
Learning Pore-scale Multi-phase Flow from Experimental Data with Graph Neural Network

Yuxuan Gu

Department of Electrical and Electronic Engineering
Imperial College London
yuxuan.gu21@imperial.ac.uk

Catherine Spurin

Department of Energy Science and Engineering
Stanford University
cspurin@stanford.edu

Gege Wen

Department of Earth Science & Engineering, I-X
Imperial College London
g.wen@imperial.ac.uk

Abstract

Understanding the process of multiphase fluid flow through porous media is crucial for many climate change mitigation technologies, including CO₂ geological storage, hydrogen storage, and fuel cells. However, current numerical models are often incapable of accurately capturing the complex pore-scale physics observed in experiments. In this study, we address this challenge using a graph neural network-based approach and directly learn pore-scale fluid flow using *micro-CT experimental data*. We propose a Long-Short-Edge MeshGraphNet (LSE-MGN) that predicts the state of each node in the pore space at each time step. During inference, given an initial state, the model can autoregressively predict the evolution of the multiphase flow process over time. This approach successfully captures the physics from the high-resolution experimental data while maintaining computational efficiency, providing a promising direction for accurate and efficient pore-scale modeling of complex multiphase fluid flow dynamics.

1 Introduction

Our society is facing unprecedented challenges in climate change, demanding swift action to accelerate the energy transition toward net-zero [1]. Understanding the process of multiphase flow through porous media is an important task as it is involved in many mitigation technologies, including CO₂ geological storage ([2, 3, 4]), hydrogen storage [5], and fuel cells [6]. To study these multiphase flow processes, scientists can utilize micro-CT scanners with synchrotron sources to image the rock pores in a nanometer-scale spatial resolution while recreating the in situ condition of gas flowing through the porous medium [7]. State-of-the-art research facilities can now generate 3D imaging data with billions of voxels with temporal resolution on the order of second [8], creating a unique opportunity to advance our understanding of fluid flow physics.

However, despite the advancement in experimental capability to obtain high-resolution experimental data, the modeling of pore-scale multiphase flow in porous media remains very challenging. Current

modeling approaches generally fall into three major classes: lattice-based models, continuum models, and pore-network models [9]. The former two approaches incorporate the Navier-Stokes equation to solve flow equations but are often highly computationally expensive [10]. Pore network models simplify porous media into interconnected pores and throats [11], offering computational efficiency but often lacking accuracy due to the complexity of the physics involved.

Machine learning-based approaches are emerging in recent pore-scale modeling studies [12, 13]. However, most existing studies aim to learn fluid flow physics from simulation data, which suffers from significant drawbacks. Due to the aforementioned computational challenges in pore-scale modeling, training datasets are often incapable of reflecting the accurate behavior of real-world multiphase flow. They are often simulated with highly simplified pore geometry (e.g., packed spheres [13]) or simplified physics (e.g., negligible viscous effects and assumptions of steady-state flow). In addition, many previous approaches were designed based on convolutional neural networks (CNNs) that are optimized for grid-like structures. As a result, they struggled with the irregular geometries of complex pore structures (see example in Appendix A).

To address the challenges, here we propose the Long-Short-Edge MeshGraphNet (LSE-MGN), a graph neural network (GNN)-based architecture for learning pore-scale fluid flow directly from experimental data (Figure 1). Inspired by the MESHGRAPHNETS [14] architecture proposed by Pfaff *et al.*, we used graph structure to represent irregular pore structures, where nodes represent fluid volume and edges represent fluid flow paths. Unlike CNNs, the graph-based representation can handle complex geometries by passing messages only between connected nodes, ignoring nodes separated by solid rock grains. Learning the fluid flow dynamics using a GNN-based architecture leverages inductive biases to focus on local interactions between fluid and pore structures. This allows for zero-shot generalisation to other boundary conditions [15] or bigger input fields through section-based training [16]. Learning directly from experimental data allows us to replicate realistic flow phenomena,

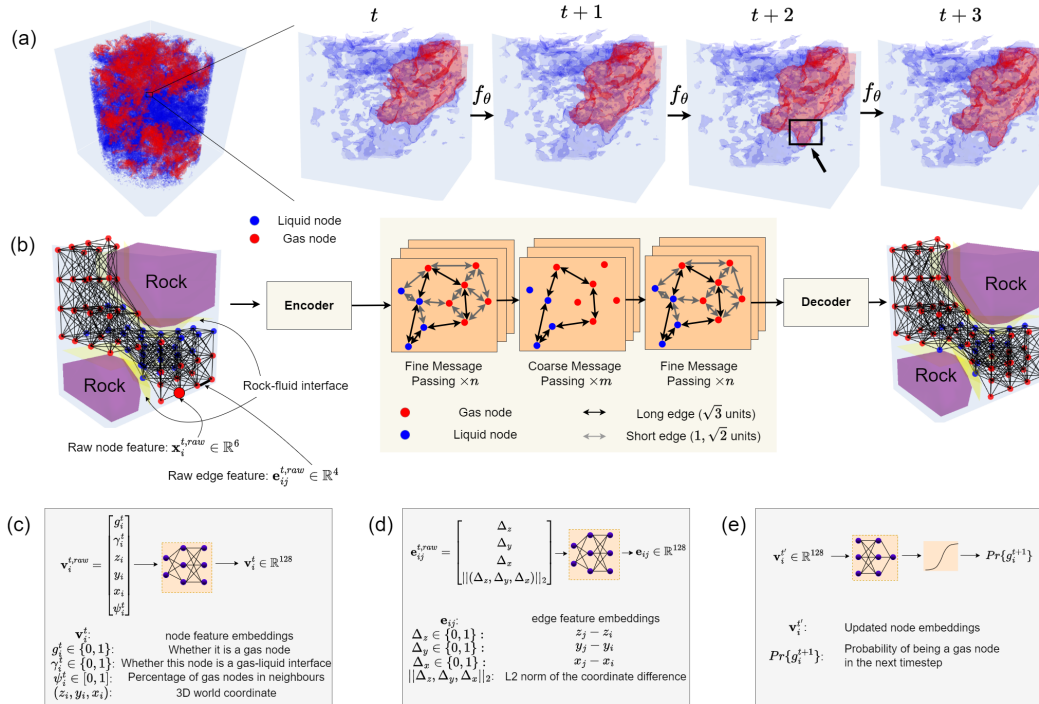


Figure 1: (a) A section in the rock is used to train a GNN model f_θ . It learns to predict the next state based on its current state. (b) Five steps in the prediction pipeline: step 1. encoder that embeds node and edge features into latent space; step 2. fine message passing with all edges; step 3. coarse message passing only with long edges; step 4. fine message passing with all edges; step 5. a decoder that outputs the final node states. (c) Node encoder. (d) Edge encoder. (e) Decoder.

including those that remain beyond the reach of simulations due to incomplete understanding. Our results demonstrate that LSE-MGN can effectively capture complex fluid behaviors and generalise well across varying boundary conditions. Our approach provides a new paradigm to leverage the extensive micro-CT experimental data to study pore-scale physics.

2 Data Description

The micro-CT multiphase flow experiment dataset used in here was collected at the Swiss Light Source, Paul Scherrer Institute. During the experiment, the rock sample was exposed to filtered polychromatic X-ray radiation originating from a 2.9 T bending magnet source, generating a spatial imaging resolution of $2.75 \mu\text{m}$ and a temporal resolution of 2 s . The experiments were conducted in a cylindrical carbonate sample, 5 mm diameter and 20 mm length. The sample was saturated with brine (deionised water doped with 15wt.% KI to improve the X-ray contrast). Then nitrogen and brine were injected simultaneously. X-ray images were acquired to capture the transition from transient effects to steady-state flow. Each image took 2 s to acquire, and had a resolution of $2.75 \mu\text{m}$. For a complete experimental description see [8]. Steady-state was determined by a stable pressure drop measured across the core. The greyscale X-ray images were segmented to extract the location of the fluids present (gas and brine). This was done using the method described in [17]. The dataset was published in Spurin *et al.* [8].

In this work, we focus on a $50 \times 80 \times 50$ cube over 300-time steps characterised by highly intermittent fluid flow, where the flow pathways through the pore space are constantly rearranging (Figure 1 (a)). Fluid intermittency is a phenomenon that current numerical modeling techniques cannot accurately capture.

3 Methodology

Graph construction. We constructed the input and output graphs using the 3D images segmented into either rock or pore spaces. As shown in Figure 1 (b), the rock spaces were excluded from the graph, and the pore spaces were represented as interconnected nodes. At time step $t \in \{0, \dots, T\}$, each point i in the pore space is represented as a node with a feature vector \mathbf{v}_i^t defined as

$$\mathbf{v}_i^t = [g_i^t, \gamma_i^t, z_i, y_i, x_i, \psi_i^t]^T, \quad (1)$$

where $g_i^t \in \{0, 1\}$ indicate the state of a node (i.e., gas or liquid), and $\gamma_i^t \in \{0, 1\}$ indicates whether the node is at a gas-liquid interface. The spatial coordinates $\{z_i, y_i, x_i\}$ specify a node’s position in the 3D pore space. ψ_i^t represents the percentage of gas nodes among the one-hop neighbors of node i . We construct bidirectional edges by connecting nodes within a fixed radius R of the current node. In this work, R is set to $\sqrt{3}$ to ensure that message passing respects the pore structure. With this setup, a node can connect to a maximum of 27 surrounding nodes, including itself, as shown in Figure 3 (b) in Appendix A. There are three distinct edge lengths, excluding the self-connection: 1, $\sqrt{2}$, and $\sqrt{3}$ units, respectively. Inspired by [14], we defined the edge attribute \mathbf{e}_{ij} as

$$\mathbf{e}_{ij} = [(z_i - z_j), (y_i - y_j), (x_i - x_j), \|((z_i - z_j), (y_i - y_j), (x_i - x_j))\|_2]^T, \quad (2)$$

which include the coordinates difference between two nodes and the l_2 norm of the difference.

Model architecture. We propose the Long-Short-Edge MeshGraphNet (LSE-MGN) architecture that predicts the gas/liquid distribution in pore spaces given the node and edge features at the previous time step. At test time, we autoregressively generate a sequence of future states given the previous state, as shown in Figure 1 (a). The model architecture is illustrated in Figure 1 (b), where we employ the "encoder-processor-decoder" structure as inspired by Graph Network-based Simulators (GNS) [15] and MultiScale-MeshGraphNets [18].

In the encoder, the node features are embedded in the latent vector of dimension 128 via a Multi-Layer Perceptron (MLP). Similarly, an edge encoder embeds the raw edge feature of dimension 4 to a latent vector of dimension 128, as shown in Figure 1 (d). The decoder predicts the fluid state at each node with a binary node-level classification, as illustrated in Figure 1 (e). The node embeddings are

transformed and summarised into a scalar value, then passed through a Sigmoid function. The final output is the probability of the node being classified as a gas node.

In the multi-scale processor, we divided edges into two categories: long edge ($\sqrt{3}$) and short edge ($0, 1, \sqrt{2}$). Nodes at both ends of the long edges are defined as pivotal nodes. Initially, node and edge features are propagated along all edges to gather information from nearby nodes. Subsequently, information is transferred only between pivotal nodes along the long edges, allowing pivotal nodes to access information from distant nodes, thereby expanding their receptive fields. Finally, messages from the pivotal nodes are ‘shared’ with nearby nodes by propagating messages along all edges again. This is achieved using n fine processors that aggregate features along all edges, then m coarse processors that propagate messages only along the long edges, and then another n fine processors along all edges. The fine message-passing processor with residual connections is defined as,

$$\mathbf{e}'_{ij} \leftarrow \mathbf{e}_{ij} + f^W(\mathbf{e}_{ij}, \mathbf{v}_i, \mathbf{v}_j), \quad \mathbf{v}'_i \leftarrow \mathbf{v}_i + f^V(\mathbf{v}_i, \sum_{j \in \mathcal{N}(i)} \mathbf{e}'_{ij}), \quad (3)$$

and the coarse processor defined as,

$$\mathbf{e}'_{ij} \leftarrow \mathbf{e}_{ij} + f^W(\mathbf{e}_{ij}, \mathbf{v}_i, \mathbf{v}_j) \cdot \mathbb{I}(|\mathbf{e}_{ij}| = \sqrt{3}), \quad \mathbf{v}'_i \leftarrow \mathbf{v}_i + f^V(\mathbf{v}_i, \sum_{j \in \mathcal{N}(i)} \mathbf{e}'_{ij}) \cdot \mathbb{I}(|\mathbf{e}'_{ij}| = \sqrt{3}). \quad (4)$$

Here \mathbf{e}'_{ij} and \mathbf{v}'_i denote the updated edge and node embeddings, respectively. \mathbf{v}_i represents the raw embedding of a target node, while \mathbf{v}_j represents the raw embedding of a neighboring node of i . $\mathbb{I}(|\mathbf{e}'_{ij}| = \sqrt{3})$ filters for long edges in coarse processors. f^W and f^V are MLP layers.

The multiscale strategy not only increases the receptive field of each node but also enhances computational efficiency by reducing the number of edges in the coarse graph to less than 30% of those in the original graph, thereby accelerating the training process. Unlike the MultiScale MeshGraphNets [18] approach, our proposed long-short-edge message passing does not require pre-computed fine and coarse graphs and two additional networks for exchanging information between these graphs, which allows easier implementation.

A comparative analysis of loss functions is provided in Appendix C, in which we found that the soft Dice BCE loss (originally defined in [19] and described in Eq. (5)), with \mathcal{L}_{BCE} specified in Eq. (8) and $\mathcal{L}_{\text{Dice}}$ in (9), demonstrates the best overall performance on our dataset.

$$\mathcal{L}_{\text{Soft BCE Dice}}(\mathbf{y}, \hat{\mathbf{y}}) = \frac{N-n}{N} \mathcal{L}_{\text{BCE}}(\mathbf{y}, \hat{\mathbf{y}}) + \frac{n}{N} \mathcal{L}_{\text{Dice}}(\mathbf{y}, \hat{\mathbf{y}}) \quad (5)$$

4 Results

The training set comprises 20 time steps during which a bubble gradually emerges, alongside 20 additional steps with minor bubble oscillation. The model is evaluated to distinguish between these two scenarios on the testing set, which includes 8-time steps with emerging bubbles and 8-time steps with minor oscillation. The model is trained for 22.5 hours using one NVIDIA A100 GPU (80GB) graphics card. Stochastic gradient descent (SGD) with momentum was used for optimisation, with a learning rate of 0.001. Figure 2 visualises the ground truth and the autoregressive rollout. The model not only accurately captures bubble emergence, but also predicts minor oscillations, as shown in Figures 4 and 6.

$$\epsilon_\tau = \frac{|\hat{A}_\tau - A_\tau|}{A_\tau} \quad (6)$$

$$\bar{\epsilon} = \frac{1}{t_T - t_0} \sum_{\tau=t_0}^{t_T} \epsilon_\tau \quad (7)$$

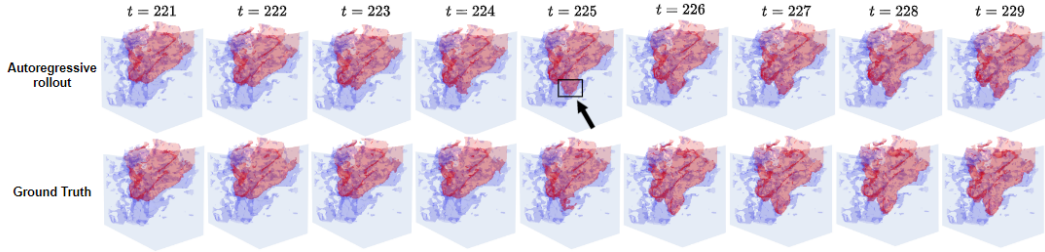


Figure 2: Visualisation (see video) of ground truth and autoregressive rollout from $t = 221$ to 229 on the **testing set with emerging bubbles**. Red parts represent gas. Blue parts represent liquid.

We define the mean surface area error, $\bar{\epsilon}$, as a metric to evaluate the performance of different loss functions, as shown in Eq. (7). In this equation, τ denotes the time step involved in the prediction, ranging from the initial time step t_0 to the final time step t_T . The term ϵ_τ , defined in Eq. (6), represents the percentage surface area error at a given time step τ . In this context, \hat{A}_τ denotes the surface area of prediction while A_τ denotes that of ground truth. The surface area calculation for each time step is detailed in Algorithm 1. Table 1 compares validation performance of different model configurations, indicating that F6C6F6 achieves the lowest mean surface area error with a reasonable training time.

Table 1: Comparison of validation performance on different model configurations based on training using a 20-time step dataset. The mean surface area error (defined as Eq. (7)) was evaluated on a 9-time step test set. FnCmFn denotes n fine, m coarse, followed by n fine message-passing layers.

Metric / Configuration	F9C0F9	F7C4F7	F6C6F6	F4C10F4
Training time (seconds/epoch)	24.88	21.72	20.02	16.74
Mean surface area error $\bar{\epsilon}$ (defined as Eq. (7))	10.40%	9.05%	8.81%	10.10%

5 Conclusion

We introduced a new paradigm for learning the physics of multiphase flow through porous media directly from experimental datasets. The LSE-MGN architecture successfully predicts the gradual emergence of bubbles and instances with minor oscillations, relying solely on local information. The results here sets the foundation for a promising direction in future pore-scale modeling.

References

- [1] V. Masson-Delmotte, P. Zhai, H.-O. Pörtner, D. Roberts, J. Skea, P. R. Shukla, A. Pirani, W. Moufouma-Okia, C. Péan, R. Pidcock, S. Connors, J.B.R. Matthews, Y. Chen, X. Zhou, M.I. Gomis, E. Lonnoy, T. Maycock, M. Tignor, and T. Waterfield. Summary for policymakers. In V. Masson-Delmotte, P. Zhai, H.-O. Pörtner, D. Roberts, J. Skea, P. R. Shukla, A. Pirani, W. Moufouma-Okia, C. Péan, R. Pidcock, S. Connors, J.B.R. Matthews, Y. Chen, X. Zhou, M.I. Gomis, E. Lonnoy, T. Maycock, M. Tignor, and T. Waterfield, editors, *Global Warming of 1.5°C. An IPCC Special Report on the impacts of global warming of 1.5°C above pre-industrial levels and related global greenhouse gas emission pathways, in the context of strengthening the global response to the threat of climate change, sustainable development, and efforts to eradicate poverty*, pages 3–24. Cambridge University Press, Cambridge, UK and New York, NY, USA, 2018. doi: 10.1017/9781009157940.001.
- [2] Tabbi Wilberforce, AG Olabi, Enas Taha Sayed, Khaled Elsaid, and Mohammad Ali Abdelkareem. Progress in carbon capture technologies. *Science of The Total Environment*, 761:143203, 2021.
- [3] Jinfeng Ma, Lin Li, Haofan Wang, Yi Du, Junjie Ma, Xiaoli Zhang, and Zhenliang Wang. Carbon capture and storage: history and the road ahead. *Engineering*, 14:33–43, 2022.

- [4] David Yang Shu, Sarah Deutz, Benedikt Alexander Winter, Nils Baumgärtner, Ludger Leenders, and André Bardow. The role of carbon capture and storage to achieve net-zero energy systems: Trade-offs between economics and the environment. *Renewable and Sustainable Energy Reviews*, 178:113246, 2023.
- [5] Nasiru Salahu Muhammed, Bashirul Haq, Dhafer Al Shehri, Amir Al-Ahmed, Mohammed Mizanur Rahman, and Ehsan Zaman. A review on underground hydrogen storage: Insight into geological sites, influencing factors and future outlook. *Energy Reports*, 8:461–499, 2022.
- [6] Martin Andersson, SB Beale, M Espinoza, Z Wu, and W Lehnert. A review of cell-scale multiphase flow modeling, including water management, in polymer electrolyte fuel cells. *Applied Energy*, 180:757–778, 2016.
- [7] Martin J Blunt, Branko Bijeljic, Hu Dong, Oussama Gharbi, Stefan Iglauer, Peyman Mostaghimi, Adriana Paluszny, and Christopher Pentland. Pore-scale imaging and modelling. *Advances in Water resources*, 51:197–216, 2013.
- [8] Catherine Spurin, Tom Bultreys, Maja Rücker, Gaetano Garfi, Christian M Schlepütz, Vladimir Novak, Steffen Berg, Martin J Blunt, and Samuel Krevor. Real-time imaging reveals distinct pore-scale dynamics during transient and equilibrium subsurface multiphase flow. *Water Resources Research*, 56(12):e2020WR028287, 2020.
- [9] Benzhong Zhao, Christopher W MacMinn, Bauyrzhan K Primkulov, Yu Chen, Albert J Valocchi, Jianlin Zhao, Qinjun Kang, Kelsey Bruning, James E McClure, Cass T Miller, et al. Comprehensive comparison of pore-scale models for multiphase flow in porous media. *Proceedings of the National Academy of Sciences*, 116(28):13799–13806, 2019.
- [10] Bartłomiej Gackiewicz, Krzysztof Lamorski, Cezary Sławiński, Shao-Yiu Hsu, and Liang-Cheng Chang. An intercomparison of the pore network to the navier–stokes modeling approach applied for saturated conductivity estimation from x-ray ct images. *Scientific reports*, 11(1): 5859, 2021.
- [11] Qingrong Xiong, Todor G Baychev, and Andrey P Jivkov. Review of pore network modelling of porous media: Experimental characterisations, network constructions and applications to reactive transport. *Journal of contaminant hydrology*, 192:101–117, 2016.
- [12] Javier E Santos, Duo Xu, Honggeun Jo, Christopher J Landry, Maša Prodanović, and Michael J Pyrcz. Poreflow-net: A 3d convolutional neural network to predict fluid flow through porous media. *Advances in Water Resources*, 138:103539, 2020.
- [13] Xu-Hui Zhou, James E McClure, Cheng Chen, and Heng Xiao. Neural network–based pore flow field prediction in porous media using super resolution. *Physical Review Fluids*, 7(7): 074302, 2022.
- [14] Tobias Pfaff, Meire Fortunato, Alvaro Sanchez-Gonzalez, and Peter W Battaglia. Learning mesh-based simulation with graph networks. *arXiv preprint arXiv:2010.03409*, 2020.
- [15] Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec, and Peter Battaglia. Learning to simulate complex physics with graph networks. In *International conference on machine learning*, pages 8459–8468. PMLR, 2020.
- [16] Tailin Wu, Qinchen Wang, Yinan Zhang, Rex Ying, Kaidi Cao, Rok Susic, Ridwan Jalali, Hassan Hamam, Marko Maucec, and Jure Leskovec. Learning large-scale subsurface simulations with a hybrid graph network simulator. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 4184–4194, 2022.
- [17] Catherine Spurin, Sharon Ellman, Dane Sherburn, Tom Bultreys, and Hamdi Tchelepi. Python workflow for segmenting multiphase flow in porous rocks. *EarthArXiv*, 2023.
- [18] Meire Fortunato, Tobias Pfaff, Peter Wirnsberger, Alexander Pritzel, and Peter Battaglia. Multiscale meshgraphnets. *arXiv preprint arXiv:2210.00612*, 2022.

- [19] Adrian Galdran, Gustavo Carneiro, and Miguel A González Ballester. On the optimal combination of cross-entropy and soft dice losses for lesion segmentation with out-of-distribution robustness. In *Diabetic Foot Ulcers Grand Challenge*, pages 40–51. Springer, 2022.
- [20] Francesco Caliva, Claudia Iriondo, Alejandro Morales Martinez, Sharmila Majumdar, and Valentina Pedoia. Distance map loss penalty term for semantic segmentation. *arXiv preprint arXiv:1908.03679*, 2019.
- [21] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.
- [22] Seyed Sadegh Mohseni Salehi, Deniz Erdogmus, and Ali Gholipour. Tversky loss function for image segmentation using 3d fully convolutional deep networks. In *International workshop on machine learning in medical imaging*, pages 379–387. Springer, 2017.

A Pore structure

As illustrated in Figure 3 (a), node 1 cannot directly transmit information to node 13 due to the obstruction posed by the solid rock between them. Instead, the information must be relayed through a sequence of intermediate nodes, such as following the path $1 \rightarrow 3 \rightarrow 5 \rightarrow 7 \rightarrow 9 \rightarrow 11 \rightarrow 13$. This routing is necessary to bypass the blockage created by the rock. Figure 3 (b) illustrates how neighbours are defined during the graph construction process. Nodes are considered as neighbors if they are within a distance of $\sqrt{3}$ units from one another. Each node can connect to up to 27 neighbouring nodes. On average, within the selected section of the pore space, each node is connected to approximately 23.2071 neighbours.

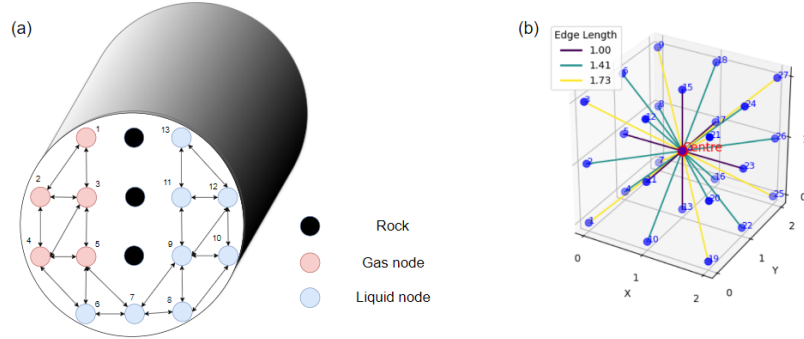


Figure 3: **(a)** An example of a cross section of a porous rock showing gas and liquid particles separated by solid rocks. **(b)** Edge formation: connect nodes within $\sqrt{3}$ units apart.

B More visualisation of the prediction

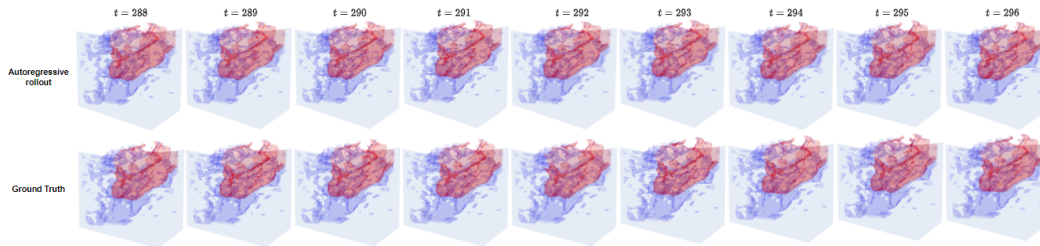


Figure 4: Visualisation (see video) of ground truth and autoregressive rollout from $t = 288$ to 296 on the **testing set with minor oscillations**. Red parts represent gas. Blue parts represent liquid.

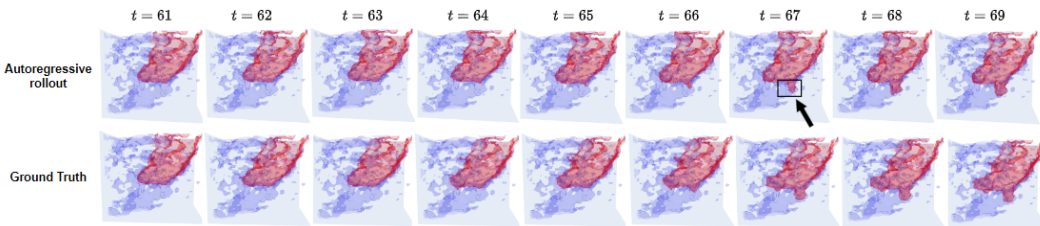


Figure 5: Visualisation (see video) of ground truth and autoregressive rollout from $t = 61$ to 69 on the **training set with emerging bubbles**. Red parts represent gas. Blue parts represent liquid.

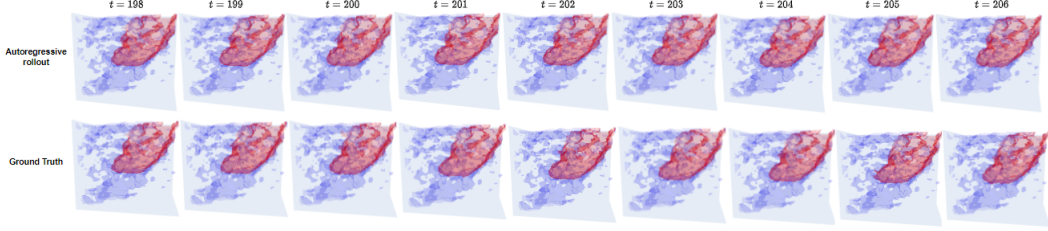


Figure 6: Visualisation (see video) of ground truth and autoregressive rollout from $t = 198$ to 206 on the **training set with minor oscillations**. Red parts represent gas. Blue parts represent liquid.

C Loss functions comparison

We compared different loss functions mentioned in [19][20][21][22] on our training and validation sets. The training set includes 20 time steps with gradually emerging bubbles, while the validation set consists of 9 time steps with similar gradual bubble emergence.

$$\mathcal{L}_{\text{BCE}}(\mathbf{y}, \hat{\mathbf{y}}) = -\frac{1}{M} \sum_{i=1}^M [y_i \log \hat{y}_i + (1 - y_i) \log (1 - \hat{y}_i)] \quad (8)$$

$$\mathcal{L}_{\text{Dice}}(\mathbf{y}, \hat{\mathbf{y}}) = 1 - \frac{2|\mathbf{y} \cup \hat{\mathbf{y}}|}{|\mathbf{y}| + |\hat{\mathbf{y}}|} = 1 - \frac{2\langle \mathbf{y}, \hat{\mathbf{y}} \rangle}{\langle \mathbf{y}, \mathbf{y} \rangle + \langle \hat{\mathbf{y}}, \hat{\mathbf{y}} \rangle} \quad (9)$$

$$\mathcal{L}_{\text{Tversky}}(\mathbf{y}, \hat{\mathbf{y}}) = 1 - \frac{2|\mathbf{y} \cup \hat{\mathbf{y}}|}{\alpha|\mathbf{y}| + \beta|\hat{\mathbf{y}}|} = 1 - \frac{\langle \mathbf{y}, \hat{\mathbf{y}} \rangle}{\langle \mathbf{y}, \mathbf{y} \rangle + \alpha\langle \mathbf{y}, \mathbf{1} - \hat{\mathbf{y}} \rangle + \beta\langle \mathbf{1} - \mathbf{y}, \hat{\mathbf{y}} \rangle} \quad (10)$$

$$\mathcal{L}_{\text{Hard BCE Dice}}(\mathbf{y}, \hat{\mathbf{y}}) = \begin{cases} \mathcal{L}_{\text{BCE}}(\mathbf{y}, \hat{\mathbf{y}}), & \text{if } n < 0.9N \\ \mathcal{L}_{\text{Dice}}(\mathbf{y}, \hat{\mathbf{y}}), & \text{Otherwise} \end{cases} \quad (11)$$

$$\mathcal{L}_{\text{Distance map + BCE}}(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{N} \sum_{i=1}^N (1 + \Phi) \odot \mathcal{L}_{\text{BCE}}(\mathbf{y}, \hat{\mathbf{y}}) \quad (12)$$

$$\mathcal{L}_{\text{Distance map + Soft Dice BCE}}(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{N} \sum_{i=1}^N (1 + \Phi) \odot \mathcal{L}_{\text{Soft BCE Dice}}(\mathbf{y}, \hat{\mathbf{y}}) \quad (13)$$

$$\mathcal{L}_{\text{focal}}(\mathbf{y}, \hat{\mathbf{y}}) = -\sum_{i=1}^N \alpha_i (1 - \hat{y}_i)^\gamma y_i \log \hat{y}_i \quad (14)$$

Eq. (8) shows the Binary Cross-Entropy (BCE) loss, which measures the difference between two probability distributions but struggles with imbalanced data. Dice loss (see Eq. (9)) addresses this imbalance by considering the intersection and union of ground truth and predictions, though it may be less accurate than BCE alone. Tversky loss (see Eq. (10)) adjusts the weighting of false positives and false negative. Combining BCE and Dice loss, as in Eq. (5) and Eq. (11), improves performance by balancing accuracy and data imbalance.

The distance map indicates the distance of points with value 0 to the nearest point with value 1, serving as a weight to penalise points near the gas-liquid boundary more heavily. It can be combined with BCE and soft Dice BCE, as shown in Eq. (12) and (13). Additionally, focal loss (see Eq. (14)) emphasises ‘hard’ examples by reducing the loss contribution of easy samples.

Table 2 compares the mean surface area error of various loss functions on training and validation sets. The training set consists of 20 time steps and the testing set consists of 9 time steps. The soft dice BCE loss achieves the lowest mean surface area error on both sets, making it the preferred choice in our work.

Table 2: Comparison of mean surface area error $\bar{\epsilon}$ using different loss functions on training and testing sets. The surface area error for each time step is calculated as shown in Eq. (6). The reported error is the average across all time steps in autoregressive rollout (see Eq. (7)).

Loss function	$\bar{\epsilon}$ (Testing)	$\bar{\epsilon}$ (Training)
Soft dice bce	0.0988	0.0567
Hard dice bce	0.1269	0.0941
Distance map + bce	0.1310	0.0840
Distance map + soft dice bce	0.1550	0.1326
Focal loss	0.1122	0.0983
Tversky loss	0.2086	0.2510
Dice loss	0.2120	0.2442
BCE loss	0.1502	0.1272

Algorithm 1 Calculate Surface Area of a 3D Mesh in One Time Step

Require: A 3D binary fluid dataset representing the spatial distribution of fluid within a volume.

Ensure: The total surface area of the generated 3D mesh.

- 1: Use the Marching Cubes algorithm to extract the 3D mesh from the input binary fluid data. This process computes a set of vertices $V = \{v_1, v_2, \dots, v_n\}$ and a set of faces $F = \{f_1, f_2, \dots, f_m\}$ that define the geometry of the mesh.
 - 2: Initialise the total surface area: $A \leftarrow 0$
 - 3: **for** each face $f \in F$ **do**
 - 4: Identify the three vertices v_1, v_2, v_3 that form the triangular face f .
 - 5: Compute the edge vectors: $\vec{a} \leftarrow v_2 - v_1$ and $\vec{b} \leftarrow v_3 - v_1$.
 - 6: Calculate the cross product of the edge vectors: $\vec{c} \leftarrow \vec{a} \times \vec{b}$, which is perpendicular to the plane of the triangle.
 - 7: Compute the area of the triangle: $A_{\text{triangle}} \leftarrow \frac{1}{2} \|\vec{c}\|$, where $\|\vec{c}\|$ is the magnitude of the cross product vector, giving twice the area of the triangle.
 - 8: Accumulate the triangle area into the total surface area: $A \leftarrow A + A_{\text{triangle}}$
 - 9: **end for**
 - 10: **return** A {Return the total surface area of the 3D mesh}
-

D Links of visualisation videos

Visualization videos for Figures 2 through 6 can be viewed by clicking the hyperlinks. If the YouTube links are not functional, the corresponding URLs are provided below for direct access.

- Figure 2: <https://www.youtube.com/watch?v=Cy0piv1anCY>
- Figure 4: <https://www.youtube.com/watch?v=hbbWUJxL3ng>
- Figure 5: https://www.youtube.com/watch?v=nm4u_G7p0cs
- Figure 6: https://www.youtube.com/watch?v=Rp_eI_iVqrY