

Article

Accelerating Numerical Simulations of CO₂ Geological Storage in Deep Saline Aquifers via Machine-Learning-Driven Grid Block Classification

Eirini Maria Kanakaki ¹, Ismail Ismail ¹ and Vassilis Gaganis ^{1,2,*}

¹ School of Mining and Metallurgical Engineering, National Technical University of Athens, 15780 Athens, Greece; ekanakaki@metal.ntua.gr (E.M.K.); iismail@mail.ntua.gr (I.I.)

² Institute of Geoenery, Foundation for Research and Technology-Hellas, 73100 Chania, Greece

* Correspondence: vgaganis@metal.ntua.gr

Abstract: The accurate prediction of pressure and saturation distribution during the simulation of CO₂ injection into saline aquifers is essential for the successful implementation of carbon sequestration projects. Traditional numerical simulations, while reliable, are computationally expensive. Machine learning (ML) has emerged as a promising tool to accelerate these simulations; however, challenges remain in effectively capturing complex reservoir dynamics, particularly in regions experiencing rapid changes in pressure and saturation. This article addresses the challenges by introducing a fully automated, data-driven ML classifier that distinguishes between regions of fast and slow variation within the reservoir. Firstly, we demonstrate the variability in pressure across different reservoir grid blocks using a simple brine injection and production scenario, highlighting the limitations of conventional acceleration approaches. Subsequently, the proposed methodology leverages ML proxies to rapidly and accurately predict the behavior of slow-varying regions in CO₂ injection simulations, while traditional iterative methods are reserved for fast-varying areas. The results show that this hybrid approach significantly reduces the computational load without compromising on accuracy. This provides a more efficient and scalable solution for modeling CO₂ storage in saline aquifers.



Citation: Kanakaki, E.M.; Ismail, I.; Gaganis, V. Accelerating Numerical Simulations of CO₂ Geological Storage in Deep Saline Aquifers via Machine-Learning-Driven Grid Block Classification. *Processes* **2024**, *12*, 2447. <https://doi.org/10.3390/pr12112447>

Academic Editor: Carlos Sierra Fernández

Received: 17 October 2024

Revised: 1 November 2024

Accepted: 5 November 2024

Published: 5 November 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: carbon capture and storage (CCS); CO₂ injection; saline aquifers; reservoir simulation; proxy models; machine learning (ML); grid blocks; classification; acceleration

1. Introduction

In recent years, the urgent need to mitigate anthropogenic carbon dioxide (CO₂) emissions into the atmosphere has reached unprecedented levels [1]. Since 1990, CO₂ emissions have surged by nearly 60% [2], primarily due to modern society's over-reliance on fossil fuels for energy production [3], coupled with the adverse effects of deforestation and unsustainable agricultural practices [3,4]. To avert the detrimental impacts of global warming and climate change, such as rising sea levels, extreme weather events, water scarcity, and biodiversity loss [5], it is imperative to redirect the rising trajectory of CO₂ emissions towards the targets outlined in the 2015 Paris Agreement [6–8], whose primary goal is to limit global temperature increase to below 1.5 °C above pre-industrial levels.

Carbon capture and storage (CCS) technology holds significant potential for tackling the pressing challenge of reducing atmospheric CO₂ concentrations [9,10]. With advancements in CCS, this technology could contribute to a 15% reduction in overall emissions by 2050 [11], representing a significant step toward mitigating climate change. One specific approach within CCS entails the permanent and long-term geological storage of captured CO₂ into suitable deep saline aquifers [9,10]. These underground porous formations offer high reservoir porosity and permeability [12], along with a large storage capacity ranging from 400 to 10,000 gigatons of CO₂ [12–14]. Consequently, they are regarded as one of

the most promising storage sites for large-scale CO₂ sequestration when compared to alternative types of reservoirs such as depleted oil and gas reservoirs or coal seams [12].

The process of CO₂ storage into brine-bearing geological formations typically involves injecting CO₂ in a supercritical state at depths exceeding 800 m [9,15]. Supercritical CO₂ has a significantly higher density than gaseous CO₂, yet much lower density and viscosity than the resident brine it displaces [15,16]. Within saline aquifers, four distinct CO₂ trapping mechanisms are encountered: trapping in structural and stratigraphic traps (structural trapping), trapping in the pore space at irreducible gas saturation (residual gas trapping), partial dissolution of CO₂ in the aqueous phase (solubility trapping), and mineral trapping [9,12]. Over time, these mechanisms exhibit a gradual increase in effectiveness, as depicted in Figure 1.

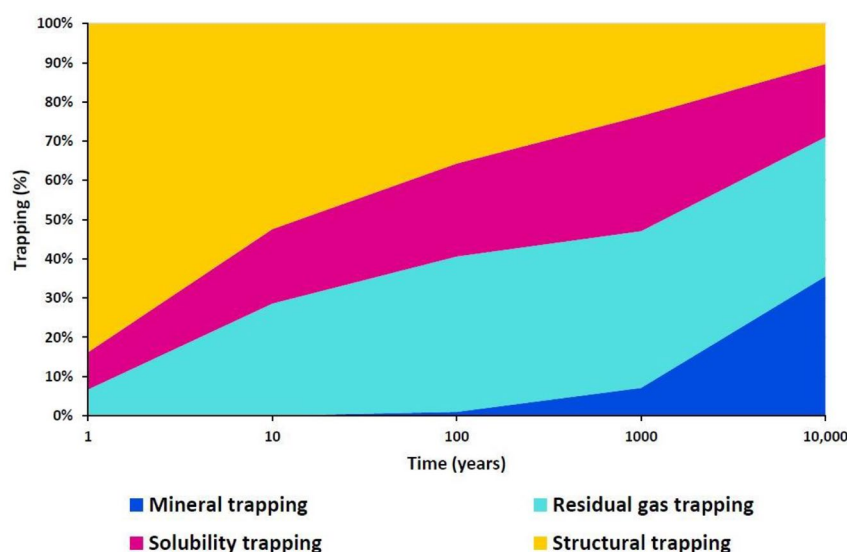


Figure 1. CO₂ trapping mechanisms of saline aquifers over time [9].

Determining the CO₂ injection policy that maximizes stored quantities while adhering to geomechanical and market constraints requires careful consideration of several key factors. Among these, pressure distribution and CO₂ saturation stand out as the two most critical elements of the optimization process. Monitoring the pressure distribution is essential for safe and efficient storage, as excessive pressures can cause caprock fractures, the reactivation of faults, and the opening of natural or artificial conduits and channels within the reservoir [17–22]. Such events elevate the risk of induced seismicity and pose serious hazards [23–25].

The occurrence of seismic activity, even at minor levels, can attract public scrutiny of CCS operations due to its perceived risks. In severe cases, induced earthquakes could cause casualties, damage infrastructure, and compromise caprock integrity, ultimately undermining the long-term goals of secure CO₂ containment. Additionally, induced seismicity raises the risk of unintended migration of CO₂ or formation of brine beyond the intended storage zone, potentially contaminating shallow freshwater aquifers [17–22,26] critical for drinking water and agriculture in nearby regions.

Given these risks, effective CO₂ injection policies must integrate mechanisms to monitor and control pressure within safe limits, minimizing potential hazards and protecting essential water resources. Alongside pressure management, monitoring CO₂ saturation is equally crucial for understanding the distribution and behavior of injected CO₂ within the reservoir. By analyzing saturation levels, operators gain insights into the containment stability and mobility of CO₂ over time. Accurate mapping of saturation levels enables engineers to predict CO₂ movement and make informed adjustments to injection strategies, ensuring both secure containment and optimal storage efficiency.

Numerical reservoir simulation [27] is a mathematical tool that can be used to predict the spatial and temporal distribution of the pressure and CO₂ plume. By numerically solving the differential and algebraic equations derived from the integration of the mass, momentum, and energy conservation principles, together with thermodynamic equilibrium, reservoir simulators can accurately describe multiphase fluid flow within deep saline aquifers. However, these computations are CPU-time intensive, especially in the case of compositional reservoir simulations, which rely on intricate Equations of State (EoS) to determine the phase distribution and phase properties in each cell [28–34]. On top of that, optimizing the injection policy requires many dozens, if not hundreds, of simulations to be executed in order to study various schedules over long time periods, thus yielding an enormous CPU workload. Hence, utilizing methods that achieve comparable accuracy to reservoir simulators while requiring less computational effort than full-order simulations is highly desirable.

Proxy models, also referred to as surrogate models or metamodels [35], serve as an efficient alternative to the time-intensive high-fidelity model (reservoir simulator), effectively bridging the gap between speed and accuracy. With their unique capability to rapidly generate predictions that closely mimic real reservoir performance within acceptable error bounds, these models prove invaluable to reservoir engineers. Simply speaking, proxy models can be thought of as functions of the form $y = f(x)$, where x stands for the model input (such as well location, initial pressure and saturation, and well operation schedule) and y is the reservoir response over space and time. The advantage is that, unlike traditional differential-equation-based methods that require iterative approaches, a proxy model, once built, can be directly evaluated for any possible reservoir configuration. This allows it to rapidly provide its response, y . According to Mohaghegh [36], proxy models fall into two main categories: traditional and smart (Figure 2).

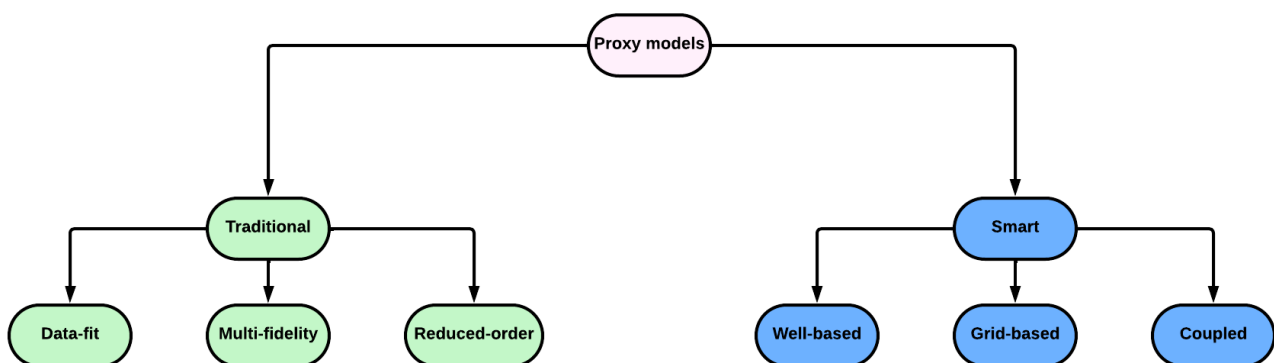


Figure 2. Proxy models' categorization according to Mohaghegh.

The traditional proxy models can be further subdivided into data-fit, multi-fidelity, and reduced-order ones. Data-fit models are non-physics-based, meaning that they do not explicitly account for the underlying physics principles. Instead, their predictions are based on statistical methods such as the interpolation or regression of the generated results from a few runs of the reservoir simulator. On the other hand, multi-fidelity models are physics-based, lower-fidelity models that are attained through coarser discretization [37] or the simplification of physics assumptions [38]. Finally, reduced-order models are also physics-based models that lower the dimensionality of the high-fidelity model by neglecting irrelevant parameters while holding the characteristics and physics over a defined space [39].

As far as smart proxy models are concerned, these models are trained using machine learning (ML) and pattern recognition techniques to generate high-fidelity models. The development of smart proxy models adopts either a well-based or a grid-based approach. Well-based smart proxy models are employed when the objective is to generate predictions for parameters associated with well locations, such as oil, gas, and water production

over time. Conversely, grid-based models are utilized when pressure and saturation predictions at a grid level are desired. Moreover, coupled smart proxy models offer a comprehensive solution by predicting results at both well-based and grid-based levels [40]. Unlike traditional proxy models, they predict the pressure and phase saturation distribution at each discretization block of the model and for each time step without sacrificing the physics, dimensionality, or temporal/spatial resolution of the original reservoir system [41].

Developing proxy models for dynamic systems of any arbitrary complexity is no easy task. The primary challenge lies in the significant variation in response variables, such as pressure and saturation, across space and time. This variability makes it difficult to accurately capture the system's behavior using a proxy model. For instance, when a well is initially shut in or activated, the pressure change around the wellbore is rapid and substantial. Additionally, generating a suitable dataset to train the model while ensuring sufficient generalization capabilities is another hurdle. However, for systems that exhibit relatively slow variations over time, as is the case with flow in porous media, specific properties of the response variable variation can be exploited to enhance the generation process, simplify the model, and improve its automation and performance.

In this paper, we present a novel computational methodology that employs ML modeling to accelerate numerical simulations of CO₂ injection into deep underground saline aquifers while maintaining accuracy. Prior studies [42–45] have investigated the application of ML to accelerate such simulations; however, these efforts typically apply ML models uniformly across all grid blocks within the reservoir model. This approach can compromise accuracy, particularly in regions where pressure and CO₂ saturation exhibit rapid changes.

Our approach addresses this limitation by introducing a unique categorization of grid cells into fast-varying and slow-varying ones—a strategy not previously explored in the literature. This classification enables a targeted computational focus: in slow-varying regions, where changes in pressure and saturation are gradual, ML-based predictions are employed to speed up simulations. Conversely, traditional iterative methods are reserved for fast-varying regions to ensure high accuracy in areas experiencing dynamic changes.

Specifically, the proposed method employs ML models to predict the future state (pressure and saturation) of each grid block in a direct way, using information held by the neighboring cells at the previous timesteps. Strictly speaking, the state of any cell in a future timestep depends on that of all grid cells in the previous timestep. However, the spatial dependence becomes weaker as the distance between the cells increases. In other words, the grid blocks surrounding a focal cell directly are expected to contribute mostly to its future state whereas grid blocks lying far away are almost uncorrelated. Therefore, the vast majority of the cells in an aquifer reservoir model are expected to depend on its neighboring grid blocks only. This is not true for cells that vary significantly over time as is the case with those close to a well (an injector or a producer), hence the traditional iterative solution method also needs to be employed for such cases.

The methodology developed to address the above-mentioned problem employs a two-stage approach. In the first stage, a fully automated ML and IQR-based classifier is used to classify the grid blocks in the simulator model into fast-varying and slow-varying ones (Figure 3). By automating this process, we significantly reduce operator dependency and ensure consistent classification across different simulation scenarios. Fast-varying grid blocks represent areas where flow in the porous medium demonstrates high spatial and temporal variance (i.e., typically close to the injectors where pressure disturbance is caused), while slow-varying grid blocks represent areas where the CO₂ and formation water flow evolve very slowly; thus, their state is mostly related to the state of their neighboring cells solely. This latter category includes cells lying far from the wells, as well as most grid blocks during the post-injection phase, often lasting several decades, during which CO₂ plume migration is solely capillarity and gravity driven. In the second stage, ML methods are employed to predict the state of the slow-varying grid blocks based on the previous states of the neighboring grid blocks by taking advantage of the slowly varying property.

Consequently, the problem, which typically involves millions of discretized equations, will be drastically downscaled as the state of the fast-varying cells will only need to be predicted by means of conventional iterative methods, thus offering a huge acceleration factor.

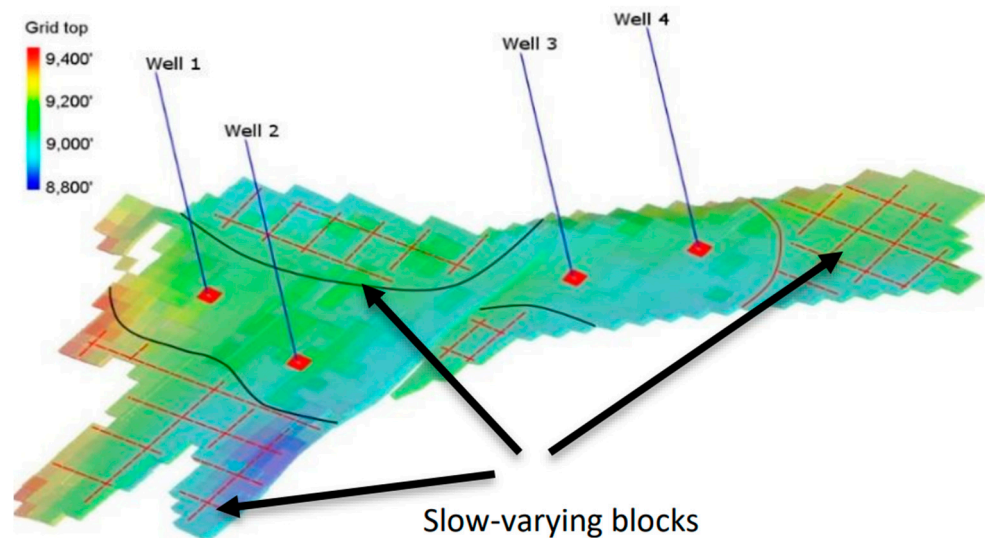


Figure 3. Distribution of fast-varying grid blocks located near wells and slow-varying grid blocks lying far from wells.

Clearly, rather than splitting the proxy modeling problem into two parts, one might consider building a huge, highly accurate proxy model. This model could incorporate any required input to accurately predict the future reservoir state. However, this approach is impractical for two main reasons. First, it is extremely challenging to ensure that the training dataset used to develop the proxy model is adequate to meet such high standards of generalization and predictive accuracy. Second, since speed is a major concern, a complex model would increase the CPU time required for generating predictions, even if the model had an explicit form.

This paper is laid out as follows: Section 2 formally establishes the need for classifying grid blocks into fast-varying and slow-varying ones when using a proxy model that predicts the pressure or phase saturation distributions based on the neighboring cells' state in the previous timesteps. Section 3 describes the proposed methodology, while Section 4 discusses the results obtained. Conclusions are presented in Section 5.

2. Proof of Concept

Accurately identifying the slow-varying and fast-varying areas within a reservoir model is essential when using machine learning (ML) proxy models that describe the state of a given grid block by its state and those of the neighboring blocks at the previous time steps. This section highlights the importance of grid block classification through a straightforward brine injection and production scenario in a simple, homogeneous saline aquifer, where pressure is the sole variable exhibiting spatial and temporal variation. Saturation is not considered at this stage as no CO₂ is injected to keep the demonstrative example simple. To maintain simplicity, pressure predictions from the ML proxy model were limited to grid blocks with six adjacent faces (interior cells), as illustrated in Figure 4, thus ignoring neighboring cells sharing a side or a point with the center block, or even cells lying even farther away than them. Boundary grid blocks with three, four, or five neighbor face tier cells, such as those lying on the top and bottom layer of the reservoir, were excluded from the analysis.

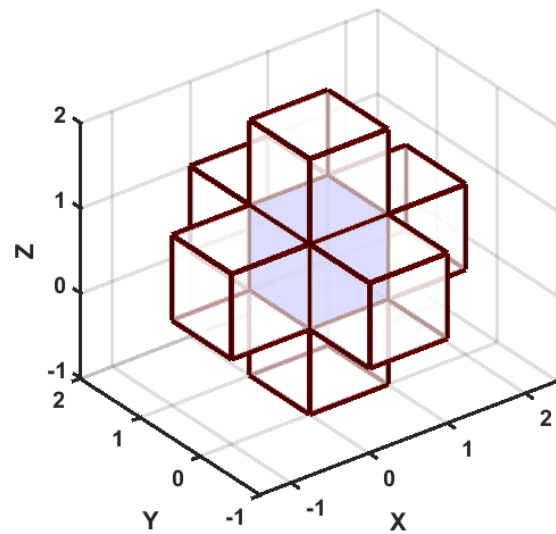


Figure 4. Focal cell that is in face contact with six adjacent cells in a three-dimensional Cartesian grid.

The aquifer system is modeled by a simple three-dimensional $25 \times 25 \times 4$ Cartesian grid with $100 \text{ m} \times 100 \text{ m} \times 75 \text{ m}$ grid blocks, each representing a volume of 0.75 million m^3 . A constant porosity of 0.2 and permeability of 50 mD are assigned to all grid blocks. To simulate brine thermodynamics, a black-oil simulator is used [46]. One injector (I1) and three producers (P1, P2, P3) are positioned at the top left, bottom left, top right, and bottom right corners, respectively, as shown in Figure 5, and they vertically penetrate all reservoir layers.

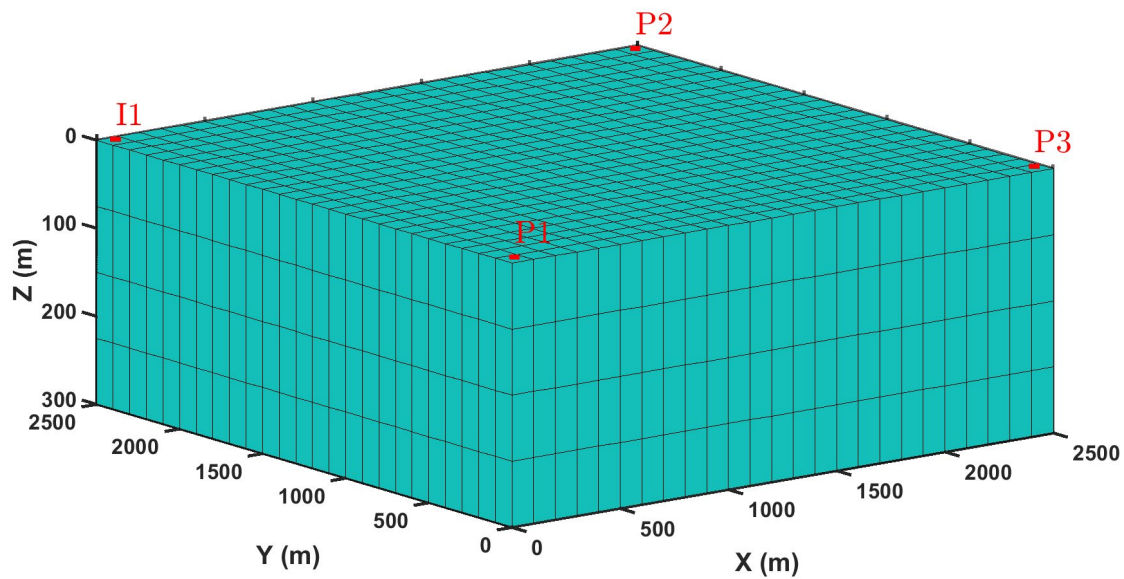


Figure 5. Visualization of the saline aquifer structure and well locations in this study investigating brine injection and production.

To train the proxy model, a spatial–temporal dataset was generated using the MATLAB Reservoir Simulation Toolbox (MRST) [47,48]. Injection and production schemes were designed with bottomhole pressure (BHP) as a well-level operational constraint, imposing maximum permissible BHP values of 5800 psi (400 bar) for injector I1 and 3626 psi (250 bar) and 4351 psi (300 bar) for producers P1, P2, and P3, respectively. The simulation spanned a 9-year period, divided into 110 one-month time steps. Initially, only producer P1 operated for 14 months, followed by the activation of injector I1 and producer P2 from month 14 to 50. Producer P3 commenced operation from month 50 until the end of the simulation.

Table 1 and Figure 6 illustrate the operational phases of the injection and production wells according to the operating scheme, and the corresponding average reservoir pressure over time. Table 1 provides specific details of well activation periods and BHP constraints, while Figure 6 graphically represents the three distinct phases characterized by pressure valleys and peaks.

Table 1. Operational timeframes and BHP constraints for injector and production wells.

Wells	Start Month	End Month	BHP Constraint
P1	0	14	3626 psi
P2	14	50	4351 psi
P3	50	110	4351 psi
I1	14	50	5800 psi

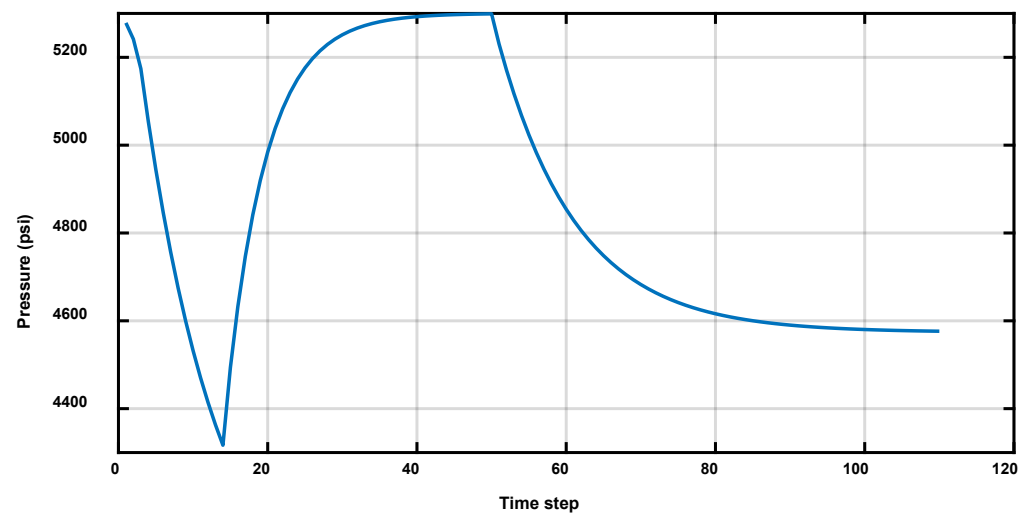


Figure 6. Average reservoir pressure over time.

It is important to note that the difficulty in our approach lies in the proper selection of the input–output information imposed on the proxy model, rather than its exact form, type, or operation principle. To this end, a conventional three-layer feedforward artificial neural network (ANN), illustrated in Figure 7, serves as the ML proxy model in this study. Given the need for rapid proxy model predictions to accelerate the overall simulation, a simple ANN architecture was adopted that rapidly produces predictions.

In detail, the input layer takes in seven features, which include the pressure values from the target cell and its six neighboring cells at the previous time step, allowing for a comprehensive capture of the spatiotemporal dynamics within the aquifer. This feature space is then passed to the intermediate hidden layer, which consists of 10 neurons. Each neuron applies a linear transformation followed by a non-linear activation function (sigmoid) to map the input data into a higher-dimensional space. Mathematically, this is represented as follows:

$$h = s(\mathbf{W}_1 \mathbf{x}_i + \mathbf{b}_1) \quad (1)$$

where $\mathbf{x}_i \in \mathbb{R}^{7 \times 1}$ represents the input pressure data for interior cell i , $\mathbf{W}_1 \in \mathbb{R}^{10 \times 7}$ represents the connection weights between the input and hidden layer, and $\mathbf{b}_1 \in \mathbb{R}^{10 \times 1}$ is the bias term for the hidden layer. The activation function $s(\cdot)$ is the non-linear sigmoid function defined as follows:

$$s(z) = \frac{1}{1 + \exp(-z)} \quad (2)$$

After processing through the hidden layer, the output is passed through a linear transformation at the output layer. The output layer contains a single neuron that predicts

the pressure of target cell i for the subsequent time step using a linear activation function. The final output $y_i \in \mathbb{R}^{1 \times 1}$ is given by the following:

$$y_i = \mathbf{W}_2 \cdot s(\mathbf{W}_1 \mathbf{x}_i + \mathbf{b}_1) + \mathbf{b}_2 \quad (3)$$

where $\mathbf{W}_2 \in \mathbb{R}^{1 \times 10}$ is the weight matrix connecting the hidden layer to the output neuron, and $\mathbf{b}_2 \in \mathbb{R}^{1 \times 1}$ is the bias for the output layer. Unlike the hidden layer, which uses the sigmoid function, the output layer applies a linear activation function without further transformation.

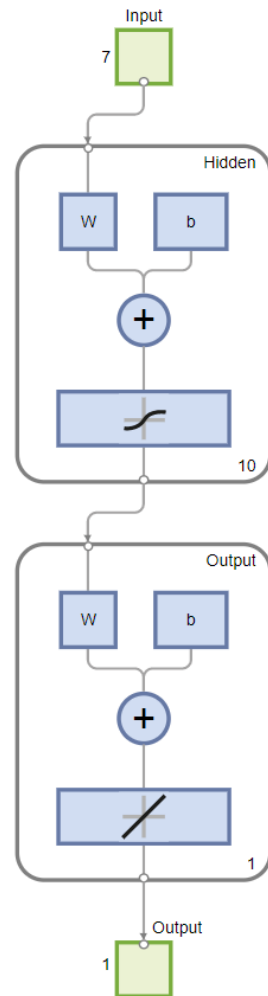


Figure 7. Feedforward ANN structure.

Once the simulation was run, the $\{(x_i, y_i)\}$ dataset was generated by collecting the pressure values predicted at each cell and timestep, and by combining them according to the cells' connectivity. Note that the need to accelerate the overall simulation implies that a simple ANN architecture needs to be chosen to ensure rapid proxy model generation. More complex models, while potentially offering improved accuracy, would incur higher computational costs, counteracting the desired efficiency gains. While more complex models might potentially enhance accuracy, the trade-off in terms of computational cost is deemed undesired given the priority of computational efficiency.

The grid-based pressure histograms derived from the spatial-temporal dataset, shown in Figure 8, demonstrate the distribution of all training input and training output pressure values. Clearly, a highly non-uniform distribution of bars is identified, directly reflecting the distinct influence of injection and production wells on each interior grid block. Factors such as cells' proximity to wells and operational history contribute to this grid-level

pressure variability. The complex dynamics of the reservoir system are reflected in this pressure distribution. The two histograms are largely similar; however, slight differences arise because the Inputs histogram represents pressure values across the entire reservoir, capturing the full spatial variation. In contrast, the second histogram, which focuses on interior cells only, naturally shows a slightly narrower range of pressure variability due to its more limited spatial scope.

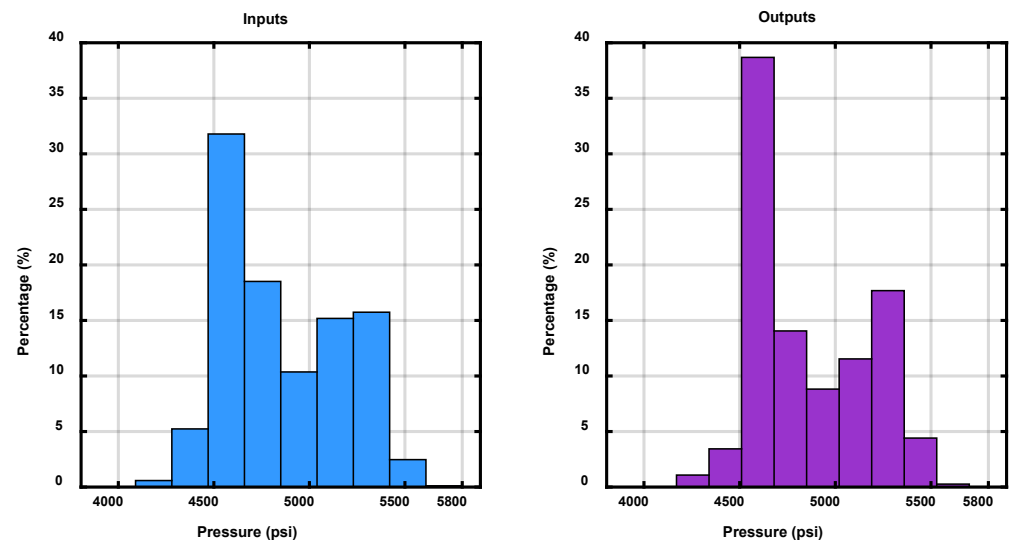


Figure 8. Grid-based pressure histogram of the spatial-temporal dataset.

To evaluate the trained ML proxy model's ability to predict the interior grid blocks' state, three key metrics were used: maximum absolute error, mean absolute error, and standard deviation of the predicted pressure error over all grid blocks, per specific timestep. The maximum absolute error identifies the largest individual prediction error, highlighting specific timeframes and reservoir areas where the model struggles most. The mean absolute error quantifies overall prediction accuracy by averaging absolute errors. Lastly, standard deviation measures error dispersion around the mean, indicating model performance consistency across grid blocks.

Figure 9 illustrates the temporal and spatial evolution of the evaluation metrics, revealing distinct patterns. Maximum absolute errors peak at time steps 15 and 51, reaching 736 and 323 psi, respectively. These coincide with the deactivation of production well P1 (time step 14) and injector I1 and producer P2 (time step 50), suggesting the model struggles to capture pressure dynamics in areas strongly affected by well-switching events. Figure 10 corroborates this, showing concentrated high errors in the interior cells near deactivated and activated wells, contrasting with lower errors in distant regions. Notably, both mean absolute error and standard deviation significantly decrease following well-switching events. This indicates an improved model performance in the ensuing low-variance flow regime as the system stabilizes post-disturbance.

This simplistic application demonstrates how the complexity of learning pressure variation patterns can differ across reservoir regions. The analysis shows that most areas experience limited spatial and temporal pressure changes, making them suitable for accurate modeling using simple ML proxies. However, certain localized regions exhibit significantly more complex pressure dynamics, requiring conventional iterative methods to accurately capture their state at specific time steps. Identifying which areas can be effectively modeled with simple approaches and which demand more advanced techniques is essential for optimizing computational efficiency and improving predictive accuracy in reservoir simulations.

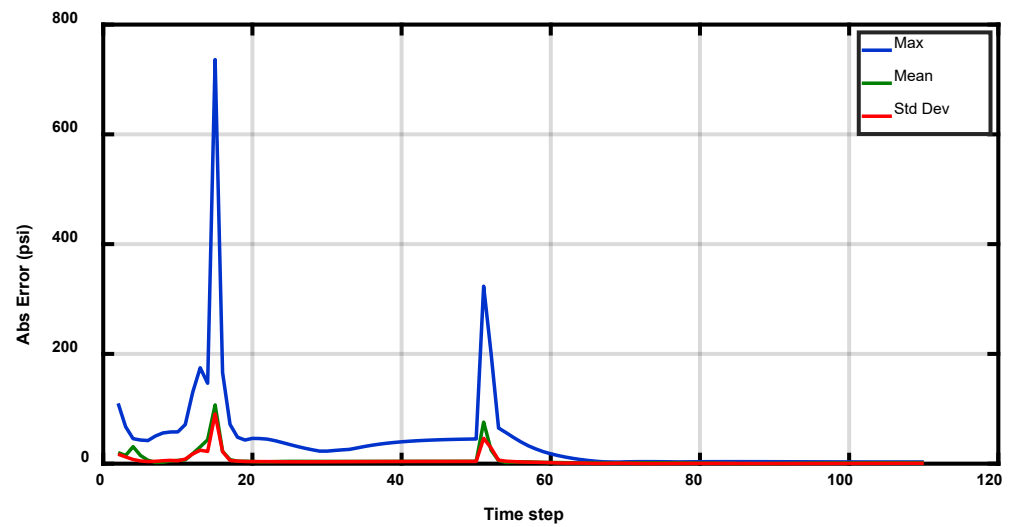


Figure 9. Max absolute error, mean absolute error, and standard deviation in ML proxy model training.

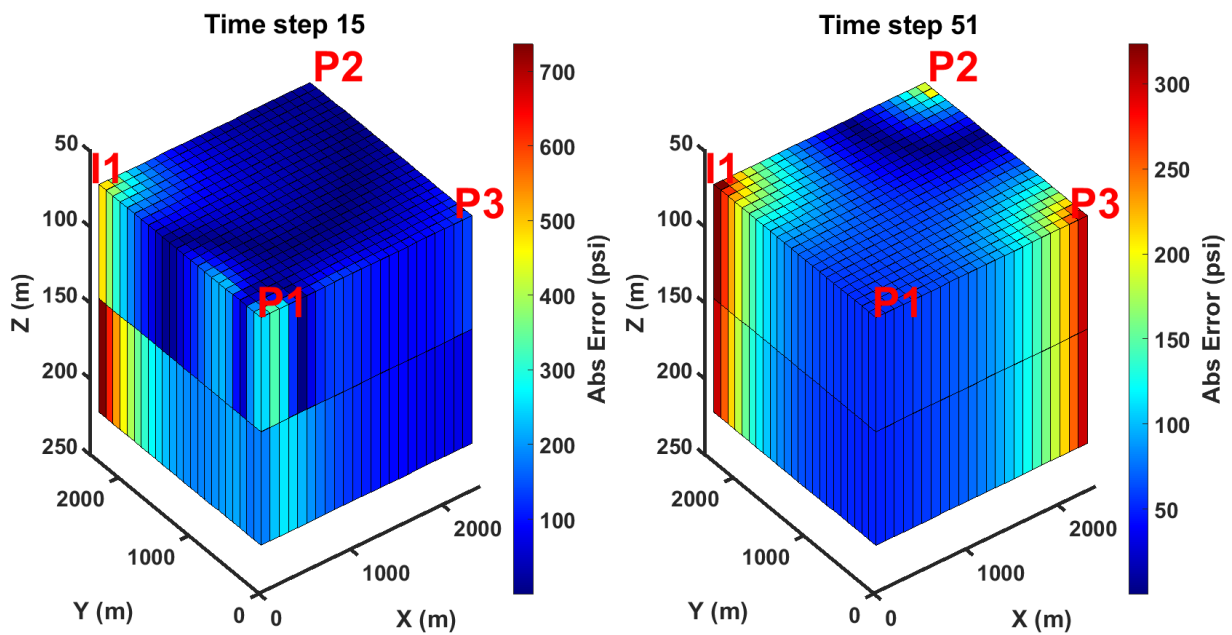


Figure 10. Spatial distribution of absolute error in interior grid blocks at time steps 15 and 51.

3. Methodology

This section introduces a novel, fully automated, machine learning (ML)-driven methodology for classifying grid blocks within a reservoir model as either fast-varying or slow-varying, with the primary objective of accelerating numerical simulations of CO₂ geological storage in deep saline aquifers.

As discussed in Section 2, prior attempts at using ML models to simulate grid-scale pressure dynamics during brine injection and production—under conditions of time-varying bottomhole pressure control and sudden well shut-ins and openings—revealed limitations in proxy models. Specifically, these models struggled to accurately estimate pressure in regions and time steps where fluctuations were highly unpredictable, particularly in areas affected by well-switching events. To address this challenge, a two-step approach is proposed, as outlined in Figure 11.

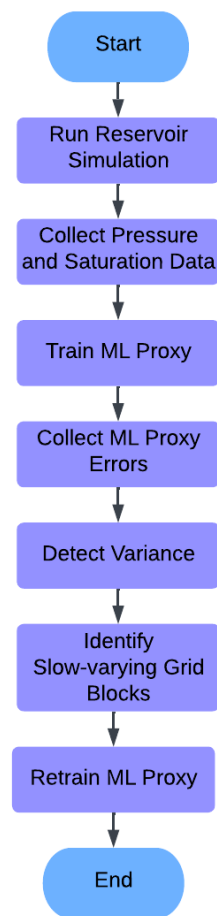


Figure 11. Outline of the proposed methodology.

In the first stage, an ML proxy model is trained to predict changes in pressure and saturation for each grid block within the reservoir. The model is trained on data generated by a numerical simulator, which include the pressure and saturation values of each grid block at successive time steps, along with the state of neighboring grid blocks. The simulation runs that generate these datasets must be designed with a high degree of generality to encompass the full spectrum of expected variability in grid block behavior. Additionally, the datasets should strictly adhere to subsurface regulatory frameworks and align with industry-standard best practices.

Once the ML proxy is trained, the next step involves collecting the proxy's prediction errors by comparing its predictions to the actual simulator results for each grid block and time step. An interquartile range (IQR)-based statistical outlier detection method is then applied to classify grid blocks based on the ML proxy's error variance across all time steps. To ensure precise classification, the IQR-based detector is applied separately to the pressure and saturation predictions' errors.

The IQR is determined as the difference between the third quartile ($Q3$) and the first quartile ($Q1$) of the error distribution. Grid blocks with errors exceeding a predefined lower threshold ($Q1 - k \times IQR$) or upper threshold ($Q3 + k \times IQR$) are classified as fast-varying. The hyperparameter k controls the sensitivity of the classification and plays a critical role in balancing computational efficiency and prediction accuracy.

A smaller value of k results in narrower thresholds, meaning that even slight variations from the median error will cause more grid blocks to be classified as fast-varying. While this approach may capture subtle variations, it can also increase the computational load because more grid blocks are forwarded to the conventional iterative non-linear solver. On the other hand, a larger k would result in wider thresholds, classifying fewer grid blocks as fast-varying and reducing computational costs but potentially missing important variations.

In practice, optimizing k is an iterative process aimed at achieving the best trade-off between accurate classification and computational cost. A well-optimized k ensures that computational resources are focused where they are most needed—on the fast-varying regions—while the slow-varying regions are efficiently handled by the proxy model, reducing overall computational burden.

In the second stage, once the grid blocks have been classified, those identified as outliers in either pressure or saturation predictions are excluded from further proxy model analysis and are passed to the conventional iterative non-linear solver. The remaining slow-varying grid blocks are used for further state prediction by the ML proxy model. At this point, the ML proxy is retrained using only the slow-varying grid blocks, which form a subset of the original dataset used for initial training. This division leads to significant computational time savings, as the simpler and faster ML proxy models handle the majority of grid blocks—those exhibiting low spatial and temporal variance—while the more computationally expensive non-linear solvers are reserved exclusively for the fast-varying grid blocks.

4. Results and Discussion

A three-dimensional Cartesian, physics-based reservoir model was developed using the MATLAB Reservoir Simulation Toolbox (MRST) [47,48] to simulate CO₂ injection and brine production in a deep saline aquifer. This model integrates characteristics observed in major commercial projects worldwide and serves as the basis for generating synthetic data for testing the classification methodology described in Section 3.

The modeled aquifer spans 2100 m in both length and width, with a maximum observed thickness of 250 m. The grid resolution is 210 m in both the x and y directions, and 25 m in the z direction, resulting in a $10 \times 10 \times 10$ grid configuration. The reservoir displays Gaussian-distributed heterogeneity, with a median porosity of 25% and a median permeability of 245 mD, reflecting the range of values typical in large-scale commercial projects. Additionally, vertical permeability is set at 20% of the horizontal permeability in the X and Y directions.

The top of the reservoir is positioned at a depth of 1925 m, determined from the weighted average of large-scale commercial cases. The reservoir is assumed to be horizontally layered and isothermal, with a maximum temperature of 100 °C. In the absence of publicly available salinity data, a salinity of 150,000 ppm was adopted, based on values from the L. Tuscaloosa Sandstone Formation in the SECARB Mississippi Pilot project [49]. This results in a top reservoir pressure of 206.1 bars. Relative permeabilities were calculated using a connate water saturation of 0.27 and a residual CO₂ saturation of 0.20, utilizing built-in MRST functions that were uniquely implemented across all simulations.

To further accelerate the simulation, the black oil modeling technique described in [41] was employed, producing the solution gas–oil ratio (altered to CO₂–brine ratio) and oil formation volume factor (altered to brine formation volume factor) as functions of pressure for the aquifer with the specified characteristics ($T = 100$ °C and salinity of 150,000 ppm). Two injection wells were placed on one side of the reservoir (ij : 1,1 and 1,10), while two producers were positioned on the opposite side (ij : 10,1 and 10,10). Each injection well was rate-controlled, injecting 0.85 million tons per year into the bottom two layers (9 and 10), corresponding to a perforated length of 50 m. The injectors were perforated at the reservoir's deepest layer only to ensure sufficient dissolution of the injected CO₂ into the brine, as the former is driven to the top due to buoyancy. A bottom hole pressure constraint of 500 bars was imposed, consistent with the maximum allowable pressure buildup. The production wells were configured to extract brine, reducing pressure and allowing for more efficient CO₂ injection and storage. They were bottom-hole pressure controlled at 400 bars each, and they were perforated in the bottom two layers (50 m long), to minimize CO₂ production.

The simulation covered a 5-year injection period, representing the initial phase of model validation and aligning with typical 25-year storage operation permit durations.

This was followed by a 5-year post-closure monitoring phase, bringing the total simulation time to 10 years. During the injection phase, the simulation utilized monthly time steps to generate results, while during the post-closure phase, quarterly intervals were used. Figure 12 demonstrates the CO₂ plume evolution at the end of the injection period (5 years) and at the end of the monitoring period (10 years) by illustrating CO₂ saturation (i.e., grid block volume fraction occupied) for all grid blocks.

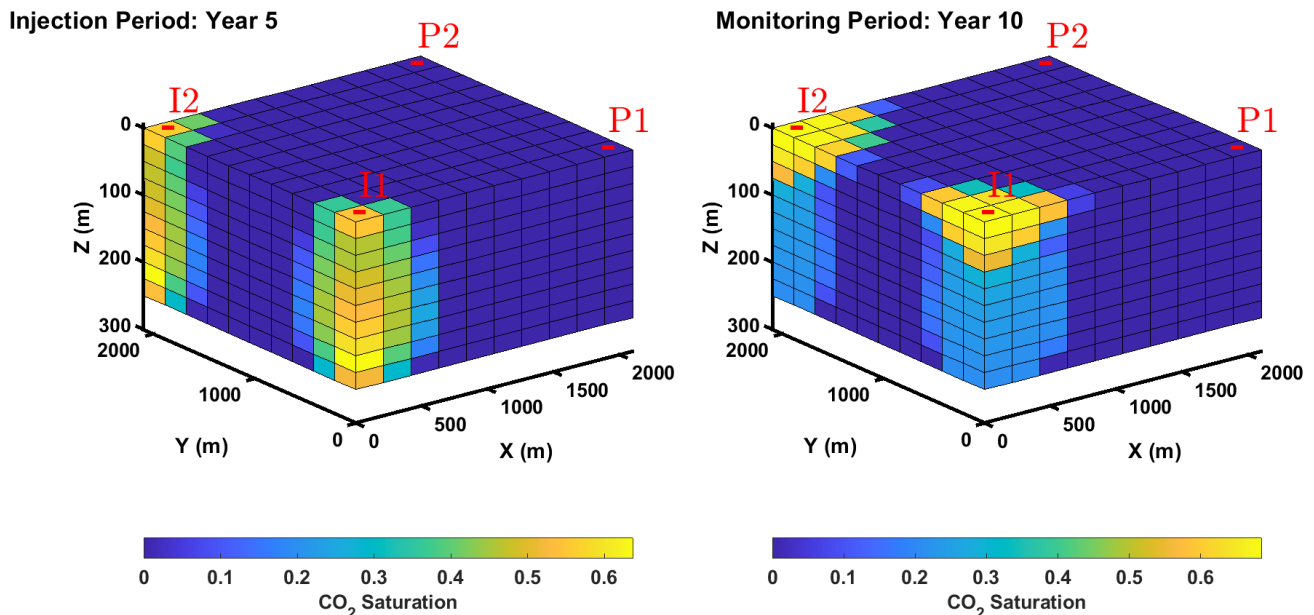


Figure 12. CO₂ saturation distribution in the reservoir at years 5 and 10.

While the proposed methodology is applicable to both interior and boundary cells, the focus is placed on interior grid blocks, in order to assess the method's effectiveness in the most challenging areas of the aquifer model. Unlike boundary cells, which typically experience flow from three to five directions, interior grid blocks can encounter fluid movement from all six faces. This increase in complexity in flow patterns and interactions with adjacent cells makes accurate prediction of CO₂ migration and brine displacement in these areas more challenging for machine learning (ML) proxies. By successfully demonstrating the efficacy of the methodology in these areas, it can be confidently asserted that it is also effective for application in boundary grid blocks.

The ML model used in the CO₂ injection study within the aquifer is a three-layer feedforward neural network (ANN), following the same architecture described in detail in Section 2. The training dataset was generated through simulation, ensuring consistency and the absence of noise. As a result, there was no requirement to apply specialized algorithms designed for handling noise or outliers, as such complexities did not pertain to this dataset.

The ANN model takes as inputs the rates of change in pressure and saturation (dP/dt and dS/dt) for both the focal cell and its six neighboring cells, which share one common face with the focal cell, over the time intervals from t_{i-2} to t_{i-1} and from t_{i-1} to t_i . Additionally, it incorporates the pressure and saturation values at t_i for both the focal cell and these adjacent grid blocks. The model's output then predicts the rate of change in pressure and saturation (dP/dt and dS/dt) for the focal cell between t_i and t_{i+1} , enabling the prediction of future reservoir behavior.

The use of derivatives, rather than only raw pressure and saturation values at individual time steps, is driven by the need to maintain time step invariance. Specifically, in reservoir simulation, time steps are not fixed, as they are often dynamically adjusted by the non-linear solver depending on convergence behavior. For instance, the solver may require more iterations to converge at a given moment, leading to variable time steps. By relying

on dP/dt and dS/dt , the proxy model remains independent of the specific time intervals, allowing it to seamlessly integrate with the solver's adaptive time-stepping algorithm. This approach preserves the model's robustness and flexibility, ensuring that predictions are not tied to a rigid time grid but rather reflect the intrinsic dynamics of the reservoir.

To further enhance the model's temporal sensitivity, inputs from three consecutive time steps— t_{i-2} , t_{i-1} , and t_i —were selected. This choice is grounded in the Taylor series expansion, a mathematical tool used to approximate complex, time-varying functions. For example, it can represent pressure and saturation changes within a reservoir grid block as a sum of their derivatives at a given point in time. The Taylor series can be expressed in its general form as follows:

$$f(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!} (x - a)^n \quad (4)$$

where $f^{(n)}(a)$ is the n th derivative of the function evaluated at point a , $n!$ is the factorial of n , and $(x - a)^n$ represents the difference between x and a .

In the context of time-dependent processes like pressure or saturation in a grid block, a represents the point in time at which the function is evaluated, which can be any relevant time step (e.g., t_i), and x represents the time point where the function is being approximated, such as a future or neighboring time step (e.g., t_{i+1}). Thus, the term $(x - a)$ translates to $(t_{i+1} - t_i)$, representing the time difference between two selected time steps.

Each derivative in the Taylor series has a specific physical meaning related to the function's evolution in time. The first derivative, $f'(t_i)$, describes the rate of change (or slope) of the function at time t_i , indicating how fast the function is changing at that point. The second derivative, $f''(t_i)$, represents the curvature or the rate at which the rate of change itself is evolving, capturing non-linear behavior or acceleration of changes over time. Higher-order derivatives account for increasingly finer details of the function's time evolution, thus accounting for more complex, higher-order effects like the changes in curvature and other non-linearly evolving processes.

By including not only the current rate of change but also previous time steps, the model implicitly accounts for higher-order temporal effects that influence reservoir behavior. This is especially significant in reservoir simulations, where fluid flow dynamics depend on both the present state and the preceding changes. By incorporating data from multiple time steps, the model enhances its accuracy by accounting for these cumulative effects, which may otherwise be overlooked when only a single time step is considered.

Figure 13 provides a detailed breakdown of the errors in the model's predictions of pressure change (dP/dt) and saturation change (dS/dt) over time. The comparison is made between two proxy models: one in which all time instances of the interior cells are used for model training, and another where only the slow-varying grid blocks—representing 71% of all these time instances (31,093 out of 44,032)—are used. These 44,032 instances correspond to the values across the 512 interior grid blocks evaluated at 86 time steps (from time step 4 to 89), for which the ML model provides predictions. The slow-varying cells were identified by the automated ML and the IQR-based classifier outlined in Section 3. As part of this IQR-based selection process, a hyperparameter value of $k = 0.5$ was applied for pressure and $k = 20$ for saturation.

Excluding 71% of the cells from the solver and handling them directly with the ANN means the solver only needs to process 29% of the problem's full dimension. This reduction effectively brings down the time complexity to between 0.29^2 and 0.29^3 of that required for the complete problem, translating to approximately 8% to 2.4% of the original computational cost. Consequently, this method can decrease the computational burden by a factor of $1/0.08 = 12$ to $1/0.024 = 42$, allowing users to conduct significantly more scenarios, evaluate them efficiently, and ultimately select the optimal one for their purposes.

Errors are evaluated using three primary metrics: maximum absolute error, mean absolute error, and standard deviation of the absolute error. These metrics are visualized for both the full set of interior cells (in red) and the slow-varying cell runs (in blue). Note that

including boundary cells with 3, 4, or 5 neighbors would result in more cells being classified as slow-varying due to their proximity to the non-flow boundaries of the reservoir.

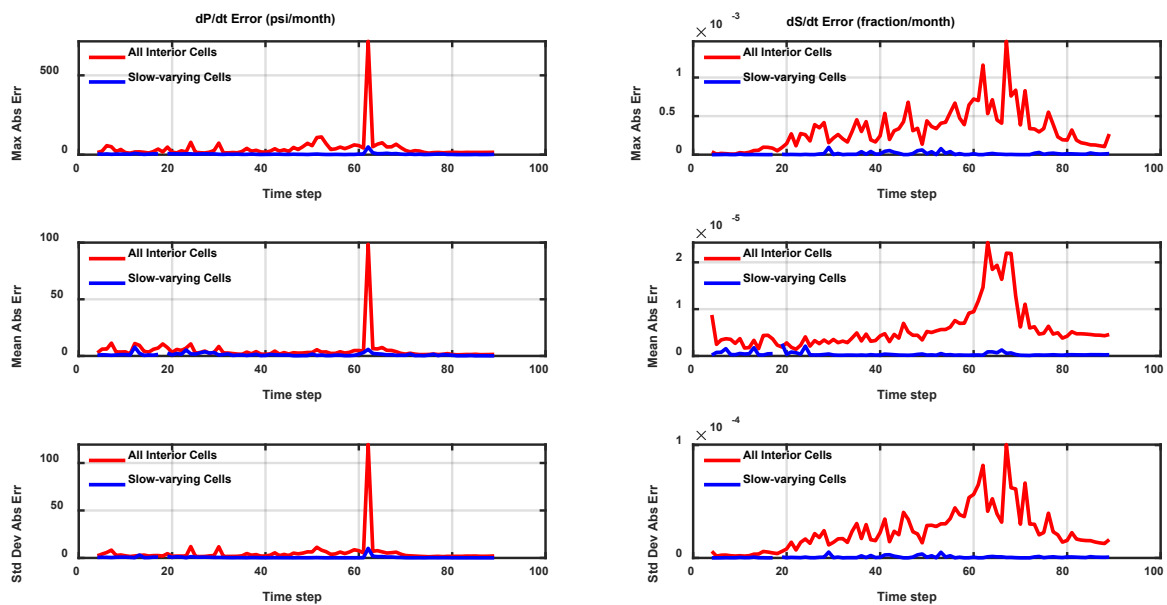


Figure 13. Comparison of dP/dt and dS/dt error metrics when the ML model is trained using all interior cells vs. slow-varying cells across time steps.

In the case where the model is trained on all interior cells, the maximum absolute error for dP/dt remains relatively low throughout most time steps, with one notable exception: a sharp spike observed at time step 62. This peak suggests that the model struggles to accurately predict pressure changes during this period due to rapid fluctuations in pressure rates between time steps 61 and 62, deviating from the general trend. The complexity arises during the transition from the injection phase to the monitoring phase, marked by the shutdown of both injection and production wells. This operational change results in significant pressure rate fluctuations, which the model finds challenging to capture accurately, causing a temporary spike in error. After time step 62, as pressure rate changes stabilize, the maximum absolute error decreases and levels off, indicating the system has returned to a more predictable state. In contrast, in the second case where the model is trained only on slow-varying cells, consistently low errors are observed without noticeable spikes. These slow-varying cells experience gradual pressure changes, and the model performs reliably under such stable conditions, demonstrating that it can handle smooth, steady-state behavior more effectively.

The application of our methodology, which categorizes grid blocks based on variance in pressure and saturation behavior, is validated by these results. The proxy model's ability to focus on slow-varying cells ensures that computational efforts are directed where they are most effective, thus reinforcing the overall aim of reducing computational cost while maintaining accuracy.

The mean absolute error for dP/dt follows a similar pattern in both cases. In the case where the model is trained on all interior cells, there is a moderate increase in error at time step 62, reflecting the difficulty of predicting pressure changes during rapid fluctuations. However, in the second case, when trained only on slow-varying cells, the mean absolute error remains consistently low and stable throughout the time steps, further demonstrating that the model is better suited to predicting pressure changes when the variations are gradual.

Similarly, in the dS/dt (CO_2 saturation change rate) error analysis, when the model is trained on all interior cells, a comparable pattern emerges. While the maximum absolute error for dS/dt starts off relatively low, it begins to fluctuate from time step 20 onward,

peaking between time steps 46 and 76. These fluctuations reflect the model's difficulty in accurately predicting CO₂ saturation changes during periods when the CO₂ plume migrates into interior cells that were previously unaffected. As CO₂ saturates new regions of the reservoir, cells that initially had no CO₂ experience sudden, large changes in saturation, leading to rapid rate changes and increased errors. This issue is not confined to regions near the wells, as it is primarily associated with the movement of the plume within the reservoir. While early time steps show greater errors close to the wells due to the initial migration of the plume, as the plume progresses further into the reservoir, errors also arise in cells located farther from the wells. After time step 76, the maximum absolute error begins to decrease, implying that the reservoir enters a more stable phase, and the model's predictions for saturation become more reliable as the system stabilizes.

In contrast, in the case where the model is trained only on slow-varying cells, the maximum absolute error remains consistently low across all time steps. There are no significant spikes in error, suggesting that the model performs reliably when predicting saturation changes in these regions. This indicates that the model is well-suited to handling areas with slow, steady-state CO₂ migration and struggles more with the complex dynamics present in cells experiencing rapid saturation changes.

Again, these results underscore the effectiveness of the classification-based methodology. By applying the ML proxy model to slow-varying grid blocks, the system can achieve reliable predictions with significantly lower computational effort, demonstrating the efficiency gains sought in this study.

The mean absolute error for dS/dt follows a similar trend to the maximum absolute error. When all interior cells are used for training, there is a gradual increase in the mean error starting around time step 20, peaking at time step 63, reflecting the model's difficulty in predicting saturation changes during periods of rapid CO₂ redistribution. After time step 71, the mean absolute error decreases, suggesting that the model regains some degree of predictive accuracy as the saturation changes slow down.

Conversely, when the model is trained only on slow-varying cells, the mean absolute error remains low and stable throughout the entire simulation, demonstrating that the model can predict CO₂ saturation changes in these regions with high accuracy. The absence of significant error increases indicates that the model performs well when saturation changes are gradual and steady.

The standard deviation of the absolute error for dS/dt reveals additional insights into the variability of the model's predictions. In the case where all interior cells are used, the standard deviation increases significantly from time step 20 onward, peaking between time steps 58 and 70. This increase in variability suggests that the model's predictions become more inconsistent across different cells during periods of rapid saturation changes. The higher standard deviation during this period indicates that certain cells in the reservoir are experiencing large, unpredictable changes in CO₂ saturation, which the model finds challenging to capture accurately.

In contrast, when the model is trained on slow-varying cells, the standard deviation remains consistently low across all time steps. This suggests that the model's predictions of saturation changes in slow-varying cells are not only more accurate but also more consistent, as the gradual nature of saturation changes in these cells allows the model to maintain steady and reliable predictions.

These findings reaffirm the benefits of focusing on slow-varying cells, as highlighted in the methodology section. The ability of the proxy model to produce consistent and accurate results in these regions aligns with the broader goal of optimizing computational efficiency in large-scale CO₂ sequestration simulations.

It is important to emphasize that although the ML model is designed to predict the pressure rate change, dP/dt , the primary objective remains the prediction of the absolute pressure P . Therefore, in addition to analyzing the error in dP/dt predictions, scatter plots were employed to evaluate the ML model's ability to predict the absolute pressure across all interior cells and the slow-varying ones. These scatter plots, presented in Figure 14,

compare the predicted pressure \hat{P} and the actual pressure \bar{P} , offering a visual insight into the model's predictive accuracy.

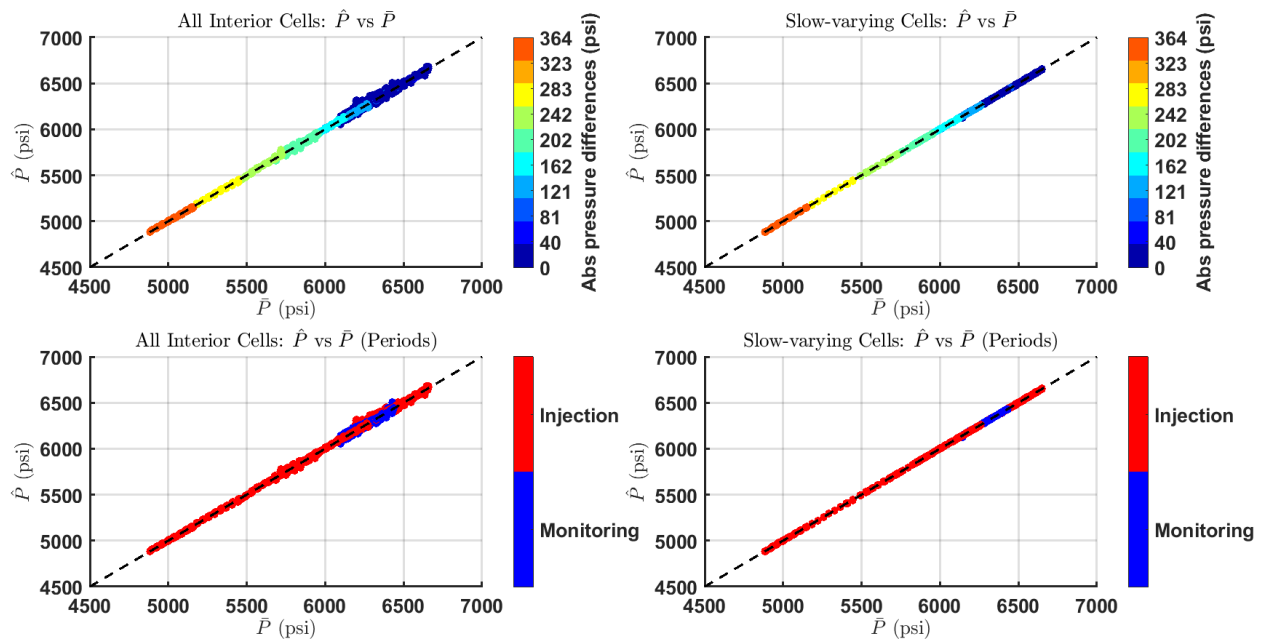


Figure 14. Comparison of predicted vs. actual pressure when the ML model is trained using all interior cells vs. slow-varying cells, differentiating absolute real pressure differences between consecutive time steps and injection and monitoring phases.

The transformation from the predicted rate of pressure change to the absolute pressure is governed by the following relationship:

$$\hat{P}_{i+1} = \bar{P}_i + \frac{d\hat{P}}{dt}_i (t_{i+1} - t_i) \quad (5)$$

In this equation, the predicted pressure \hat{P}_{i+1} at the next time step is computed based on the actual pressure \bar{P}_i at the current time step and the predicted rate of change $\frac{d\hat{P}}{dt}_i$, scaled by the time interval $(t_{i+1} - t_i)$.

In Figure 14, the points in the plots are color-coded based on the absolute pressure differences between consecutive time steps, with blue representing smaller differences and red indicating larger ones. In other words, the more reddish the point color, the greater the pressure change in the timestep, which needs to be predicted by the ML model. Furthermore, the scatter plots distinguish between the injection and monitoring periods, where red denotes the injection phase and blue represents the monitoring phase.

For the case where the model is trained on all interior cells, the scatter plots on the left indicate that the predicted values generally follow a linear relationship with the actual values, closely aligning along the $\hat{P} = \bar{P}$ line, shown as a dashed black line. However, while the plots may suggest better performance at lower pressure levels, this might be misleading. The appearance of fewer errors at lower pressures is not due to improved model accuracy, but rather to the fact that fewer data points fall into this range during the early stages of CO₂ injection, where pressure builds up rapidly. In fact, the model still struggles to predict pressure for fast-varying cells in these early stages, but the smaller number of data points makes this difficulty less apparent. The model's primary challenge lies in predicting rapid pressure changes, rather than being inherently more accurate at lower pressures. Nevertheless, accurate predictions during the phase of rapid pressure buildup are critical, as they help identify when the injection rate is approaching or exceeding the aquifer's

capacity. Excessive pressure can lead to induced seismicity or leakage into overlying formations, posing significant risks.

As time progresses and more data points accumulate at higher pressure levels, the model's errors become more evident. The cooler colors in the scatter plot indicate smaller pressure differences between consecutive time steps, showing that the model's difficulty lies in capturing the increasingly non-linear behavior of the reservoir, particularly as it tries to handle fast-varying cells. It is clear that the model's accuracy is primarily influenced by how well it can manage sudden changes in pressure, rather than pressure magnitude itself.

The lower-left scatter plot, which differentiates between the injection and monitoring phases, highlights further aspects of the model's performance. During the injection phase, shown in red points, the model faces challenges, particularly at higher pressures, where the spread of data points is more pronounced. This is largely due to the rapid and unpredictable pressure fluctuations that occur during CO₂ injection, making it difficult for the model to maintain accuracy. However, the spread of points is more a reflection of the fast pressure changes during injection, rather than just the pressure levels themselves.

In the monitoring phase, represented by blue points, where fewer data points are present, there is still a noticeable spread and deviation from the $\hat{P} = \bar{P}$ line. This deviation occurs because the monitoring phase includes the transition from the injection period to the monitoring period, particularly around time step 62, when all wells are shut down. During this transition, the pressure dynamics shift rapidly as the system adjusts to the cessation of injection, leading to complex pressure variations that the model struggles to capture, especially in fast-varying cells. Note that accurate predictions in the monitoring phase are equally as crucial as those in the injection phase. Reliable predictions are essential for assessing sequestration effectiveness and mitigating potential leakage risks throughout the entire CO₂ storage process.

In contrast, when the model is trained on slow-varying cells, as can be seen in the scatter plots on the right, its performance improves significantly across all pressure ranges. These cells experience more gradual pressure changes, leading to tighter alignment between the predicted and actual values, regardless of pressure. This improvement is due to the exclusion of fast-varying cells, which exhibit rapid and unpredictable pressure fluctuations. By focusing only on slow-varying cells, the model is relieved of the challenges associated with rapid pressure changes, resulting in more accurate and consistent predictions across both the injection and monitoring phases.

Figure 15 provides further insight into the ML model's performance, this time focusing on CO₂ saturation predictions \hat{S} compared to actual values \bar{S} . The transformation from dS/dt to S is analogous to Equation (5), but with saturation replacing pressure. The scatter plots indicate that predictions for all interior cells are notably more accurate than those for pressure, with data points closely clustering around the ideal $\hat{S} = \bar{S}$ line, even during periods of more rapid saturation changes. This suggests that the model can generally handle saturation dynamics effectively, which contrasts with the greater challenges observed in predicting pressure changes. Given this higher accuracy in saturation predictions, more lenient thresholds were applied for the classifier in the case of saturation, allowing the inclusion of a broader range of cells in the slow-varying saturation category. Note that since grid blocks identified as outliers in either pressure or saturation predictions are excluded from further proxy model analysis, this explains why less grid blocks remain in the slow-varying saturation category, as shown in Figure 15, despite the more lenient thresholds.

Figure 16 offers a final comparison of the errors in pressure and saturation predictions over the entire 10-year period, showing how these errors accumulate in the two cases. In the first case, where all grid blocks are considered and the proxy model operates independently from the non-linear solver, the errors for both pressure and saturation steadily increase over time. The maximum, mean, and standard deviation of the errors rise continuously, with the accumulation becoming so large that the predictions generated by the model become unreliable.

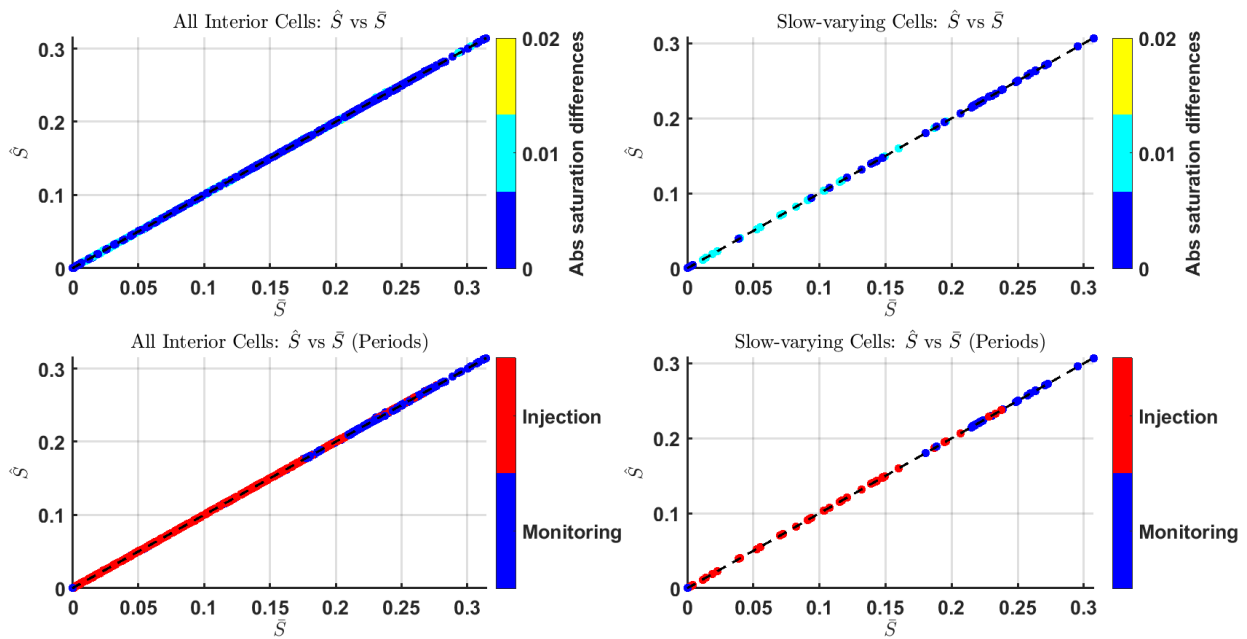


Figure 15. Comparison of predicted vs. actual saturation when the ML model is trained using all interior cells vs. slow-varying cells, differentiating absolute real saturation differences between consecutive time steps and injection and monitoring phases.

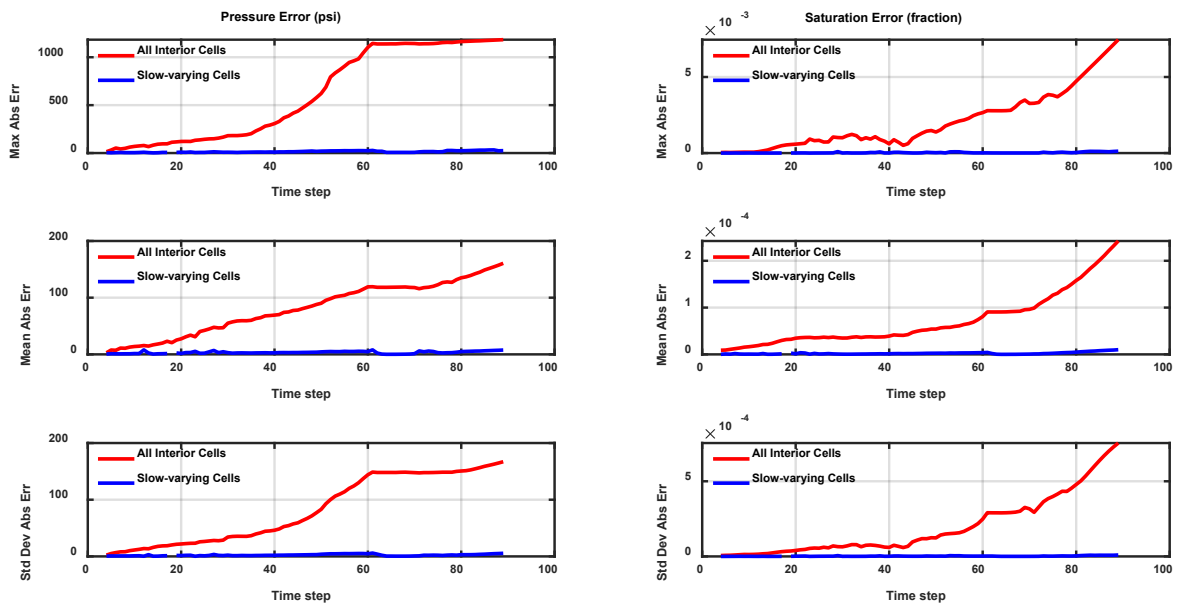


Figure 16. Comparison of cumulative error metrics for P and S when the ML model is trained using all interior cells vs. slow-varying cells.

For instance, at the midpoint of the simulation (5 years), the mean error in pressure for the case where all interior cells are considered reaches approximately 120 psi, while the mean error in saturation reaches 9.07×10^{-5} in fraction. In contrast, in the case where only the slow-varying cells are considered, the mean error in pressure is substantially lower at about 8 psi, and the error in saturation is around 4.16×10^{-6} fraction. This stark difference highlights the improved accuracy achieved by focusing on slow-varying cells.

By the end of the 10-year simulation, the accumulation of errors becomes even more pronounced. In the case where all interior cells are used, the error in pressure escalates to approximately 161 psi, and the saturation error increases to about 0.0002 fraction. However, in the slow-varying cell case, the pressure error remains much lower at around 8 psi, while

the saturation error is limited to 9.98×10^{-6} fraction. These numbers demonstrate that the application of the proposed methodology results in significantly lower error accumulation, making it a more reliable option for long-term predictions.

This error accumulation can be better understood by examining how pressure and saturation predictions evolve at each time step. Each predicted state at time step t_{i+1} , denoted as \hat{P}_{i+1} and \hat{S}_{i+1} , is computed using the following formulas:

$$\hat{P}_{i+1} = \hat{P}_i + \frac{dP}{dt}_i (t_{i+1} - t_i) \quad (6)$$

$$\hat{S}_{i+1} = \hat{S}_i + \frac{dS}{dt}_i (t_{i+1} - t_i) \quad (7)$$

While the error in one time step may seem negligible, it is carried forward and compounded in subsequent time steps, leading to a cumulative effect. Over a series of time steps, the compounding effect becomes significant. As time progresses, this accumulation results in a large deviation from the true system state, especially since there is no feedback from the non-linear solver to correct the trajectory.

In contrast, the second approach utilizes a hybrid methodology wherein regions of the grid exhibiting rapid changes in pressure and saturation are selectively excluded from ML-based predictions. Instead, the non-linear solver is employed in these regions to provide more accurate estimates of P_i and S_i at critical time steps, mitigating error accumulation in the areas most prone to dynamic changes. For grid blocks where pressure and saturation vary slowly, the ML model continues to make predictions, as the risk of significant error accumulation is considerably lower in these zones. As illustrated in Figure 16, this hybrid approach yields significantly more stable and accurate results over time, with errors remaining consistently lower compared to the first case. The strategic exclusion of fast-varying regions from ML-based predictions is therefore highly effective in maintaining model fidelity, demonstrating the advantage of adaptive modeling in reducing long-term error accumulation.

It should further be noted that the results obtained are limited by the size and complexity of the selected ML model. In this specific example, the model used was a single hidden-layer ANN containing 10 neurons, thus rapidly increasing its response speed, while limiting its learning capacity. Clearly, by increasing the size of the network, either by adding more neurons to the hidden layer or by introducing additional hidden layers, the model's learning capacity can be enhanced.

5. Conclusions

This article introduces a novel and highly efficient approach to accelerating numerical simulations of CO₂ geological storage in deep saline aquifers by integrating machine learning (ML) proxy models into traditional reservoir simulation workflows. The core innovation of this approach lies in the strategic classification of reservoir grid blocks based on their dynamic behavior in terms of pressure and saturation changes. Specifically, the proposed method differentiates between fast-varying and slow-varying regions within the reservoir, which allows for a targeted allocation of computational resources, significantly enhancing the overall efficiency of the simulation process without sacrificing accuracy or precision in key areas.

The classification methodology presented in this work follows a two-stage process. In the first stage, an ML proxy model is developed and trained using the dataset from grid blocks within distinct regions of the reservoir, such as interior or boundary cells. Given the need for rapid, yet reliable, predictions, the chosen ML proxy model is deliberately simple, striking an optimal balance between computational speed and predictive accuracy. Following the initial training, the ML model is further refined to focus on slow-varying grid blocks—those exhibiting minimal spatial and temporal variations. At this stage, the ML proxy is retrained using only the slow-varying subset of grid blocks, significantly

improving computational efficiency. By isolating the slow-varying blocks, which constitute the majority of the grid, the ML proxy model can efficiently handle their predictions, while more resource-intensive non-linear solvers are applied exclusively to the fast-varying blocks. This division between fast and slow dynamics leads to significant time savings, as computationally expensive calculations are avoided for regions where they are unnecessary.

The identification of fast-varying regions is conducted using an ML and interquartile range (IQR)-based outlier detection technique, which automatically flags grid blocks where the proxy model's predictions deviate substantially from the expected range based on the overall model behavior. These flagged grid blocks, as already mentioned, are then classified as fast-varying and treated with conventional iterative solvers. Fast-varying grid blocks are typically located around critical regions such as wells, where rapid changes in pressure and saturation often occur due to well activation or deactivation, or significant fluctuations in injection and production rates. These regions are challenging to identify due to complex interactions between well operations and reservoir properties, such as well perforation details, injection rates, wellbore inclination, and petrophysical characteristics like permeability anisotropy. The automation of this identification process through machine learning greatly reduces the possibility of human error and offers substantial time savings over manual methods.

Importantly, while this study utilized a fully automated ML and IQR-based classifier to distinguish fast- and slow-varying regions, other classification techniques could also have been employed. The classifier chosen demonstrated a satisfactory performance in detecting regions with low spatial and temporal variance, but future work could explore alternative models to enhance detection accuracy or further optimize computational speed. This highlights the adaptability of the framework to different ML approaches, depending on the specific requirements of the reservoir model or the operational scenario.

Fast-varying cells need to be handled regularly by the non-linear solver of the differential equations governing the flow while honoring the values predicted by the proxy model. Although acceleration is guaranteed due to the vast reduction in the number of cell variables to be predicted, it is recommended that smart numbering must be assigned to those cells. This will allow the system matrix to remain banded rather than just sparse, so as to take full advantage of the linear solver available.

Another notable aspect of this work is the flexibility of the proposed methodology, which is not limited to CO₂ injection into deep saline aquifers. The approach can be readily adapted to other subsurface storage applications, such as CO₂ injection into depleted oil reservoirs. In these cases, an additional variable—such as oil saturation—would need to be predicted to account for multi-phase flow dynamics. Furthermore, for CO₂ injection into aquifers, the model could be extended to predict the amount of CO₂ dissolved in the aqueous phase, a key factor in long-term storage security. However, in this study, the focus was intentionally narrowed to the prediction of pressure and CO₂ saturation to validate the methodology and demonstrate its effectiveness in classifying reservoir dynamics.

It is important to note that this study only serves as an initial validation of the proposed hybrid methodology, limited to a single scheduling scenario to explore the foundational capabilities of the method. Future iterations will require larger datasets, and the simulation runs generating these datasets must be designed with a high degree of generality to encompass the full spectrum of expected variability in grid block behavior. This means that the input data used to train the ML models should be derived from simulations that capture a broad range of subsurface conditions, including various injection and production scenarios, and petrophysical properties such as permeability and porosity distributions. This ensures that the proxy models are robust and capable of generalizing to diverse operational conditions encountered in field-scale projects. Moreover, it is imperative that the datasets strictly adhere to subsurface regulatory frameworks and align with industry-standard best practices for CO₂ storage and reservoir simulation. Compliance with these standards ensures that the methodology remains applicable to real-world projects, where operational safety, regulatory requirements, and long-term containment integrity are paramount. Adherence

to regulatory frameworks also ensures that the results produced by the simulations can be used in formal reporting and decision-making processes, which are critical for the approval and monitoring of CO₂ storage projects.

The proposed method is particularly viable for CO₂ injection into saline aquifers, where accurate pressure estimation near wells is critical for maintaining CO₂ containment integrity. In such projects, pressure management is closely tied to geomechanical stability, as excessive pressures can lead to fracturing or caprock failure, potentially compromising the containment of injected CO₂. Since the grid blocks in which wells are perforated are classified as fast-varying under this methodology, the rigorous, non-linear solvers are applied to these critical regions, ensuring that bottom-hole pressure (BHP) estimates remain highly accurate. This is essential for managing well integrity and mitigating risks related to geomechanical issues, such as induced seismicity or the migration of CO₂ through fractures. The proposed approach thus provides a robust solution for ensuring the precision of near-wellbore simulations while simultaneously reducing computational costs across the broader reservoir model.

In summary, the proposed ML-based approach provides a highly efficient and flexible solution for simulating CO₂ storage in geological formations. By automating the classification of grid blocks into fast- and slow-varying categories, this method enables the selective application of computational resources, delivering substantial time savings while ensuring that critical areas, such as those near wells, receive the rigorous attention needed for accurate pressure estimation. The versatility of this methodology extends beyond CO₂ injection into aquifers, offering the potential for application in a wide range of subsurface storage scenarios. Future work could further refine the classifier models used or extend the approach to include additional variables, such as CO₂ dissolution in water or multiphase flow considerations in depleted reservoirs. Overall, this method represents a significant advancement in the field of reservoir simulation, with important implications for the efficiency and accuracy of CO₂ storage operations and their contribution to global climate mitigation efforts through improved carbon sequestration strategies.

Author Contributions: Conceptualization, E.M.K. and V.G.; methodology, E.M.K.; software, E.M.K. and I.I.; validation, E.M.K. and V.G.; formal analysis, E.M.K.; investigation, E.M.K.; data curation, E.M.K.; writing—original draft preparation, E.M.K. and I.I.; writing—review and editing, V.G.; visualization, E.M.K.; supervision, V.G. All authors have read and agreed to the published version of the manuscript.

Funding: The development of the thermodynamic model used in this research was supported by the Hellenic Foundation for Research and Innovation (HFRI) under the 3rd Call for HFRI PhD Fellowships (fellowship number 61/513800).

Data Availability Statement: No new data were created.

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

References

1. Nunes, L.J.R. The Rising Threat of Atmospheric CO₂: A Review on the Causes, Impacts, and Mitigation Strategies. *Environments* **2023**, *10*, 66. [[CrossRef](#)]
2. Kartha, S.; Kemp-Benedict, E.; Ghosh, E.; Nazareth, A.; Gore, T. *The Carbon Inequality Era: An Assessment of the Global Distribution of Consumption Emissions Among Individuals from 1990 to 2015 and Beyond*; Oxfam GB: Oxford, UK, 2020.
3. West, T.O.; Marland, G. A synthesis of carbon sequestration, carbon emissions, and net carbon flux in agriculture: Comparing tillage practices in the United States. *Agric. Ecosyst. Environ.* **2002**, *91*, 217–232. [[CrossRef](#)]
4. Khan, R.; Alabsi, A.A.N.; Muda, I. Comparing the effects of agricultural intensification on CO₂ emissions and energy consumption in developing and developed countries. *Front. Environ. Sci.* **2023**, *10*, 1065634. [[CrossRef](#)]
5. Xu, J.; Grumbine, R.E.; Shrestha, A.; Eriksson, M.; Yang, X.; Wang, Y.; Wilkes, A. The melting Himalayas: Cascading effects of climate change on water, biodiversity, and livelihoods. *Conserv. Biol.* **2009**, *23*, 520–530. [[CrossRef](#)] [[PubMed](#)]
6. Michaelowa, A.; Hermwille, L.; Obergassel, W.; Butzengeiger, S. Additionality revisited: Guarding the integrity of market mechanisms under the Paris Agreement. *Clim. Policy* **2019**, *19*, 1211–1224. [[CrossRef](#)]
7. Ekardt, F.; Wieding, J.; Zorn, A. Paris Agreement, Precautionary Principle and Human Rights: Zero Emissions in Two Decades? *Sustainability* **2018**, *10*, 2812. [[CrossRef](#)]

8. Huang, Z.; Huang, Y.; Zhang, S. The Possibility and Improvement Directions of Achieving the Paris Agreement Goals from the Perspective of Climate Policy. *Sustainability* **2024**, *16*, 4212. [[CrossRef](#)]
9. Ismail, I.; Gaganis, V. Carbon Capture, Utilization, and Storage in Saline Aquifers: Subsurface Policies, Development Plans, Well Control Strategies and Optimization Approaches—A Review. *Clean Technol.* **2023**, *5*, 609–637. [[CrossRef](#)]
10. Ismail, I.; Gaganis, V. Well Control Strategies for Effective CO₂ Subsurface Storage: Optimization and Policies. *Mater. Proc.* **2023**, *15*, 74.
11. IEA. *Net Zero by 2050: A Roadmap for the Global Energy Sector*; IEA: Paris, France, 2021.
12. Luo, A.; Li, Y.; Chen, X.; Zhu, Z.; Peng, Y. Review of CO₂ Sequestration Mechanism in Saline Aquifers. *Nat. Gas Ind. B* **2022**, *9*, 383–393. [[CrossRef](#)]
13. Gassara, O.; Estublier, A.; Garcia, B.; Noirez, S.; Cerepi, A.; Loisy, C.; le Roux, O.; Petit, A.; Rossi, L.; Kennedy, S.; et al. The Aquifer-CO₂ Leak Project: Numerical Modeling for the Design of a CO₂ Injection Experiment in the Saturated Zone of the Saint-Emilion (France) Site. *Int. J. Greenh. Gas Control* **2021**, *104*, 103196. [[CrossRef](#)]
14. Celia, M.A.; Bachu, S.; Nordbotten, J.M.; Bandilla, K.W. Status of CO₂ Storage in Deep Saline Aquifers with Emphasis on Modeling Approaches and Practical Simulations. *Water Resour. Res.* **2015**, *51*, 6846–6892. [[CrossRef](#)]
15. Nordbotten, J.M.; Celia, M.A.; Bachu, S. Injection and Storage of CO₂ in Deep Saline Aquifers: Analytical Solution for CO₂ Plume Evolution during Injection. *Transp. Porous Media* **2005**, *58*, 339–360. [[CrossRef](#)]
16. Doughty, C.; Pruess, K. Modeling Supercritical Carbon Dioxide Injection in Heterogeneous Porous Media. *Vadose Zone J.* **2004**, *3*, 837–847. [[CrossRef](#)]
17. Phade, A.; Gupta, Y. Reservoir Pressure Management Using Waterflooding: A Case Study. In All Days. In Proceedings of the SPE Western Regional and Pacific Section AAPG Joint Meeting, Bakersfield, CA, USA, 31 March–2 April 2008; OnePetro: Richardson, TX, USA, 2008.
18. Lee, J.-Y.; Weingarten, M.; Ge, S. Induced Seismicity: The Potential Hazard from Shale Gas Development and CO₂ Geologic Storage. *Geosci. J.* **2016**, *20*, 137–148. [[CrossRef](#)]
19. Rutqvist, J.; Birkholzer, J.T.; Tsang, C.-F. Coupled Reservoir–Geomechanical Analysis of the Potential for Tensile and Shear Failure Associated with CO₂ Injection in Multilayered Reservoir–Caprock Systems. *Int. J. Rock Mech. Min. Sci.* **2008**, *45*, 132–143. [[CrossRef](#)]
20. Rutqvist, J.; Birkholzer, J.; Cappa, F.; Tsang, C.-F. Estimating Maximum Sustainable Injection Pressure during Geological Sequestration of CO₂ Using Coupled Fluid Flow and Geomechanical Fault-Slip Analysis. *Energy Convers. Manag.* **2007**, *48*, 1798–1807. [[CrossRef](#)]
21. Carroll, S.; Hao, Y.; Aines, R. Transport and Detection of Carbon Dioxide in Dilute Aquifers. *Energy Procedia* **2009**, *1*, 2111–2118. [[CrossRef](#)]
22. Bachu, S. CO₂ Storage in Geological Media: Role, Means, Status and Barriers to Deployment. *Prog. Energy Combust. Sci.* **2008**, *34*, 254–273. [[CrossRef](#)]
23. Alkan, H.; Rivero, F.F.; Burachok, O.; Kowollik, P. Engineering design of CO₂ storage in saline aquifers and in depleted hydrocarbon reservoirs: Similarities and differences. *First Break* **2021**, *39*, 69–80. [[CrossRef](#)]
24. Tiwari, P.K.; Chidambaram, P.; Azahree, A.I.; Das, D.P.; Patil, P.A.; Low, Z.; Chandran, P.K.; Tewari, R.D.; Hamid, M.K.A.; Yaakub, M.A. Safeguarding CO₂ Storage in a Depleted Offshore Gas Field with Adaptive Approach of Monitoring, Measurement and Verification MMV. In Proceedings of the SPE Middle East Oil & Gas Show and Conference, Virtual, 1 December 2021.
25. Patil, P.A.; Hamimi, A.M.; Abu Bakar, M.A.B.; Das, D.P.; Tiwari, P.K.; Chidambaram, P.; Jalil, M.A.B.A. Scrutinizing Wells Integrity for Determining Long-Term Fate of a CO₂ Sequestration Project: An Improved and Rigorous Risk Assessment Strategy. In Proceedings of the International Petroleum Technology Conference, Riyadh, Saudi Arabia, 23 February 2022.
26. Rasool, M.H.; Ahmad, M.; Ayoub, M. Selecting Geological Formations for CO₂ Storage: A Comparative Rating System. *Sustainability* **2023**, *15*, 6599. [[CrossRef](#)]
27. Kanakaki, E.M.; Samnioti, A.; Koffa, E.; Dimitrellou, I.; Obetzanov, I.; Tsiantis, Y.; Kiomourtzi, P.; Gaganis, V.; Stamataki, S. Prospects of an Acid Gas Re-Injection Process into a Mature Reservoir. *Energies* **2023**, *16*, 7989. [[CrossRef](#)]
28. Ahmed, T. *Reservoir Engineering Handbook*, 4th ed.; Elsevier: Amsterdam, The Netherlands, 2010; ISBN 9780080966670.
29. Ahmed, T. *Equations of State and PVT Analysis*; Elsevier: Amsterdam, The Netherlands, 2013.
30. Whitson, C.H. Characterizing hydrocarbon plus fractions. *Soc. Pet. Eng. J.* **1983**, *23*, 683–694. [[CrossRef](#)]
31. Kanakaki, E.M.; Gaganis, V. Automated Equations of State Tuning Workflow Using Global Optimization and Physical Constraints. *Liquids* **2024**, *4*, 261–277. [[CrossRef](#)]
32. Nghiem, L.; Sammon, P.; Grabenstetter, J.; Ohkuma, H. Modeling CO₂ Storage in Aquifers with a Fully-Coupled Geochemical EOS Compositional Simulator. In Proceedings of the SPE/DOE Symposium on Improved Oil Recovery, Tulsa, OK, USA, 17–21 April 2004; Paper Number SPE-89474-MS.
33. Kumar, A.; Ozah, R.; Noh, M.; Pope, G.A.; Bryant, S.; Sepheerhoori, K.; Lake, L.W. Reservoir Simulation of CO₂ Storage in Deep Saline Aquifers. *SPE J.* **2005**, *10*, 336–348. [[CrossRef](#)]
34. Kanakaki, E.M.; Samnioti, A.; Gaganis, V. Enhancement of Machine-Learning-Based Flash Calculations near Criticality Using a Resampling Approach. *Computation* **2024**, *12*, 10. [[CrossRef](#)]
35. Bahrami, P.; Sahari Moghaddam, F.; James, L.A. A Review of Proxy Modeling Highlighting Applications for Reservoir Engineering. *Energies* **2022**, *15*, 5247. [[CrossRef](#)]

36. Mohaghegh, S.D. *Data-Driven Analytics for the Geological Storage of CO₂*; CRC Press: Boca Raton, FL, USA; Taylor & Francis Group: Boca Raton, FL, USA, 2018; ISBN 978-1-315-28081-3.
37. Arridge, S.R.; Kaipio, J.P.; Kolehmainen, V.; Schweiger, M.; Somersalo, E.; Tarvainen, T.; Vauhkonen, M. Approximation Errors and Model Reduction with an Application in Optical Diffusion Tomography. *Inverse Probl.* **2006**, *22*, 175–195. [[CrossRef](#)]
38. March, A.; Willcox, K. Provably Convergent Multifidelity Optimization Algorithm Not Requiring High-Fidelity Derivatives. *AIAA J.* **2012**, *50*, 1079–1089. [[CrossRef](#)]
39. Cozad, A.; Sahinidis, N.V.; Miller, D.C. Learning Surrogate Models for Simulation-Based Optimization. *AIChE J.* **2014**, *60*, 2211–2227. [[CrossRef](#)]
40. Gholami, V.; Mohaghegh, S.D.; Maysami, M. Smart Proxy Modeling of SACROC CO₂-EOR. *Fluids* **2019**, *4*, 85. [[CrossRef](#)]
41. Mohaghegh, S.D.; Abdulla, F.; Abdou, M.; Gaskari, R.; Maysami, M. Smart Proxy: An Innovative Reservoir Management Tool. In *Case Study of a Giant Mature Oilfield in the UAE*; OnePetro: Abu Dhabi, United Arab Emirates, 2015.
42. Mohaghegh, S.D.; Amini, S.; Gholami, V.; Gaskari, R.; Bromhal, G. Grid-Based Surrogate Reservoir Modeling (SRM) for Fast Track Analysis of Numerical Reservoir Simulation Models at the Grid Block Level. In Proceedings of the SPE Western Regional Meeting, Bakersfield, CA, USA, 19–23 March 2012; Society of Petroleum Engineers: Richardson, TX, USA, 2012; SPE-153844.
43. Amini, S.; Mohaghegh, S.D.; Gaskari, R.; Bromhal, G.S. Pattern Recognition and Data-Driven Analytics for Fast and Accurate Replication of Complex Numerical Reservoir Models at the Grid Block Level. In Proceedings of the SPE Intelligent Energy Conference and Exhibition, Utrecht, The Netherlands, 1–3 April 2014; Society of Petroleum Engineers: Richardson, TX, USA, 2014; SPE-167897-MS.
44. Yan, B.; Chen, B.; Harp, D.R.; Jia, W.; Pawar, R.J. A robust deep learning workflow to predict multiphase flow behavior during geological CO₂ sequestration injection and Post-Injection periods. *J. Hydrol.* **2022**, *607*, 127542. [[CrossRef](#)]
45. Amiri, B.; Jahanbani Ghahfarokhi, A.; Rocca, V.; Ng, C.S.W. Optimization of Offshore Saline Aquifer CO₂ Storage in Smeaheia Using Surrogate Reservoir Models. *Algorithms* **2024**, *17*, 452. [[CrossRef](#)]
46. Ismail, I.; Fotias, S.P.; Avgoulas, D.; Gaganis, V. Integrated Black Oil Modeling for Efficient Simulation and Optimization of Carbon Storage in Saline Aquifers. *Energies* **2024**, *17*, 1914. [[CrossRef](#)]
47. Lie, K.-A. *An Introduction to Reservoir Simulation Using MATLAB/GNU Octave: User Guide for the MATLAB Reservoir Simulation Toolbox (MRST)*; Cambridge University Press: Cambridge, UK, 2019.
48. Lie, K.-A.; Møyner, O. (Eds.) *Advanced Modeling with the MATLAB Reservoir Simulation Toolbox*; Cambridge University Press: Cambridge, UK, 2021.
49. Advanced Resources International, Inc. *Geologic Storage Capacity for CO₂ of The Lower Tuscaloosa Group and Woodbine Formations*; Advanced Resources International, Inc.: Arlington, VA, USA, 2009.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.