17th International Conference on Greenhouse Gas Control Technologies, GHGT-17

20th -24th October 2024 Calgary, Canada

Unveiling the Potential of MeshGraphNets for Predicting Subsurface Evolution in Carbon Storage Projects

Paul Holcomb[a,b]; Alex Sun[a,b]; Chung Yan Shih[c]; Guoxiang Liu[c]; Hema Siriwardane[a]

[a]*National Energy Technology Laboratory, 3610 Collins Ferry Road, Morgantown, WV 26505, USA*

[b]*NETL Support Contractor, 3610 Collins Ferry Road, Morgantown, WV 26505, USA*

[c]*National Energy Technology Laboratory, 626 Cochran Mill Road, Pittsburgh, PA 15236, USA;*

Abstract

Carbon capture and storage (CCS) technology is critical for mitigating climate change but requires effective subsurface reservoir management to ensure safe containment of injected CO_2. Accurate predictions of reservoir pressure and saturation are essential for assessing long-term CCS performance. Traditional numerical simulations, while effective, are computationally intensive, time-consuming, and constrained by data discretization. Previous work has shown the effectiveness of MeshGraphNets (MGN), a graph-based machine learning framework introduced by Pfaff et al.[1], as an innovative alternative for predicting reservoir behavior[2]. MGN leverages graph neural networks (GNNs) and mesh representations to model complex geological formations, offering superior adaptability across different discretizations and reservoir configurations.

Classic MGN implementations utilize an autoregressive technique to predict future behavior based on current predictions, but this technique is hampered by error accumulation over time. To enhance the model accuracy in time-series predictions, this study implemented a multi-step rollout strategy that integrates autoregressive predictions during training to stabilize prediction of saturation over time. Using the Illinois Basin – Decatur Project (IBDP) dataset, comprising 100 simulations of CO_2 injection, pressure, and saturation changes, the framework demonstrated its ability to learn spatial dependencies and temporal dynamics. With inputs including permeabilities, porosities, and injection rates, MGN accurately predicted CO_2 plume evolution over time, even with limited training data. Moreover, the addition of a multi-step rollout procedure during training improved the ability of MGN to predict stably over time by ~15%.

This research positions MGN, enhanced with multi-step rollout capabilities, as a robust and efficient tool for CCS applications. It advances the field by enabling precise, computationally efficient predictions of reservoir behavior, providing a foundation for the broader adoption of machine learning frameworks in CCS and other geoscience domains.

Keywords: Deep Learning; Graph Neural Network; MeshGraphNets

1. Introduction

Achieving global net-zero carbon emissions by mid-century is a cornerstone of international efforts to combat climate change, as outlined in frameworks such as the Paris Agreement [3]. Carbon capture and storage (CCS) is recognized as one of the most promising technologies to meet these goals by enabling the secure, long-term storage of CO_2 in deep geological reservoirs [4]. To ensure the safe and efficient deployment of CCS, it is critical to accurately predict subsurface reservoir behavior, particularly pressure and saturation dynamics, which play key roles in assessing injection feasibility and storage integrity. However, traditional numerical simulations, which solve complex partial

differential equations to model subsurface flow and transport processes, face challenges in scalability due to their computational expense, time-intensive nature, and reliance on domain expertise [5]. These limitations pose significant obstacles to the rapid analysis and optimization of CCS projects.

The emergence of machine learning (ML) and deep learning (DL) offers new avenues to complement traditional simulation methods. These techniques excel in identifying patterns within large datasets and generating predictive models, providing significant reductions in computational costs and analysis time while retaining predictive accuracy. In particular, graph neural networks (GNNs) have gained attention as a flexible and effective approach for modeling systems with inherent graph structures, such as geological formations. GNNs can capture spatial relationships and dependencies within geological data, offering a powerful tool for subsurface modeling [6].

Building on the capabilities of GNNs, MeshGraphNets (MGN), introduced by Pfaff et al. [1], provide a robust framework for analyzing graph-structured data. By incorporating mesh-based representations, commonly used in finite element methods, MGN bridges physics-informed modeling with data-driven learning. This framework is particularly suited for CCS applications, as it can adapt to varying discretizations and capture complex interactions within geological formations. Using data from the Illinois Basin – Decatur Project (IBDP), a landmark CCS initiative, the multi-year research plan evaluates the scalability, efficiency, and predictive accuracy of MGN in modeling CCS reservoir dynamics. This study focuses on the feasibility of improving the predictive accuracy of MGN across time using a customized multi-step rollout (MGN-MR) procedure during training. As such, this paper focuses specifically on the prediction of changes in subsurface saturation during injection and compares the two proposed techniques.

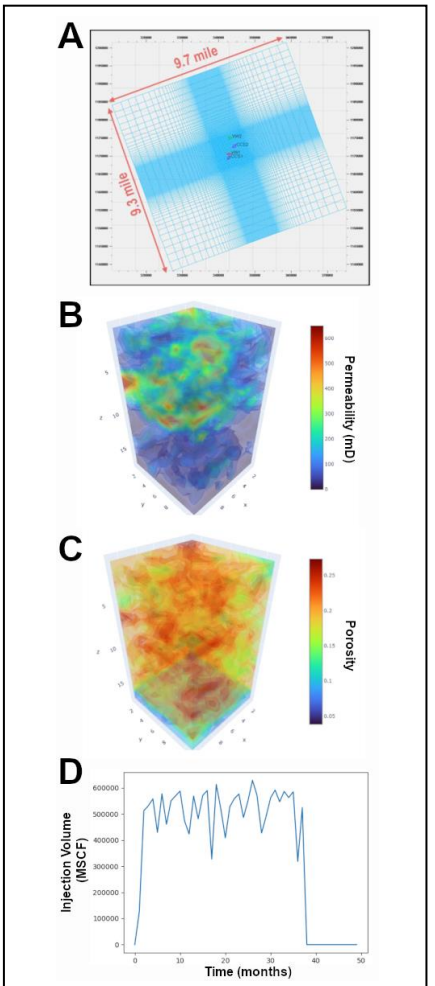

Figure 1: A summary of the IBDP dataset including graph extents (A), permeability and porosity for one realization (B and C, respectively), and the injection volume over time into the reservoir (D).

2. Study Site – Illinois Basin Decatur Project (IBDP)

The Illinois Basin – Decatur Project (IBDP) was selected for this study due to its comprehensive data availability and its alignment with the SMART Initiative [7], [8]. This pilot project successfully injected one million metric tons of CO_2 into the Mt. Simon Sandstone, a saline reservoir in Decatur, Illinois, beginning on November 17, 2011, at a rate of approximately 1,000 metric tons per day. By November 2014, the project reached its injection goal. Notably, the IBDP demonstrated superior capacity, injectivity, and containment potential, exceeding initial expectations [7], [9], [10], [11], [12].

The Mt. Simon Sandstone, the primary CO_2 storage formation in both the Illinois Basin and the broader Midwest region, is more than 457 meters thick at the IBDP site. It is capped by the Eau Claire Formation, an impermeable layer providing an effective seal. The upper portion of the Mt. Simon Sandstone was deposited in a tidally influenced environment, while the lower 183 meters is arkosic sandstone from a braided river and alluvial fan system, offering enhanced secondary porosity due to feldspar grain dissolution.

IBDP holds the distinction of being the world's first bioenergy carbon capture and storage (BECCS) project, integrating a full-scale industrial CCS system using CO_2 captured from Archer Daniels Midland's (ADM) ethanol fermentation process. The project infrastructure consisted of a compression and dehydration facility, a CO_2 delivery pipeline, an injection well, a monitoring well, a geophysical test well, and an environmental monitoring system.

Real-time monitoring of the project was facilitated by custom software systems such as the Real-Time Acquisition and Control (RTAC) system

and the Well Watcher Connect system. These systems continuously recorded data on wellhead pressure, downhole conditions, and CO_2 volumes, ensuring rigorous operational monitoring and compliance with reporting requirements.

The CO_2 injector (CCS1) had three perforation zones, while additional monitoring took place through a deep observation well (VW1) with multi-depth sensors. Additionally, microseismic data from downhole arrays in the injection and monitoring wells provided critical insights into subsurface behavior.

The SMART team generated 100 simulations of IBDP CO_2 pressure and saturation changes. The IBDP dataset used for this study was comprised of 100 ensemble simulations, each having a grid size of 126 × 125 × 110 (x, y, z) and included 3D permeability, porosity, 50 monthly model outputs for pressure and saturation distributions, and 1,521 daily time steps for injection data aggregated into monthly injection rates (Figure 1). The MGN (20 and 50 time steps) and MGN-MR (20 time steps) models were tested using this data to quantify their effectiveness in predicting saturation changes over time.

3. Method

3.1. Graph Neural Networks (GNNs)

Graph Neural Networks (GNNs) are a class of deep learning models designed to operate on graph-structured data [13], [6]. They excel at capturing relationships between nodes (vertices) and their connections (edges), making them effective for problems where data is inherently represented as graphs, such as social networks, molecular structures, and geological formations.

A graph is defined as $G = (V, E)$, where V is the set of nodes, and E is the set of edges. Each node $i \in V$ has associated features $x_i \in R^d$, where d is the dimension of a node feature vector. Each edge $e \in E$ may have features $e_{ij} \in R^k$, where k is the dimension of an edge feature vector. The goal of a GNN is to learn node representations, node links, or graph-level representations. The features as "messages" can be aggregated and passed through the connected network [14]. Each node v receives information from its neighbors. The hidden state at each node $h_i^{(t+1)}$ is updated based on:

$$h_i^{(\ell)} = \psi\left(h_i^{(\ell-1)}, \text{AGGREGATE}_{j\in N(i)}\phi\left(h_i^{(\ell-1)}, h_j^{(\ell-1)}, e_{ij}\right)\right) \quad (1)$$

3.2. where ψ and ϕ are learnable functions, often implemented as a Multi-Layer Perceptron (MLP), and ℓ is layer index. MeshGraphNets (MGN)

MeshGraphNets is a machine learning framework specifically designed to model physical systems represented as graphs, as introduced by Pfaff et al. [1]. It combines the flexibility of GNNs with the structured representation of mesh-based data, making it suitable for scientific and engineering simulations. Inherited from GNNs, the MeshGraphNets framework processes graph-structured data, where a graph $G = (V, E)$, consists of nodes V , edges E , and associated features.

The architecture of MeshGraphNets consists of three main stages (Figure 2):

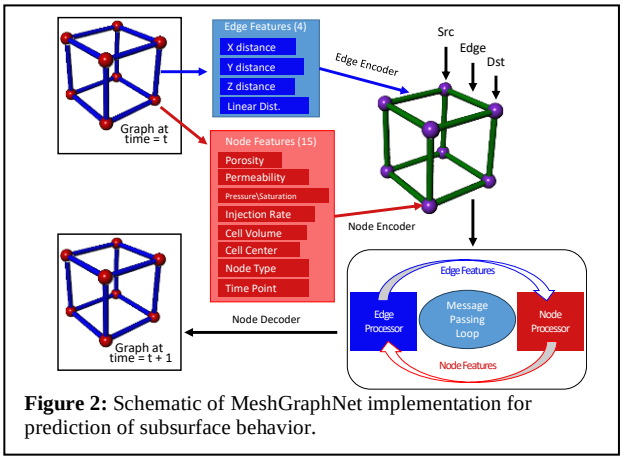

Figure 2: Schematic of MeshGraphNet implementation for prediction of subsurface behavior.

Encoding

Node and edge features are encoded into latent embeddings using neural networks:

$$h_i^{(0)} = \mathrm{ENC}_v(x_i), \quad h_{ij}^{(0)} = \mathrm{ENC}_e(e_{ij}) \tag{2}$$

where ENC_v and ENC_e are learnable encoders, and x_i and e_{ij} are node and edge features, respectively.

Processing

Message passing updates node and edge embeddings iteratively. For each time step t, messages are computed and aggregated:

$$m_{ij}^{(t)} = \phi\left(h_i^{(t-1)}, h_j^{(t-1)}, h_{ij}^{(t-1)}\right) \tag{3}$$

$$h_i^{(t)} = \psi\left(h_i^{(t-1)}, \mathrm{AGGREGATE}_{j\in\mathcal{N}(i)}\left(m_{ij}^{(t)}\right)\right) \tag{4}$$

where ϕ and ψ are MLPs, and AGGREGATE combines incoming messages (e.g., by summation).

Decoding

The processed node embeddings are decoded to predict target properties:

$$y_i = \mathrm{DEC}_v\left(h_i^{(T)}\right) \tag{5}$$

where DEC_v is a learnable decoder, and y_i is the prediction for node i.

In the processing stage, MeshGraphNets performs message passing to iteratively update node and edge embeddings. Traditionally, node updates are made at each time step t based on the previous state at $t-1$. However, this approach can struggle to capture long-term dependencies, as the error or change at each step might not provide a sufficient signal for learning long-term behaviors. As a consequence, when this model is used for inferencing in an autoregressive manner, small errors in prediction can accumulate over time, pushing the prediction further and further from the target behavior.

3.3. MeshGraphNets with Multi-step Rollout (MGN-MR)

To address this drawback in using the MGN model, a multi-step rollout procedure during training is implemented. In this approach, rather than updating the model directly at each timestep based on the previous timestep's state, the system simulates a multi-step trajectory using a rollout of k steps, and the error is computed over these k steps to guide learning (Figure 3). This allows the model to account for the accumulation of changes over a longer horizon, providing richer signals for learning:

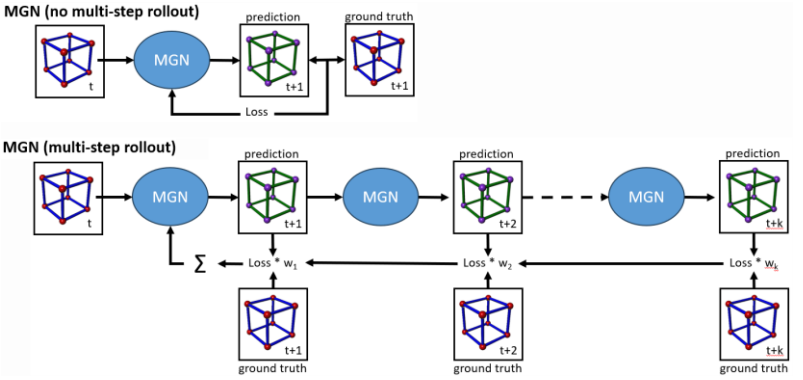

Figure 3: Diagram of loss calculation for standard MeshGraphNets model (MGN, top) and MeshGraphNets with Multi-step Rollout (MGN-MR, bottom)

$$h_i(t+k) = \mathrm{ROLL\_OUT}\left(h_i(t), \{h_j(t): v_j \in \mathcal{N}(v_i)\}, k\right) \tag{6}$$

The error over the multi-step rollout is computed by comparing the predicted final state $\widehat{h_i}(t+k)$ with the true observed state $h_i(t+k)$. This error is typically computed using a loss function like Root Mean Squared Error (RMSE):

$$\mathrm{RMSE}(t+k) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left\| h_i^{(t+k)} - \hat{\mathrm{h}}_{\mathrm{i}}^{(t+k)} \right\|^2} \quad (7)$$

The total error is accumulated over all timesteps and rollouts as:

$$L_{\mathrm{total}} = \sum_t \sum_{k=1}^{K} \mathrm{w_t} * RMSE(t,k) \quad (8)$$

where k is the maximum number of rollout steps, and $RMSE(t,k)$ represents the error computed from the predicted node embeddings at timestep $t+k$ compared to the true target. This allows the model to learn long-term behaviors by utilizing the error signal from the entire rollout trajectory and has been shown previously to improve overall prediction accuracy in GNN models both in subsurface prediction [15] and pressure and flow modeling in the cardiovascular system [16]. In this study, edges contain the x, y, and z distances between two nodes. The node features include time point, porosity, permeability, saturation, as well as cell-specific metrics such as cell volume, cell center, and node type.

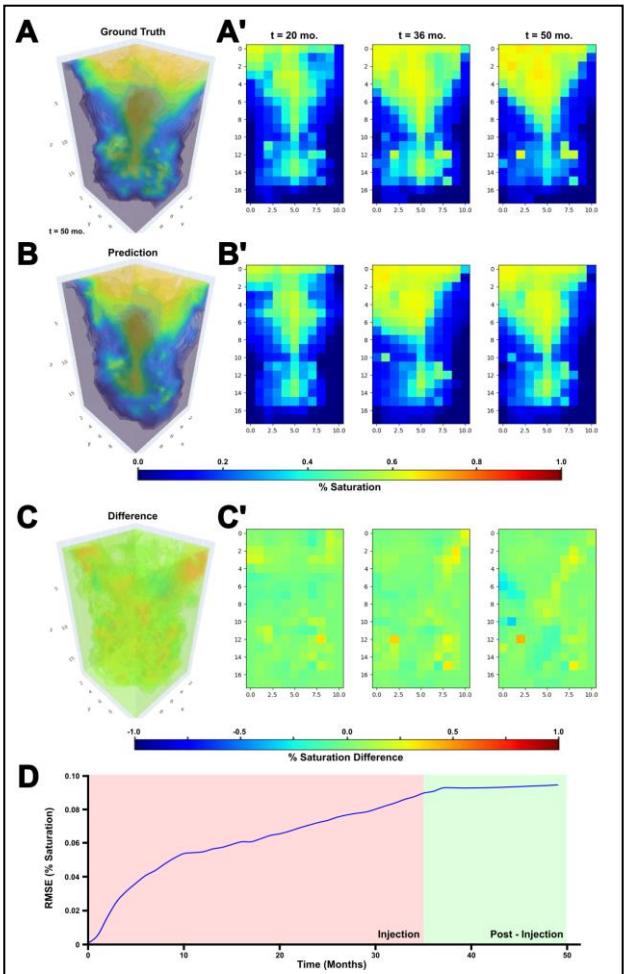

Figure 4: Results of MGN model trained on 50 time points. Ground truth (A) and predicted (B) values were compared in 3D as well as at three time points in 2D (YZ plane, A' and B') to determine the accuracy of the prediction (C, C'). Root mean squared error (RMSE) in prediction increased over time during injection (D, pink region) and stabilized during post-injection (D, green region).

3.4. Computational Hardware

All training was performed on a custom-built PC with an AMD Ryzen 9 7950X3D 16-Core processor, 64GB of DDR5 5600 RAM, and a NVIDIA GeForce RTX 4070 Ti OC GPU with 12GB of GDDR6X RAM. Models were run in an emulated Ubuntu workspace on Windows 11 using WSL2 (Linux-5.15.133.1) and a custom Anaconda environment running Python 3.10.13 and NVIDIA Modulus v0.8.0.

3.5. Data

Of the 100 realizations provided by the IBDP dataset, the last 20 realizations (80-100) could not be used due to scaling differences. The remaining 80 realizations were separated into 64 training, 8 validation, and 8 test realizations; this was accomplished by taking the first 8 of every 10 realizations as training data, with the remaining 2 realizations divided into the validation and test datasets. As mentioned previously, each realization was comprised of 50 monthly timepoints, representing 36 months of injection and 14 months post-injection. The full extents of the data were 126 × 125 × 110 cells, with 11 cell locations representing the injection locations. To conduct proof-of-concept experiments while maintaining the temporal dynamics of the system, in this study we used a subvolume of 11 × 11 × 18 centered around the injection cells.

4. Results

4.1. MGN Saturation Prediction (50 time points)

To establish a benchmark for performance evaluation on saturation prediction over time, a MGN model was trained using all 50 time points in the IBDP training dataset. Ground truth and predicted values were compared both in 3D at the 50-month time point (Figure 4, A and B, respectively) and as single slices through the YZ plane at the level of the injection well. Three time points were considered: 20 months after injection started, the end of injection at 36 months, and at 50 months (14 months post-injection) (Figure 4, A' and B', respectively). The average difference between the ground truth and predicted values across all realizations at all time points was found to be 0.012 ± 0.072, with an average RMSE of 7.35% (Table 1). As the prediction of saturation within the volume starts from zero at all points, we differentiated between the average RMSE of zero values—the model's ability to accurately predict where CO_2 should not be—and nonzero average RMSE, which measures the ability of the model to predict the correct percent saturation at locations that are nonzero. The RMSE values for zero and nonzero locations for the 50 time point model were 1.69% and 9.14%, respectively (Table 1). As prediction in this method is done in an autoregressive manner, with the input for the model at the current time point being the predicted output from the previous time point, it is imperative to understand the performance of the system as prediction proceeds through time. To accomplish this, the average RMSE value for saturation was calculated after each time point prediction. The average RMSE steadily increased over the injection period, reaching an asymptote at 36 months when injection was stopped. Beyond 36 months, the RMSE of the predictions appears relatively constant, suggesting that the model performs well for predicting the evolution of CO_2 saturation under steady-state conditions (Figure 4D).

4.2. MGN Saturation Prediction (20 and 36 time points)

To address the increasing average RMSE values during prediction in a timely manner, we trained a second MGN model using only the first 20 time points of the IBDP dataset. The benefits of reducing the number of time points in this scenario are threefold. First, we dramatically reduced the amount of time necessary to train the model from 3 hours 37 minutes for the full 50 time points to 1 hour 30 minutes for 20 time points. Second, we focused the model on the time period where the temporal increase in RMSE is most profound while eliminating the potential confound of including post-injection data with markedly different physical characteristics. Finally, reducing the number of time points afforded the opportunity to determine the accuracy of the MGN model in forward prediction; while the MGN model was trained on 20 time points in this case,

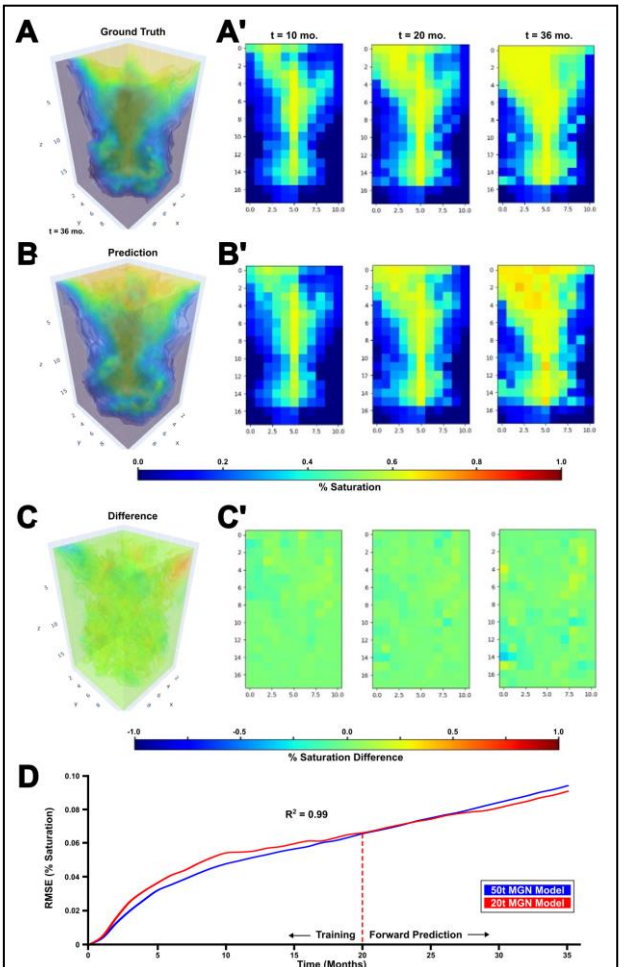

Figure 5: Results of MGN model trained on 20 time points and used to predict out to month 36. Agreement between ground truth (A) and predicted (B) values was high for both 3D as well as three time points in 2D (YZ plane, A' and B') when the difference between the two datasets is taken (C, C'). A high degree of correlation exists in average RMSE over time (D) between the MGN model trained with 50 time points (blue) and the 20 time point model (red), both in the training space (left of dotted red line) and in the forward prediction of 21-36 months from the 20 time point model (right of dotted red line).

predictions were performed out to 36 months. The difference between the predicted values and ground truth for this new model was quantified both volumetrically (Figure 5A-C) and in single YZ slices across several time points (Figure 5A'-C') and found to be 0.009±0.062. The average RMSE for this model was 6.3%, with the zero RMSE and nonzero RMSE being 2.9% and 7.87% respectively (Table 1). Though not explored quantitatively, the reduction in RMSE between this model and the full 50 time point model is assumed not to derive from a greater accuracy in the model, but rather to a lower number of non-zero cells early in injection. This assumption is strengthened by a comparison of the average RMSE over time from the predicted results of the reduced model to those of the model trained on 50 time points, which shows good correlation between the two models, both for the first 20 timesteps and for the forward prediction to 36 timesteps (R = 0.99 for both comparisons). These results suggest that using 20 time points for training rather than the original 50 time points is sufficient to produce a model that accurately reflects the behavior of this system, and that the accuracy of the model is maintained when predicting beyond the time on which it was trained.

4.3. MGN-MR Saturation Prediction (20 and 36 time points)

In order to determine if the multi-step rollout procedure is effective, the MGN-MR model was trained on the same dataset as the 20 time point MGN model using the same hyperparameters and random seed to minimize variability. A drawback to this technique that was identified prior to any training is the length of time necessary to train the model; compared to 1 hour 30 minutes to train the MGN model, the MGN-MR model required 6 hours 17 minutes to train. This increase in training time is owing to the need to pass each batch of graphs through the model 5 times rather than the single pass for the MGN model. Inferencing time, however, remains unchanged (~93 seconds for 8 realizations with 50 time points, 0.23 seconds per time point). As in the previous model, the MGN-MR results were quantified both volumetrically (Figure 6 A, B) and in 2D YZ slices through the injection nodes (Figure 6 A', B') with the average difference between the ground truth and prediction found to be 0.012 ± 0.051 (Figure 6 C, C'). The average RMSE of the MGN-MR model was 5.27%, with a zero RMSE of 1.14% and a nonzero RMSE of 6.83% (Table 1). This represents a 16.3% increase in accuracy with the MGN-MR model over the MGN model alone. The average improvement in temporal prediction between the MGN and MGN-MR model was 0.92% (Figure 6D, red vs green) but was heavily skewed toward a larger difference at later time points. Analysis of the difference in RMSE between the MGN and MGN-MR model showed a linear divergence in RMSE over time (Figure 6E, R^2 = 0.918) suggesting that the MGN-MR model exerts a stabilizing influence on temporal prediction.

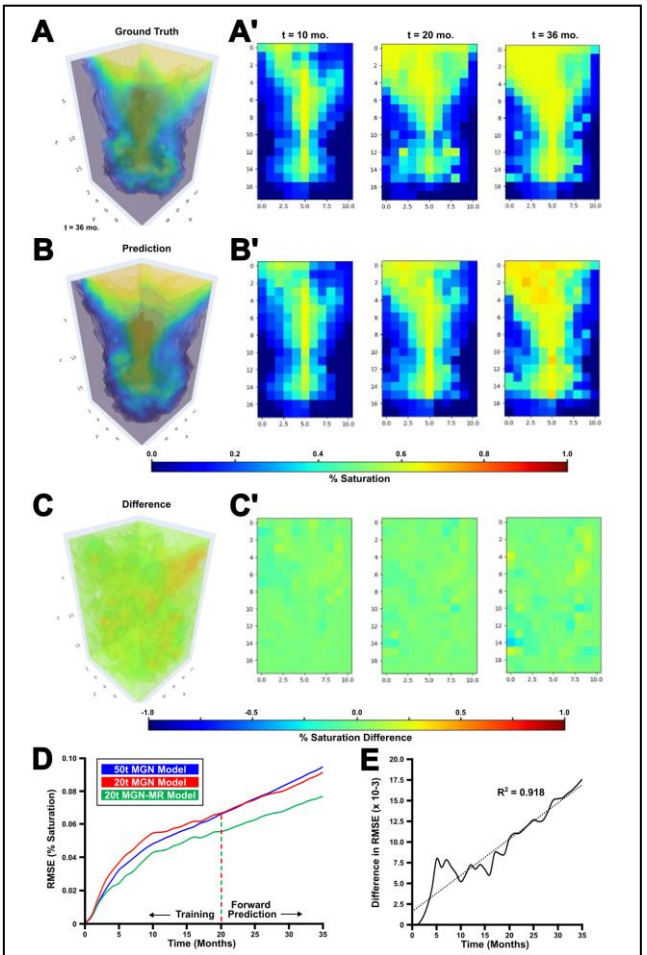

Figure 6: Comparison of MGN-MR model prediction with both ground truth and MGN prediction. Ground truth (A) and predicted (B) values for MGN-MR were similar for both 3D and three time points in 2D (YZ plane, A' and B') as shown by low variability between the two datasets (C, C'). The MGN-MR model (D, green) outperformed the standard MGN models (D, blue and red) in RMSE and showed similar consistency to the MGN model in forward prediction (right of the dashed red/green line). The difference in RMSE between the MGN and MGN-MR models follows a linear trend (E, R^2 = 0.918).

5. Conclusion

The reliable prediction of subsurface conditions is vital to the analysis of feasibility, cost, and risk for carbon capture and storage. These predictions have classically been the domain of numeric solvers, but recent advances in artificial intelligence have provided avenues for both more rapid and more accurate solutions. Graph neural network models, and particularly the MeshGraphNets model, are particularly suited for this regime thanks to their ability to handle unstructured data, such as that commonly found in geological systems. However, the application of MGN has been hampered by the "temporal drift" during autoregressive predictions that causes error to build over time. This study supports the application of multi-step rollout during MGN training as a reliable method to partially mitigate this drawback in MGN. An average improvement of 16.3% in saturation RMSE value was observed in the MGN-MR model compared to the standard MGN model. Additionally, the MGN-MR model displayed greater stability in forward prediction over time. Future efforts will explore improving the performance of the MGN-MR model to stabilize prediction over time and increase training speed. In conclusion, this work provides a pathway to greater confidence in using MGN for subsurface prediction and positions the MGN model, enhanced with multi-step rollout capabilities, as a robust and efficient tool for CCS applications.

Table 1

	Average RMSE	Zeros RMSE	Nonzeros RMSE
MGN (50 time points)	0.0735	0.0169	0.0914
MGN (20 time points)	0.063	0.029	0.0787
MGN-MR (20 time points)	0.0527	0.0114	0.0683

Acknowledgements

This work was supported by US Department of Energy Office of Fossil Energy and Carbon Management project—the Science-informed Machine Learning for Accelerating Real-Time Decisions in Subsurface Applications (SMART) Initiative. This conference proceeding paper was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of the presenter do not necessarily state or reflect those of the United States Government or any agency thereof.

References

[1] T. Pfaff, M. Fortunato, A. Sanchez-Gonzalez, and P. W. Battaglia, "Learning Mesh-Based Simulation with Graph Networks," Jun. 18, 2021, *arXiv*: arXiv:2010.03409. doi: 10.48550/arXiv.2010.03409.

[2] P. Holcomb, C. Y. Shih, A. Sun, G. Liu, and H. Siriwardane, "Predicting CO_2 Plume Migration in Carbon Storage Projects using Graph Neural Networks." Accessed: Dec. 13, 2024. [Online]. Available: https://www.nvidia.com/gtc/sessions/oil-and-gas/[3] United Nations Climate Change, "The Paris Agreement | UNFCCC." Accessed: Dec. 07, 2024. [Online]. Available: https://unfccc.int/process-and-meetings/the-paris-agreement

[4] Intergovernmental Panel on Climate Change (IPCC), Ed., Climate Change 2022 - Mitigation of Climate Change: Working Group III Contribution to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change. Cambridge: Cambridge University Press, 2023. doi: 10.1017/9781009157926.

[5] X. Wu, C. Y. Shih, G. Liu, M. Mark-Moser, and P. Wingo, "A Machine Learning-Based Field-Scale Reservoir Analysis Method," presented at the Proceedings of the 16th Greenhouse Gas Control Technologies Conference (GHGT-16), 2022, pp. 23–24.

[6] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A Comprehensive Survey on Graph Neural Networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 4–24, Jan. 2021, doi: 10.1109/TNNLS.2020.2978386.
[7] R. A. Bauer, R. Will, S. E. Greenberg, and S. G. Whittaker, "Illinois Basin–Decatur Project," in *Geophysics and Geosequestration*, M. Wilson, M. Landrø, and T. L. Davis, Eds., Cambridge: Cambridge University Press, 2019, pp. 339–370. doi: 10.1017/9781316480724.020.
[8] R. J. Finley, "An overview of the Illinois Basin – Decatur Project," *Greenh. Gases Sci. Technol.*, vol. 4, no. 5, pp. 571–579, 2014, doi: 10.1002/ghg.1433.
[9] "Illinois State Geological Survey (ISGS), Illinois Basin - Decatur Project (IBDP) Well Information, April 30, 2021. Midwest Geological Sequestration Consortium (MGSC) Phase III Data Sets. DOE Cooperative Agreement No. DE-FC26-05NT42588. - EDX." Accessed: Dec. 09, 2024. [Online]. Available: https://edx.netl.doe.gov/dataset/illinois-state-geological-survey-isgs-illinois-basin-decatur-project-ibdp-well-information
[10] "Illinois Basin – Decatur Project," *Greenh. Gases Sci. Technol.*, vol. 4, no. 5, pp. 569–570, 2014, doi: 10.1002/ghg.1467.
[11] "Illinois State Geological Survey (ISGS), Illinois Basin - Decatur Project (IBDP) MVA GIS Locations File Geodatabase and Aerial Imagery, April 30, 2021. Midwest Geological Sequestration Consortium (MGSC) Phase III Data Sets. DOE Cooperative Agreement No. DE-FC26-05NT42588. - EDX." Accessed: Dec. 09, 2024. [Online]. Available: https://edx.netl.doe.gov/dataset/illinois-state-geological-survey-isgs-illinois-basin-decatur-project-ibdp-mva-gis-location-data
[12] "Illinois State Geological Survey (ISGS), Illinois Basin - Decatur Project (IBDP) Selected Reports, April 30, 2021. Midwest Geological Sequestration Consortium (MGSC) Phase III Data Sets. DOE Cooperative Agreement No. DE-FC26-05NT42588. - EDX." Accessed: Dec. 09, 2024. [Online]. Available: https://edx.netl.doe.gov/dataset/illinois-state-geological-survey-isgs-illinois-basin-decatur-project-ibdp-selected-reports
[13] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The Graph Neural Network Model," *IEEE Trans. Neural Netw.*, vol. 20, no. 1, pp. 61–80, Jan. 2009, doi: 10.1109/TNN.2008.2005605.
[14] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural Message Passing for Quantum Chemistry," Jun. 12, 2017, *arXiv*: arXiv:1704.01212. doi: 10.48550/arXiv.1704.01212.
[15] L. Pegolotti, et al, "Learning Reduced-Order Models for Cardiovascular Simulations with Graph Neural Networks," *Computers in Biology and Medicine*, vol. 168, Jan. 2024, arXiv:2303.07310. doi: 10.1016/j.compbiomed.2024.107676.
[16] T. Wu, et al. "Learning Large-scale Subsurface Simulations with a Hybrid Graph Network Simulator," Jun. 15, 2022, *arXiv*: arXiv:2206.07680v1. doi: 10.1145/3534678.3539045