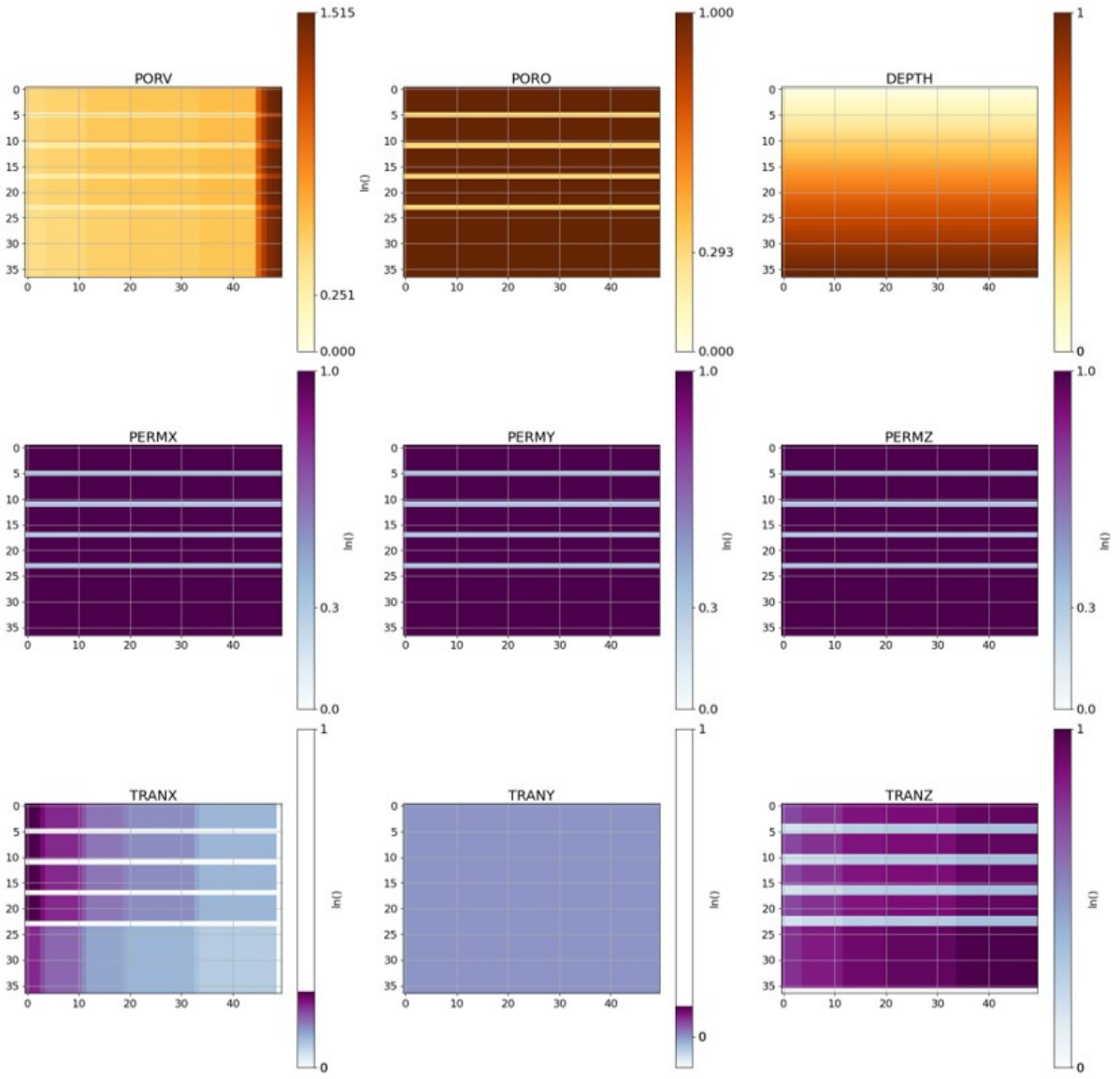

Figure 4 *Static properties of a general 2D sample from the variable injection ensemble (constant across the ensemble). It shows the general structure with horizontal shale layers found in every 2D sample. PORV increases steadily in the x-direction. DEPTH increases in the increasing z-direction. PORO influences the input property PORV. PERMY is preconfigured nonzero, but is not encoded in any edge weight. Transmissibility depends on the cell dimension in its specific direction, making TRANY zero in the xz-dimensional grid. TRANX/TRANZ are influenced by PERMX/PERMZ, PORO and PORV. The variable shale ensemble is identical in the non-shale surroundings, but has varying PORO and randomly decaying PERMX/PERMZ in the shale layers.*

further out in time auto-regressively. The overall results in this experiment show that the model does respond accurately to parametric variability in the geological features and performs particularly well for saturation predictions, as expected.

Unseen Geologies

To test the transfer learning capabilities of the model to new geological structures which the model has not seen during training, we construct three different permeability fields in the z direction. The three fields which we tested on are given in figure 9. The corresponding pressure and saturation values which the model has predicted are given in figures 10, 11 and 12. We expect saturation values to be

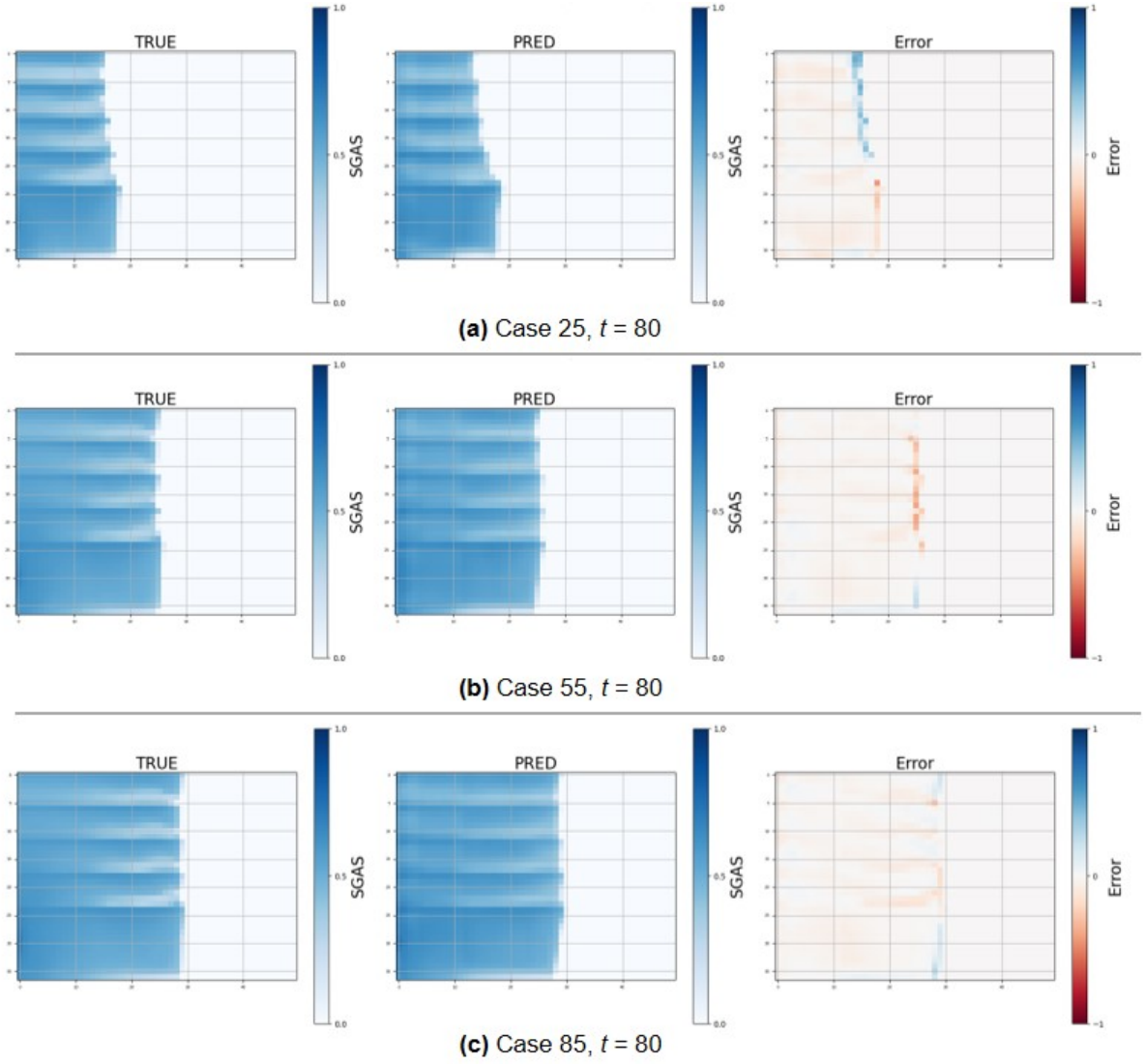

Figure 5 *Gas saturation (SGAS) simulation, roll-out prediction and error at t = 80 for the variable injection ensemble. a) Low injection rate, b) medium injection rate, c) high injection rate*

predicted better than pressure variables as saturation is governed by a hyperbolic partial differential equation which is more dependent on local dynamics, whereas pressure is governed by an elliptical partial differential equation which is more affected by long distance interactions. Graph neural networks, in their nature, capture only short distance interactions unless significantly rewired or a large number of message-passing layers are stacked on top of one and another, hence lending themselves to saturation predictions.

As we can see from each of the cases, the new unseen shapes of the permeability/porosity fields are captured accurately. The flow learns to go around boundaries where there is a sharp gradient in the field. There is some leakage in areas of narrow low permeability channels however, as when multiple GCN layers are stacked, they end up propagating information over the narrow low permeability channels. A potential fix to this is to drastically reduce information propagation when this behaviour is encountered with a scaling factor. The networks capture magnitudes of pressure and saturation values well with errors within flow fronts almost resulting in zero error.

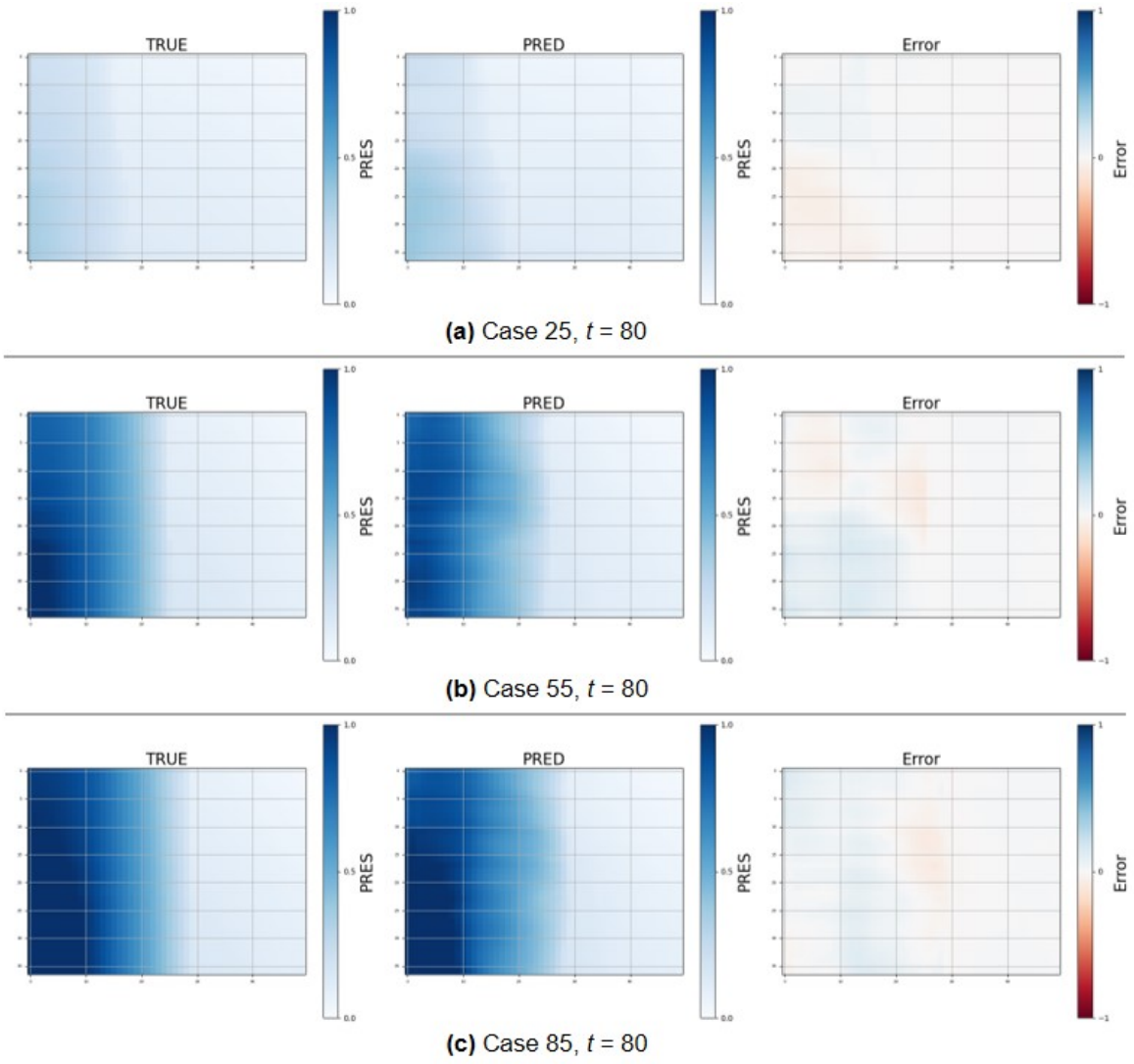

Figure 6 *Pressure (PRES) simulation, roll-out prediction and error at t = 80 for the variable injection ensemble. a) Low injection rate, b) medium injection rate, c) high injection rate*

3D Unseen Geological

We finally test the GNN architecture for 3D cases. Here we generate 100 different geological fields of permeability and porosities using Perlin noise algorithms and create inactive cells at various regions to better approximate real field cases in a synthetic manner. We connect each cell (node) to all its directly adjacent cells in 3D and remove edges when there is an inactive cell. Furthermore, we only test for propagation of saturation in the case as pressure is an elliptic variable. The results for various layers for different timesteps ($0-9$ months) are given in figures 13, 14, 15, 16, and 17. We can see from these results that the saturation propagation fronts are captured accurately. The magnitudes are also within the same range for the prediction and ground truth values from the simulator. These results are more accurate in comparison to other convolution or neural operator-based methods as we see the flow is dynamically able to adapt to the geometry of the underlying grid. For instance, in the results presented, we show the saturation predictions on a completely unseen permeability and porosity field and the model has no prior information during training of this specific configuration, but is still able to predict the direction and boundaries of the flow. Most of the errors appear near the fronts of the flows which signals that the model is not able to capture fully the rate of flow, suggesting we may need additional rate information like in the variable injection rate experiments. Still, this stands as a significant milestone for transfer learning on 3D reservoir grids.

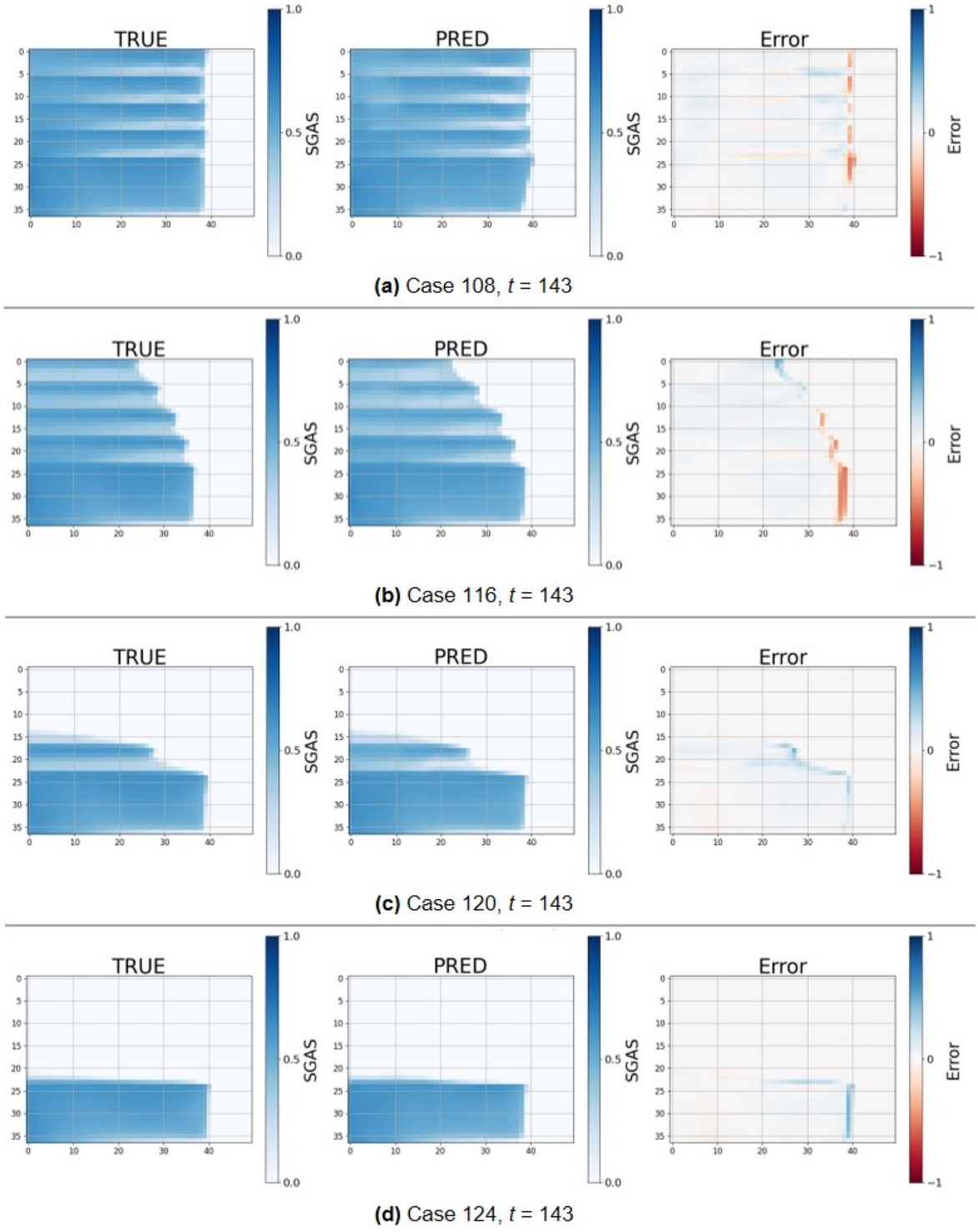

Figure 7 *Gas saturation (SGAS) simulation, roll-out predictions and errors of 4 cases from the validation set of the variable shale composition ensemble, in order of decreasing permeability. a) Permeability shale ∼14mD, b) Permeability shale ∼0.7mD, c) Permeability shale ∼0.04mD, d) Permeability shale ∼0.005mD)*

Conclusion

In this work, we have presented a Graph Neural Network approach to predict solutions to high-fidelity simulators used in modelling CO_2 injection into subsurface reservoirs. We have further studied the abil-

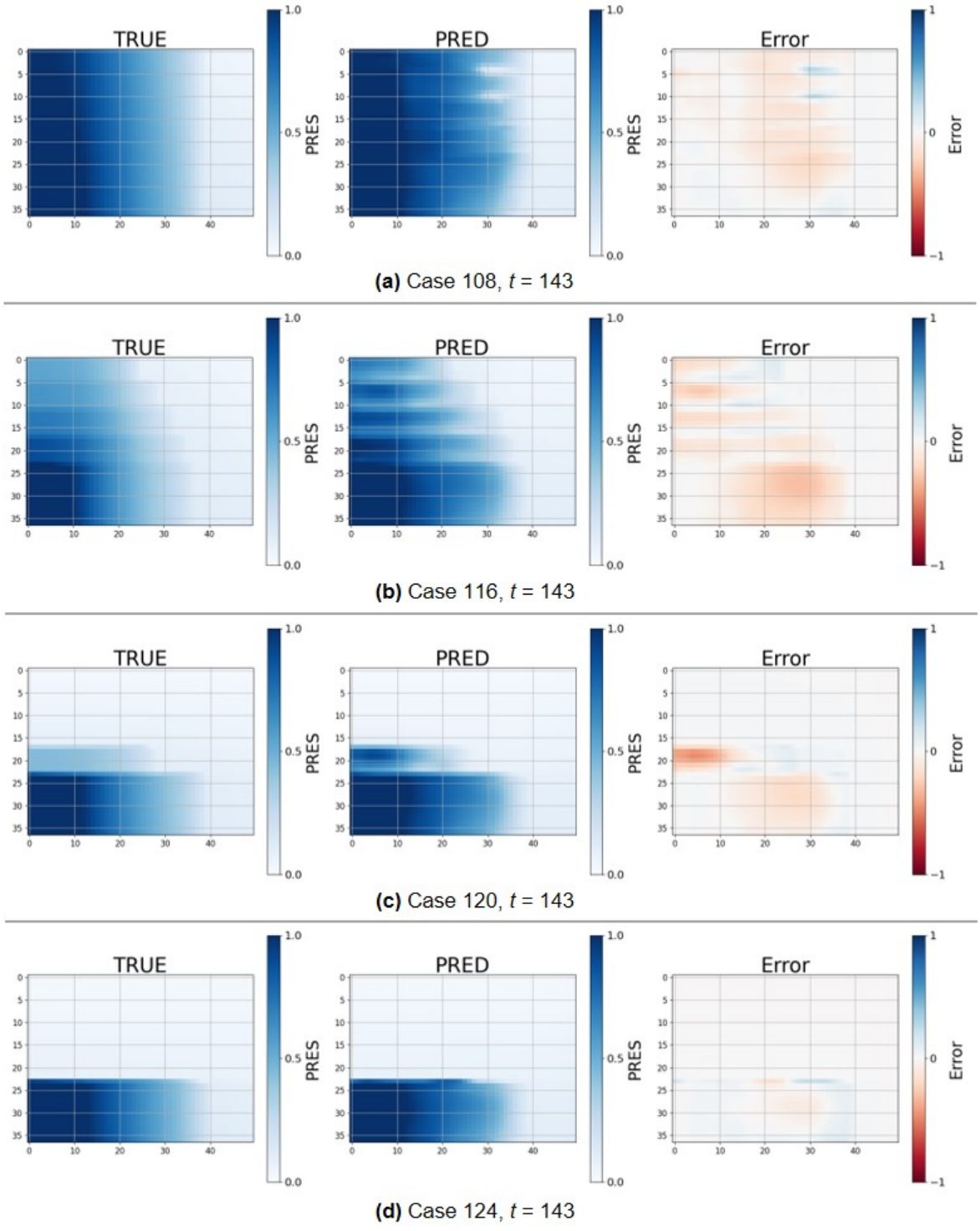

Figure 8 *Pressure (PRES) simulation, roll-out predictions and errors of 4 cases from the validation set of the variable shale composition ensemble, in order of decreasing permeability. a) Permeability shale ∼14mD, b) Permeability shale ∼0.7mD, c) Permeability shale ∼0.04mD, d) Permeability shale ∼0.005mD)*

ity of GNNs to generalize to new geological configurations which they have not been exposed to during the training procedure and showed that they are able to accurately capture flow fronts and magnitudes for unseen permeability and porosity fields, in both a 2D and 3D context. We also observe that the message-passing nature of GNNs are able to accurately capture local dynamics in the form of different flow rates for variable inject rates.

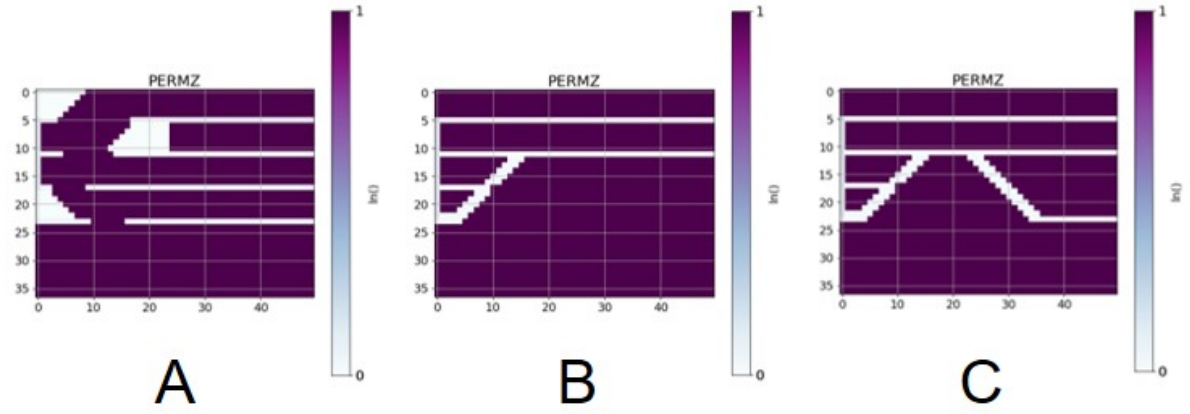

Figure 9 *Permeability in z-direction (PERMZ) for structurally modified cases of the P7 dataset. Reservoir A includes holes in the shale layers allowing easy propagation of CO2, combined with zero permeability blocks resembling impermeable rock formations. Reservoir B contains a diagonal shale layer to evaluate the diagonal flow. Reservoir C extends B with a mirrored shale layer to evaluate the effect of the CO2 flow downward.*

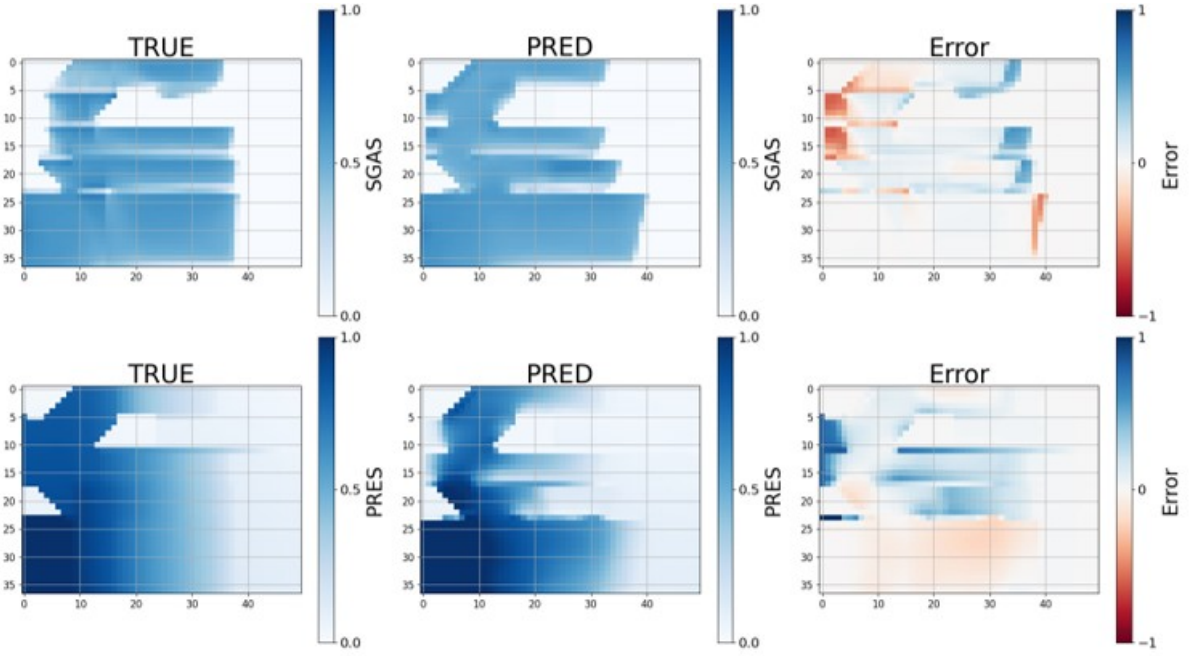

Figure 10 *Simulation, roll-out predictions and errors of untrained reservoir case A at final injection phase state at t = 143.*

In the future, we aim to extend our work on 3D grids by studying truly unstructured grids, which GNNs have a strong inductive bias towards, and introduce variable injection rates to our analysis. We also aim to analyse whether GNNs are able to accurately capture flow when trained on a fixed set of well location configurations and are tested on a new set of unseen well locations, whether it be adding additional wells or relocating existing ones.

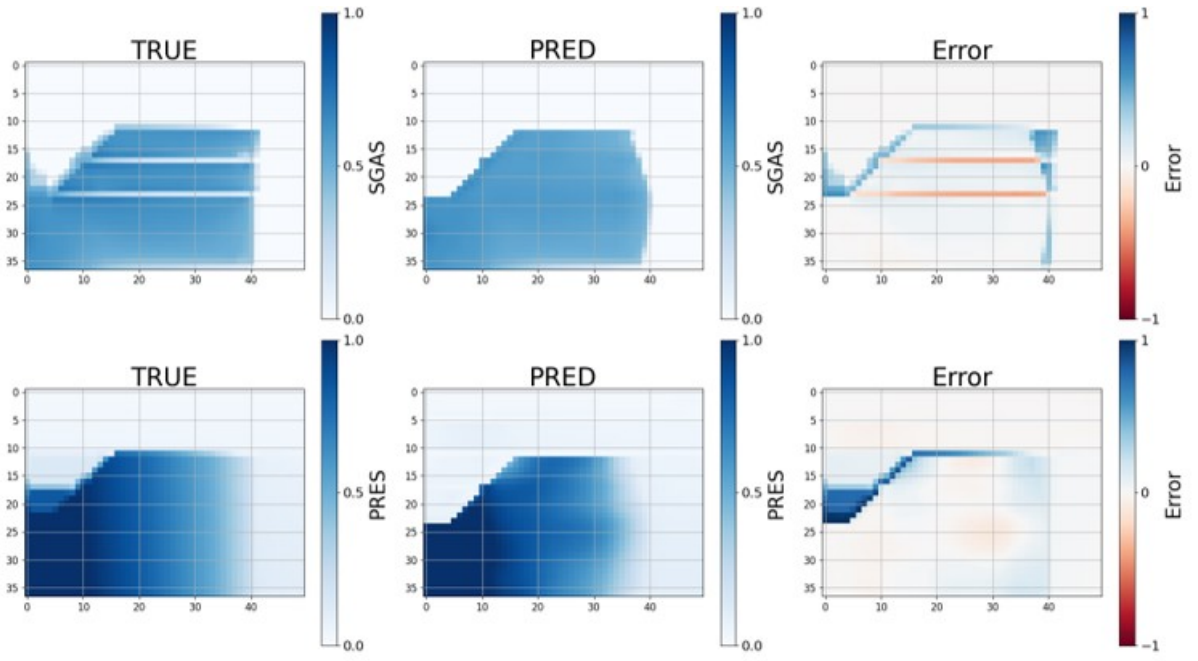

Figure 11 *Simulation, roll-out predictions and errors of untrained reservoir case B at final injection phase state at t = 143.*

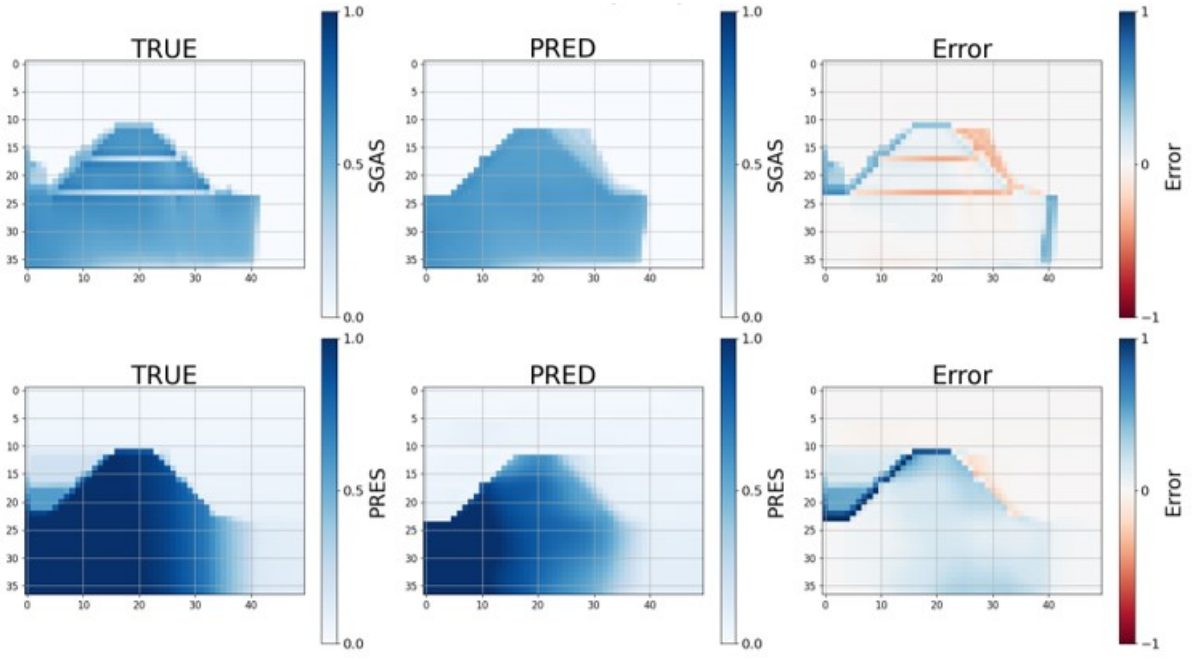

Figure 12 *Simulation, roll-out predictions and errors of untrained reservoir case C at final injection phase state at t = 143.*

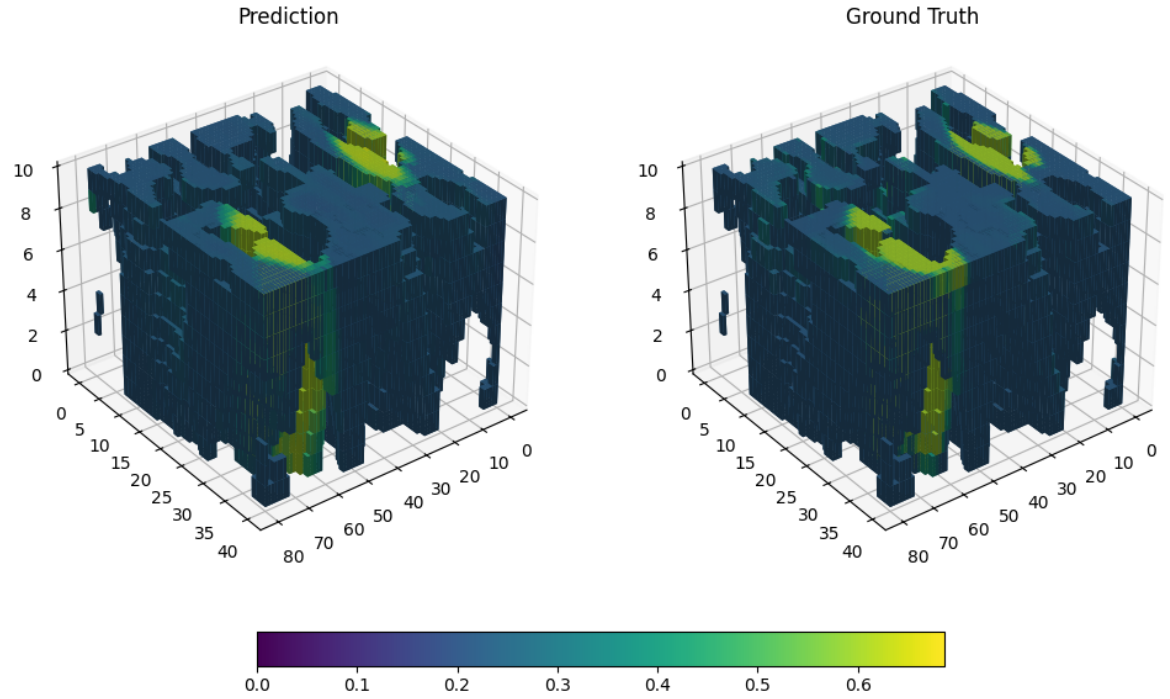

Figure 13 *3D simulation case for case 23. Saturation prediction for an unseen permeability/porosity field. Full grid prediction for time step 4.*

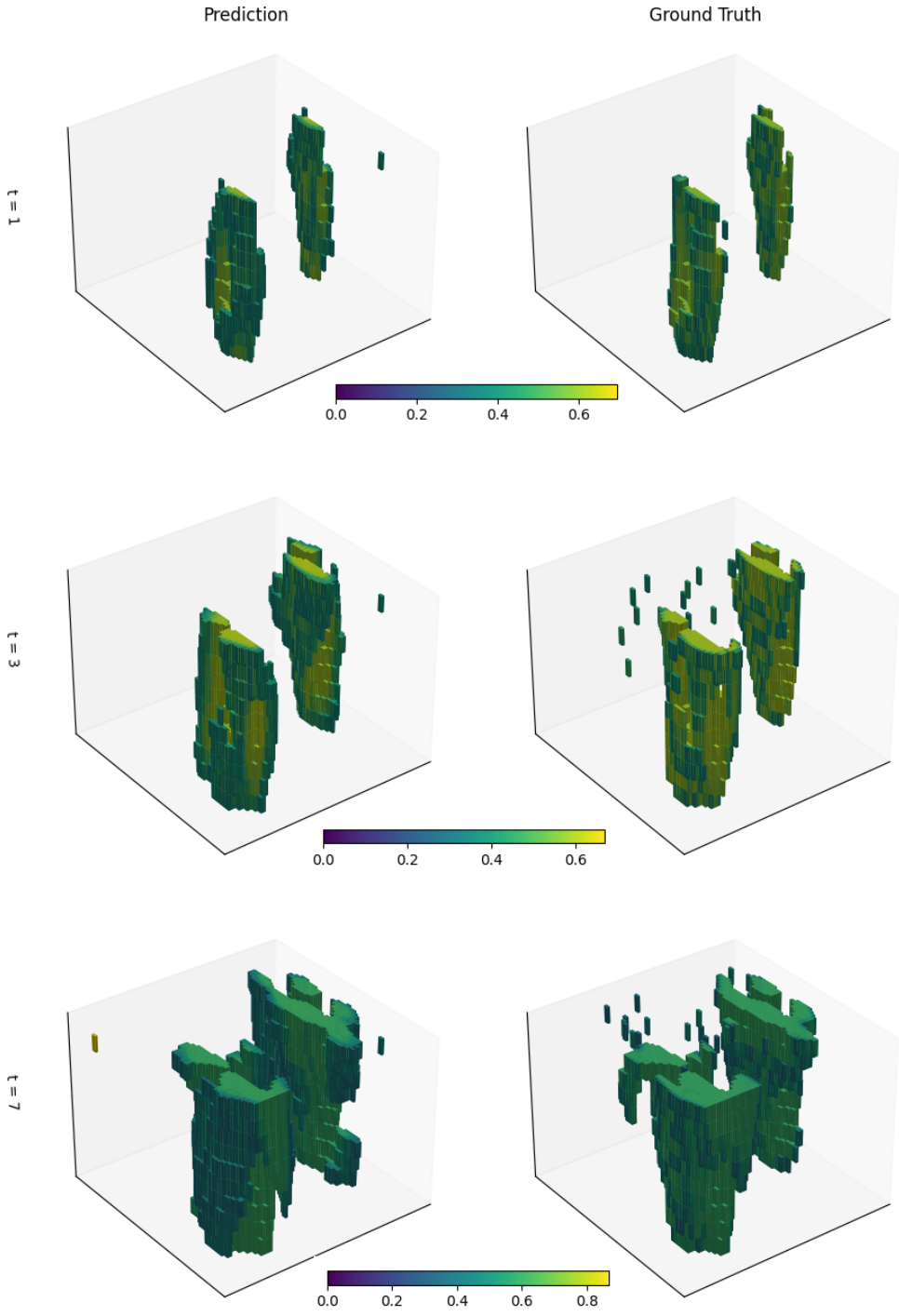

Figure 14 *Saturation predictions for case 23 to demonstrate flow in the reservoir. Values of saturation above 0.3 are displayed to show fonts and magnitudes of flow.*

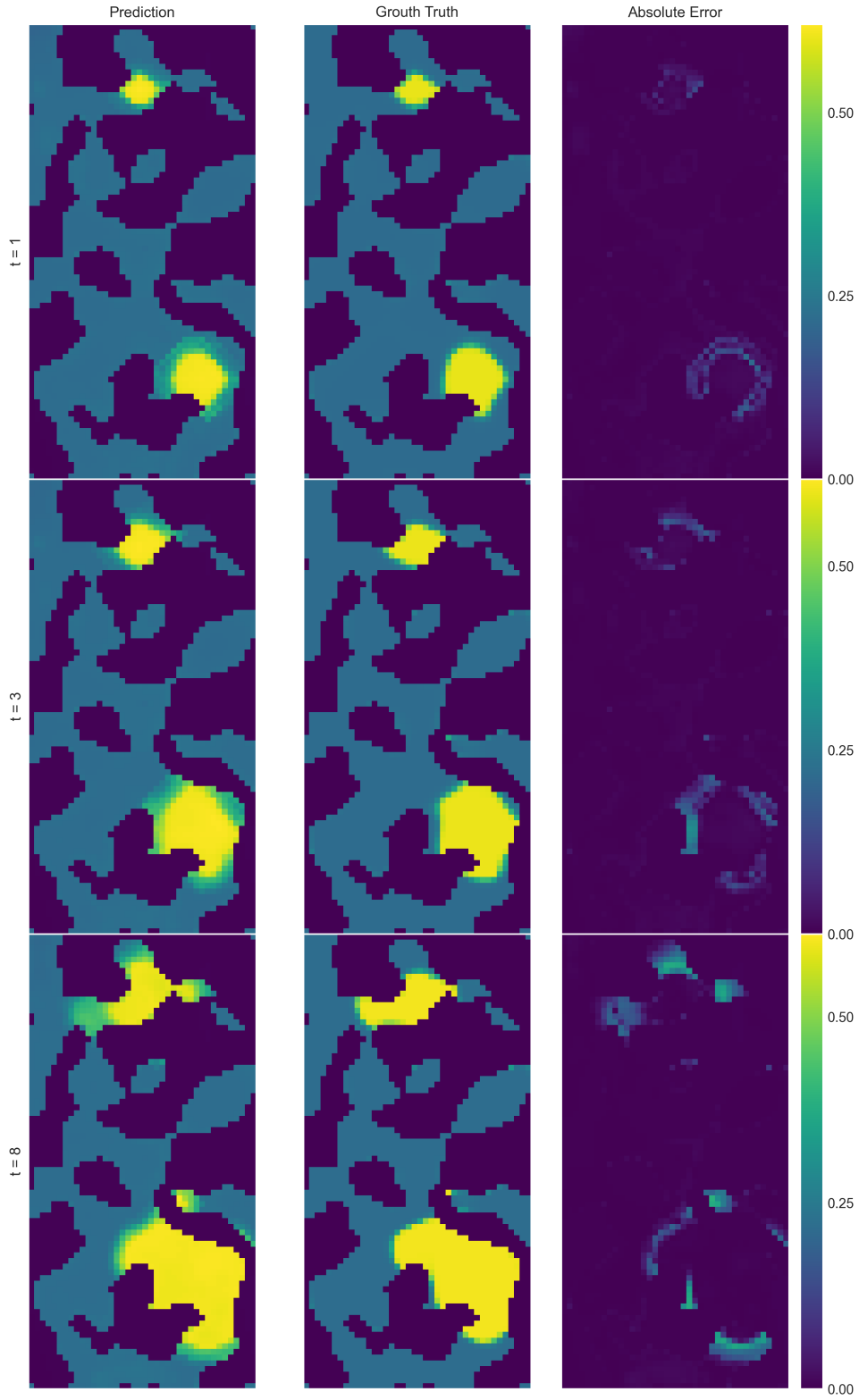

Figure 15 *3D simulation case for case 23. Saturation prediction for an unseen permeability/porosity field. Layer 2 of 10 for the 3D grid.*

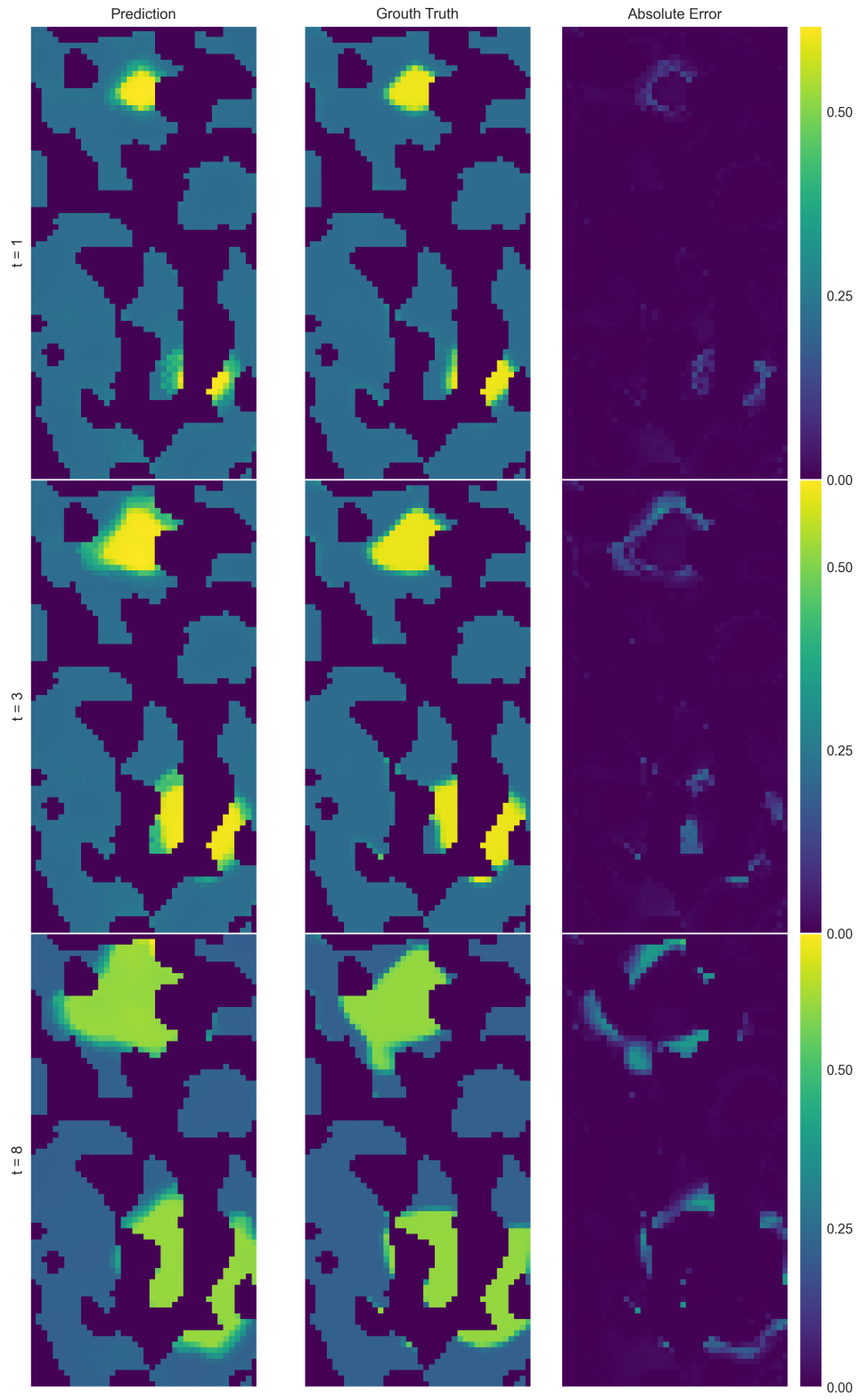

Figure 16 *3D simulation case for case 23. Saturation prediction for an unseen permeability/porosity field. Layer 5 of 10 for the 3D grid.*

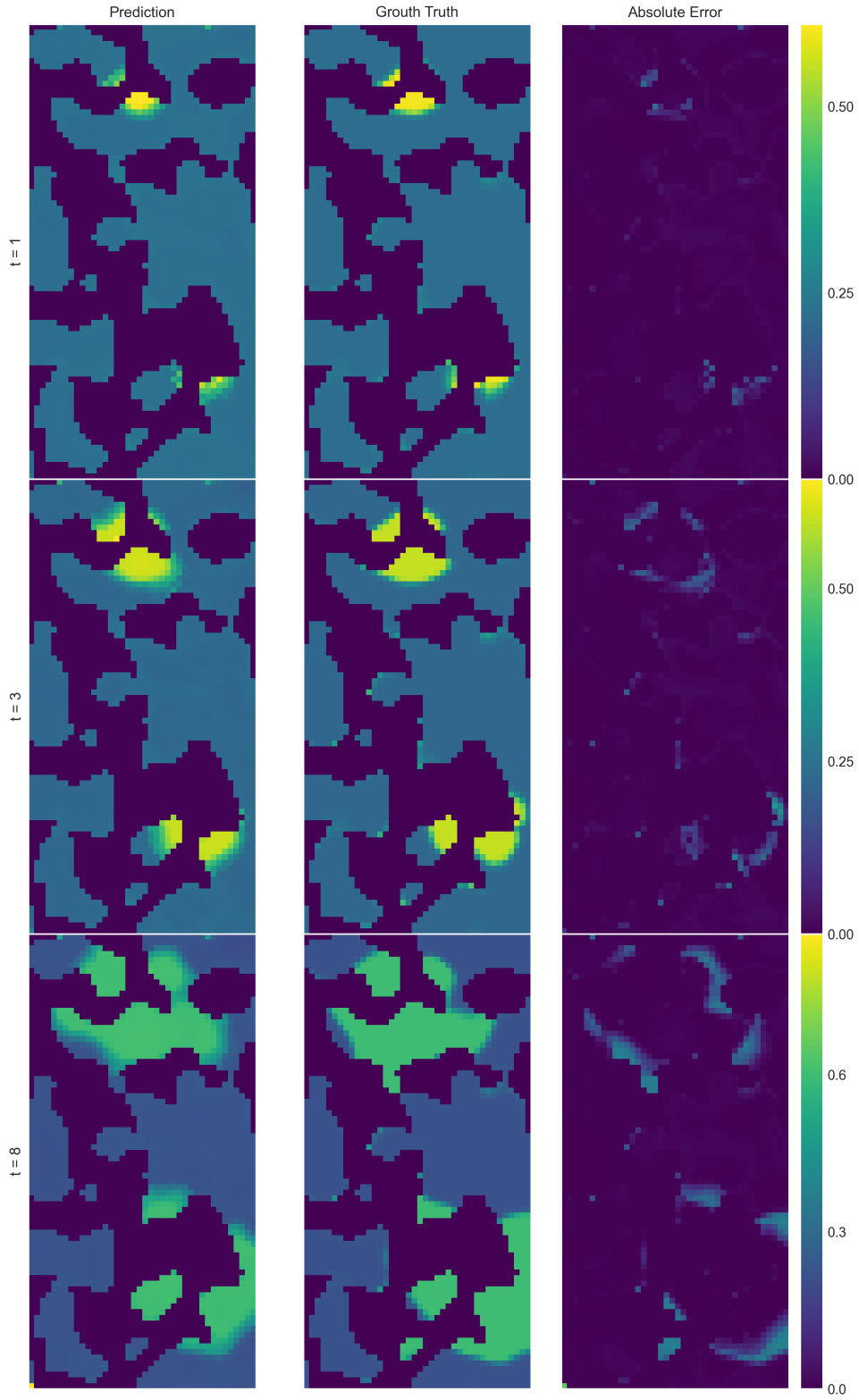

Figure 17 *3D simulation case for case 23. Saturation prediction for an unseen permeability/porosity field. Layer 8 of 10 for the 3D grid.*

References

Atadeger, A., Sheth, S., Vera, G., Banerjee, R. and Onur, M. [2022] Deep Learning-Based Proxy Models to Simulate Subsurface Flow of Three-Dimensional Reservoir Systems. **2022**(1), 1–32.

Bronstein, M.M., Bruna, J., Cohen, T. and Veličković, P. [2021] Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges.

Chen, M., Wei, Z., Huang, Z., Ding, B. and Li, Y. [2020] Simple and Deep Graph Convolutional Networks. In: III, H.D. and Singh, A. (Eds.) *Proceedings of the 37th International Conference on Machine Learning*, *Proceedings of Machine Learning Research*, 119. PMLR, 1725–1735.

Coutinho, E., Dall'Aqua, M. and Gildin, E. [2021] Physics-Aware Deep-Learning-Based Proxy Reservoir Simulation Model Equipped With State and Well Output Prediction. *Frontiers in Applied Mathematics and Statistics*, **7**.

Gori, M., Monfardini, G. and Scarselli, F. [2005] A new model for learning in graph domains. In: *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, 2. 729–734 vol. 2.

Jiang, J. and Guo, B. [2023] Graph Convolutional Networks for Simulating Multi-phase Flow and Transport in Porous Media.

Jin, Z.L., Liu, Y. and Durlofsky, L.J. [2020] Deep-learning-based surrogate model for reservoir simulation with time-varying well controls. *Journal of Petroleum Science and Engineering*, **192**, 107273.

Kipf, T.N. and Welling, M. [2017] Semi-Supervised Classification with Graph Convolutional Networks. In: *International Conference on Learning Representations*.

Li, Q., Han, Z. and Wu, X.m. [2018] Deeper Insights Into Graph Convolutional Networks for Semi-Supervised Learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, **32**(1).

Li, Z., Kovachki, N.B., Azizzadenesheli, K., liu, B., Bhattacharya, K., Stuart, A. and Anandkumar, A. [2021] Fourier Neural Operator for Parametric Partial Differential Equations. In: *International Conference on Learning Representations*.

Sathujoda, S. and Sheth, S.M. [2023] Physics-informed Localized Learning for Advection-Diffusion-Reaction Systems. In: *ICML Workshop on New Frontiers in Learning, Control, and Dynamical Systems*.

Scarselli, F., Gori, M., Tsoi, A.C., Hagenbuchner, M. and Monfardini, G. [2009] The Graph Neural Network Model. *IEEE Transactions on Neural Networks*, **20**(1), 61–80.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P. and Bengio, Y. [2018] Graph Attention Networks. In: *International Conference on Learning Representations*.

Watter, M., Springenberg, J., Boedecker, J. and Riedmiller, M. [2015] Embed to Control: A Locally Linear Latent Dynamics Model for Control from Raw Images. In: Cortes, C., Lawrence, N., Lee, D., Sugiyama, M. and Garnett, R. (Eds.) *Advances in Neural Information Processing Systems*, 28. Curran Associates, Inc.

Wen, G., Li, Z., Azizzadenesheli, K., Anandkumar, A. and Benson, S.M. [2022] U-FNO – An enhanced Fourier neural operator-based deep-learning model for multiphase flow.