Closed-Loop CO_2 Storage Control With History-Based Reinforcement Learning and Latent Model-Based Adaptation

Sofianos Panagiotis Fotias [1,*], Vassilis Gaganis [1,2]

May 19, 2026

[1]School of Mining and Metallurgical Engineering, National Technical University of Athens, Athens 157 73, Greece
[2]Institute of Geoenergy, Foundation for Research and Technology, Chania, 73100, Greece
*Correspondence: sfotias@metal.ntua.gr

Abstract

Closed-loop management of geological CO_2 storage requires control policies that adapt to uncertain reservoir behavior while relying on observations that are realistically available during operation. This work formulates CO_2 injection and brine-production control as a partially observable sequential decision problem and studies deployable deep reinforcement-learning controllers trained with high-fidelity reservoir simulation. We first compare privileged-state, well-only, history-conditioned, masking-curriculum, and asymmetric teacher-student model-free policies in order to quantify the value of temporal well-response information and training-time privileged simulator states. We then evaluate a latent model-based adaptation pipeline that reuses nominal latent dynamics and retunes controllers under known injector failure, leakage-induced dynamics and reward shift, and compartmentalized reservoir connectivity. The results show that history-conditioned policies recover nearly all of the privileged-state performance while using only deployable well-level information, and that latent model-based retuning outperforms direct model-free retuning under the same scenario-specific real-simulator budget in the abnormal operating cases. The proposed framework therefore provides a simulator-budget-aware alternative to repeated online history matching and re-optimization for closed-loop CO_2 storage control.

Keywords: CCS; Deep Reinforcement Learning; Transformers; Partial Observability; Closed-Loop Optimization;

1 Introduction

Achieving net-zero emissions by mid-century requires rapid deployment of near-zero carbon technologies across sectors [1]. Carbon Capture, Utilization and Storage (CCUS) is widely considered a key option because it can mitigate emissions from hard-to-abate industries and enable deep reductions where alternatives are limited [2, 3]. In geological storage, captured CO_2 is injected into subsurface formations for permanent containment; suitable sites must provide adequate pore volume and injectivity, alongside a competent seal to prevent unwanted migration [4, 5]. Saline aquifers and depleted hydrocarbon fields are among the most mature storage candidates, supported by extensive project experience and monitoring data [6–11]. Despite this maturity, storage performance and containment remain strongly influenced by uncertain geology and nonlinear multiphase flow, motivating decision-making frameworks that can adapt as information becomes available.

Operational CCS planning and management rely heavily on numerical simulation to evaluate injectivity,

pressure buildup, plume evolution, and trapping mechanisms over the project life cycle [12–14]. Building on simulation-based forecasting, optimizing the operational schedule is essential for maximizing sequestration while meeting constraints and regulatory guidelines [15–18]. In practice, CCS strategies often aim to sustain high CO_2 injection while regulating pressure buildup through brine production; producers may be shut in after significant CO_2 breakthrough, and operations are typically curtailed when bottomhole pressure approaches safety limits [19–21]. In open systems with stratigraphic traps, pressure can also be relieved through brine migration to neighboring formations [22]. These operational choices define the main decision variables for subsurface management. In the present work, we focus on time-varying CO_2 injection and brine-production controls. Common objectives include maximizing stored CO_2, improving trapping quality (e.g., immobile-to-mobile ratios), and economic measures such as NPV, while respecting pressure and containment constraints. Since each candidate control schedule requires a fully implicit simulation, the resulting objectives are expensive, nonlinear, and naturally treated as black-box functions in constrained multi-objective optimization [23].

Classic reservoir optimization is often conducted in an open-loop manner, where a control schedule is optimized against a fixed digital twin of the subsurface [24]. This is limiting in CCS because the true system response is uncertain before operations begin, and decisions should ideally incorporate newly acquired monitoring data over time. Open-loop approaches can partially address uncertainty by optimizing over ensembles of geological realizations, but this increases computational cost substantially and can drive overly conservative solutions when constraints must hold across all models. Surrogate modeling and Bayesian optimization have been widely adopted to reduce simulator calls and quantify uncertainty in expensive black-box objectives [25–27], including explicit distinctions between aleatoric and epistemic uncertainty [28,29]. Nevertheless, the sequential nature of CCS operations motivates a closed-loop viewpoint in which controls are updated as new information arrives. Our previous work explored uncertainty-aware open-loop optimization for CCS through Bayesian optimization with permutation-invariant Gaussian-process surrogates [30]. The present study moves from that pre-operational setting to sequential closed-loop decision making.

Closed-loop reservoir management (CLRM) addresses sequential subsurface decision-making by assimilating newly acquired data, updating geological models, and re-optimizing controls over time [31–36]. Its practical use in CCS, however, is limited by the repeated cost of history matching and simulation-based optimization, especially when large ensembles of plausible geological models must be maintained [37–40]. In this work, we formulate closed-loop CO_2 storage control as a partially observable sequential decision problem. Policies are trained offline on prior digital-twin realizations and evaluated on a held-out target reservoir under information regimes that distinguish between privileged simulator fields and deployable well-level observations [41–43]. The first part of the paper compares model-free SAC controllers across observability regimes, from privileged-state access to well-history-based deployment. The second part introduces a latent model-based adaptation workflow for abnormal operating conditions, including injector-control loss, leakage-induced dynamics and reward shift, and compartmentalized connectivity. All abnormal-scenario comparisons are reported under explicit real-simulator interaction budgets.

2 Related Work

2.1 Closed-loop reservoir management (CLRM)

CLRM frameworks have been developed to enable sequential decision-making by repeatedly assimilating production data (history matching) and re-optimizing operational controls [31–36]. Within CLRM, production optimization has been addressed using derivative-free methods [44–47], gradient-based approaches [48–50], and ensemble-based formulations [51–53]. A persistent challenge is that history matching is

itself a difficult inverse problem [37–40] and must be repeated throughout operations, increasing the computational burden and expanding the ensemble of plausible models. A related extension is closed-loop field development (CLFD), where drilling decisions are optimized jointly with operational control [54–56].

2.2 Representative model selection and uncertainty propagation in CLRM

Because CLRM can produce large ensembles of updated models, several works have focused on selecting representative subsets that preserve key flow responses while reducing simulation cost. Shirangi and Durlofsky proposed low-dimensional representations that combine permeability-based and flow-based descriptors and studied subset-selection accuracy using differences in flow response [57]. They further introduced closed-loop optimization with sample validation, where the representative set is adaptively expanded if a validation criterion is not met [58]. For uncertainty propagation across multiple stages of reservoir modeling and development, Arnold et al. and Hutahaean et al. used neighborhood sampling methods (NAB) to sample posterior distributions of history-matched models, enabling uncertainty quantification for subsequent optimization steps [59–62].

2.3 Deep reinforcement learning for reservoir control

Deep reinforcement learning has been increasingly explored as an alternative to repeated online optimization by learning policies that map observations directly to control actions. Early work applied value-based and on-policy methods to subsurface control problems, including SARSA for SAGD [63] and Q-learning-based approaches for waterflooding [64,65]. Subsequent studies compared modern DRL algorithms against heuristic optimizers and investigated training across multiple geological realizations to account for uncertainty [66]. Policy-gradient methods have also been used with high-dimensional state representations (e.g., pressure/saturation images) [67, 68].

More CLRM-aligned workflows train agents on sets of prior models selected to represent geological uncertainty, and compare DRL against evolutionary strategies [69, 70]. Soft Actor-Critic (SAC) has become a common choice for continuous well-control problems [71], and convolutional architectures have been shown to be advantageous when dense spatial reservoir states are available as inputs [41]. Related representation-learning pipelines embed reservoir states into latent spaces (e.g., via VAEs) to enable policy learning and transfer to unseen but similar environments [72]. Hybrid evolutionary-RL frameworks have also been proposed to improve policy diversity and robustness [73].

2.4 Partial observability and sequence models for CLRM-like control

In practical CCS/CLRM settings, the control problem is naturally partially observable: operators primarily observe well measurements and sparse monitoring rather than full-field states. This motivates POMDP formulations and history-dependent policies that integrate temporal context. Nasir et al. explicitly formulated CLRM as a POMDP and introduced sequence-based policies using gated transformer blocks (GTrXL) to process observation histories, demonstrating improved control under partial observability relative to baseline CLRM strategies [42, 43]. Their subsequent work incorporated practical operational considerations through action and reward shaping and extended the approach to multi-asset settings [74, 75]. DRL has also been applied in closed-loop field development (CLFD), where sequential drilling decisions are optimized jointly with controls [76, 77].

3 Problem Formulation and Model-Free Methodology

3.1 Sequential CCS control problem

We consider closed-loop reservoir management for geological CO_2 storage as a finite-horizon sequential decision-making problem. At each control step, the agent selects injection and production controls, the

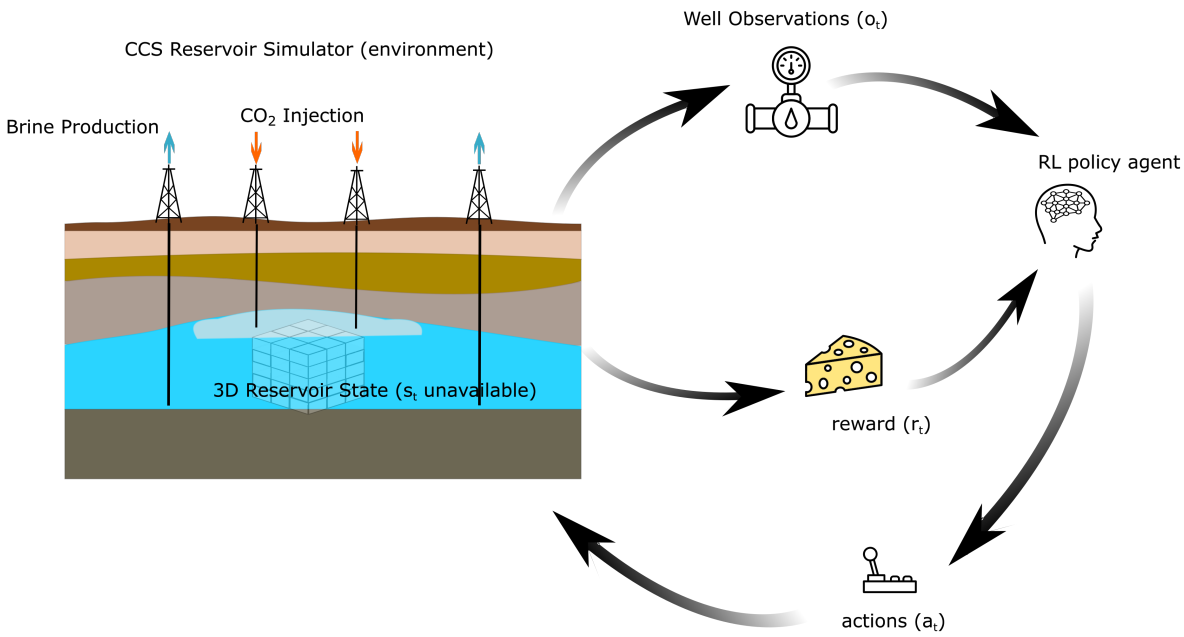

Figure 1: Closed-loop CCS control formulation considered in this work. The reservoir simulator evolves the latent subsurface state under the applied actions, while the agent receives only well-level observations and reward signals. The full spatial reservoir state remains latent to the deployed controller.

reservoir simulator advances the system dynamics, and new observations are returned for the next decision. The objective is to maximize storage-related operational performance while satisfying the physical and operational limits of the reservoir system.

In the present study, the action vector is continuous and contains eleven control variables,

$$a_t \in \mathbb{R}^{11}, \tag{1}$$

corresponding to the rate settings of eight producer wells and three CO_2 injector wells. The policy operates in a normalized action space, and each action component is subsequently mapped to its corresponding physical range before being written to the simulation deck. Producer controls are mapped to admissible production rate intervals, while injector controls are mapped to admissible gas injection rate intervals. This normalization allows all model variants to share the same action parameterization and learning setup.

Each decision is applied over one control interval of two years, implemented in the simulator through a time step of 730 days. An episode consists of 20 such control updates, giving rise to a long-horizon closed-loop management problem. After each control update, the simulator outputs are used both to construct the next observation and to evaluate the reward.

The reward function is designed to reflect the main operational trade-offs in CO_2 storage. At each decision step, we evaluate a storage-related cash-flow surrogate that rewards retained injected CO_2 while penalizing gas production and brine handling. In addition, a small bonus is assigned when the net storage rate remains within a desired operating window, emulating realistic CCS project needs. Denoting by r_t the resulting scalar reward, the learning objective is to maximize the expected return

$$J(\pi) = \mathbb{E}_\pi \left[\sum_{t=0}^{T-1} r_t \right], \tag{2}$$

where $T = 20$ is the episode length and the expectation is taken over simulator transitions and, when applicable, reservoir-model uncertainty. Table 1 summarizes the reward and constraint components used

Table 1: Reward and constraint components used by the simulator environment. The same base reward is used across the nominal settings; Scenario 2 adds the leakage penalty.

Component	Implementation role	Used in
Retained CO_2 value	Present-value reward based on injected minus produced gas	All scenarios
Produced-gas penalty	Penalizes CO_2 production/breakthrough through the cash-flow term	All scenarios
Brine-handling penalty	Penalizes produced brine treatment cost	All scenarios
Net-storage operating bonus	Adds a small bonus when sequestration rate lies in the target interval	All scenarios
Leakage penalty	Penalizes gas saturation above threshold in the leakage region	Scenario 2
Action bounds	Maps normalized actions to admissible producer/injector rate ranges	All scenarios
Simulator/deck limits	Enforces well and operational limits during OPM/Flow execution	All scenarios

by the environment. Figure 1 provides a schematic overview of the sequential CCS control loop, including the interaction between controls, simulator evolution, and available observations.

3.2 POMDP formulation

The CCS control problem is naturally formulated as a partially observable Markov decision process (POMDP),

$$\mathcal{P} = (\mathcal{S}, \mathcal{A}, \mathcal{O}, P, \Omega, R, \gamma), \tag{3}$$

where $s_t \in \mathcal{S}$ denotes the latent reservoir state, $a_t \in \mathcal{A}$ the control action, $o_t \in \mathcal{O}$ the observation available to the agent, $P(s_{t+1} \mid s_t, a_t)$ the simulator-induced transition kernel, $\Omega(o_t \mid s_t)$ the observation model, $R(s_t, a_t)$ the reward function, and γ the discount factor used by the reinforcement-learning algorithm.

The latent state contains the full physical information required to evolve the system under the numerical simulator, including pressure and saturation distributions together with the effect of the applied controls. In contrast, the agent does not necessarily observe this full state. In practical closed-loop operation, decisions must be made from sparse monitoring information, primarily well responses and limited field measurements, rather than from complete spatial flow fields. This distinction is central to the present work. To make this explicit, we define the policy input through an observation map

$$o_t = h(s_t), \tag{4}$$

and study multiple choices of $h(\cdot)$ corresponding to the information regimes defined below. Across all model-free variants, the action space, simulator dynamics, and reward definition are unchanged; only the policy information set differs. This isolates the effect of observability while retaining the same control task. Because training uses prior digital twins and evaluation uses the target reservoir, the experiments also include a train-test reality gap. Figure 2 summarizes the resulting observability and reality gaps.

3.3 Baseline extremes of observability

We use two endpoint baselines to define the observability range. The privileged-state benchmark is a diagnostic upper reference that uses dense target-reservoir simulator fields together with well observations. The well-only baseline is the most restrictive deployable controller and uses only the current well-observation tensor during training on prior digital twins and evaluation on the target reservoir. The later variants retain the deployable observation interface but add either temporal context or asymmetric privileged training signals.

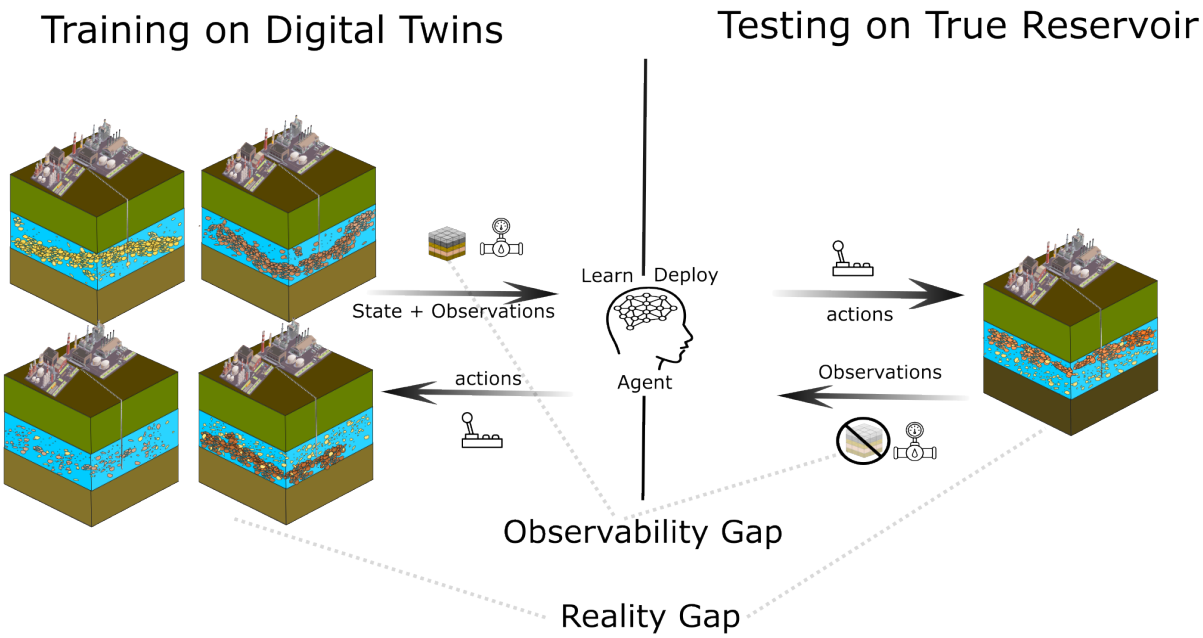

Figure 2: Training-deployment setting considered in this work. Policies are trained on ensembles of prior digital twins, whereas deployment takes place on the true reservoir using only practical field observations. This creates both a *reality gap*, due to mismatch between training models and the true system, and an *observability gap*, because dense simulator states available during training are not available during real operation.

3.3.1 Privileged-state benchmark

The privileged-state benchmark uses the target-reservoir observation

$$o_t^{\text{priv}} = \left(x_t^{\text{field}},\, y_t^{\text{well}}\right), \tag{5}$$

where x_t^{field} is the dense simulator-derived reservoir tensor and y_t^{well} is the most recent well-response summary. The spatial tensor is

$$x_t^{\text{field}} \in \mathbb{R}^{2\times 4\times 163\times 120}, \tag{6}$$

with channels corresponding to normalized pressure and a saturation-related field. The well-observation tensor is

$$y_t^{\text{well}} \in \mathbb{R}^{9\times 30}. \tag{7}$$

It contains nine intra-interval samples over the latest two-year control interval. Each sample contains the following well measurements: gas injection rate, gas breakthrough rate, water production rate and well bottomhole pressures. This intra-interval sampling provides a secondary time level within each control step, but it does not by itself provide memory across previous control decisions.

Architecturally, the privileged actor and critics use two observation branches. The spatial branch applies a 3D convolutional encoder to x_t^{field}, with four convolutional layers followed by global average pooling. The well branch applies a one-dimensional convolution over the nine intra-interval samples of y_t^{well}. The resulting spatial and well features are concatenated and passed through two fully connected layers before the SAC actor and critic heads. For the critics, the action is concatenated with the encoded observation before value estimation. A schematic is shown in Figure 3.

Table 2: Model-free information regimes considered in Section 3.

Variant	Policy input	Privileged training signal	Role
Privileged-state	Spatial fields + wells	Policy observes dense target state	Oracle upper reference
Well-only	Current well tensor	None	Sparse deployable baseline
History-conditioned	Well history + current wells	None	Tests temporal context
Masked-critic curriculum	Current wells	Critic receives masked spatial state	Tests critic-side privileged dependence
Teacher-student	Well history + current wells	Teacher critics / distillation	Privileged value supervision

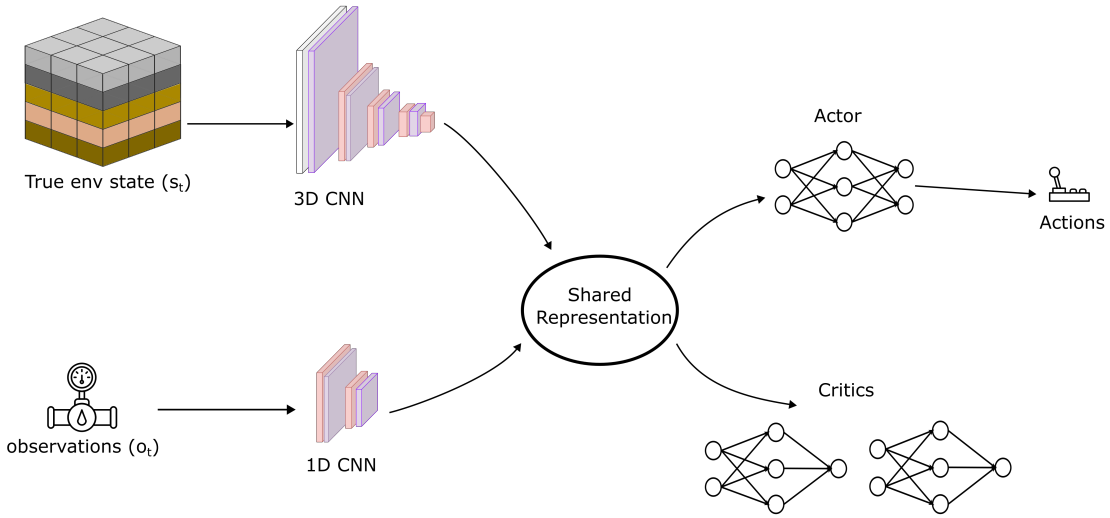

Figure 3: Baseline information regimes. The privileged-state benchmark uses dense simulator fields and well observations, whereas the well-only baseline removes the spatial branch and acts only from well-level observations.

3.3.2 Well-only baseline

The well-only baseline removes the dense simulator state and uses

$$o_t^{\text{well}} = y_t^{\text{well}}. \tag{8}$$

This controller is trained on prior digital-twin realizations and evaluated on the held-out target reservoir using only sparse well-level observations.

The actor and critics retain only the one-dimensional well-observation encoder and the final fully connected SAC heads. Because no spatial pressure/saturation tensor or inter-step memory is provided, this baseline isolates the difficulty of acting from the current well response alone.

3.4 History-conditioned sequence model

The well-only baseline uses only the most recent well-response tensor and is therefore limited in a partially observable setting. Similar current well measurements may correspond to different reservoir states depending on the preceding control sequence, pressure evolution, and displacement history. The history-conditioned variant addresses this limitation by conditioning the policy and critics on a finite history of public well observations.

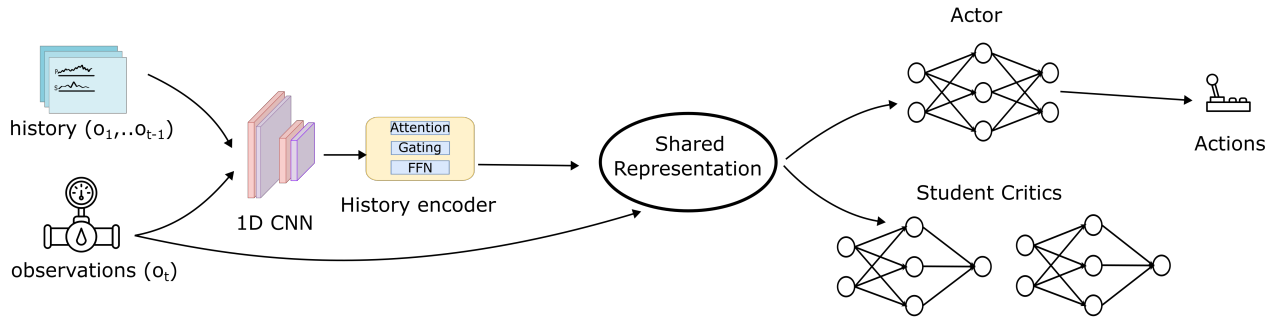

Figure 4: History-conditioned model. Current well observations and a rolling history of public well observations are encoded through a shared 1D convolutional front-end. The history embeddings are processed by a gated transformer-style sequence encoder, whose final output is fused with the current observation embedding and used by the SAC actor and twin critics.

The policy input is extended from

$$o_t^{\text{well}} = y_t^{\text{well}} \tag{9}$$

to

$$o_t^{\text{hist}} = \left(y_t^{\text{well}},\, H_t\right), \tag{10}$$

where

$$H_t = \left[y_{t-L+1}^{\text{well}}, \ldots, y_t^{\text{well}}\right] \in \mathbb{R}^{L\times 9\times 30} \tag{11}$$

is a rolling history of the most recent L decision-step observations. In this work, $L = 20$, equal to the episode horizon, so the controller can access the full sequence of well responses accumulated during an episode. At reset, the history buffer is initialized with zeros and is then updated after each simulator step.

Each well-observation block $y_t^{\text{well}} \in \mathbb{R}^{9\times 30}$ contains nine uniformly sampled intra-interval snapshots of the 30 well variables described previously. Rather than reprocessing the full raw history directly, each block is first mapped to a compact embedding by a shared one-dimensional convolutional encoder. In the implementation, the tensor is interpreted as 30 well-variable channels over nine intra-interval samples. A one-dimensional convolution with kernel size 3 and 64 output channels is followed by a ReLU activation and average pooling over the intra-interval dimension, giving

$$e_t = f_{\text{conv}}\left(y_t^{\text{well}}\right) \in \mathbb{R}^{64}. \tag{12}$$

Applying the same encoder to every element of the history gives a sequence of embeddings

$$E_t = [e_{t-L+1}, \ldots, e_t] \in \mathbb{R}^{L\times 64}. \tag{13}$$

Temporal dependencies across control steps are modeled using a gated transformer-style encoder applied to the explicitly stored history tensor. Importantly, this module is not used as a stateful recurrent policy: no hidden state is carried across environment steps or through the replay buffer. Instead, each replay-buffer sample already contains the finite history H_t, and the sequence encoder deterministically maps this history to a control feature. For an input embedding sequence E_t, the attention block computes multi-head self-attention over the decision-step history,

$$A_t = \text{MHA}(\text{LN}(E_t))\,. \tag{14}$$

The attention output is then combined with the incoming representation through a GRU-gated residual connection,

$$\tilde{E}_t = \text{GRUGate}(E_t, A_t), \tag{15}$$

rather than by a standard additive residual connection. Here, the GRU cell is used only as a gating mechanism for the residual pathway, not as an external recurrent memory. The same gated residual structure is applied to the feedforward block,

$$\bar{E}_t = \mathrm{GRUGate}\left(\tilde{E}_t,\, \mathrm{FFN}\Big(\mathrm{LN}(\tilde{E}_t)\Big)\right). \tag{16}$$

The feedforward module uses a hidden dimension of 256 and maps back to the 64-dimensional embedding space. In the present implementation, one gated transformer-style block is used, with four attention heads of dimension 16. The final history feature is taken as the last output of the sequence encoder,

$$h_t = \bar{E}_{t,L} \in \mathbb{R}^{64}. \tag{17}$$

This design should therefore be interpreted as finite-history sequence encoding for off-policy SAC rather than as recurrent-state learning. The temporal information is supplied by the environment through H_t, while the gated transformer-style encoder learns how to summarize that finite observation history.

The current observation is also encoded separately using the same convolutional encoder, giving $e_t^{\mathrm{cur}} = f_{\mathrm{conv}}(y_t^{\mathrm{well}})$. The actor representation is then formed by concatenating the current and history features,

$$\phi_t^{\mathrm{hist}} = [e_t^{\mathrm{cur}},\, h_t]\,, \tag{18}$$

and passing the resulting 128-dimensional vector through two fully connected layers of width 256. The actor outputs the mean and standard deviation of the Gaussian SAC policy. The twin critics use the same history-conditioned representation, concatenated with the action, before estimating the corresponding state-action values. Because the encoder uses only $(H_t, y_t^{\mathrm{well}})$, the policy preserves the deployable observation interface defined in Section 3.2. Figure 4 summarizes the architecture.

3.5 Curriculum masking of privileged state information

The masking-curriculum variant keeps the actor well-only but allows the critics to use a spatial simulator tensor whose information content is progressively reduced during training. This creates an asymmetric actor-critic diagnostic: the policy remains deployable, while the value estimators are tested under decreasing access to privileged state information. Figure 5 summarizes the setup.

As in the previous variants, the environment returns a dense spatial reservoir tensor together with a well-observation tensor,

$$o_t = \left(x_t^{\mathrm{field}},\, y_t^{\mathrm{well}}\right), \tag{19}$$

with

$$x_t^{\mathrm{field}} \in \mathbb{R}^{2\times 4\times 163\times 120}, \qquad y_t^{\mathrm{well}} \in \mathbb{R}^{9\times 30}. \tag{20}$$

Before the spatial tensor is used by the learning algorithm, however, a random masking operator is applied:

$$\tilde{x}_t^{\mathrm{field}} = M_p\left(x_t^{\mathrm{field}}\right), \tag{21}$$

where $M_p(\cdot)$ denotes stochastic masking with ratio $p \in [0, 1]$. Specifically, a fraction p of the nonzero entries of the spatial tensor is selected uniformly at random and set to zero. Hence, $p = 0$ corresponds to the unmodified spatial state, whereas $p = 1$ corresponds to full masking, i.e., a zero spatial tensor. The well observations remain unchanged throughout.

A key design choice in this experiment is that the actor and critics do not receive the same inputs. The actor is intentionally restricted to well observations only and is therefore parameterized as

$$\pi_\theta(a_t \mid y_t^{\mathrm{well}}), \tag{22}$$

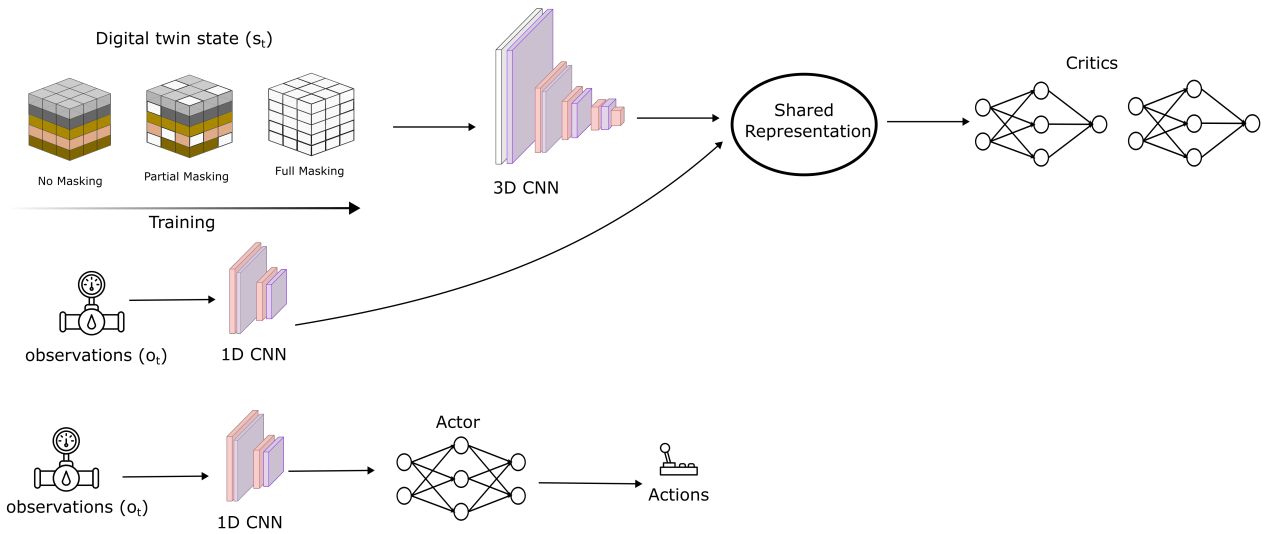

Figure 5: Masked-critic curriculum. During training, the critic receives progressively masked spatial simulator information, ranging from no masking to full masking, while the actor remains restricted to public well observations throughout. This yields an asymmetric actor-critic setup that gradually transitions from privileged to fully deployable training.

using the same one-dimensional convolutional encoder employed in the well-only baseline. In contrast, both critics receive the masked spatial tensor together with the well observations and action,

$$Q_{\phi_i}\left(\tilde{x}_t^{\text{field}}, y_t^{\text{well}}, a_t\right), \qquad i \in \{1, 2\}, \tag{23}$$

and process the spatial component through a 3D convolutional encoder before combining it with well features and the action. This yields an asymmetric actor-critic setting: the policy remains deployable from well data alone, while the critics are allowed to exploit additional spatial information during training, albeit in progressively degraded form.

The masking ratio is not fixed, but instead follows a curriculum over training epochs,

$$p \in \{0.00,\ 0.25,\ 0.50,\ 0.75,\ 1.00\}. \tag{24}$$

Training begins with $p = 0$, so that the critics observe the full privileged spatial state. The masking level is then increased in stages until the critics are trained with fully masked spatial input. Importantly, the replay buffer is reset whenever the masking ratio changes. This prevents transitions collected under one observability regime from being mixed with transitions collected under another and thereby reduces distribution mismatch during off-policy updates. During evaluation, the masking ratio is fixed at

$$p_{\text{test}} = 1.0, \tag{25}$$

so that test-time performance reflects the fully deployable regime in which no spatial simulator field is available.

We use this curriculum as a diagnostic stress test of critic-side information dependence. Since the actor is always restricted to y_t^{well}, the deployment interface does not change across masking levels. The experiment therefore measures whether progressively removing privileged critic information destabilizes learning or whether well-based features are sufficient to support the deployable actor.

3.6 Asymmetric teacher-student sequence model

The final model-free variant combines history-based public control, asymmetric critic learning, and auxiliary latent distillation from privileged simulator states. Unlike the masking curriculum, privileged infor-

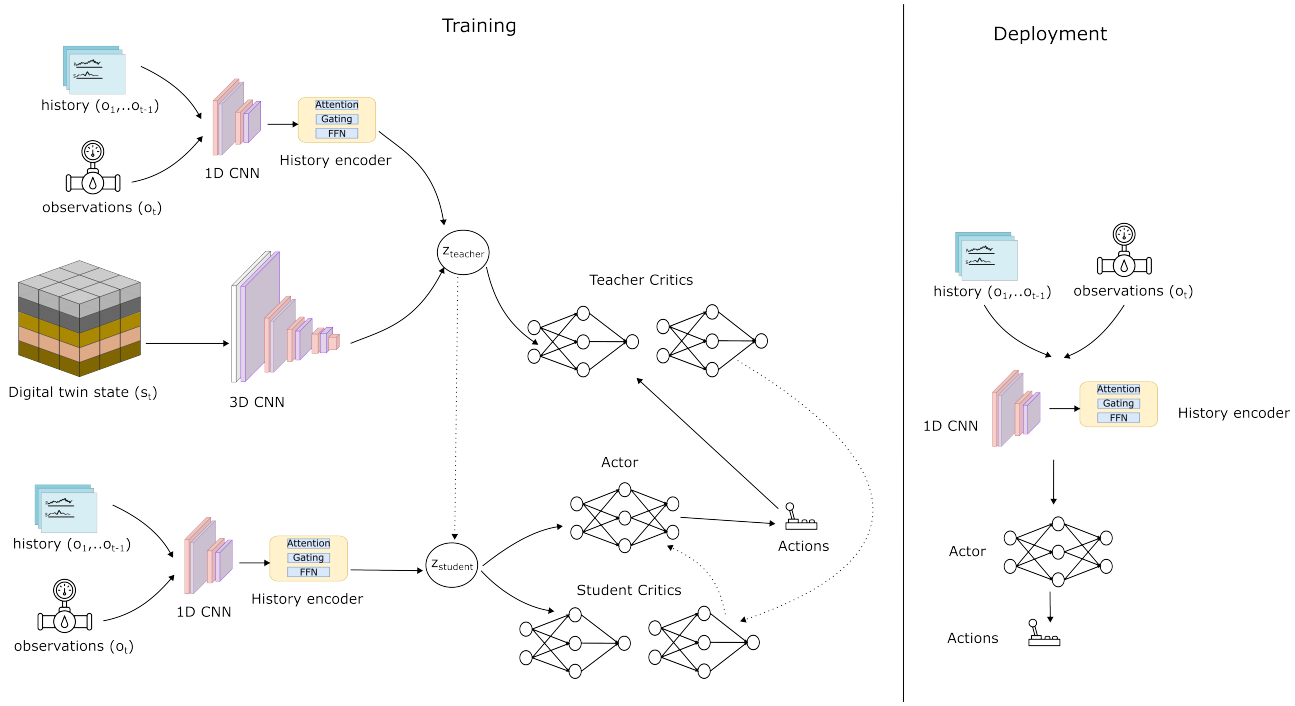

Figure 6: Asymmetric teacher-student model. During training, privileged teacher critics use dense spatial simulator fields together with public well-history information. Student critics and the actor use only public current and historical well observations. The actor is updated through the student critics, while an auxiliary distillation loss encourages the public history encoder to capture information contained in the privileged spatial branch. At deployment, only the public history encoder and actor are used.

mation is not gradually removed; it is confined to teacher critics, while the actor and student critics use only public well-history observations. Figure 6 summarizes the setup.

At each decision step, the simulator provides

$$o_t = \left(x_t^{\text{field}},\, y_t^{\text{well}},\, H_t\right), \tag{26}$$

where x_t^{field} is the dense spatial simulator state, y_t^{well} is the current well-observation tensor, and H_t is the rolling history of public well observations. The public observation available to the deployable policy is

$$o_t^{\text{pub}} = \left(H_t,\, y_t^{\text{well}}\right), \tag{27}$$

whereas the privileged training observation used by the teacher critics is

$$o_t^{\text{teach}} = \left(x_t^{\text{field}},\, H_t,\, y_t^{\text{well}}\right). \tag{28}$$

The actor uses the same finite-history idea introduced in the previous subsection, but with a larger latent dimension. A shared one-dimensional convolution maps each 9×30 well-observation block to a d-dimensional embedding, with $d = 128$ in the present implementation. The sequence of historical embeddings is processed by a gated transformer-style encoder. The final history embedding is then combined with the current-observation embedding by addition,

$$\phi_t^{\text{pub}} = f_{\text{hist}}\left(H_t, y_t^{\text{well}}\right) \in \mathbb{R}^{128}. \tag{29}$$

This public feature is passed to a policy head that outputs the mean and standard deviation of the Gaussian SAC policy. The two teacher critics receive the privileged observation and action. Each teacher

critic has a spatial branch and a public-history branch. The spatial branch maps the dense simulator tensor to a privileged latent vector,

$$z_{t,i}^{\text{teach}} = f_{3D,i}\left(x_t^{\text{field}}\right), \tag{30}$$

while the public-history branch maps the well-history observation to

$$h_{t,i}^{\text{teach}} = f_{\text{hist},i}^{Q}\left(H_t, y_t^{\text{well}}\right). \tag{31}$$

The teacher critic value is then computed as

$$Q_i^{\text{teach}}\left(o_t^{\text{teach}}, a_t\right) = q_i^{\text{teach}}\left(z_{t,i}^{\text{teach}}, h_{t,i}^{\text{teach}}, a_t\right), \qquad i \in \{1,2\}. \tag{32}$$

Thus, the teacher critics can exploit dense pressure/saturation information when constructing value estimates during training.

In parallel, two student critics are trained using only public inputs,

$$Q_i^{\text{stud}}\left(o_t^{\text{pub}}, a_t\right) = q_i^{\text{stud}}\left(f_{\text{hist},i}^{\text{stud}}\left(H_t, y_t^{\text{well}}\right), a_t\right), \qquad i \in \{1,2\}. \tag{33}$$

Only the teacher critics use x_t^{field}; the actor and student critics share the public observation interface o_t^{pub}. This is important because the actor update is computed through the student critics, not through the privileged teacher critics. As a result, privileged information can improve the training target, but the policy gradient remains tied to value estimates that are realizable from deployment-time observations.

Training proceeds as follows within each SAC update. First, the teacher critics are updated using the standard SAC critic loss on privileged observations. Then, a teacher TD target is constructed using the target teacher critics and the next action sampled from the public actor:

$$y_t^{\text{teach}} = r_t + (1-d_t)\gamma\left[\min_{i\in\{1,2\}} Q_{i,\text{target}}^{\text{teach}}\left(o_{t+1}^{\text{teach}}, a_{t+1}\right) - \alpha \log \pi\left(a_{t+1} \mid o_{t+1}^{\text{pub}}\right)\right]. \tag{34}$$

The student critics are then trained to regress to this privileged target:

$$\mathcal{L}_Q^{\text{stud}} = \sum_{i=1}^{2}\left\|Q_i^{\text{stud}}\left(o_t^{\text{pub}}, a_t\right) - y_t^{\text{teach}}\right\|_2^2. \tag{35}$$

Finally, the actor is updated using the student critics:

$$\mathcal{L}_\pi = \mathbb{E}\left[\alpha \log \pi\left(a_t \mid o_t^{\text{pub}}\right) - \min_{i\in\{1,2\}} Q_i^{\text{stud}}\left(o_t^{\text{pub}}, a_t\right)\right]. \tag{36}$$

This update structure separates the source of the training target from the source of the actor gradient: teacher critics provide privileged TD targets, student critics approximate those targets from public observations, and the actor learns through the public student branch.

An auxiliary distillation loss is added to encourage the public history representation to contain information correlated with the privileged spatial latent. The student distillation pathway shares the actor history encoder and applies a distillation head to produce

$$z_t^{\text{stud}} = g_{\text{dist}}\left(f_{\text{hist}}\left(H_t, y_t^{\text{well}}\right)\right). \tag{37}$$

The target latent is obtained from the spatial encoder of the first teacher critic,

$$z_t^{\text{teach}} = f_{3D,1}\left(x_t^{\text{field}}\right), \tag{38}$$

with gradients stopped through the teacher branch. In the present implementation, projected and normalized student and teacher latents are matched using an InfoNCE-style contrastive loss,

$$\mathcal{L}_{\mathrm{NCE}} = -\frac{1}{B}\sum_{b=1}^{B}\log\frac{\exp\left(\langle p_s(z_b^{\mathrm{stud}}), p_t(z_b^{\mathrm{teach}})\rangle/\tau\right)}{\sum_{j=1}^{B}\exp\left(\langle p_s(z_b^{\mathrm{stud}}), p_t(z_j^{\mathrm{teach}})\rangle/\tau\right)}, \tag{39}$$

where p_s is trainable, p_t is kept frozen, and τ is the contrastive temperature. A scalar value-alignment head is also trained from the student latent to match the detached student-side value estimate. The auxiliary distillation objective is therefore

$$\mathcal{L}_{\mathrm{dist}} = \mathcal{L}_{\mathrm{NCE}} + \beta\mathcal{L}_{\mathrm{val}}. \tag{40}$$

The total actor-side loss used when distillation is active is

$$\mathcal{L}_{\mathrm{actor,total}} = \mathcal{L}_{\pi} + w_{\mathrm{dist}}(e)\mathcal{L}_{\mathrm{dist}}, \tag{41}$$

where $w_{\mathrm{dist}}(e)$ is linearly annealed over the final part of training. In the reported configuration, distillation is ramped from epoch 22 to epoch 30 with final weight 0.1. At test time, only the public history encoder and policy head are used; teacher critics, student critics, spatial encoders, and distillation heads are discarded.

This variant therefore combines the public history interface of Section 3.4 with asymmetric privileged value supervision, making it the most structured deployable model-free controller considered here.

4 Latent model-based adaptation

Section 3 considered deployable model-free controllers trained directly with high-fidelity simulator interaction. We now test whether a latent model-based controller can retain similar real-simulator performance while reducing the amount of scenario-specific simulator interaction needed for adaptation. The fixed pipeline maps public well-history observations to a latent control state, trains a latent world model on simulator-derived transitions, and updates a latent actor-critic pair through Dreamer-style imagined rollouts.

Within this fixed pipeline, we consider four latent world-model backbones, described in the next subsection: a probabilistic GRU, a geometrically regularized GRU, an RSSM, and a Koopman-inspired model. These backbones are evaluated under the same overall training and control framework across all scenarios. A schematic overview of the full model-based pipeline is shown in Figure 7.

4.1 Latent world-model backbones

The role of the world model is to predict the next deployable latent state and reward from the current latent state and action. Let z_t denote the latent state obtained from the public history encoder and a_t the normalized control action. Each world model approximates

$$(z_t, a_t) \mapsto (z_{t+1}, r_t), \tag{42}$$

and is trained on latent transition tuples extracted from prior simulator interaction. The same trained world model is then used to generate imagined rollouts for actor-critic learning in latent space.

We evaluate four world-model backbones, each representing a different inductive bias about the latent dynamics. The first is a probabilistic GRU model. It uses a recurrent hidden state to summarize recent latent-action history and predicts a diagonal Gaussian distribution over the next latent state together with a deterministic reward estimate:

$$h_{t+1} = \mathrm{GRU}(z_t, a_t, h_t), \qquad z_{t+1} \sim \mathcal{N}(\mu_t, \sigma_t^2), \qquad \hat{r}_t = r_\theta(h_{t+1}). \tag{43}$$

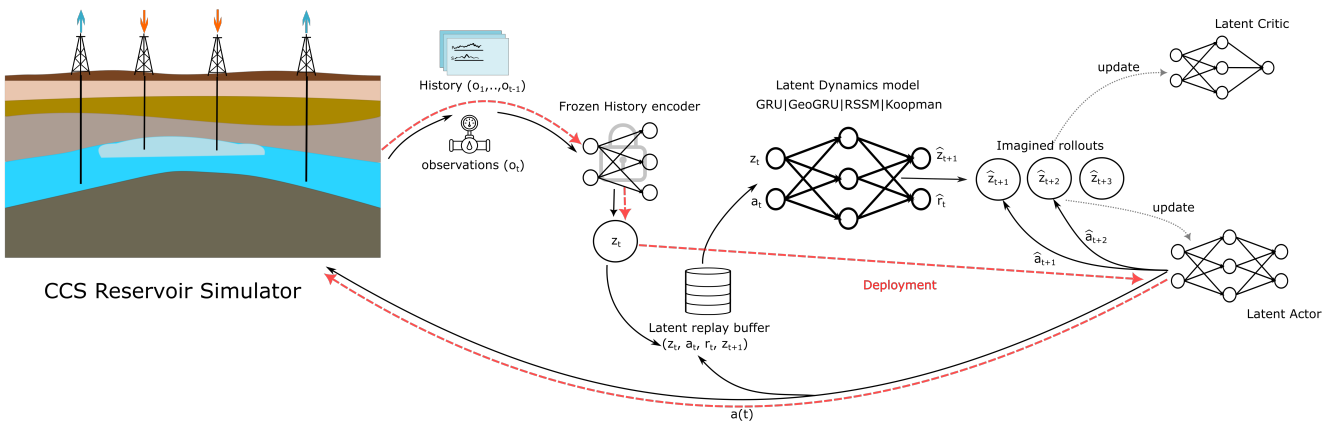

Figure 7: Model-based pipeline used in this work. Public observations are mapped to a deployable latent state through a frozen history encoder, latent transitions are stored in a replay buffer, and a latent dynamics model is trained to support Dreamer-style imagined rollouts for the latent actor and critic. The same latent actor is also deployed back in the real CCS simulator.

This model is flexible and can represent path-dependent latent dynamics, but imagined rollouts may accumulate uncertainty over long horizons.

The second backbone is a geometrically regularized GRU. It uses the same probabilistic GRU transition model, but augments the latent prediction loss with a decoder-induced geometric term. The purpose is to penalize latent errors according to their effect after decoding back toward spatial reservoir fields, rather than treating all latent directions as equally important. This introduces a physically motivated bias: errors that correspond to larger decoded pressure/saturation discrepancies are weighted more strongly. The trade-off is that this model requires a trained decoder and additional computation during world-model training.

The third backbone is a recurrent state-space model (RSSM). It separates the latent dynamics into a deterministic recurrent state and a stochastic state. The deterministic component propagates information through a GRU, while prior and posterior stochastic variables are trained with a KL-regularized reconstruction objective:

$$h_{t+1} = f_\theta(h_t, z_t, a_t), \qquad z^s_{t+1} \sim p_\theta(z^s_{t+1} \mid h_{t+1}), \tag{44}$$

with a posterior conditioned on the observed next latent during training. This structure is closer to Dreamer-style world modeling and is intended to improve multi-step imagination under uncertainty, although it introduces additional modeling complexity and KL balancing.

The fourth backbone is a Koopman-inspired model. It assumes that latent dynamics can be approximated by a controlled linear transition plus a small nonlinear residual:

$$\hat{z}_{t+1} = Az_t + Ba_t + \epsilon_\theta(z_t, a_t). \tag{45}$$

This model is less expressive than the recurrent alternatives, but it provides a stabilizing bias for imagined rollouts because most of the transition is constrained to be linear in latent space. The residual term preserves some flexibility for nonlinear deviations.

All four backbones are trained under the same offline procedure and are used within the same Dreamer-style control loop. Recurrent models are trained on fixed-length latent sequences, whereas the Koopman-inspired model is trained on one-step transitions. In all cases, an ensemble of world models is used, and imagined next states are sampled from one ensemble member during rollout. The comparison is therefore not intended as a hyperparameter search over world models, but as a test of how different latent-dynamics assumptions affect control retention and adaptation.

4.2 Scenario 0: nominal model-based retention of model-free performance

Scenario 0 is the nominal retention test. No abnormality is introduced, the control interface is unchanged, and the reward is the same as in the model-free experiments. The purpose is to determine whether the fixed latent model-based pipeline of Figure 7 can preserve the real-simulator performance of the deployable history-based model-free reference. We evaluate two public latent sources: the student history encoder obtained after teacher-student training, and the same encoder after post-hoc distillation toward the privileged spatial encoder. These are referred to as pre-finetuning and post-finetuning student latents, respectively.

The world model is pretrained offline on public latent transition tuples

$$(z_t, a_t, r_t, z_{t+1}), \tag{46}$$

extracted from prior simulator interaction. Recurrent backbones are trained on fixed-length latent sequences, whereas the Koopman-inspired model is trained on one-step transitions. After offline pretraining, the world model is inserted into a latent Dyna-style training loop with imagined rollouts. At each epoch, the latent actor collects a limited amount of fresh real simulator experience from training digital twins, encoded through the frozen public history encoder. These real latent transitions are used to continue updating the world model. The actor and critic are then trained using imagined trajectories generated by the latent dynamics model from sampled latent start states.

Evaluation is performed by deploying the learned latent actor back into the high-fidelity OPM/Flow simulator through the same public encoder. Real transitions used during model-based training are collected only from training digital-twin realizations, while performance is evaluated on the held-out target deck. Thus, Scenario 0 preserves the same train/test logic as the model-free study: prior realizations are used for learning, and the target environment is used for evaluation. Scenario 0 is therefore a viability benchmark: the world model need not be an exact surrogate of the full simulator, but it must be accurate enough in the public student latent space to support imagined rollouts that preserve real-simulator control performance.

4.3 Abnormal-scenario retuning protocol

Scenarios 1-3 evaluate adaptation after the nominal setting changes. In each abnormal scenario, the comparison is between two retuning strategies under the same scenario-specific real-simulator interaction budget. The first is direct model-free retuning in the abnormal environment. The second is latent model-based retuning, which reuses the nominal latent structure and updates the controller through a combination of limited real abnormal transitions and imagined rollouts. The abnormal scenarios differ only in the source of mismatch: a known loss of injector control authority, leakage-induced dynamics and reward shift, or changed reservoir connectivity.

For Scenarios 2 and 3, which involve abnormal physics rather than a known action clamp, we use a common residual-adapter protocol. A small abnormal-transition dataset is collected from the modified environment and encoded with the frozen public history encoder, giving tuples (z_t, a_t, r_t, z_{t+1}). The nominal Scenario 0 world model is frozen and used as prior structure. A residual adapter then corrects the nominal next-latent and reward predictions,

$$\hat{z}_{t+1}^{\text{corr}} = \hat{z}_{t+1}^{\text{nom}} + \Delta z_\psi, \qquad \hat{r}_t^{\text{corr}} = \hat{r}_t^{\text{nom}} + \Delta r_\psi. \tag{47}$$

The corrected model is inserted into the latent Dyna-style retuning loop, where the actor is warm-started from the nominal deployable controller and updated using corrected imagined rollouts together with the limited abnormal real-transition budget.

4.4 Scenario 1: known loss of one injector control channel

Scenario 1 introduces an operational abnormality in which one injector control channel is no longer available for optimization. The reservoir physics and reward definition remain the same as in the nominal

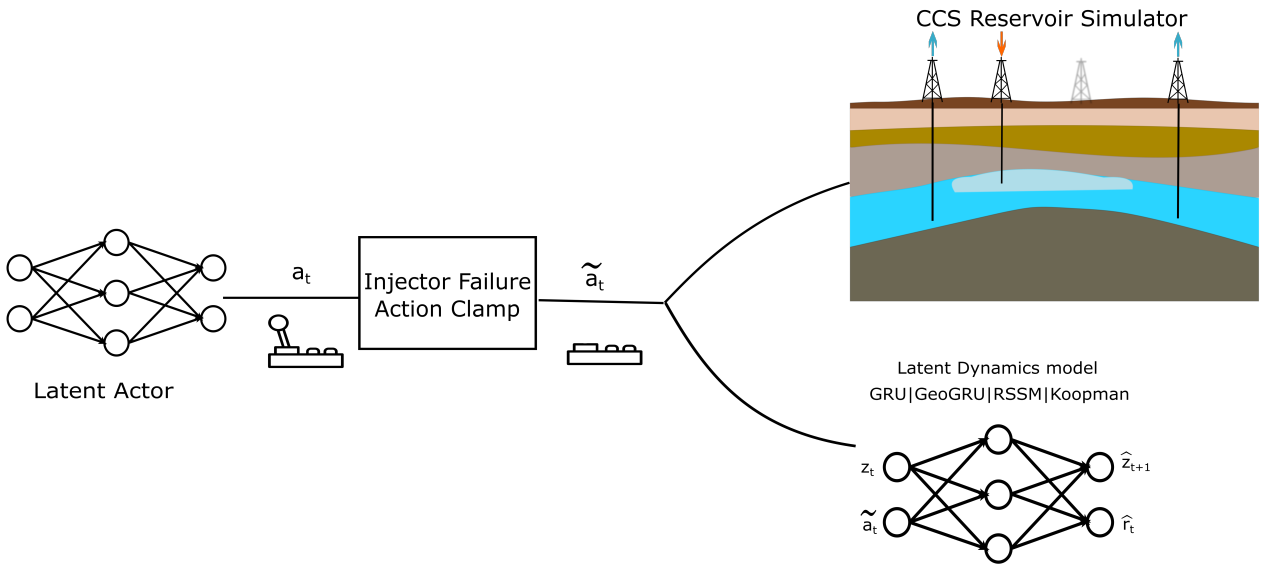

Figure 8: Scenario 1 methodology. The actor outputs an 11-dimensional nominal action, after which the failed-injector mask fixes the I_3 control component to its minimum admissible value. The same masked action is used in real simulator interaction, latent world-model updates, imagined rollouts, and evaluation, keeping the real and imagined failed-control regimes consistent.

case, but the controller loses authority over injector I_3. This scenario therefore tests adaptation to a known degradation of the control interface, rather than adaptation to an unknown geological or physical mismatch.

The original 11-dimensional action parameterization is retained for compatibility with the nominal policy and latent world models. The failed control channel is handled by imposing a deterministic action mask on the last action component, corresponding to injector I_3. In normalized action space, this component is fixed to

$$a_{t,10} = -1, \tag{48}$$

which maps to the minimum admissible physical injection rate after unnormalization. The same convention is enforced in the simulator deck construction, where the injection rate of I_3 is overwritten by its lower bound. Thus, the policy may still output an 11-dimensional action, but the failed coordinate is clamped before the action is used by either the real simulator or the latent dynamics model.

For model-based retuning, the same action mask is applied during real abnormal data collection, latent world-model updates, imagined rollouts, and evaluation. This keeps the real and imagined failed-control regimes consistent. The scenario therefore asks whether latent model-based retuning can make better use of limited abnormal-regime simulator interaction than direct model-free retuning when the actuator loss is known and can be imposed explicitly.

4.5 Scenario 2: leakage-induced dynamics and reward shift

Scenario 2 considers an abnormal-physics setting in which the nominal reservoir assumptions no longer describe the target environment. The action space and deployable observation interface remain unchanged, but the simulator is modified to represent a leakage pathway. As a result, the controller faces both a transition-dynamics shift and a reward shift. The public observations are still given by the current well responses and their rolling history, but the reward now includes an additional leakage penalty based on gas saturation in a designated leakage region. Once the leakage indicator exceeds a prescribed threshold, the reward is reduced accordingly.

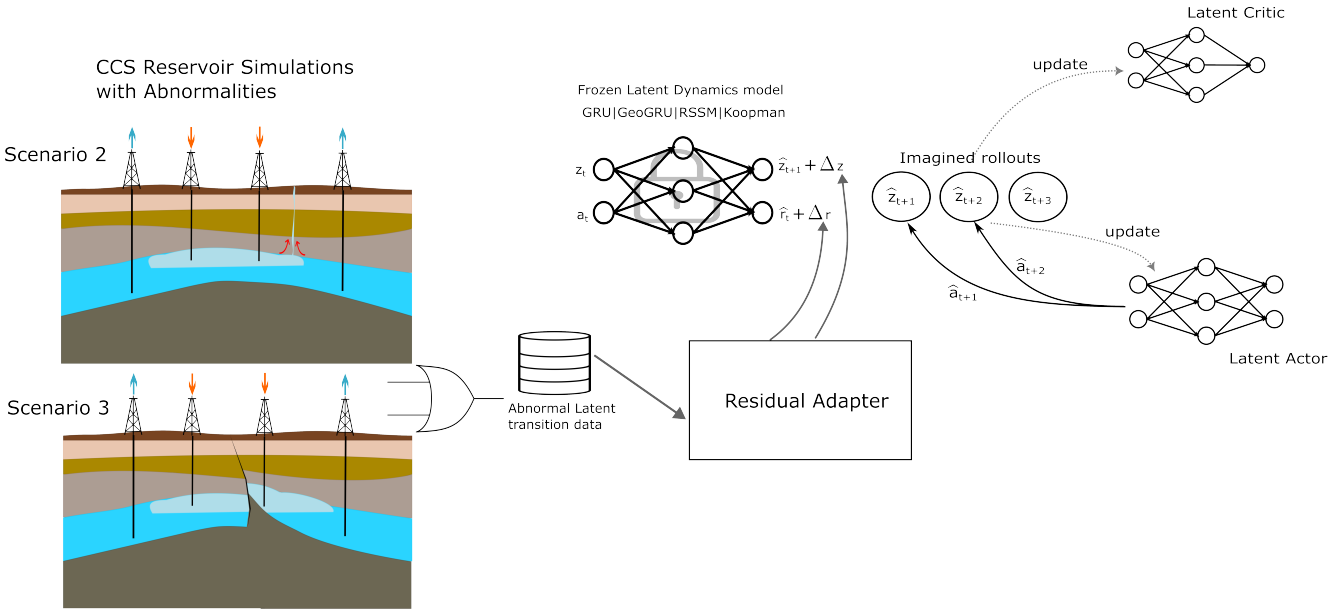

Figure 9: Residual world-model adaptation for Scenarios 2 and 3. Abnormal latent transitions are encoded using the frozen public history encoder. A residual adapter corrects nominal next-latent and reward predictions, and the corrected model is used for imagined rollouts during latent actor-critic retuning.

Unlike Scenario 1, the abnormality is not a known action clamp that can be imposed exactly in both real and imagined interaction. Instead, the nominal latent dynamics model has become partially incorrect: the same public latent state and action may now lead to different latent transitions and rewards because leakage changes the underlying simulator response and the operational objective.

Scenario 2 applies the residual-adapter protocol of Section 4.3 to a leakage setting. Here the adapter must correct both latent transition dynamics and reward predictions because leakage changes the simulator response and introduces an additional penalty term.

Scenario 2 therefore tests residual model-based adaptation under a combined dynamics-and-reward shift.

4.6 Scenario 3: compartmentalized reservoir with changed connectivity

Scenario 3 considers a second abnormal-physics setting, in which the nominal reservoir model is assumed to be overly connected. Under the revised interpretation, the reservoir is more compartmentalized: communication between some injector-producer pairs is reduced because transmissibility across compartments is strongly restricted. The action space, reward definition, and deployable observation interface remain unchanged, but the same control sequence now produces different pressure propagation and flow responses. Thus, unlike Scenario 2, the main abnormality is a transition-dynamics shift rather than a combined dynamics-and-reward shift.

Scenario 3 uses the same residual-adapter protocol defined in Section 4.3. Since the reward definition is unchanged, the adapter primarily corrects latent transition dynamics induced by the revised connectivity pattern; reward correction is retained only for consistency with the general adapter formulation. It therefore tests residual model-based adaptation under a structural connectivity shift with an unchanged reward.

Table 3: Simulator-interaction budgets used in the reported comparisons. Real steps denote high-fidelity simulator control steps; imagined steps are latent model rollouts and do not require simulator calls.

Study	Controller family	Epochs	Real steps/epoch	Total real steps	Imagined steps/epoch	Test episodes/epoch
Model-free observability endpoints	Privileged / well-only SAC	30	6000	180000	–	1
Model-free deployable variants	History, masking, teacher-student SAC	30	3000	90000	–	1
Scenario 0 nominal model-based	Latent Dreamer-style control	20	200	4000	4000	1
Scenarios 1–3 model-based retuning	Latent Dreamer-style retuning	20	40	800	3200	1
Scenarios 1–3 model-free retuning	Direct SAC retuning	20	40	800	–	1

Table 4: Model-free controller performance under different observability regimes. Best and final values are evaluation returns from the TensorBoard logs.

Method	Deployment input	Privileged training signal	Deployable	Best return	Final return
Privileged-state benchmark	Spatial state + wells	Policy observes spatial state	No	19.612	19.604
Well-only baseline	Current well tensor	None	Yes	18.816	18.345
History-conditioned model	Well history + current wells	None	Yes	19.652	19.622
Masked-critic curriculum	Current wells	Critic receives gradually masked spatial state	Yes	19.619	18.948
Teacher-student model	Well history + current wells	Privileged teacher critics / distillation	Yes	19.620	19.616

5 Results

We present the results in two stages. First, we compare the model-free controllers under different observability regimes in order to quantify the value of temporal context and privileged training information. Second, we examine whether the model-based pipeline can (i) retain the performance of the deployable model-free reference under nominal conditions and (ii) adapt more effectively than direct model-free retuning under abnormal operating conditions. For the abnormal-scenario figures, epoch 0 corresponds to the nominal controller evaluated in the abnormal environment before any scenario-specific updates.

Because each training run requires repeated high-fidelity OPM/Flow simulations, the reported curves and tables should be interpreted as simulator-budget-controlled comparisons rather than full random-seed sensitivity studies. Table 3 reports the real-simulator and imagined-rollout budgets used in each experiment. For Scenarios 1-3, direct model-free retuning and latent model-based retuning use the same scenario-specific real-simulator interaction budget; the model-based variants additionally reuse nominal latent structure learned before the abnormality is introduced, which is part of the proposed adaptation mechanism rather than an additional abnormal-scenario data source.

For Scenarios 1-3, each subsection discusses the scenario-level behavior, while Table 6 reports the best final model-based variant against direct model-free retuning and Appendix A gives the full per-variant values.

5.1 Model-free results under varying observability

Figure 10 compares the test performance of the five model-free variants. The corresponding best and final evaluation returns are reported in Table 4. The well-only baseline is clearly the weakest and most unstable controller, confirming that decisions based only on the most recent well-response tensor are too information-limited for robust closed-loop control. In contrast, the privileged-state benchmark rapidly reaches high performance and remains stable, as expected when dense simulator fields are available to the policy.

The key result, however, is that the history-conditioned model closes almost all of the gap to the privileged benchmark while remaining fully deployable. Its performance rises steadily and stabilizes in the same high-return regime as the privileged model. This shows that much of the information lost under partial observability can be recovered from temporal structure in the public observation stream alone. The teacher-student model reaches essentially the same final performance band and does so with similarly smooth learning dynamics, indicating that asymmetric training with privileged critics can yield a deployable controller that remains competitive with the strongest history-based model.

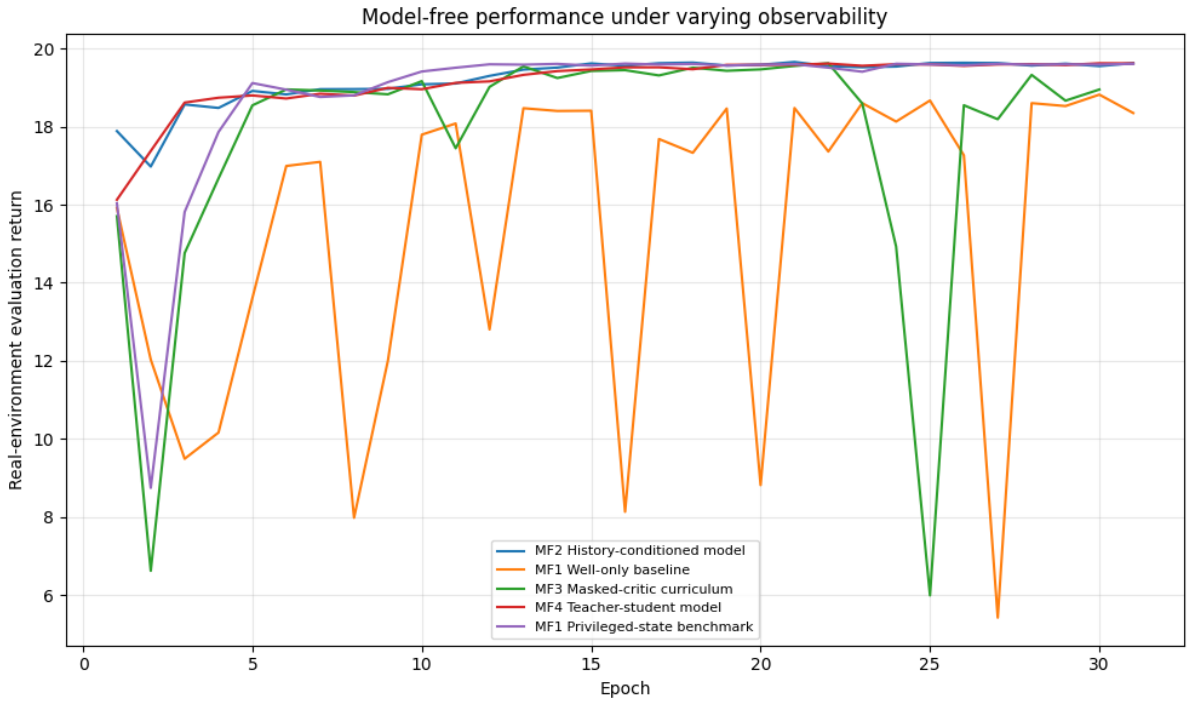

Figure 10: Test return as a function of training epoch for the five model-free variants. The well-only baseline is the weakest and most unstable controller, whereas the history-conditioned and teacher-student models reach the same high-performance regime as the privileged-state benchmark. The masking-curriculum variant achieves strong returns but is noticeably less stable.

The masking-curriculum variant provides a more mixed picture. It achieves strong returns for a substantial portion of training, which suggests that gradually withdrawing privileged critic information is a meaningful idea. However, it is noticeably less stable than the history-conditioned and teacher-student models, with pronounced late-training degradation. Taken together, these results support two main conclusions. First, temporal context is the dominant ingredient for overcoming partial observability in this setting. Second, privileged information is still useful during training, but its benefits are most convincing when incorporated through asymmetric teacher-student learning rather than through the masking curriculum alone.

5.2 Scenario 0: nominal model-based retention

Figure 11 evaluates the model-based pipeline under nominal conditions. Table 5 reports the corresponding best and final returns and expresses the final values as retention relative to the deployable model-free teacher-student reference. The purpose of this experiment is not to outperform the model-free reference, but to determine whether latent imagination can preserve its real-environment performance. The results show that this is indeed possible. Several model-based variants remain in the same high-performance regime as the deployable model-free controller, demonstrating that the fixed latent model-based pipeline is viable in the base case.

A clear pattern also emerges with respect to the latent source. The controllers built on pre-finetuning latents are consistently stronger and more stable than most of their post-finetuning counterparts. In particular, the pre-finetuning variants remain tightly clustered near the top of the plot throughout training, whereas post-finetuning variants exhibit larger performance spread and, in some cases, marked degradation. This suggests that latent representations optimized for distillation or auxiliary objectives are not automatically the best representations for downstream model-based control. In other words, nominal

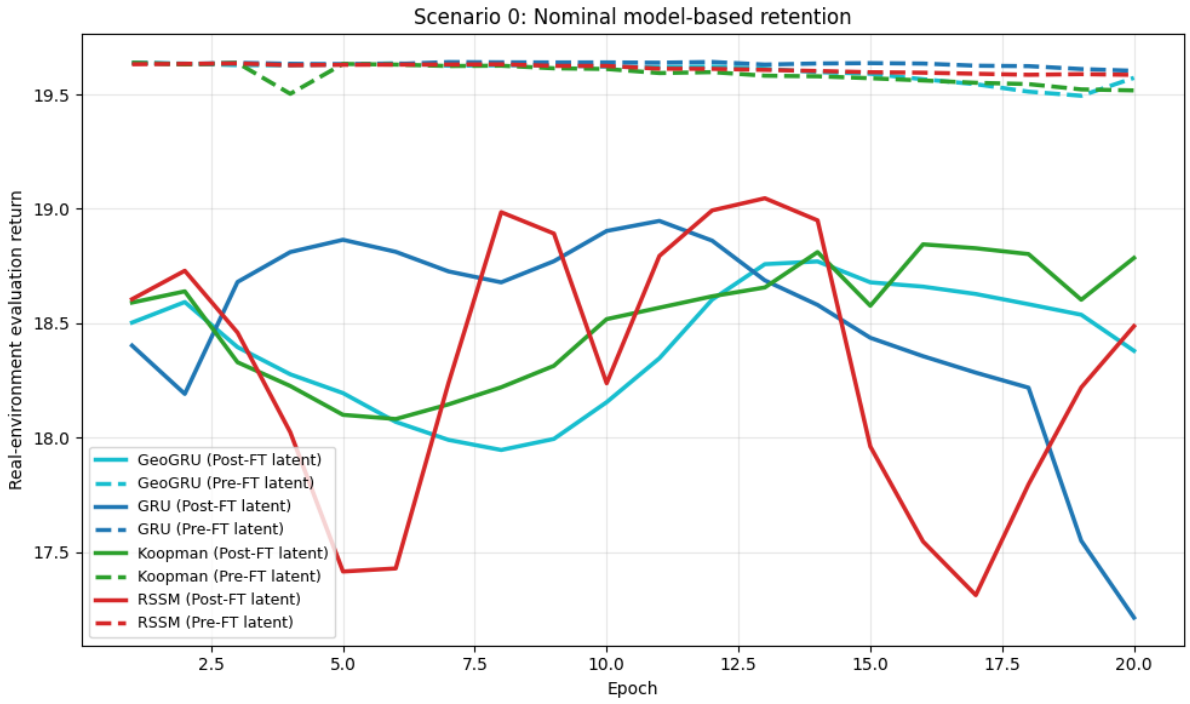

Figure 11: Scenario 0: nominal model-based retention. Real-environment evaluation return as a function of training epoch for the different world-model backbones and latent sources. Several model-based variants remain in the same high-performance regime as the deployable model-free reference, while clear differences also emerge between pre-finetuning and post-finetuning latents.

control retention depends not only on the world-model backbone, but also on the structure of the latent space on which it is trained.

From the perspective of the paper, Scenario 0 therefore establishes the main prerequisite for the later abnormal cases: dream-based latent control can remain competitive in the real simulator under nominal physics, provided that the latent representation and world-model backbone are chosen appropriately.

5.3 Scenario 1: adaptation under known injector failure

Figure 12 shows the effect of losing one injector control channel. This is the easiest of the abnormal scenarios conceptually, since the abnormality is operational and explicitly known rather than latent and hidden. Even so, the difference between model-based and model-free adaptation is pronounced. After the shared nominal initialization at epoch 0, the model-based variants rapidly improve and converge to a high-performance regime, whereas direct model-free retuning deteriorates sharply and remains far below all model-based alternatives.

Thus, when the actuator loss is known and can be imposed consistently in real and imagined interaction, latent model-based retuning uses the limited abnormal simulator budget much more effectively than direct SAC retuning. The modest spread among model-based variants suggests that Scenario 1 mainly tests adaptation to a known control constraint rather than correction of deep latent-dynamics error.

5.4 Scenario 2: adaptation under leakage-induced dynamics and reward shift

Figure 13 presents the most challenging and, arguably, most informative abnormal case. Here the nominal assumptions fail not only at the level of state evolution, but also at the level of the operational objective, since leakage introduces an additional penalty term. This is reflected in the much larger spread across

Table 5: Scenario 0 nominal model-based retention results. The deployable model-free teacher-student final return is 19.616; the retention column reports final return relative to this reference.

World model	Latent source	Best return	Final return	Retention (%)
GRU	Pre-FT	19.642	19.604	99.9
RSSM	Pre-FT	19.636	19.586	99.8
GeoGRU	Pre-FT	19.639	19.573	99.8
Koopman	Pre-FT	19.639	19.518	99.5
Koopman	Post-FT	18.845	18.786	95.8
RSSM	Post-FT	19.047	18.487	94.2
GeoGRU	Post-FT	18.770	18.379	93.7
GRU	Post-FT	18.947	17.213	87.7

Table 6: Abnormal-scenario retuning summary. For each scenario, the model-based row reports the best final-performing model-based variant. The last column is the final-return difference relative to direct model-free retuning under the same scenario-specific real-simulator budget.

Scenario	Abnormality	Best final MB variant	MF retuning final	MB best return	MB final return	Final gain
1	Known injector-control loss	GRU Post-FT	11.573	17.725	17.682	+6.109
2	Leakage dynamics + reward shift	Koopman Post-FT	-5.584	10.990	10.563	+16.147
3	Compartmentalized connectivity shift	RSSM Pre-FT	12.514	14.894	14.414	+1.900

model-based variants and the substantial difficulty of direct model-free retuning.

The best model-based variants recover to clearly positive high-return regimes, while direct model-free retuning remains poor throughout training. Unlike Scenario 1, however, performance depends strongly on the latent source and world-model backbone. This indicates that leakage adaptation is not only a policy-retuning problem; it also tests whether the residual adapter can correct a meaningful dynamics-and-reward mismatch.

5.5 Scenario 3: adaptation under compartmentalized connectivity shift

Figure 14 considers a different kind of physics mismatch. Specifically, the reservoir is more compartmentalized than assumed in the nominal digital twins. Unlike Scenario 2, the reward structure is unchanged, so the challenge lies primarily in the transition dynamics rather than in both dynamics and objective. The results again favor model-based adaptation, although the margin over model-free retuning is smaller and more nuanced than in the leakage scenario.

Most model-based variants improve substantially over the nominal initialization and remain above the model-free retuning curve for the majority of training. However, the scenario is clearly more heterogeneous than Scenario 1, and the relative ranking of world-model/latent combinations is less uniform. Some variants remain strong and stable, while others display noticeable fluctuations or partial degradation. This indicates that the compartmentalized setting is difficult in a different way from leakage: the problem is not an explicit new penalty, but a structural revision of flow connectivity that must be absorbed by the adapted latent dynamics.

The main conclusion is nevertheless consistent with the earlier abnormal scenarios. Even when the control interface is unchanged and the reward is nominal, correcting the nominal latent dynamics through the model-based pipeline remains more effective than direct model-free retuning. The advantage is smaller than in Scenario 2, but it is still systematic enough to support the broader adaptation argument of the paper.

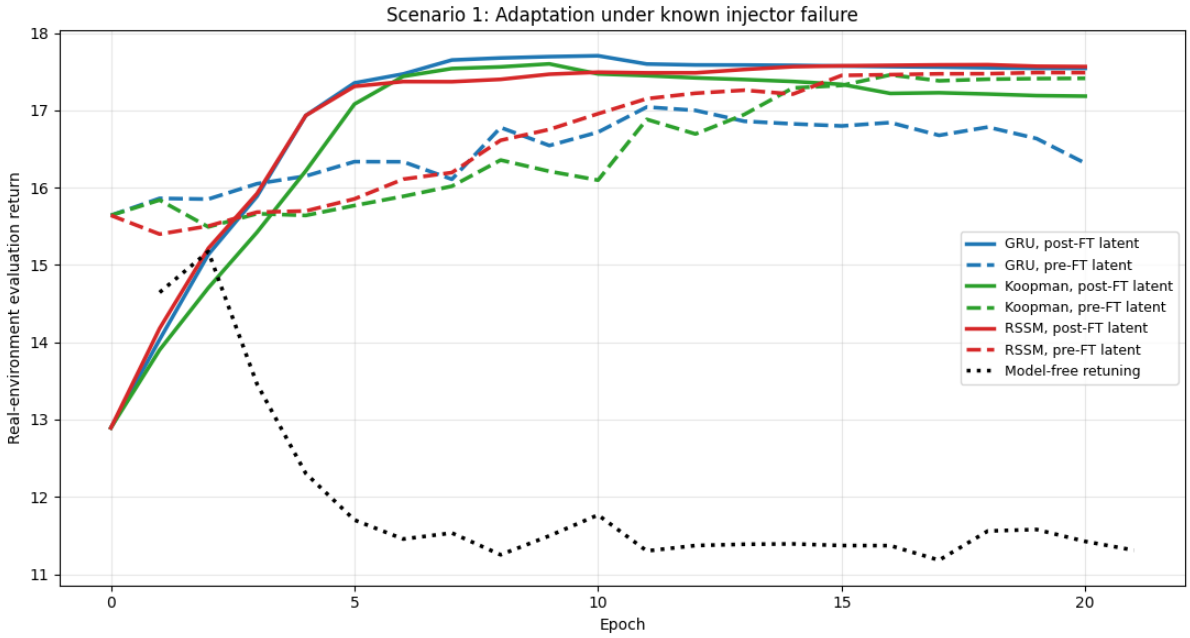

Figure 12: Scenario 1: adaptation under known injector failure. Real-environment evaluation return as a function of training epoch for model-based retuning and direct model-free retuning. Epoch 0 corresponds to the nominal controller before any scenario-specific updates. Model-based retuning rapidly recovers and clearly outperforms direct model-free retuning throughout training.

5.6 Overall interpretation of the results

Across the abnormal scenarios, the benefit of model-based retuning is largest when the mismatch changes the reward and dynamics simultaneously, as in leakage, and smaller when the mismatch is primarily structural, as in compartmentalization. The results also show that no single world-model backbone or latent source dominates uniformly. In particular, the contrast between pre-finetuning and post-finetuning latent representations suggests that latent design is part of the adaptation problem itself, rather than a purely technical implementation detail.

6 Conclusions

This work formulated closed-loop CO_2 storage control as a partially observable sequential decision problem and evaluated deployable model-free and latent model-based controllers under nominal and abnormal operating conditions. The results show that well-history conditioning is sufficient to recover nearly all of the privileged-state performance while preserving deployment feasibility. They also show that latent model-based adaptation can use limited abnormal-scenario simulator interaction more effectively than direct model-free retuning, especially when the mismatch alters both dynamics and reward.

A broader conclusion is that latent representation design matters throughout the workflow. The comparison between pre-finetuning and post-finetuning latents shows that the latent space learned for model-free control is not neutral with respect to downstream model-based control and adaptation. Similarly, no single world-model backbone dominated uniformly across all scenarios, which suggests that robustness in this setting emerges from the interaction between encoder, latent geometry, and adaptation mechanism rather than from a single universally best model class.

Overall, the results support a practical message for CCS closed-loop control. Strong deployable policies can be learned from public well-level information when temporal context is modeled explicitly, and these policies can later be adapted effectively through latent model-based pipelines when the control interface

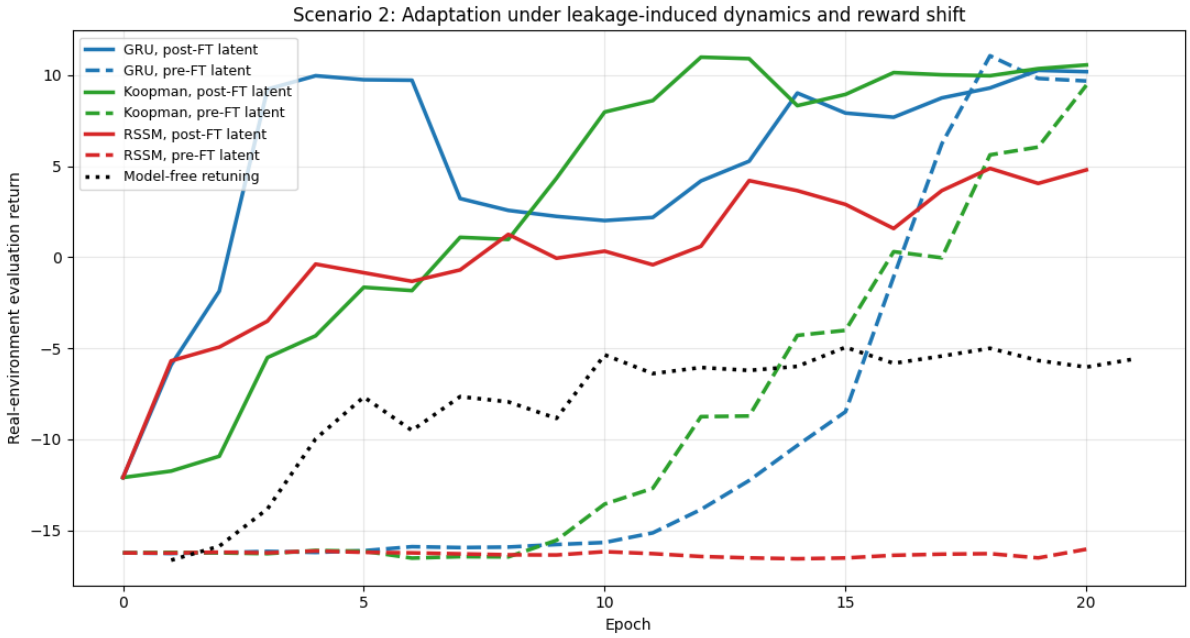

Figure 13: Scenario 2: adaptation under leakage-induced dynamics and reward shift. Real-environment evaluation return as a function of training epoch for model-based retuning and direct model-free retuning. Epoch 0 denotes the nominal controller before adaptation. The best model-based variants recover strongly despite the combined dynamics and reward shift, whereas model-free retuning remains in a poor regime.

or the reservoir physics deviate from nominal assumptions. This makes the proposed framework a viable alternative to repeated history matching and re-optimization in settings where online adaptation must remain computationally tractable.

The present results should be interpreted as simulator-budget-controlled comparisons rather than exhaustive robustness studies. Future work should therefore examine multi-seed variability, richer monitoring streams, online updating of the public encoder, and abnormal scenarios in which the type or location of the mismatch is not known in advance. Additional extensions include latent representations explicitly co-designed for both model-free control and downstream model-based adaptation, more realistic abnormality detection and partial diagnosis settings, and richer closed-loop decision problems involving structural uncertainty updates, field-development actions, and multi-stage operational planning. These directions would further strengthen the connection between deployable deep reinforcement learning and practical CCS reservoir management.

Credit author statement

Conceptualization, S.F.; methodology, S.F.; software, S.F.; validation, S.F. and V.G.; resources, S.F.; writing—original draft preparation, S.F.; writing—review and editing, S.F., V.G.; visualization, S.F. All authors have read and agreed to the final version of the manuscript.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

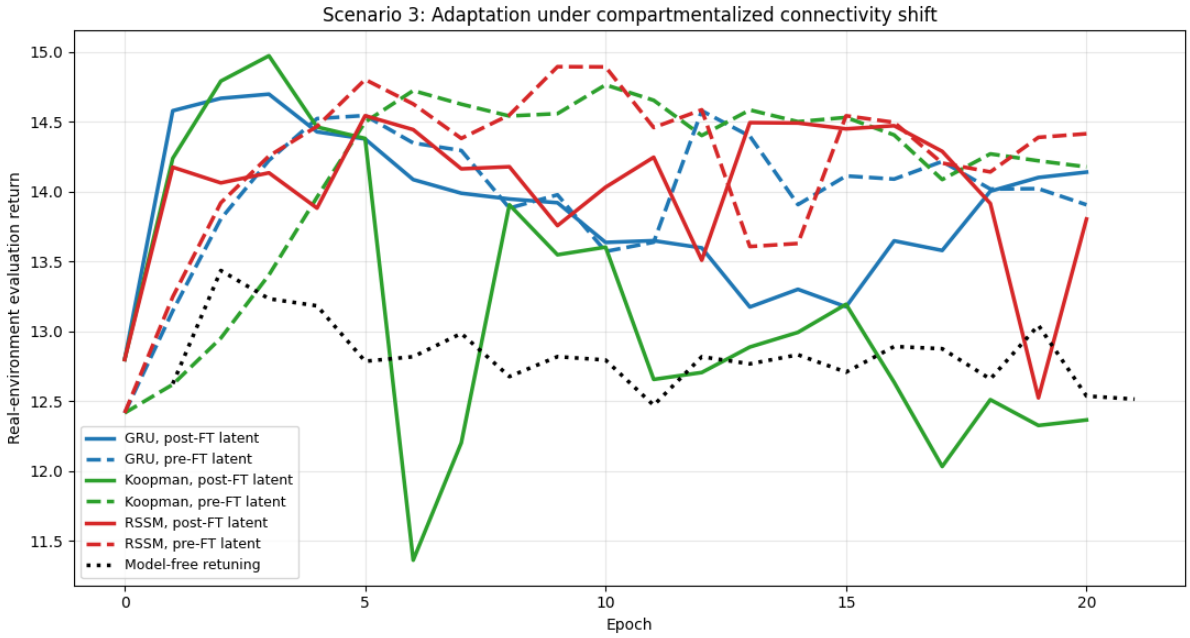

Figure 14: Scenario 3: adaptation under compartmentalized connectivity shift. Real-environment evaluation return as a function of training epoch for model-based retuning and direct model-free retuning. Epoch 0 denotes the nominal controller before adaptation. Most model-based variants outperform model-free retuning, although the spread across latent sources and world-model backbones is more nuanced than in Scenario 2.

Funding

The resources were granted with the support of GRNET as part of the project StorageCO_2. The publication of the article in OA mode was financially supported by HEAL-Link.

Data Availability Statement

Project code can be found at https://github.com/flammmes/Closed-Loop-Reservoir-Management

A Detailed abnormal-scenario retuning results

This appendix reports the full per-variant abnormal-scenario retuning values that support the scenario-level summary in Table 6. These tables are kept outside the main text because the main claims depend on the scenario-level comparison between direct model-free retuning and the best final-performing model-based retuning variant under the same scenario-specific real-simulator budget.

Table 7: Full Scenario 1 retuning results under known injector-control loss.

Run	Family	Best return	Final return
MB GRU Post-FT	Model-based	17.725	17.682
MB Koopman Post-FT	Model-based	17.683	17.242
MB RSSM Post-FT	Model-based	17.578	17.500
MB Koopman Pre-FT	Model-based	17.492	17.465
MB RSSM Pre-FT	Model-based	17.471	17.319
MB GRU Pre-FT	Model-based	17.328	17.164
MF retuning	Model-free	14.645	11.573

Table 8: Full Scenario 2 retuning results under leakage-induced dynamics and reward shift. Runs with missing evaluation values are omitted from the table.

Run	Family	Best return	Final return
MB GRU Pre-FT	Model-based	11.066	9.683
MB Koopman Post-FT	Model-based	10.990	10.563
MB GRU Post-FT	Model-based	10.266	10.192
MB Koopman Pre-FT	Model-based	9.449	9.449
MB RSSM Post-FT	Model-based	4.883	4.800
MB RSSM Pre-FT	Model-based	-16.034	-16.034
MF retuning	Model-free	-4.941	-5.584

Table 9: Full Scenario 3 retuning results under compartmentalized connectivity shift.

Run	Family	Best return	Final return
MB Koopman Post-FT	Model-based	14.972	12.365
MB RSSM Pre-FT	Model-based	14.894	14.414
MB Koopman Pre-FT	Model-based	14.764	14.177
MB GRU Post-FT	Model-based	14.697	14.139
MB GRU Pre-FT	Model-based	14.578	13.906
MB RSSM Post-FT	Model-based	14.543	13.803
MF retuning	Model-free	13.435	12.514

References

[1] Sean M Jarvis and Sheila Samsatli. Technologies and infrastructures underpinning future co2 value chains: A comprehensive review and comparative analysis. *Renewable and Sustainable Energy Reviews*, 85:46–68, 2018.

[2] Paolo Gabrielli, Matteo Gazzani, and Marco Mazzotti. The role of carbon capture and utilization, carbon capture and storage, and biomass to enable a net-zero-co2 emissions chemical industry. *Industrial & Engineering Chemistry Research*, 59(15):7033–7045, 2020.

[3] Mai Bui, Graeme Douglas Puxty, Matteo Gazzani, Salman Masoudi Soltani, and Carlos Pozo. The role of carbon capture and storage (ccs) technologies in a net-zero carbon future. 2021.

[4] SA Rackley and SA Rackley. Introduction to geological storage. *Carbon Capture and Storage; Elsevier: Amsterdam, The Netherlands*, pages 285–304, 2017.

[5] Lola Tomić, Vesna Karović Maričić, Dušan Danilović, and Miroslav Crnogorac. Criteria for co2 storage in geological formations. *Podzemni radovi*, (32):61–74, 2018.

[6] Xiaoyan Ji and Chen Zhu. Co2 storage in deep saline aquifers. In *Novel materials for carbon dioxide mitigation technology*, pages 299–332. Elsevier, 2015.

[7] Stefan Bachu. Review of co2 storage efficiency in deep saline aquifers. *International Journal of Greenhouse Gas Control*, 40:188–202, 2015.

[8] Karsten Michael, Alexandra Golab, Valeriya Shulakova, Jonathan Ennis-King, Guy Allinson, Sandeep Sharma, and Toby Aiken. Geological storage of co2 in saline aquifers—a review of the experience from existing storage operations. *International journal of greenhouse gas control*, 4(4):659–667, 2010.

[9] Sarah Hannis, Jiemin Lu, Andy Chadwick, Sue Hovorka, Karen Kirk, Katherine Romanak, and Jonathan Pearce. Co2 storage in depleted or depleting oil and gas fields: what can we learn from existing projects? *Energy Procedia*, 114:5680–5690, 2017.

[10] Erfan Mohammadian, Badrul Mohamed Jan, Amin Azdarpour, Hossein Hamidi, Nur Hidayati Binti Othman, Aqilah Dollah, Siti Nurliyana Binti Che Mohamed Hussein, and Rozana Azrina Binti Sazali. Co 2-eor/sequestration: Current trends and future horizons. In *Enhanced Oil Recovery Processes-New Technologies*. IntechOpen, 2019.

[11] Zhaowen Li, Mingzhe Dong, Shuliang Li, and Sam Huang. Co2 sequestration in depleted oil and gas reservoirs—caprock characterization and storage capacity. *Energy Conversion and Management*, 47(11-12):1372–1382, 2006.

[12] Ismail Ismail and Vassilis Gaganis. Carbon capture, utilization, and storage in saline aquifers: Subsurface policies, development plans, well control strategies and optimization approaches—a review. *Clean Technologies*, 5(2):609–637, 2023.

[13] Karsten Pruess, Julio García, Tony Kovscek, Curt Oldenburg, Jonny Rutqvist, Carl Steefel, and Tianfu Xu. Code intercomparison builds confidence in numerical simulation models for geologic disposal of co2. *Energy*, 29(9-10):1431–1444, 2004.

[14] Holger Class, Anozie Ebigbo, Rainer Helmig, Helge K Dahle, Jan M Nordbotten, Michael A Celia, Pascal Audigane, Melanie Darcis, Jonathan Ennis-King, Yaqing Fan, et al. A benchmark study on problems related to co 2 storage in geologic formations: summary and discussion of the results. *Computational geosciences*, 13:409–434, 2009.

[15] Abdullah Cihan, Jens T Birkholzer, and Marco Bianchi. Optimal well placement and brine extraction for pressure management during co2 sequestration. *International Journal of Greenhouse Gas Control*, 42:175–187, 2015.

[16] David A Cameron and Louis J Durlofsky. Optimization of well placement, co2 injection rates, and brine cycling for geological carbon sequestration. *International Journal of Greenhouse Gas Control*, 10:100–112, 2012.

[17] Stefan Bachu. Co2 storage in geological media: Role, means, status and barriers to deployment. *Progress in energy and combustion science*, 34(2):254–273, 2008.

[18] Heleen De Coninck, Todd Flach, Paul Curnow, Peter Richardson, Jason Anderson, Simon Shackley, Gudmundur Sigurthorsson, and David Reiner. The acceptability of co2 capture and storage (ccs) in europe: An assessment of the key determining factors: Part 1. scientific, technical and economic dimensions. *International Journal of Greenhouse Gas Control*, 3(3):333–343, 2009.

[19] Karl W Bandilla and Michael A Celia. Active pressure management through brine production for basin-wide deployment of geologic carbon sequestration. *International Journal of Greenhouse Gas Control*, 61:155–167, 2017.

[20] Thomas A Buscheck, Jeffrey M Bielicki, Joshua A White, Yunwei Sun, Yue Hao, William L Bourcier, Susan A Carroll, and Roger D Aines. Pre-injection brine production in co2 storage reservoirs: An approach to augment the development, operation, and performance of ccs while generating water. *International Journal of Greenhouse Gas Control*, 54:499–512, 2016.

[21] Steven T Anderson and Hossein Jahediesfanjani. Estimating the net costs of brine production and disposal to expand pressure-limited dynamic capacity for basin-scale co2 storage in a saline formation. *International Journal of Greenhouse Gas Control*, 102:103161, 2020.

[22] Yang Wang, Yaqin Xu, and Keni Zhang. Investigation of co2 storage capacity in open saline aquifers with numerical models. *Procedia Engineering*, 31:886–892, 2012.

[23] Kalyanmoy Deb, Karthik Sindhya, and Jussi Hakanen. Multi-objective optimization. In *Decision sciences*, pages 161–200. CRC Press, 2016.

[24] Jahedul Islam, Pandian M Vasant, Berihun Mamo Negash, Moacyr Bartholomeu Laruccia, Myo Myint, and Junzo Watada. A holistic review on artificial intelligence techniques for well placement optimization problem. *Advances in engineering software*, 141:102767, 2020.

[25] Alison Cozad, Nikolaos V Sahinidis, and David C Miller. Learning surrogate models for simulation-based optimization. *AIChE Journal*, 60(6):2211–2227, 2014.

[26] Peter I Frazier. A tutorial on bayesian optimization. *arXiv preprint arXiv:1807.02811*, 2018.

[27] Robert J Adler. An introduction to continuity, extrema, and related topics for general gaussian processes. IMS, 1990.

[28] Armen Der Kiureghian and Ove Ditlevsen. Aleatory or epistemic? does it matter? *Structural safety*, 31(2):105–112, 2009.

[29] Eyke Hüllermeier and Willem Waegeman. Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine learning*, 110(3):457–506, 2021.

[30] Sofianos Panagiotis Fotias, Ismail Ismail, and Vassilis Gaganis. Optimization of well placement in carbon capture and storage (ccs): Bayesian optimization framework under permutation invariance. *Applied Sciences*, 14(8):3528, 2024.

[31] DR Brouwer, G Nœvdal, JD Jansen, Erland H Vefring, and CPJW Van Kruijsdijk. Improved reservoir management through optimal control and continuous model updating. In *SPE Annual Technical Conference and Exhibition?*, pages SPE–90149. SPE, 2004.

[32] Inegbenose Aitokhuehi and Louis J Durlofsky. Optimizing the performance of smart wells in complex reservoirs using continuously updated geological models. *Journal of Petroleum Science and Engineering*, 48(3-4):254–264, 2005.

[33] Pallav Sarma, Louis J Durlofsky, Khalid Aziz, and Wen H Chen. Efficient real-time reservoir management using adjoint-based optimal control and model updating. *Computational Geosciences*, 10:3–36, 2006.

[34] Jan-Dirk Jansen, SD Douma, Dr R Brouwer, PMJ Van den Hof, OH Bosgra, and AW Heemink. Closed-loop reservoir management. In *SPE Reservoir Simulation Conference?*, pages SPE–119098. SPE, 2009.

[35] Chunhong Wang, Gaoming Li, and Albert C Reynolds. Production optimization in closed-loop reservoir management. *SPE journal*, 14(03):506–523, 2009.

[36] Vladislav Bukshtynov, Oleg Volkov, Louis J Durlofsky, and Khalid Aziz. Comprehensive framework for gradient-based optimization in closed-loop reservoir management. *Computational Geosciences*, 19:877–897, 2015.

[37] Mohsen Dadashpour, David Echeverria Ciaurri, Tapan Mukerji, Jon Kleppe, and Martin Landrø. A derivative-free approach for the estimation of porosity and permeability using time-lapse seismic and production data. *Journal of Geophysics and Engineering*, 7(4):351–368, 2010.

[38] Yimin Liu and Louis J Durlofsky. Multilevel strategies and geological parameterizations for history matching complex reservoir models. *SPE Journal*, 25(01):081–104, 2020.

[39] Yan Chen and Dongxiao Zhang. Data assimilation for transient flow in geologic formations via ensemble kalman filter. *Advances in Water Resources*, 29(8):1107–1122, 2006.

[40] Alexandre A Emerick and Albert C Reynolds. Ensemble smoother with multiple data assimilation. *Computers & Geosciences*, 55:3–15, 2013.

[41] Kai Zhang, Zhongzheng Wang, Guodong Chen, Liming Zhang, Yongfei Yang, Chuanjin Yao, Jian Wang, and Jun Yao. Training effective deep reinforcement learning agents for real-time life-cycle production optimization. *Journal of Petroleum Science and Engineering*, 208:109766, 2022.

[42] Yusuf Nasir and Louis J Durlofsky. Deep reinforcement learning for optimal well control in subsurface systems with uncertain geology. *Journal of Computational Physics*, 477:111945, 2023.

[43] Emilio Parisotto, Francis Song, Jack Rae, Razvan Pascanu, Caglar Gulcehre, Siddhant Jayakumar, Max Jaderberg, Raphael Lopez Kaufman, Aidan Clark, Seb Noury, et al. Stabilizing transformers for reinforcement learning. In *International conference on machine learning*, pages 7487–7498. PMLR, 2020.

[44] Zyed Bouzarkouna, Didier Yu Ding, and Anne Auger. Well placement optimization with the covariance matrix adaptation evolution strategy and meta-models. *Computational Geosciences*, 16:75–92, 2012.

[45] Obiajulu J Isebor, Louis J Durlofsky, and David Echeverría Ciaurri. A derivative-free methodology with local and global search for the constrained joint optimization of well locations and controls. *Computational Geosciences*, 18:463–482, 2014.

[46] Yusuf Nasir, Wei Yu, and Kamy Sepehrnoori. Hybrid derivative-free technique and effective machine learning surrogate for nonlinear constrained well placement and production optimization. *Journal of Petroleum Science and Engineering*, 186:106726, 2020.

[47] Jérôme E Onwunalu and Louis J Durlofsky. Application of a particle swarm optimization algorithm for determining optimum well location and type. *Computational Geosciences*, 14:183–198, 2010.

[48] Ahmed H Alhuthali, Akhil Datta-Gupta, Bevan Yuen, and Jerry P Fontanilla. Optimal rate control under geologic uncertainty. In *SPE Improved Oil Recovery Conference?*, pages SPE–113628. SPE, 2008.

[49] Zhe Liu and Albert C Reynolds. A sequential-quadratic-programming-filter algorithm with a modified stochastic gradient for robust life-cycle optimization problems with nonlinear state constraints. *SPE Journal*, 25(04):1938–1963, 2020.

[50] Pengju Wang, Michael Litvak, and Khalid Aziz. Optimization of production operations in petroleum fields. In *SPE Annual Technical Conference and Exhibition?*, pages SPE–77658. SPE, 2002.

[51] R-M-M Fonseca, Olwijn Leeuwenburgh, Ernesto Della Rossa, PM Van den Hof, and J-D-D Jansen. Ensemble-based multiobjective optimization of on/off control devices under geological uncertainty. *SPE Reservoir Evaluation & Engineering*, 18(04):554–563, 2015.

[52] RM M Fonseca, Olwijn Leeuwenburgh, PMJ MJ Van den Hof, and JD D Jansen. Improving the ensemble-optimization method through covariance-matrix adaptation. *Spe Journal*, 20(01):155–168, 2015.

[53] KR Ramaswamy, RM Fonseca, Olwijn Leeuwenburgh, MM Siraj, and PMJ Van den Hof. Improved sampling strategies for ensemble-based optimization. *Computational Geosciences*, 24:1057–1069, 2020.

[54] Mathias C Bellout, David Echeverría Ciaurri, Louis J Durlofsky, Bjarne Foss, and Jon Kleppe. Joint optimization of oil well placement and controls. *Computational Geosciences*, 16:1061–1079, 2012.

[55] Lianlin Li, Behnam Jafarpour, and M Reza Mohammad-Khaninezhad. A simultaneous perturbation stochastic approximation algorithm for coupled well placement and control optimization under geologic uncertainty. *Computational Geosciences*, 17:167–188, 2013.

[56] Fahim Forouzanfar and Albert C Reynolds. Joint optimization of number of wells, well locations and controls using a gradient-based algorithm. *Chemical Engineering Research and Design*, 92(7):1315–1328, 2014.

[57] Mehrdad G Shirangi and Louis J Durlofsky. A general method to select representative models for decision making and optimization under uncertainty. *Computers & geosciences*, 96:109–123, 2016.

[58] Mehrdad G Shirangi and Louis J Durlofsky. Closed-loop field development under uncertainty by use of optimization with sample validation. *SPE Journal*, 20(05):908–922, 2015.

[59] Dan Arnold, Vasily Demyanov, Mike Christie, Alexander Bakay, and Konstantin Gopa. Optimisation of decision making under uncertainty throughout field lifetime: A fractured reservoir example. *Computers & Geosciences*, 95:123–139, 2016.

[60] Junko Hutahaean, Vasily Demyanov, and Mike Christie. Reservoir development optimization under uncertainty for infill well placement in brownfield redevelopment. *Journal of Petroleum Science and Engineering*, 175:444–464, 2019.

[61] Malcolm Sambridge. Geophysical inversion with a neighbourhood algorithm—i. searching a parameter space. *Geophysical journal international*, 138(2):479–494, 1999.

[62] Malcolm Sambridge. Geophysical inversion with a neighbourhood algorithm—ii. appraising the ensemble. *Geophysical Journal International*, 138(3):727–746, 1999.

[63] Gavin A Rummery and Mahesan Niranjan. *On-line Q-learning using connectionist systems*, volume 37. University of Cambridge, Department of Engineering Cambridge, UK, 1994.

[64] Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8:279–292, 1992.

[65] Farzad Hourfar, Hamed Jalaly Bidgoly, Behzad Moshiri, Karim Salahshoor, and Ali Elkamel. A reinforcement learning approach for waterflooding optimization in petroleum reservoirs. *Engineering Applications of Artificial Intelligence*, 77:98–116, 2019.

[66] Hongze Ma, Gaoming Yu, Yuehui She, and Yongan Gu. Waterflooding optimization under geological uncertainties by using deep reinforcement learning algorithms. In *SPE Annual Technical Conference and Exhibition?*, page D031S043R001. SPE, 2019.

[67] Ruslan Miftakhov, Abdulaziz Al-Qasim, and Igor Efremov. Deep reinforcement learning: reservoir optimization from pixels. In *International Petroleum Technology Conference*, page D021S052R002. IPTC, 2020.

[68] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[69] Atish Dixit and Ahmed H ElSheikh. Stochastic optimal well control in subsurface reservoirs using reinforcement learning. *Engineering Applications of Artificial Intelligence*, 114:105106, 2022.

[70] Volodymyr Mnih. Asynchronous methods for deep reinforcement learning. *arXiv preprint arXiv:1602.01783*, 2016.

[71] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. PMLR, 2018.

[72] Zhongzheng Wang, Kai Zhang, Jinding Zhang, Guodong Chen, Xiaopeng Ma, Guojing Xin, Jinzheng Kang, Hanjun Zhao, and Yongfei Yang. Deep reinforcement learning and adaptive policy transfer for generalizable well control optimization. *Journal of Petroleum Science and Engineering*, 217:110868, 2022.

[73] Zhong-Zheng Wang, Kai Zhang, Guo-Dong Chen, Jin-Ding Zhang, Wen-Dong Wang, Hao-Chen Wang, Li-Ming Zhang, Xia Yan, and Jun Yao. Evolutionary-assisted reinforcement learning for reservoir real-time production optimization under uncertainty. *Petroleum Science*, 20(1):261–276, 2023.

[74] Yusuf Nasir and Louis J Durlofsky. Practical closed-loop reservoir management using deep reinforcement learning. *SPE Journal*, 28(03):1135–1148, 2023.

[75] Yusuf Nasir and Louis J Durlofsky. Multi-asset closed-loop reservoir management using deep reinforcement learning. *Computational Geosciences*, 28(1):23–42, 2024.

[76] Jincong He, Meng Tang, Chaoshun Hu, Shusei Tanaka, Kainan Wang, Xian-Huan Wen, and Yusuf Nasir. Deep reinforcement learning for generalizable field development optimization. *SPE Journal*, 27(01):226–245, 2022.

[77] Yusuf Nasir, Jincong He, Chaoshun Hu, Shusei Tanaka, Kainan Wang, and XianHuan Wen. Deep reinforcement learning for constrained field development optimization in subsurface two-phase flow. *Frontiers in Applied Mathematics and Statistics*, 7:689934, 2021.