Fluid Dynamics &
Materials Processing

Tech Science Press

http://doi.org/10.32604/fdmp.2026.087218

ARTICLE

Benchmarking Physical-Parameter Conditioning Strategies for Data-Driven Hydro-Mechanical Field Forecasting

Zongzheng Jiao[1], Shuaikang Yang[1], Junlong Yin[1], Shaohui Wang[2] and Minpo Jung[1,*]

[1]Department of Computer Information Engineering, College of Engineering, Graduate School, Youngsan University, Yangsan, Republic of Korea
[2]College of Civil Engineering, Guangxi Polytechnic Vocational Technical School, Nanning, China
*Corresponding Author: Minpo Jung. Email: minpo@ysu.ac.kr

Received: 12 June 2026; Accepted: 19 August 2026; Published: 04 September 2026

ABSTRACT: Hydro-mechanical (HM) simulations of porous-media systems—such as those used in geotechnical engineering, groundwater flow, formation consolidation, and underground-structure safety assessment—become computationally expensive when large material- and load-parameter spaces must be explored for design optimization, uncertainty quantification, or real-time decision support. Although data-driven surrogate models can accelerate such analyses, it remains unclear whether explicitly conditioning a history-based predictor on physical parameters offers a meaningful advantage over learning directly from the temporal evolution of the physical fields. This study systematically benchmarks five physical-parameter conditioning strategies—token concatenation, feature-wise linear modulation (FiLM), weak FiLM, adaptive instance normalization (AdaIN), and gated residual conditioning (GRC)—implemented within a common convolutional neural network (CNN)–Transformer backbone. The evaluation uses a 108-case OpenGeoSys consolidation dataset and a unified protocol comprising a strict case-level split, five independent training repetitions, random-condition and zero-condition controls, an α-ablation study, and equal-budget, mechanism-specific hyperparameter tuning. The results show that the temporal evolution of the field variables contains most of the information required for forecasting, while the physical parameters primarily provide an auxiliary correction signal. Among the explicitly conditioned models, GRC provides the best overall combination of predictive accuracy and robustness. Under the shared training protocol, however, the unconditional CNN-Transformer slightly outperforms all conditioned variants, while the long short-term memory (LSTM) baseline achieves the lowest overall prediction error. These findings are specific to the present fixed-node, single-geometry benchmark and should not be interpreted as a universal ranking of surrogate-model architectures.

KEYWORDS: Hydro-mechanical coupling; data-driven field forecasting; physical-parameter conditioning; OpenGeoSys; CNN-Transformer; gated residual conditioning; surrogate-model benchmarking

1 Introduction

Hydro-mechanical (HM) coupling in porous media underpins many applications in geotechnical engineering, groundwater flow, formation consolidation, and the safety assessment of underground structures. Biot's consolidation theory [1] and Terzaghi's one-dimensional consolidation theory [2] established the foundations for coupled analyses of pore pressure and solid-skeleton deformation, while open-source platforms such as OpenGeoSys enable reliable simulations of complex thermo-hydro-mechanical (THM) and thermo-hydro-mechanical-chemical (THMC) processes [3]. However, design optimization, uncertainty quantification, and real-time decision support often require repeated simulations across many combinations

Copyright © 2026 The Authors. Published by Tech Science Press. This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

of material properties, loading conditions, and permeability values, resulting in substantial computational cost. Data-driven surrogate models offer a promising means of accelerating these analyses.

Machine learning has increasingly been applied to fluid mechanics and other complex physical systems [4]. Physics-informed learning [5,6], DeepONet [7], and the Fourier neural operator (FNO) [8] have further expanded data-driven approaches to partial differential equations, while the integration of scientific knowledge and machine learning has emerged as an important direction in engineering-system modeling [9].

Recent studies have extended neural operators and other surrogate architectures to subsurface flow, geological CO_2 storage, and reservoir simulation. Nested FNO, Fourier-enhanced multiple-input neural operator (Fourier-MIONet), U-Net-enhanced Deep Operator Network (U-DeepONet), FNO-based reservoir proxies, and graph U-Net frameworks have been used to predict pressure, saturation, CO_2 migration, and multiphase-flow fields [10–14]. U-Net-enhanced Fourier neural operator (U-FNO) and feature-modulated variants have also been developed for multiphase flow in porous media [15,16]. Collectively, these studies demonstrate the growing importance of FNO, DeepONet, and graph neural networks in parameterized physical-field modeling. In HM applications, a recent physics-informed surrogate for unsaturated soil combined sparse measurements with mass- and force-balance constraints [17]. Although that physics-constrained formulation differs from the history-based sequence-forecasting task considered here, it further illustrates the increasing interest in data-driven HM surrogates.

DeepONet, FNO, and graph neural networks represent distinct surrogate-modeling paradigms and are not directly compared in this study. Instead, we isolate a narrower question: how should physical parameters be introduced into a fixed-node, sequence-based spatiotemporal predictor for multi-case HM consolidation? Despite growing work on parameterized field surrogates, targeted comparisons of conditioning strategies under a common backbone, data split, and training protocol remain limited. Consequently, the relative accuracy, robustness, and optimization sensitivity of different parameter-injection mechanisms are not yet well understood.

To address this gap, we use a unified lightweight convolutional–Transformer hydro-mechanical network (LCT-HMNet) backbone and treat Young's modulus E, Poisson's ratio ν, applied load, and permeability as explicit conditioning variables. Five injection strategies are benchmarked to determine how physical-parameter conditioning affects forecasting accuracy, run-to-run variability, and sensitivity to optimization.

The main contributions of this benchmark study are as follows:

(1) We conduct a controlled comparison of five physical-parameter conditioning strategies within the same fixed-node CNN-Transformer backbone, using a strict case-level split and a common baseline training protocol.
(2) We evaluate parameter use, optimization sensitivity, and training variability through five-seed repeated experiments, an α-ablation study, random-condition shuffling, zero-condition retraining, and equal-budget mechanism-specific tuning.
(3) We report pooled inverse-standardized metrics, variable-specific errors in physical units, representative spatial fields, and conditional inference times, while explicitly restricting the conclusions to a single fixed-node, shared-mesh OpenGeoSys consolidation benchmark.

Fig. 1 summarizes the overall workflow.

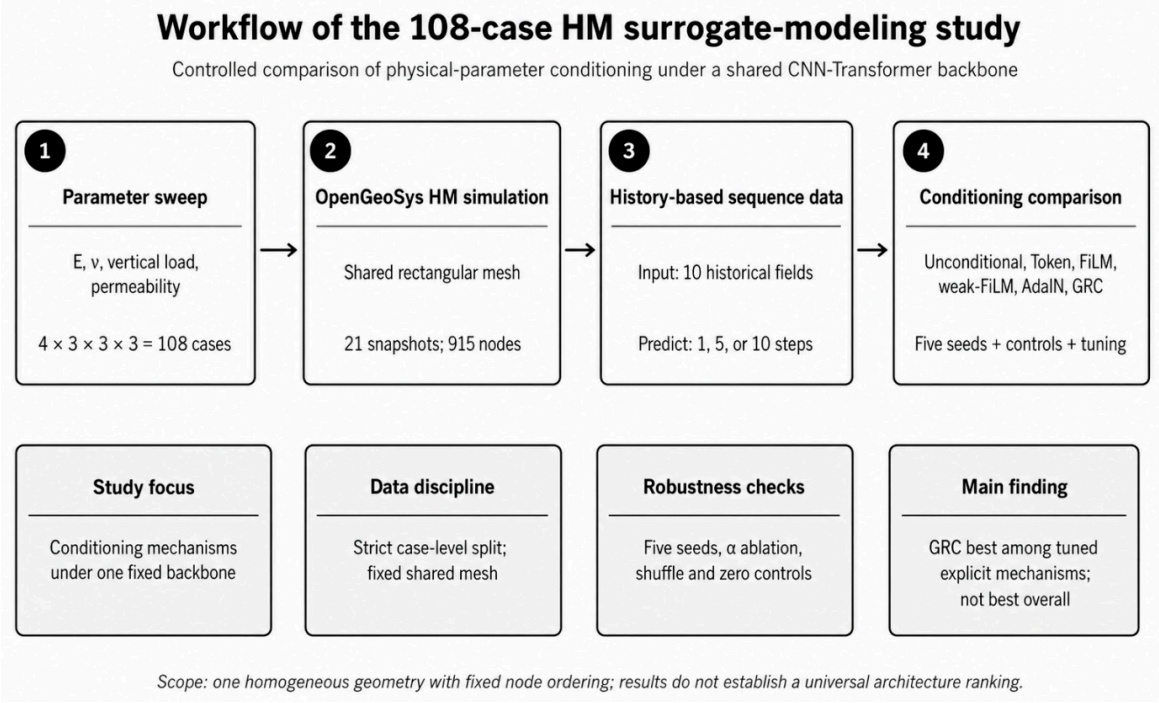

Figure 1: Overall workflow, robustness analyses, and scope of the 108-case HM surrogate-modeling study.

2 Dataset and Methods

2.1 OpenGeoSys Parameter Sweep and Dataset

HM field data were generated using the OpenGeoSys consolidation benchmark. The parameter sweep varied Young's modulus E, Poisson's ratio ν, the vertical load applied at the top boundary, and permeability. A full-factorial design produced 108 cases ($4 \times 3 \times 3 \times 3$). Each case contains 21 time steps on a mesh with 915 nodes and includes three field variables: pressure p and the horizontal and vertical displacement components u_x and u_y. All cases use the same finite-element mesh and node ordering, so a given node index corresponds to the same spatial location in every case and at every time step. Table 1 lists the parameter values, and Table 2 summarizes the dataset.

For reproducibility, only E, ν, top load, and permeability were varied across the 108 simulations; all other OpenGeoSys settings were held constant. The public release referenced in Availability of Data and Materials contains the project files, meshes, processed datasets, data splits, preprocessing scripts, training scripts, and run-level results.

The benchmark domain is a two-dimensional rectangle 0.1 m wide and 1.0 m high, spanning x = 0–0.1 m and y = −1–0 m. It is discretized into 412 six-node quadratic triangular elements (Visualization Toolkit (VTK) cell type 22) with 915 mesh points. Pressure is approximated using first-order shape functions and displacement using second-order shape functions, with an integration order of four. The constitutive model is isotropic linear elasticity, with E and ν varied as specified in Table 1.

The HM problem is solved using a staggered fixed-stress coupling scheme with a stabilization parameter of 1.0, fixed-stress treatment over each time step, and no more than two global coupling iterations. In the archived project, the Biot coefficient is 1.0, fluid viscosity is 1.0×10^{-3} Pa·s, fluid density is 1000 kg·m^{-3}, solid density and porosity are set to 0, the reference temperature is 293.15 K, and body force is neglected. These values reproduce the selected consolidation benchmark and are not intended to represent a general soil parameter set.

The initial pressure and displacement fields are zero. The top boundary is drained (p = 0), while the remaining hydraulic boundaries satisfy the natural no-flow condition. Horizontal displacement is fixed along the left and right boundaries, vertical displacement is fixed at the bottom, and a spatially uniform vertical Neumann load is applied at the top. Backward Euler integration is performed from t = 0 to 10 s using 20 fixed increments of 0.5 s, yielding 21 snapshots including the initial state. Separate Newton solvers for pressure and displacement allow up to 100 nonlinear iterations. The linear systems are solved using conjugate gradients with diagonal preconditioning, a maximum of 10,000 iterations, and a tolerance of 10^{-16}. The pressure process uses a relative DeltaX tolerance of 10^{-15}, the displacement process uses component-wise absolute tolerances of 10^{-14}, and the global coupling criteria use absolute tolerances of 10^{-14}. The exact OpenGeoSys release used for the original parameter sweep was not preserved in the archived metadata; however, the project, mesh, boundary-condition, and solver files are included in the reproducibility package.

Table 1: Parameter ranges in the 108-case OpenGeoSys HM dataset.

Parameter	Description	Unit	Values	Levels
E	Young's modulus	Pa	20,000, 26,667, 33,333, 40,000	4
ν	Poisson's ratio	—	0.15, 0.20, 0.25	3
top load	Vertical load at the top boundary	Pa	-500, -750, -1000	3
permeability	Permeability	m^2	5×10^{-11}, 1×10^{-10}, 2×10^{-10}	3

Table 2: Dataset summary and sequence-forecasting tasks.

Item	Value
Number of cases	108
Number of nodes	915
Time steps per case	21
Field variables	pressure, displacement_x, displacement_y
Training/validation/test cases	76/16/16
in10_out1 samples	1188
in10_out5 samples	756
in10_out10 samples	216

2.2 Sequence-Forecasting Tasks and Data Split

Each 21-step simulation was transformed into sequence-forecasting samples using a sliding window. Every sample contains 10 historical HM field snapshots and targets the subsequent 1, 5, or 10 time steps, denoted in10_out1, in10_out5, and in10_out10, respectively. To prevent leakage across temporal windows from the same simulation, the data were split strictly at the case level: 76 cases for training, 16 for validation, and 16 for testing. None of the parameter combinations assigned to the test set appeared in the training set; the test set therefore assesses generalization to unseen combinations within the sampled parameter grid. Fig. 2 illustrates the sample-construction procedure and the three forecasting horizons.

The dataset is sufficiently large for a controlled comparison of conditioning strategies on a shared mesh, but it cannot support a universal ranking of deep surrogate architectures. For the long-horizon in10_out10 task, each test case yields only two sliding windows, resulting in 32 test samples. The corresponding findings should therefore be interpreted as indicative evidence of long-horizon behavior rather than a statistically exhaustive assessment.

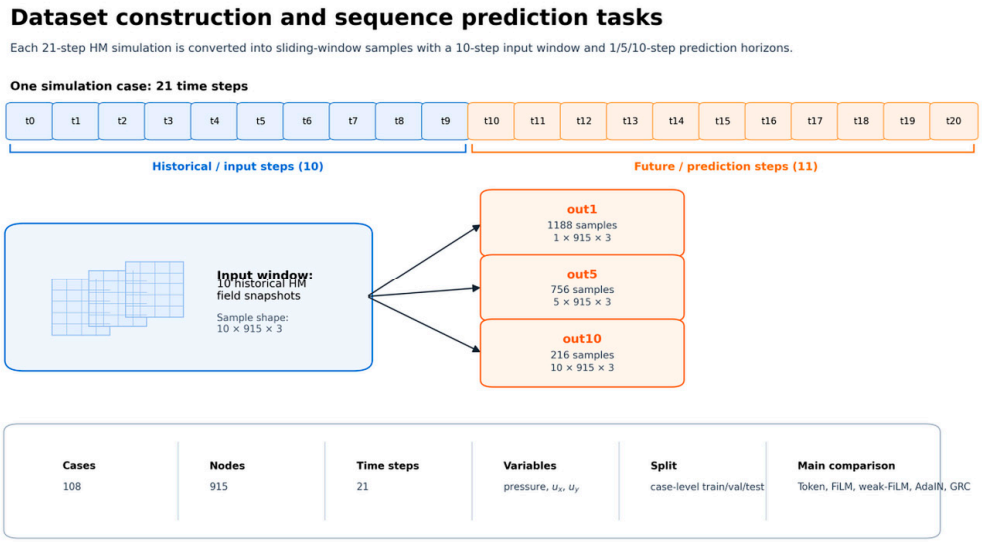

Figure 2: Construction of the dataset and the three history-based sequence-forecasting tasks.

2.3 LCT-HMNet Backbone and GRC Conditioning

The LCT-HMNet backbone comprises three modules: (i) a lightweight separable one-dimensional convolutional spatial encoder, (ii) a Transformer temporal encoder with learnable positional embeddings [18], and (iii) a linear prediction head. At each time step, the spatial encoder maps the nodal field snapshot (915 × 3) to a temporal token of dimension d_model = 128. A two-layer, four-head self-attention encoder then models dependencies across the 10 historical time steps. Finally, the prediction head decodes the temporal features into nodal fields for the next N time steps.

Here, "lightweight" refers specifically to the compact CNN-Transformer backbone, which uses separable convolution and a shallow Transformer encoder. The benchmark baselines comprise a multilayer perceptron (MLP), long short-term memory (LSTM), a gated recurrent unit (GRU), and a temporal convolutional network (TCN). GRU and LSTM contain approximately 2.87 and 3.24 million parameters, respectively, compared with approximately 4.12 million for LCT-HMNet-GRC. The term therefore describes the backbone design rather than implying that the complete model has the smallest parameter count.

Five conditioning strategies were implemented on this common backbone. Fig. 3 shows the architecture and data flow of gated residual conditioning (GRC), defined as follows:

$$\begin{aligned} &\mathbf{h}_p = \mathrm{Dropout}\left[\mathrm{GELU}\left(\mathbf{W}_p \mathrm{LN}(\tilde{\mathbf{c}}) + \mathbf{b}_p\right)\right] \\ &\mathbf{c} = \tanh\left(\mathbf{W}_c \mathbf{h}_p + \mathbf{b}_c\right), \mathbf{g} = \sigma\left(\mathbf{W}_g \mathbf{h}_p + \mathbf{b}_g\right) \\ &\mathbf{z}'_t = \mathbf{z}_t + \alpha(\mathbf{g} \odot \mathbf{c}), t = 1, \ldots, 10 \end{aligned} \tag{1}$$

where $\tilde{c}$ denotes the z-score-normalized physical-parameter vector; h_p is the shared 128-dimensional condition embedding; c is the bounded condition vector; g is the sigmoid gate; and z_t is the 128-dimensional spatial token at historical time step *t*. The matrices W_p, W_c, and W_g and their corresponding bias vectors are learnable, ⊙ denotes element-wise multiplication, and α controls the global conditioning strength. The residual α(g ⊙ c) is broadcast to all 10 historical tokens. Learnable temporal positional embeddings are added after the residual update, and the conditioned sequence is then passed to the Transformer encoder, as shown in Fig. 3.

GRC is motivated by the structure of Biot-type consolidation. Under fixed geometry and boundary conditions, E, permeability, and applied load primarily affect pressure-dissipation rates, structural stiffness, and displacement magnitudes rather than the topology of the evolving fields. We therefore use the parameters to produce a bounded residual correction instead of allowing them to strongly rewrite the spatiotemporal features. The hyperbolic tangent bounds the condition vector, the sigmoid gate learns feature-wise sensitivity to the parameters, and the small coefficient α limits the magnitude of the injected residual. Section 3.2 evaluates this design choice through an α-ablation study.

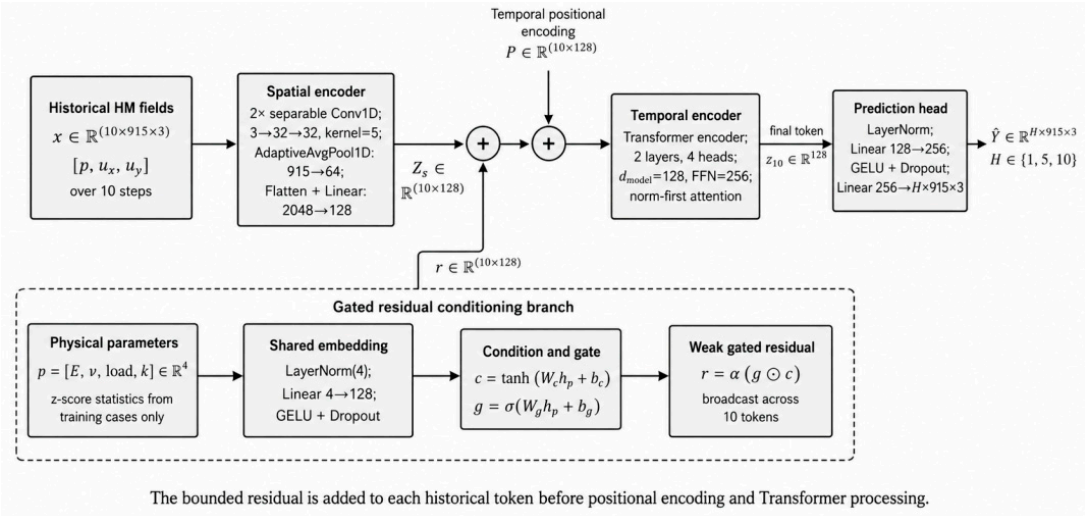

Figure 3: Architecture and data flow of LCT-HMNet-GRC.

2.4 Training Configuration and Evaluation Metrics

To minimize manual-tuning bias, all models in the shared benchmark used the same case-level split and baseline hyperparameter configuration. Normalization statistics for both field variables and conditioning parameters were computed from the training cases only. Models were trained by minimizing mean squared error on z-score-standardized fields. During validation and testing, predictions and targets were transformed back to physical scale before pooled MAE, RMSE, and R^2 were computed across all samples, forecast steps, nodes, and variables. Because pressure and displacement have different units—and pressure dominates the pooled numerical scale—we also report variable-specific errors in physical units. To determine whether the shared configuration disproportionately disadvantages certain conditioning strategies, FiLM, AdaIN, and GRC were additionally evaluated under the equal-budget tuning protocol described in Section 2.5. Results from the shared protocol and the independently tuned models are reported separately because they address different questions. Table 3 summarizes the training and evaluation settings.

Table 3: Training configuration and evaluation protocol.

Configuration	Setting
Data split	Case-level split
Input length	10 historical time steps
Output length	1/5/10 future time steps
Batch size	32
Training epochs	120

Table 3: *Cont.*

Configuration	Setting
Optimizer	AdamW (lr = 0.001, weight decay = 0.0001)
Spatial encoder	Lightweight separable 1-D CNN
Temporal encoder	Transformer encoder: 2 layers, 4 heads, d_model = 128
GRC α (shared protocol)	0.05
Evaluation metrics	Pooled inverse-standardized MAE, RMSE, R^2; variable-specific physical-unit errors
Implementation framework	PyTorch 2.12.0 reproduction environment; primary single-run seed = 42
Repeated seeds	42, 777, 1234, 2024, 3407; split_seed = 42
Equal-budget tuning	48 seed-42 configurations; selection by validation RMSE; five-seed rerun

2.5 Equal-Budget Mechanism-Specific Tuning

To account for mechanism-specific optimization sensitivity, FiLM, AdaIN, and GRC were assigned equal search budgets. For FiLM and AdaIN, the search combined learning rates of 1×10^{-4}, 3×10^{-4}, and 1×10^{-3}; weight decays of 1×10^{-5} and 1×10^{-4}; and either default or near-identity conditioner initialization, yielding 12 configurations per mechanism. For GRC, the same learning rates and weight decays were combined with $\alpha \in \{0.02, 0.05, 0.10, 0.20\}$, yielding 24 configurations. All 48 configurations were initially trained with seed = 42. For each mechanism, the configuration with the lowest validation RMSE was selected and then retrained using five seeds while retaining the same case-level split. Test-set performance was not used for model selection.

3 Results

3.1 Comparison of Conditioning Strategies

Table 4 and Fig. 4 summarize the original single-run benchmark for the primary in10_out5 task. In this run, GRC achieved the lowest MAE (0.886) and RMSE (2.887) among the explicitly conditioned LCT-HMNet variants and improved upon the corresponding unconditional run (RMSE = 3.147). Considering all models in Table 4, GRC produced the lowest MAE, whereas GRU achieved the lowest RMSE (2.666). We retain these single-run results for continuity with the initial study, but base the principal claims regarding average performance and stability on the five-seed results in Section 3.6.

The relatively large single-run errors of standard FiLM and AdaIN reflect performance under the shared default configuration and should not be interpreted as evidence of general inferiority. Because these mechanisms were not independently tuned in this comparison, the conclusions are restricted to the tested implementations and hyperparameters.

The comprehensive comparison of conditioning strategies focuses on in10_out5. For in10_out1 and in10_out10, we report only the final GRC model to characterize its behavior across forecasting horizons. The differences in parameter count across the three tasks arise from the output dimension of the prediction head, which varies with the number of forecast steps; the backbone and GRC branch are unchanged.

Table 4: Original single-run comparison of baseline models and LCT-HMNet variants using pooled inverse-standardized metrics.

Task	Model	MAE	RMSE	R^2	Parameters
in10_out5	MLP	1.858045	5.515110	0.998618	5,300,765
in10_out5	LSTM [19]	0.974505	2.874766	0.999625	3,242,525
in10_out5	GRU [20]	0.996050	2.665681	0.999677	2,874,525
in10_out5	TCN [21]	1.112638	3.136698	0.999553	2,874,013

Table 4: *Cont.*

Task	Model	MAE	RMSE	R^2	Parameters
in10_out5	LCT-HMNet v1 (unconditional)	0.999678	3.146960	0.999550	4,090,639
in10_out5	Token concatenation	1.061297	4.748903	0.998975	4,140,951
in10_out5	weak-FiLM (α = 0.10)	0.975511	3.251190	0.999520	4,124,311
in10_out5	Standard FiLM [22]	1.281746	6.618742	0.998010	4,124,311
in10_out5	AdaIN	1.206773	8.572730	0.996661	4,124,311
in10_out5	LCT-HMNet-GRC	0.885865	2.886948	0.999621	4,124,311
in10_out1	LCT-HMNet-GRC	0.746827	2.426064	0.999734	1,302,451
in10_out10	LCT-HMNet-GRC	1.371187	4.584588	0.999047	7,651,636

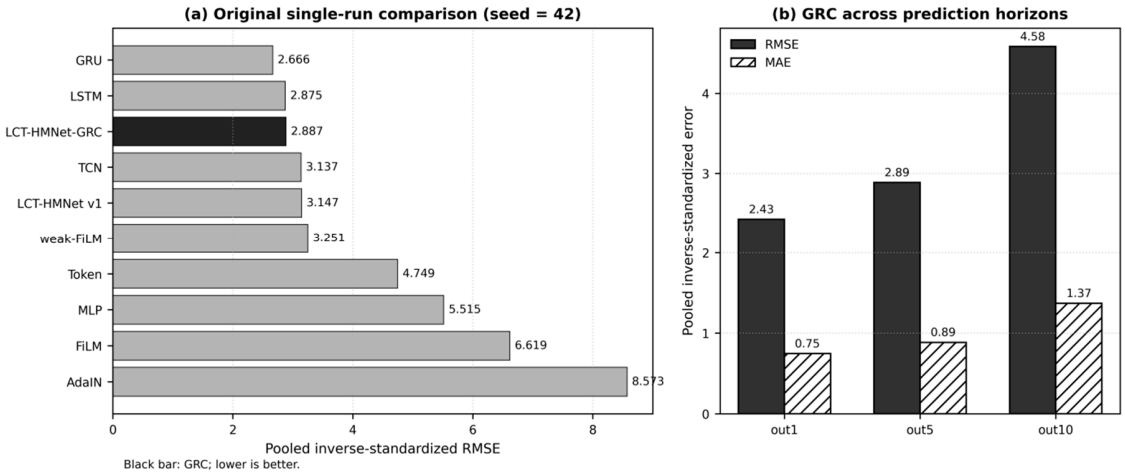

Figure 4: Original single-run comparison for in10_out5 and GRC performance across three forecasting horizons.

3.2 Effect of the GRC Conditioning Strength α

Table 5 and Fig. 5 show the effect of α on in10_out5. Across the tested values, both MAE and RMSE increased as α increased, with the lowest errors at α = 0.05. This pattern supports the hypothesis that physical parameters are most useful as a weak residual correction in the present shared-mesh benchmark. The result is specific to this GRC implementation and does not imply that α = 0.05 is optimal for other datasets or backbones.

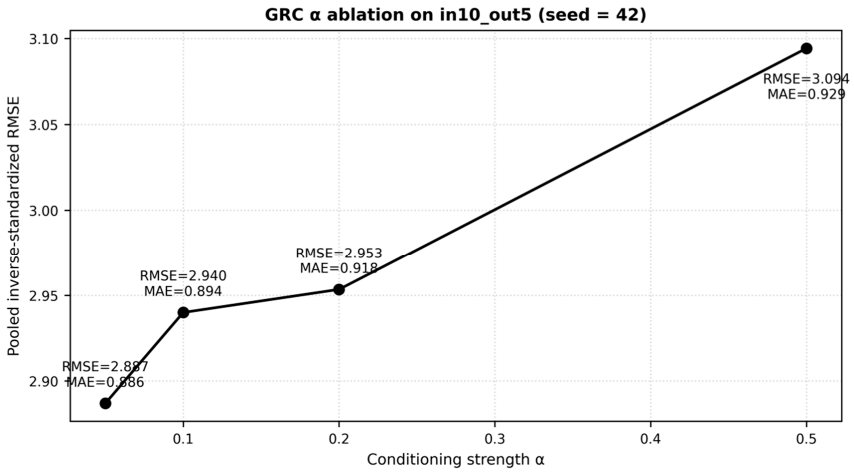

Figure 5: GRC α-ablation results for in10_out5 (seed = 42).

Table 5: GRC α-ablation results for in10_out5 (seed = 42).

Variant	α	MAE	RMSE	R^2	Parameters
LCT-HMNet-GRC	0.05	0.885865	2.886948	0.9996	4,124,311
LCT-HMNet-GRC	0.10	0.893740	2.939968	0.9996	4,124,311
LCT-HMNet-GRC	0.20	0.918162	2.953439	0.9996	4,124,311
LCT-HMNet-GRC	0.50	0.929445	3.094338	0.9996	4,124,311

3.3 Random-Condition Control

To determine whether GRC uses the physical conditioning variables, we randomly permuted the case parameters among test samples while keeping the trained model weights and historical field inputs fixed. This procedure breaks the correspondence between each field sequence and its parameter vector. Five independent permutations were evaluated, and the results are reported in Table 6.

Table 6: Random-condition control for in10_out5 (seed = 42).

Model	Condition Input	MAE	RMSE	R^2
LCT-HMNet-GRC	Correct case parameters	0.885865	2.886948	0.999621
LCT-HMNet-GRC	Randomly permuted parameters (5 runs)	0.942604 ± 0.010789	3.075606 ± 0.043809	0.999570 ± 0.000012

Relative to the correctly matched parameter inputs, random permutation increased MAE and RMSE by approximately 6.4% and 6.5%, respectively. Thus, the GRC branch does exploit case-specific parameter information. The moderate magnitude of the degradation indicates, however, that the historical field sequence already encodes most of the relevant state and that the physical parameters act primarily as an auxiliary correction signal.

3.4 Representative Field Predictions

Fig. 6 compares the reference fields, GRC predictions, and absolute errors for two unseen test cases (case_003 and case_104). For each selected in10_out5 sample, the fifth and final forecast step is shown. Displacement magnitude was calculated from the predicted components as $|u| = \sqrt{(u_x^2 + u_y^2)}$. The predictions reproduce the dominant vertical consolidation pattern, while the error maps reveal localized discrepancies in regions with steep spatial gradients. The original 915-node shared-mesh fields were retained; only the annotations, case descriptors, and color-bar arrangement were redrawn to improve readability.

3.5 Variable-Specific Errors and Inference Efficiency

Table 7 reports variable-specific errors in physical units and relative RMSE. Pressure and vertical displacement both have relative RMSE values below 1% and represent the dominant dynamic response in this consolidation benchmark. By contrast, the test-set range of horizontal displacement is only 2.337×10^{-7} m—approximately 1/42,500 of the vertical-displacement range—and its absolute error is on the order of 10^{-9} m. This disparity reflects the nearly one-dimensional nature of the consolidation problem, in which lateral displacement is negligible. Horizontal displacement is retained to preserve the complete model output, but it should not be assigned the same physical significance as pressure or vertical displacement.

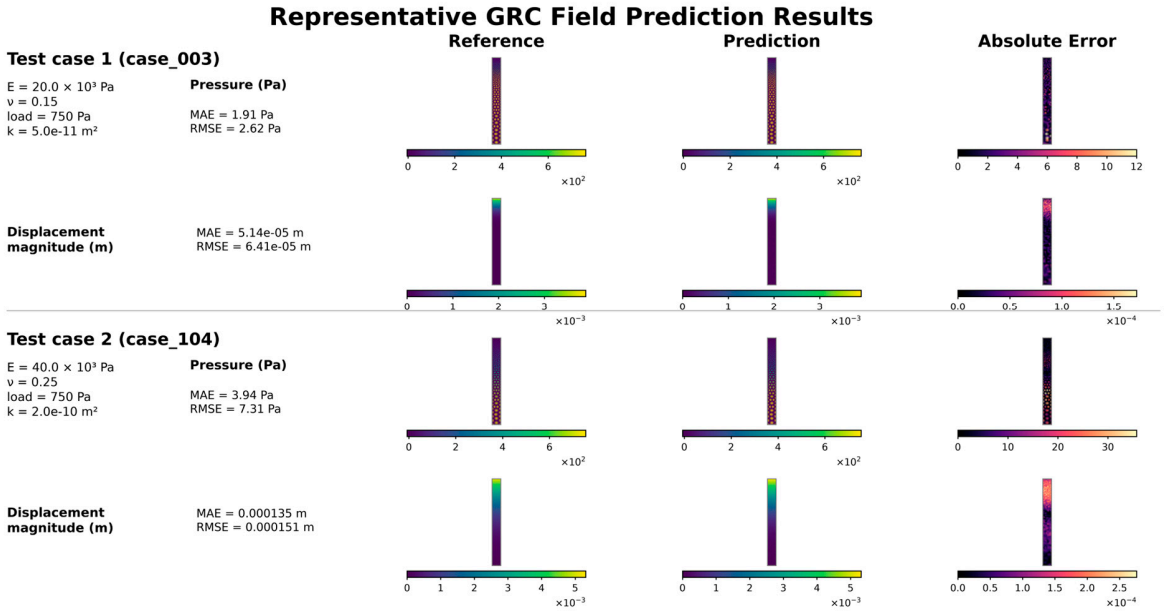

Figure 6: Reference fields, GRC predictions, and absolute errors for case_003 and case_104 at the fifth forecast step of selected in10_out5 samples. Displacement magnitude is defined as $|u| = \sqrt{(u_x^2 + u_y^2)}$.

Table 7: Variable-specific errors and relative RMSE for LCT-HMNet-GRC in physical units (in10_out5, selected original run).

Field Variable	Test-Set Dynamic Range	MAE	RMSE	Relative RMSE
Pressure	999.998 Pa	2.658 Pa	5.000 Pa	0.500%
Horizontal displacement	2.337×10^{-7} m	3.237×10^{-9} m	4.163×10^{-9} m	1.781%
Vertical displacement	9.930×10^{-3} m	5.533×10^{-5} m	7.594×10^{-5} m	0.765%

Table 8 provides reference inference times. These values should not be interpreted as an end-to-end speedup over OpenGeoSys. The surrogate requires the first 10 ground-truth historical fields as input, whereas a full OpenGeoSys simulation generates the entire sequence from the initial condition. The table therefore characterizes only the millisecond-scale response time of the surrogate when historical fields are available or can be updated online; it does not support a claim that the model replaces a complete numerical simulation.

Table 8: Reference OpenGeoSys simulation and surrogate inference times on a central processing unit (CPU) (in10_out5).

Method	Time
Full 21-step OpenGeoSys simulation (representative case)	2.595 s
LCT-HMNet-GRC inference for one in10_out5 sample (requires the first 10 historical fields)	2.806 ms
Estimated time for three sliding-window inferences covering steps 10–20	≈8.4 ms

Note: This comparison is a reference inference-time cost and is not equivalent to an end-to-end speedup. The surrogate model depends on available historical field inputs, whereas a complete OGS simulation generates the full sequence directly from the initial condition.

3.6 Five-Seed Repeated Comparisons and Zero-Condition Retraining

Table 9 summarizes the five-seed in10_out5 experiments using training seeds 42, 777, 1234, 2024, and 3407 with split_seed = 42. LSTM achieved the lowest overall mean MAE (0.883 ± 0.059) and RMSE (2.552 ± 0.227). Within the CNN-Transformer family, the unconditional backbone yielded the lowest mean MAE (0.890 ± 0.098) and RMSE (2.636 ± 0.406), followed by GRC (0.920 ± 0.140 and 2.790 ± 0.611, respectively). GRC therefore performed best among the explicitly conditioned models under the shared default protocol, but explicit conditioning did not reduce the average error below that of the unconditional backbone when 10 ground-truth historical snapshots were provided. GRC also exhibited greater run-to-run variability than the unconditional model. The repeated weak-FiLM runs used $\alpha = 0.05$, whereas the single-run entry in Table 4 used $\alpha = 0.10$. The five-seed token-concatenation results were obtained using a reconstructed training script with a slightly different parameter count from the original implementation and are included only as supplementary sensitivity evidence. Given the small number of seeds, we make no formal claims of statistical significance.

Table 9: Five-seed results for in10_out5 using pooled inverse-standardized metrics (sample standard deviation; split_seed = 42).

Model or Mechanism	Parameters	MAE (mean ± std.)	RMSE (mean ± std.)	R^2 (mean ± std.)
LSTM [19]	3,242,525	0.883 ± 0.059	2.552 ± 0.227	0.999702 ± 0.000055
LCT-HMNet v1 (unconditional)	4,090,639	0.890 ± 0.098	2.636 ± 0.406	0.999678 ± 0.000099
GRU [20]	2,874,525	1.065 ± 0.044	2.716 ± 0.122	0.999664 ± 0.000030
LCT-HMNet-GRC	4,124,311	0.920 ± 0.140	2.790 ± 0.611	0.999633 ± 0.000159
Weak FiLM ($\alpha = 0.05$)	4,124,311	0.981 ± 0.105	3.422 ± 0.547	0.999457 ± 0.000166
Standard FiLM [22]	4,124,311	1.123 ± 0.115	5.277 ± 0.831	0.998710 ± 0.000422
Token concatenation*	4,107,927	1.457 ± 0.413	6.026 ± 2.160	0.998181 ± 0.001329
AdaIN	4,124,311	1.595 ± 0.412	9.792 ± 1.221	0.995590 ± 0.001110

Note: *The five-seed results for token concatenation were obtained using a reconstructed training script with a slightly different parameter count from the original implementation; therefore, these results are included only as supplementary sensitivity evidence.

Table 10 summarizes the zero-condition retraining control. The GRC architecture and parameter count were unchanged, but every conditioning vector was set to zero during training. Relative to the original run trained with the correct parameters, MAE increased by 10.5% (0.886 to 0.979), while RMSE increased by 0.7% (2.887 to 2.906). Together with the random-condition experiment, this result confirms that GRC uses the physical parameters and that the observed effect cannot be attributed solely to the additional trainable weights. Nevertheless, the five-seed results in Table 9 show that parameter conditioning does not necessarily improve average forecasting accuracy relative to the unconditional backbone. Because the zero-condition control was conducted for only one seed, it should be regarded as supporting rather than definitive evidence.

Table 10: Zero-condition retraining control for GRC (in10_out5, seed = 42).

Condition	Parameters	MAE	RMSE	R^2	Best Epoch
Correct parameters	4,124,311	0.885865	2.886948	0.999621	92
Zero condition	4,124,311	0.979130	2.906035	0.999616	119

3.7 Equal-Budget Mechanism-Specific Tuning Results

The equal-budget search completed all 48 planned seed-42 configurations. Table 11 reports the configurations selected by validation RMSE. Near-identity initialization was selected for both FiLM and AdaIN, whereas GRC favored $\alpha = 0.02$. These selections demonstrate that the conditioning strategies have distinct optimization requirements and that the shared default configuration should not be treated as individually optimal.

Table 11: Equal-budget search spaces and validation-selected configurations.

Mechanism	Search Budget	Selected lr	Selected Weight Decay	Initialization/α	Best Validation RMSE (seed = 42)
FiLM	3 lr × 2 wd × 2 init = 12	0.001	0.0001	Near identity	3.04713
AdaIN	3 lr × 2 wd × 2 init = 12	0.0003	0.00001	Near identity	4.55989
GRC	3 lr × 2 wd × 4 α = 24	0.001	0.00001	$\alpha = 0.02$	2.48840

Table 12 and Fig. 7 present five-seed results after the best configuration for each mechanism was selected. Independent tuning reduced the mean RMSE of FiLM from 5.277 to 4.729 (10.4%) and that of AdaIN from 9.792 to 6.824 (30.3%), confirming substantial sensitivity to initialization and optimization. The selected GRC configuration achieved an RMSE of 2.866 ± 0.434, close to but slightly higher than its shared-protocol mean of 2.790 ± 0.611. Nevertheless, GRC retained the lowest five-seed mean RMSE among the three independently tuned conditioning mechanisms. The resulting conclusion is mechanism-specific rather than universal: within the present backbone and equal tuning budget, bounded residual conditioning was less error-prone than the tested FiLM and AdaIN implementations.

Table 12: Five-seed results after equal-budget mechanism-specific tuning (sample standard deviation; split_seed = 42).

Mechanism	Mean MAE	Mean RMSE ± SD	RMSE 95% CI Half-Width	Mean R^2	RMSE Change from Shared Protocol
GRC	0.946	2.866 ± 0.434	0.539	0.999620	+2.7%
FiLM	1.178	4.729 ± 0.881	1.094	0.998956	−10.4%
AdaIN	1.513	6.824 ± 0.846	1.051	0.997858	−30.3%

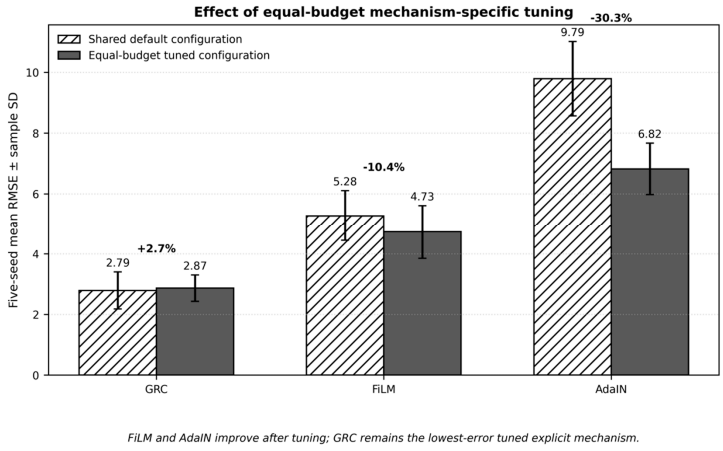

Figure 7: Five-seed mean RMSE before and after equal-budget mechanism-specific tuning. Error bars indicate the sample standard deviation.

4 Discussion

The repeated experiments refine the interpretation of the original single-run ranking. LSTM achieved the lowest overall five-seed mean MAE and RMSE, and the unconditional CNN-Transformer also outperformed GRC in mean error. GRC nevertheless produced the lowest mean error among the explicitly conditioned variants under the shared default protocol. The evidence therefore does not support a general claim that explicit physical-parameter conditioning is beneficial. Rather, when 10 ground-truth historical fields are available, the unconditional backbone already captures most of the state information needed for forecasting, while bounded residual conditioning perturbs the learned representation less than token concatenation, weak FiLM, FiLM, or AdaIN.

The random-condition and zero-condition controls confirm that GRC uses E, ν, load, and permeability. Their contribution is nevertheless secondary: mismatched or zeroed conditioning inputs degraded the selected GRC run, whereas the unconditional model remained more accurate on average across five seeds. This combination suggests that the conditioning branch learns case-specific corrections whose benefit depends on initialization and optimization. The strong performance of LSTM is also consistent with the ordered, largely monotonic consolidation sequence and with the online forecasting setting, in which 10 ground-truth snapshots already describe the evolving system state.

The α-ablation and equal-budget tuning results provide complementary evidence. The original run favored $\alpha = 0.05$ over larger values, whereas the independent validation search selected $\alpha = 0.02$; both findings support weak residual injection. Mechanism-specific tuning substantially improved FiLM and AdaIN, confirming that strong modulation is sensitive to optimization, but GRC retained the lowest mean RMSE among the three tuned conditioning strategies. This behavior is physically plausible for the present benchmark: all cases share the same geometry, mesh, boundary conditions, and broad patterns of pressure dissipation and displacement accumulation, while the varied parameters mainly alter rates and magnitudes rather than field topology. This interpretation remains specific to the benchmark and should not be generalized without further evidence.

Several limitations should be considered. First, the 108 cases use a single rectangular, homogeneous geometry with fixed node ordering; consequently, the one-dimensional CNN representation is not expected to transfer directly to variable meshes or complex geometries. Second, although FiLM, AdaIN, and GRC received equal-budget mechanism-specific tuning, token concatenation and weak FiLM were retained mainly as shared-protocol comparison variants. The repeated token-concatenation experiments used a reconstructed implementation, and the repeated weak-FiLM configuration used a different α value from the original single-run experiment. Third, the benchmark does not include rigorously tuned neural-operator, graph-based, or coordinate-aware baselines; therefore, no architecture-wide ranking is claimed. Fourth, parameter-level holdout, cold-start prediction, missing-history scenarios, and learning-curve analyses remain to be investigated. Fifth, in10_out10 contains only 32 test windows, so its long-horizon result is indicative rather than statistically exhaustive. Finally, the exact OpenGeoSys executable version used for the original parameter sweep was not recorded, although the complete project, mesh, boundary-condition, and solver files are publicly available.

5 Conclusions

Using a 108-case OpenGeoSys HM dataset and a common CNN-Transformer backbone, this study benchmarked five physical-parameter conditioning strategies: token concatenation, FiLM, weak FiLM, AdaIN, and gated residual conditioning (GRC).

Across five training seeds, LSTM achieved the lowest overall mean MAE and RMSE. Within the CNN-Transformer family, the unconditional backbone also slightly outperformed GRC in average error, whereas GRC performed best among the explicitly conditioned models. Equal-budget tuning substantially improved FiLM and AdaIN, but tuned GRC retained the lowest five-seed mean RMSE among GRC, FiLM, and AdaIN. The random-condition and zero-condition controls confirmed that GRC uses the physical parameters; however, the historical field sequence remained the primary source of predictive information, and explicit conditioning did not improve average accuracy over the unconditional backbone in this setting. For the selected GRC run, the relative RMSE was below 1% for both pressure and vertical displacement.

The principal contribution is therefore a reproducible benchmark and a clearly defined boundary result: in history-based forecasting on this fixed-node, shared-mesh OpenGeoSys consolidation dataset, physical parameters act mainly as an auxiliary correction signal, and weak gated residual injection provides the lowest error among the independently tuned explicit conditioning strategies. These findings neither establish a universal ranking of surrogate architectures nor demonstrate a complete replacement for OpenGeoSys. Broader conclusions will require rigorously tuned neural-operator and graph-based baselines, parameter-level extrapolation, cold-start forecasting, and variation in geometry and mesh topology.

Acknowledgement: The authors thank the OpenGeoSys community for developing and maintaining the open-source numerical simulation framework.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: Zongzheng Jiao: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Writing—original draft, and Writing—review and editing. Shuaikang Yang: Data curation, Visualization, and Software. Junlong Yin: Data curation, Investigation, and Validation. Shaohui Wang: Methodology, Supervision, and Writing—review and editing. Minpo Jung: Supervision, Project administration, and Writing—review and editing. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The processed dataset, case-level training/validation/test split, source code, training scripts, configuration files, 108 OpenGeoSys case directories, run-level five-seed results, equal-budget tuning summary, and reproducibility notes are publicly available in the GitHub release v1.0.0: https://github.com/zongzhengjiao-ux/FDMP_HM_Surrogate/releases/tag/v1.0.0.

Ethics Approval: Not applicable.

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

References

1. Biot MA. General theory of three-dimensional consolidation. J Appl Phys. 1941;12(2):155–64. [CrossRef].
2. Terzaghi K. Theoretical soil mechanics. New York, NY, USA: John Wiley & Sons; 1943. [CrossRef].
3. Kolditz O, Bauer S, Bilke L, Böttcher N, Delfs JO, Fischer T, et al. OpenGeoSys: an open-source initiative for numerical simulation of thermo-hydro-mechanical/chemical (THM/C) processes in porous media. Environ Earth Sci. 2012;67(2):589–99. [CrossRef].
4. Brunton SL, Noack BR, Koumoutsakos P. Machine learning for fluid mechanics. Annu Rev Fluid Mech. 2020;52:477–508. [CrossRef].
5. Raissi M, Perdikaris P, Karniadakis GE. Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J Comput Phys. 2019;378:686–707. [CrossRef].
6. Karniadakis GE, Kevrekidis IG, Lu L, Perdikaris P, Wang S, Yang L. Physics-informed machine learning. Nat Rev Phys. 2021;3(6):422–40. [CrossRef].

7. Lu L, Jin P, Pang G, Zhang Z, Karniadakis GE. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. Nat Mach Intell. 2021;3(3):218–29. [CrossRef].
8. Li Z, Kovachki N, Azizzadenesheli K, Liu B, Bhattacharya K, Stuart A, et al. Fourier neural operator for parametric partial differential equations. arXiv:2010.08895. 2020.
9. Willard J, Jia X, Xu S, Steinbach M, Kumar V. Integrating scientific knowledge with machine learning for engineering and environmental systems. ACM Comput Surv. 2022;55(4):1–37. [CrossRef].
10. Wen G, Li Z, Long Q, Azizzadenesheli K, Anandkumar A, Benson SM. Real-time high-resolution CO_2 geological storage prediction using nested Fourier neural operators. Energy Environ Sci. 2023;16(4):1732–41. [CrossRef].
11. Jiang Z, Zhu M, Lu L. Fourier-MIONet: Fourier-enhanced multiple-input neural operators for multiphase modeling of geological carbon sequestration. Reliab Eng Syst Saf. 2024;251:110392. [CrossRef].
12. Diab W, Al Kobaisi M. U-DeepONet: U-Net enhanced deep operator network for geologic carbon sequestration. Sci Rep. 2024;14:21298. [CrossRef].
13. Badawi D, Gildin E. Neural operator-based proxy for reservoir simulations considering varying well settings, locations, and permeability fields. Comput Geosci. 2025;196:105826. [CrossRef].
14. Jiang J, Chen J, Yang Z. A multigrid graph U-Net framework for simulating multiphase flow in heterogeneous porous media. arXiv:2412.12757. 2024.
15. Wen G, Li Z, Azizzadenesheli K, Anandkumar A, Benson SM. U-FNO—An enhanced Fourier neural operator-based deep-learning model for multiphase flow. Adv Water Resour. 2022;163:104180. [CrossRef].
16. Abdellatif A, Menke HP, Doster F, Singh K, Elsheikh AH. Feature-modulated UFNO for improved prediction of multiphase flow in porous media. arXiv:2511.20543. 2025.
17. Ng CWW, Zhou Q, Zhang Q. A novel surrogate model for hydro-mechanical coupling in unsaturated soil with incomplete physical constraints. Comput Geotech. 2025;180:107091. [CrossRef].
18. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is all you need. In: Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017); 2017 Dec 4–9; Long Beach, CA, USA.
19. Hochreiter S, Schmidhuber J. Long short-term memory. Neural Comput. 1997;9(8):1735–80. [CrossRef].
20. Cho K, van Merriënboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, et al. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP); 2014 Oct 25–29; Doha, Qatar. p. 1724–34. [CrossRef].
21. Bai S, Kolter JZ, Koltun V. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. arXiv:1803.01271. 2018.
22. Perez E, Strub F, De Vries H, Dumoulin V, Courville A. FiLM: visual reasoning with a general conditioning layer. Proc AAAI Conf Artif Intell. 2018;32(1):1–10. [CrossRef].