Advanced Energy Systems and Intelligent Geoscience (AESIG)
https://www.mospbs.com/journal/aesig

• *Article* •

An Intelligent Water Injection Decision-Making and Production Optimization Method Based on SAC Deep Reinforcement Learning

Tianyu Yang[1], Xiang Rao[1,*], Shuhui Xu[2]

[1] College of Petroleum Engineering, Yangtze University, Wuhan, 430100, China.
[2] Jiangsu Co-Innovation Center of Efficient Processing and Utilization of Forest Resources, Nanjing Forestry University, Nanjing 210037, China.
[*]**Corresponding Author:** Xiang Rao **Email:** raoxiang0103@163.com
Received: 21 April 2026 **Accepted:** 22 May 2026

Abstract: To address the failure of static water injection strategies in late-stage oilfield development caused by strong reservoir heterogeneity, this paper proposes an intelligent water injection and dynamic production optimization method using Soft Actor-Critic (SAC) deep reinforcement learning. By formulating waterflooding optimization as a Markov Decision Process (MDP), a Finite Volume Method (FVM) simulator is dynamically coupled with a deep learning environment. Departing from traditional methods that rely solely on wellhead data, this model uses high-dimensional oil saturation images to capture the waterflood front's topological evolution. A Convolutional Neural Network (CNN) extracts spatial features to output optimal real-time allocation weights for multiple injection wells, constrained by total injection volume. Tested on complex heterogeneous configurations—including "four-injector, five-producer" and "four-injector, nine-producer" patterns—the SAC agent demonstrated remarkable convergence stability and exploration efficiency. The model autonomously establishes an adaptive strategy that controls water cut, suppresses water channeling in high-permeability streaks, and intelligently redirects hydrodynamic energy to unswept zones. Compared to conventional uniform injection, this method significantly expands macroscopic sweep volume and reduces remaining oil saturation, offering a novel paradigm for the real-time, closed-loop management of complex reservoirs.

Keywords: Deep reinforcement learning; Soft Actor-Critic (SAC); Intelligent water injection decision-making; Closed-loop reservoir management; Spatial heterogeneity; Production optimization

1 Introduction

In the fields of reservoir engineering and multiphase fluid dynamics, waterflooding is one of the most effective strategies for maintaining formation pressure, controlling fluid migration, and enhancing oil recovery (EOR). As oilfield development enters its middle and late stages, subsurface fluid flow exhibits high degrees of nonlinearity and spatial heterogeneity, making traditional static development strategies inadequate for managing the complex evolution of fluid fronts. Consequently, closed-loop reservoir

DOI: https://doi.org/10.58244/aesig.263879

management (CLRM), which dynamically adjusts well control parameters based on real-time formation dynamics, has emerged as a core paradigm in modern reservoir development (Jansen et al., 2009). Brouwer and Jansen (2004) pioneered the use of optimal control theory for the dynamic waterflooding optimization of smart wells, establishing the theoretical foundation for proactive flow control. However, when dealing with complex three-dimensional multiphase flow partial differential equations (PDEs), traditional adjoint gradient methods and global search-based black-box optimization algorithms (e.g., genetic algorithms) encounter severe bottlenecks. These include prohibitive computational costs during numerical simulation and a susceptibility to falling into local optima, making it difficult to satisfy the real-time requirements of dynamic control (Foroud et al., 2018).

In recent years, the rapid development of machine learning and deep learning has driven a paradigm shift in fluid mechanics and computational geosciences (Brunton et al., 2020). Extensive research has been dedicated to constructing efficient physics-based surrogate models using deep neural networks to replace computationally expensive traditional numerical solvers. For instance, Zhu and Zabaras (2018) and Mo et al. (2019) utilized deep convolutional encoder-decoder networks, achieving significant advancements in the dynamic uncertainty quantification of multiphase flow in heterogeneous media. Tang et al. (2020) and Guo et al. (2018) successfully implemented data assimilation and life-cycle production forecasting for subsurface flow problems using deep learning and support vector regression. Nwachukwu et al. (2018) applied machine learning for the rapid evaluation of well placements in heterogeneous reservoirs. Additionally, physics-informed neural networks (PINNs) have demonstrated remarkable potential in solving both forward and inverse complex flow problems (Cai et al., 2021; Karniadakis et al., 2021; Lou et al., 2021). Building upon this, recent structural innovations in neural networks, such as Boundary-Integral Type Neural Networks (BINN) (Rao, Liu, Fu, et al., 2025) and Physics-informed Kolmogorov–Arnold networks (PIKAN) (Rao, Liu, He, et al., 2025), have further enhanced the modeling accuracy of flow problems in highly anisotropic and heterogeneous porous media. Alongside classical deep learning advancements, quantum computing and quantum machine learning are emerging as cutting-edge paradigms to overcome the prohibitive computational bottlenecks in reservoir simulation. Recent breakthroughs include the performance evaluation of variational quantum linear solvers for reservoir flow equations (Rao, 2024b) and the pioneering application of quantum computing algorithms in streamline-based water-flooding simulations (Rao, 2024a). Furthermore, hybrid quantum-classical physics-informed neural networks have been developed to solve reservoir seepage equations (Rao et al., 2026), and efficient quantum neural network models have demonstrated significant potential in related fields such as predicting CO2 sequestration (Rao et al., 2024). However, despite the excellent performance of these data-driven models in terms of predictive accuracy and computational speed, they predominantly rely on supervised learning or static surrogate modeling. They lack the capability for sequential decision-making within dynamic physical environments, rendering them difficult to apply directly to closed-loop real-time production optimization under complex engineering constraints.

Deep reinforcement learning (DRL) provides a groundbreaking decision-making framework for

solving continuous control tasks through continuous trial-and-error interactions between an agent and the environment (Mnih et al., 2015; Schulman et al., 2017). In the field of fluid mechanics, DRL has been widely and successfully applied to active flow control and the discovery of flow field strategies (Garnier et al., 2021; Rabault et al., 2019; Ren et al., 2021; Ye & Elsheikh, 2025). In petroleum engineering, DRL has progressively emerged as a key technology for overcoming the bottlenecks of traditional production optimization and well control. Hourfar et al. (2019) conducted early work by constructing a reinforcement learning-based waterflooding optimization framework, demonstrating the method's potential in handling complex reservoir dynamics. Nasir and Durlofsky (2023a, 2023b) proposed a DRL method for optimal well control in subsurface systems under geological uncertainty, and developed a practical DRL-based framework for closed-loop reservoir management. Yan and Zhong (2025) successfully extended DRL to optimal hydraulic fracturing design for real-time production optimization. Recently, numerous scholars have conducted extensive and fruitful research on the refined regulation of waterflooding and water injection in complex well networks. Hu et al. (2024) proposed an efficient DRL-based flow scheduling method for constant-pressure zonal water injection, significantly improving water injection efficiency; Wang et al. (2023) developed an evolution-assisted reinforcement learning algorithm specifically designed to solve real-time reservoir production optimization problems under uncertainty; Ding et al. (2023) utilized cropped well-group samples to build a reinforcement learning model, achieving optimal control of oil well production under complex conditions. For nonlinear constrained optimization in continuous action spaces, the Soft Actor-Critic (SAC) algorithm exhibits exceptional robustness and sample exploration efficiency by introducing policy entropy while maximizing expected returns (Fujimoto et al., 2018; Haarnoja et al., 2018). Xin et al. (2024) demonstrated the superior performance of the SAC-based DRL algorithm in maximizing oilfield production; furthermore, Chen et al. (2025) proposed a worst-case-based safe SAC reinforcement learning approach, effectively addressing waterflooding reservoir production optimization under stringent nonlinear engineering constraints.

Despite the significant progress achieved by the aforementioned reinforcement learning-based production optimization models, the current paradigm still faces two key limitations. First, the vast majority of existing models rely solely on pure dynamic well data (e.g., liquid production rate and bottom-hole flowing pressure) as state inputs, which fails to accurately capture the spatial heterogeneity of subsurface multiphase flow and the topological evolution of the waterflooding front. Second, purely tabular data-driven models are often decoupled from rigorous physical conservation laws of fluids and geological constraints. As pointed out by Miftakhov et al. (2020), conducting reinforcement learning optimization directly from "pixel-level" spatial feature inputs represents a frontier direction for precisely capturing the evolution of subsurface fluid dynamics.

To bridge the aforementioned research gap, this paper proposes an intelligent water injection decision-making and production optimization method based on SAC deep reinforcement learning. We formulate the production optimization problem as a Markov Decision Process (MDP) and construct a cross-language coupled environment that enables dynamic interaction between the deep learning framework and a finite volume

method (FVM) reservoir numerical simulator. Within this framework, the agent directly utilizes high-dimensional oil saturation evolution images fed back by the environment as state inputs. It employs convolutional neural networks (CNNs) under the Actor-Critic architecture to extract the spatially heterogeneous features of the subsurface waterflooding front. Subject to the total water injection quota, the agent continuously outputs the optimal dynamic allocation weights for multiple water injection wells, thereby achieving optimal waterflooding control throughout the production life-cycle. By integrating a cutting-edge maximum entropy reinforcement learning algorithm with rigorous physical numerical simulations, this paper aims to provide a novel solution paradigm—one that features both physics-awareness and highly efficient sequential decision-making capabilities—for intelligent water injection decision-making under complex heterogeneous reservoir conditions.

2 Methodology

2.1 Markov Decision Process (MDP) Formulation for Production Optimization

A waterflooding strategy comprises a sequence of decisions across multiple time steps; consequently, production optimization can be naturally formulated as an MDP (Hourfar et al., 2019) which has a strong correlation with the gained financial profit, is maximized. Fortunately, due to the recent progresses in the computational tools and also expansion of the calculating facilities, utilization of non-conventional optimization methods is feasible to achieve the desired goals. In this paper, waterflooding optimization problem has been defined and formulated in the framework of Reinforcement Learning (RL. As illustrated in Figure 1, within this framework, the agent acts as a controller that adjusts the production strategy based on the real-time reservoir state, while

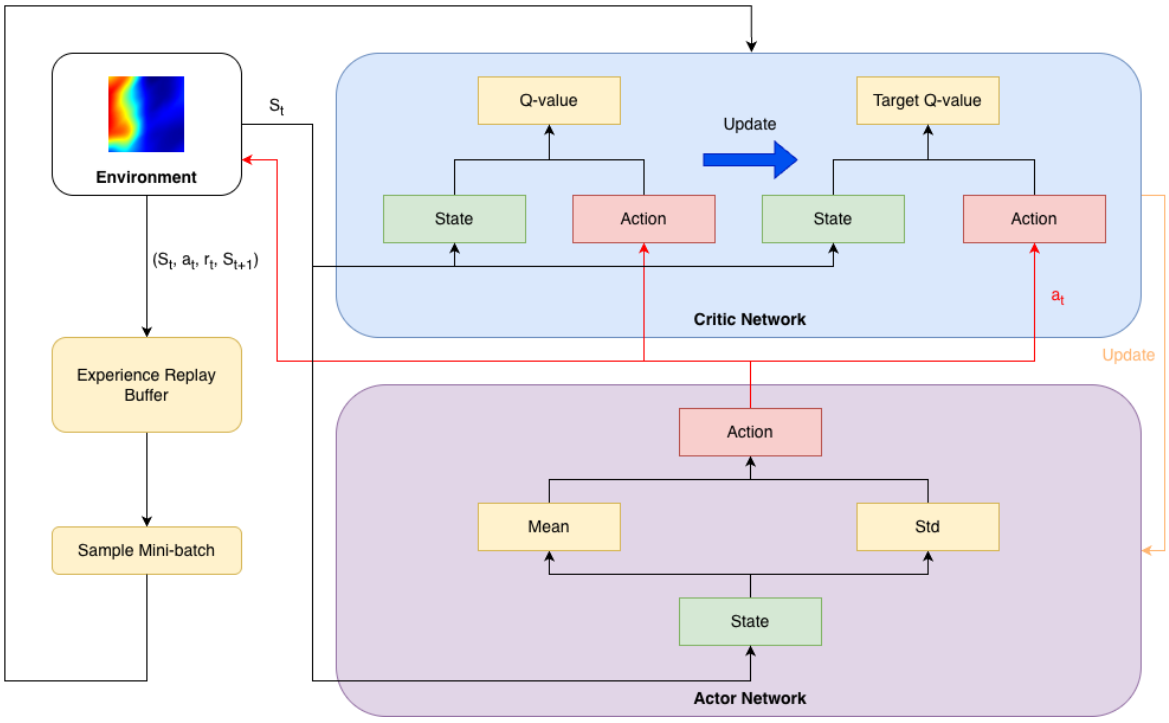

Figure 1. Reinforcement learning-based production optimization framework under the Actor-Critic architecture

the reservoir numerical simulator serves as the environment, providing real-time state evolution and reward feedback. The key elements of the MDP are defined as follows:

1. State Space: The design of the state space must concurrently consider the observability of the system and the diversity of variables. The selected state variables must provide sufficient information correlated with the reward function, enabling the agent to comprehend the current situation and take corrective measures. For the water injection optimization problem in this paper, since pure dynamic wellhead production data fail to reflect subsurface heterogeneity, we employ the spatial features of the oil saturation distribution as the state input (Miftakhov et al., 2020). At the t-th control step, the environment outputs and feeds back an image tensor, i.e., $S_t \in [0,255]^{64\times64\times3}$. This visual state matrix allows the algorithm to intuitively observe the evolution characteristics of the displacement front.

2. Action Space: The design of actions depends on the specific task environment. For production optimization problems, well control parameters are typically defined as actions. In our model, the actions are formulated as the water injection allocation weights for the 4 water injection wells. Considering that the SAC algorithm yields the most stable outputs within the continuous interval of $[-1,1]$, the raw action vector output by the agent at the t-th step is $a_t \in [-1,1]^4$. To ensure compliance with physical conservation laws and engineering constraints during the underlying environment interaction, the raw action vector is transformed into a smooth allocation probability p_t^i summing to 1 via a maximum-value-biased Softmax mechanism:

$$p_t^i = \frac{\exp(a_t^i - \max(a_t))}{\sum_{j=1}^{4}\exp(a_t^j - \max(a_t))} \tag{1}$$

Subsequently, this proportion vector is multiplied by the total water injection quota Q_{total} and then safely discretized and rounded.

3. Reward Function: The design of the reward function is the core determinant of whether the agent can successfully accomplish the target task; it must guarantee that the agent's objective is to maximize cumulative oil production. After completing the implicit equation solving at the t-th stage, the environment returns the oil production for this specific stage. In this paper, the reward r_t is defined as the scaled value of the oil production at the current control stage:

$$r_t = \eta \cdot \Delta Q_{oil}^t \tag{2}$$

where is the scaling factor, empirically set to 1/1000 in this study. This parameter is utilized to adjust the numerical scale of the reward, thereby promoting the stability of the network training process.

2.2 Policy Learning Based on Soft Actor-Critic

SAC is a state-of-the-art deep reinforcement learning algorithm that has demonstrated exceptional performance in solving continuous control tasks. Unlike typical RL algorithms that solely optimize the expected return, the optimization objective of SAC is to simultaneously maximize both the expected return and the policy entropy (Haarnoja et al., 2018). Its objective function can be expressed as:

$$\pi^* = \arg\max_{\pi} \mathbb{E}\left[\sum_t \gamma^t \left(r_t + \alpha H(\pi(\cdot|s_t))\right)\right] \tag{3}$$

where π^* is the optimal policy; π is the policy function; $\mathbb{E}[\cdot]$ is the expectation operator; γ is the reward discount factor (set to 0.99 in this paper), utilized to balance future and immediate

rewards; α is the temperature coefficient balancing exploration and exploitation (set to "auto" mode for dynamic automated adjustment in this study); and $H(\cdot)$ denotes the entropy term of the policy, which encourages the agent to explore a broader action space.

The SAC algorithm leverages an Actor-Critic architecture. Given that the state input in this paper consists of high-dimensional oil saturation images, the agent employs convolutional neural networks (CNNs) for feature extraction. During the policy iteration phase, data from the experience replay buffer $\mathcal{D}$ are utilized to alternately update the parameters of the Critic and Actor networks (Mnih et al., 2015).

For the Critic network Q_θ, its objective is to accurately evaluate the expected return of the current state-action pair. Its parameters θ are updated by minimizing the loss function based on the temporal difference (TD) error:

$$J_Q(\theta_j)=\mathbb{E}_{(s_t,a_t,r_t,s_{t+1})\sim\mathcal{D}}\left[\left(Q_{\theta_j}(s_t,a_t)-y(r_t,s_{t+1})\right)^2\right],\ j\in\{1,2\} \tag{4}$$

where $J_Q(\theta_j)$ is the loss function of the j-th Critic network; θ_j represents the weight parameters of the j-th Critic network, and $j\in\{1,2\}$ indicates the adoption of a double Q-network mechanism; $Q_{\theta_j}(s_t,a_t)$ is the value estimation of the current state-action pair by the Critic network; and $y(r_t,s_{t+1})$ is the target value.

The target value $y(r_t,s_{t+1})$ incorporates a clipped double Q-learning mechanism and an entropy regularization term to mitigate the overestimation of action values (Fujimoto et al., 2018):

$$y(r_t,s_{t+1})=r_t+\gamma\left(\min_{j=1,2}Q_{\theta_{targ\_j}}(s_{t+1},a_{t+1})-\alpha\log\pi_\phi(a_{t+1}|s_{t+1})\right) \tag{5}$$

where $\theta_{\text{targ\_}j}$ represents the weight parameters of the j-th target Critic network (typically obtained through a soft update of θ_j); a_{t+1} is the next action sampled according to the current policy π_ϕ at state s_{t+1}; and $\pi_\phi(a_{t+1}|s_{t+1})$ is the probability density of outputting action a_{t+1} given state s_{t+1}.

For the Actor network π_ϕ, its objective is to find an action distribution capable of maximizing both the value evaluated by the Critic and the policy entropy. The loss function with respect to its parameters ϕ is formulated as the divergence between the policy entropy and the predicted Q-value:

$$J_\pi(\phi)=\mathbb{E}_{s_t\sim\mathcal{D},a_t\sim\pi_\phi(\cdot|s_t)}\left[\alpha\log\pi_\phi(a_t|s_t)-\min_{j=1,2}Q_{\theta_j}(s_t,a_t)\right] \tag{6}$$

where $J_\pi(\phi)$ is the loss function of the Actor network; ϕ denotes the weight parameters of the Actor network; $\pi_\phi(\cdot|s_t)$ represents the parameterized Actor policy network; and $\min_{j=1,2}Q_{\theta_j}(s_t,a_t)$ represents taking the minimum of the evaluation values from the two Critic networks, serving as a conservative estimate of the current policy's value.

2.3 Interactive Training Framework and Physical Environment Decoupling

Since reservoir evolution simulation involves the complex computation of partial differential equations (PDEs), the data generation rate of the numerical simulator is significantly slower than that of typical RL environments (Foroud et al., 2018)which is the case for most commercial hydrocarbon reservoir simulators. The selected optimization algorithms have been divided in two categories. The first category consists of those algorithm that use approximated gradients, namely, simultaneous perturbation stochastic approximation (SPSA. To enhance training efficiency and robustness, this paper constructs a cross-language coupled framework featuring

dynamic process synchronization.

In the global optimization outer loop, the agent invokes the reservoir numerical simulator in the background via the subprocess module. Each episode comprises T=10 water injection stages. At each control step, the agent's action commands are dispatched through file I/O. Upon reading the commands, the solver executes a single-step advancement and writes the oil production and spatial saturation features of the current stage to the hard disk. In the inner loop, all generated experience tuples (s_t, a_t, r_t, s_{t+1}) are stored in an experience replay buffer with a capacity of 50,000. Once 500 steps of warm-up data have been collected, the algorithm initiates gradient descent updates on the weights of the Critic and Actor networks using mini-batches of size 128, thereby realizing the data-driven evolution of the optimal policy under physical simulation constraints. To clearly demonstrate the integration of the aforementioned environment decoupling mechanism and the policy learning process, the complete workflow of the cross-language interactive deep reinforcement learning training in this paper is presented in Algorithm 1.

Algorithm 1. Pseudocode of cross-language interactive SAC algorithm

Initialize: an empty experience replay buffer $\mathcal{D}$, actor network π_ϕ, critic network $Q_{\theta_{1,2}}$, target critic network $Q_{\theta_{targ\ 1,2}}$

Input: max episodes E_{max}, time horizon T, water injection limit Q_{total}, save frequency f

1: for episode e = 1 to E_{max} do

2: /* Cross-language environment initialization /

3: Clean history data directories and awaken MATLAB reservoir simulator via system subprocess

4: Monitor and read the initial oil saturation state s_1 via NumPy byte stream

5: for timestep t = 1 to T do

6: Obtain action $a_t = \pi_\phi(s_t)$

7: Apply Softmax and discrete rounding to a_t to satisfy Q_{total} constraints

8: Execute a_t by writing the control scheme to local file for MATLAB, receive r_t, and reach a new reservoir state s_{t+1}

9: Add the experience (s_t, a_t, r_t, s_{t+1}) to $\mathcal{D}$

10: $s_t = s_{t+1}$

11: / Learning process of the RL agent */

12: if size of $\mathcal{D} \geq 500$ then

13: for each training step do

14: Randomly sample a mini-batch of transitions $\{(s_t, a_t, r_t, s_{t+1})\}$ from $\mathcal{D}$

15: Update the critic network parameters with $\theta_j \leftarrow \theta_j - \lambda_Q \nabla_{\theta_j} J_Q(\theta_j)$, for $j \in \{1,2\}$

16: Update the actor network parameters with $\phi \leftarrow \phi - \lambda_\pi \nabla_\phi J_\pi(\phi)$

17: Update the target critic network parameters with soft update mechanism

18: Close MATLAB subprocess

19: if e mod f = 0 then

20: Save the optimal policy

21: Output: optimal policy

3 Dataset and Experimental Setup

3.1 Construction of the Interactive Reservoir Numerical Simulation Environment

Unlike traditional supervised learning based on static datasets, reinforcement learning requires the agent to conduct continuous "trial-and-error" and exploration within a dynamic environment. Based on the finite volume method (FVM), this study leverages a MATLAB numerical solver and a Python deep learning framework to construct a cross-language dynamic interactive reservoir environment. (Figure 2.)

The well pattern layout in the simulated area of this experiment adopts a "four injectors and five producers" configuration (as illustrated in Figure 2). During each production control cycle, the total allowable water injection rate of the system is strictly limited to 300m^3/d. Each complete development cycle comprises 10 control stages. At the end of each stage, the simulator implicitly solves the multiphase flow equations and outputs the current formation oil saturation field in the form of a $64 \times 64 \times 3$ RGB image matrix, which serves as the environmental state input for the agent. Simultaneously, it outputs the oil production for that specific stage, serving as the reward signal for the agent's policy evolution. The core parameters of the interactive numerical simulation environment are listed in Table 1.

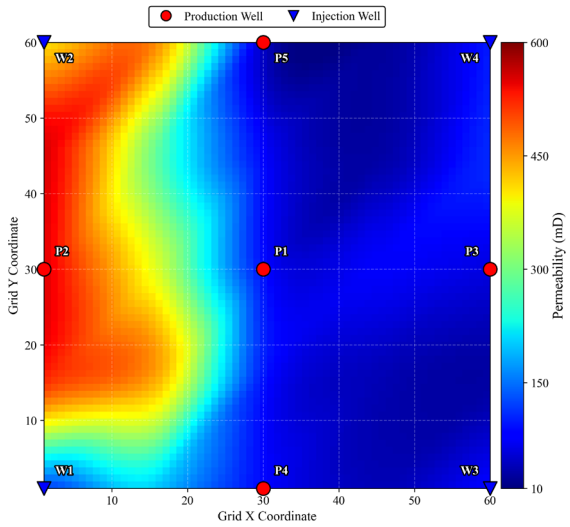

Figure 2. Well pattern layout of "four injectors and five producers" in the reservoir simulation area

Table 1. Key parameters for generating the reservoir simulation dataset

Parameter	Value	Parameter	Value
Porosity	0.2	permeability/ mD	10 - 600
Initial pressure/ MPa	50	Oil saturation	[0.2, 0.8]
Constant BHP / Mpa	45	Total injection rate/	300
Total time/ days	300	Time step/ days	30
Injectors	4	Producers	5

3.2 Experimental Environment and Model Hyperparameter Settings

The reinforcement learning control model proposed in this study is implemented using the PyTorch deep learning framework and the Stable-Baselines3 (SB3) algorithm library. The Soft Actor-Critic (SAC) algorithm is employed for the global optimization of the waterflooding strategy. Given that the reservoir environmental state is represented as 2D images with spatial topological structures, the model utilizes a convolutional neural network (CNN) feature extractor to directly extract the heterogeneous evolution features of the waterflooding front from the saturation field. During the training phase, the capacity of the experience replay buffer is set to 50,000. To prevent the model from converging to local optima during the initial stage and to accumulate sufficient evolutionary diversity of the physical environment,

the agent executes purely random exploration for the initial 500 steps (equivalent to 50 episodes). Subsequently, gradient updates for the policy network and value network are performed by randomly sampling mini-batches of size 128 from the replay buffer, with an update frequency of 10 gradient descent iterations executed after each episode. Additionally, the temperature coefficient of the SAC algorithm is set to an auto-tuning mode to achieve an optimal balance between the exploration of the unknown state space and the exploitation of existing high-return policies. The detailed hyperparameter configuration for model training is summarized in Table 2.

Table 2. Hyperparameter settings for model training

Category	Parameter	Value
Setup	Hardware	Auto (CPU/CUDA)
	Framework	PyTorch
	Algorithm	Soft Actor-Critic
Training	Resolution	64×64
	Normalization	[−1, 1]
	Learning rate	3×10^{-4}
	Steps per episode	10
	Total timesteps	50,000 (5,000 episodes)
	Replay buffer size	50,000
	Learning starts	500
	Batch size	128
	Discount factor	0.99
	Train frequency	1 episode
	Gradient steps	10
	Entropy coefficient	Auto

4 Results and Analysis

4.1 Analysis of the Training Process and Convergence Characteristics

Under the deep reinforcement learning framework, the agent continuously iterates its water injection control policy through sustained interaction with the heterogeneous reservoir environment. Figure 3 illustrates the learning curve of the SAC agent over nearly 5000 episodes. (Figure 3.)

A distinct policy evolution process can be

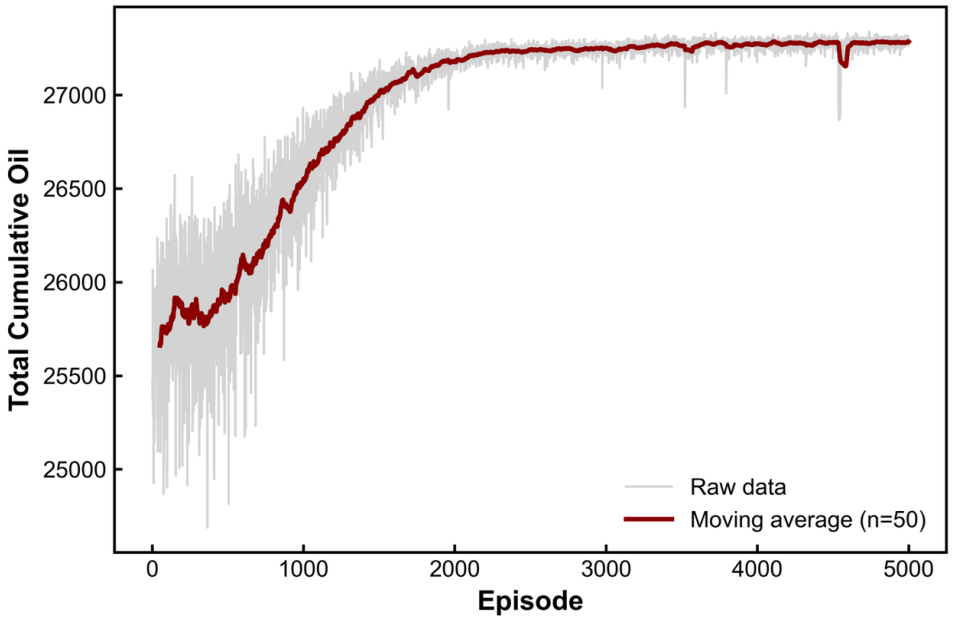

Figure 3. Learning curve of total cumulative oil production during the SAC model training process

observed throughout the training phase. In the initial stage of training, the agent is in an extensive exploration phase. Lacking an effective understanding of the complex subsurface multiphase flow dynamics, its output well control parameters exhibit high randomness, resulting in severe fluctuations in the system's total cumulative oil production (primarily distributed within the range of 25,000 to 26,250). As the training iterations progress, the SAC algorithm, leveraging its maximum entropy regularization mechanism, effectively achieves policy convergence toward high-reward regions while maintaining its exploration capability. The data indicate that after approximately 4000 training episodes, the model has fundamentally grasped the fluid migration dynamics within the formation. In the final stages of training, as the policy network approaches optimality, the total cumulative oil production stabilizes at a high-level plateau between 27,200 and 27,310. These results demonstrate the superior convergence stability of the SAC algorithm when addressing high-dimensional, nonlinear flow control problems.

4.2 Optimization Performance and Mechanism Analysis of the Waterflooding Strategy

The SAC agent autonomously learns and establishes an adaptive water-control and oil-stabilization policy. The agent can actively suppress the water injection intensity in high-permeability zones and redistribute fluid kinetic energy to medium- and low-permeability zones. This effectively delays the uneven breakthrough of the fluid front and significantly expands the macroscopic sweep volume of the injected water within the reservoir.

As illustrated in Figure 4, during the initial production stage, the water injection rates of injection wells Inj-1 and Inj-2 are maintained at a relatively low level (below 25 m^3/d), while the injection rates of Inj-3 and Inj-4 remain at a high level (above 100 m^3/d). As production proceeds, significant dynamic adjustments in the injection rates of all wells occur during the 8th to 10th control steps. This indicates that the reinforcement learning algorithm is capable of adaptively optimizing the water injection scheme based on dynamic changes in the reservoir, thereby maximizing the cumulative oil production.

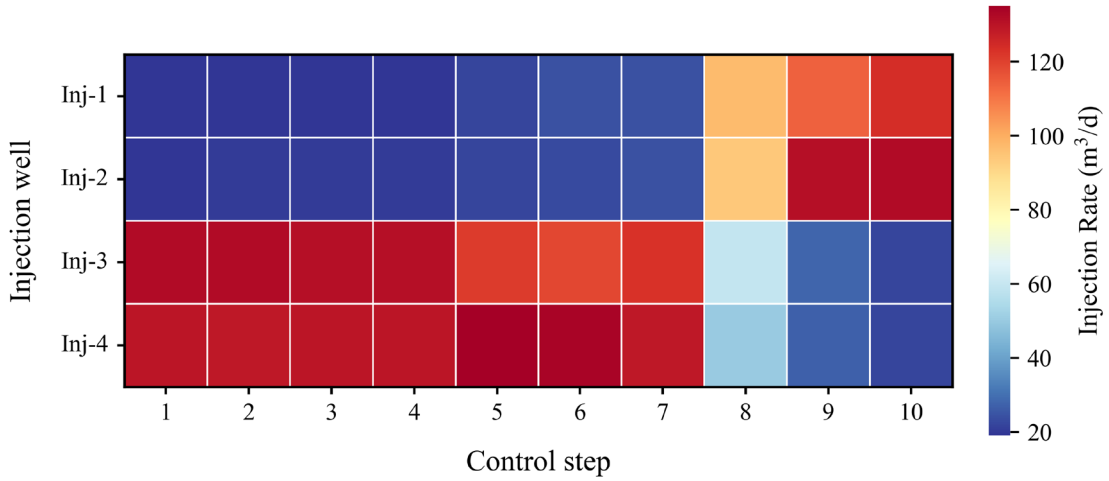

Figure 4. Dynamic evolution of water injection rates for individual injection wells across control steps under the reinforcement learning policy

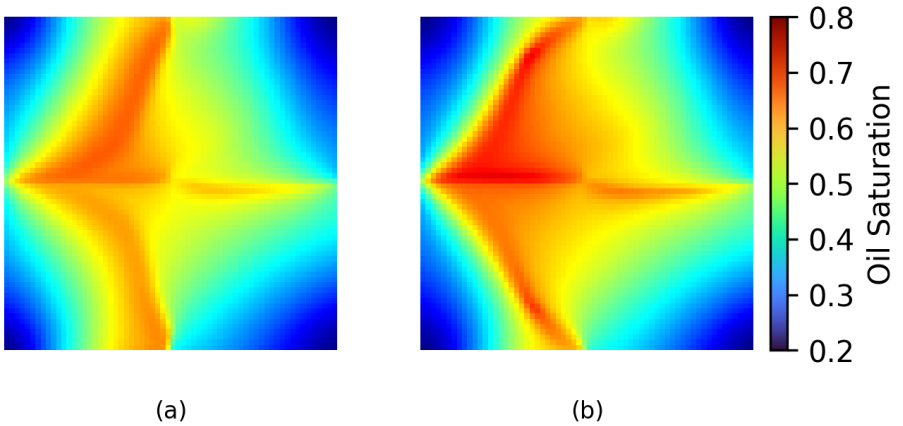

Figure 5. Comparison of reservoir remaining oil saturation field distributions under different water injection strategies after 300 days of production: (a) reinforcement learning optimization policy; (b) uniform allocation strategy

Figure 5 compares the remaining oil saturation field distributions after 300 days of operation under different water injection strategies. Specifically, Figure 5(a) displays the results of adopting the reinforcement learning optimization policy, while Figure 5(b) shows the results of applying an equal daily injection rate across the 4 injection wells (the uniform allocation strategy). A comparative analysis reveals that, compared to the conventional uniform allocation strategy, the overall remaining oil saturation in the reservoir under the reinforcement learning policy is significantly lower. This demonstrates that the proposed strategy can more effectively mobilize the remaining oil and enhance the displacement efficiency.

5 Model Validation under Complex Well Patterns

5.1 Layout and Environmental Configuration of the 4-Injector 9-Producer Complex Well Pattern

To further validate the robustness and generalization capabilities of the Soft Actor-Critic (SAC)-based Deep Reinforcement Learning (DRL) intelligent water injection decision-making model under more complex spatial topological structures, this study designed and established a “4-injector 9-producer” complex well pattern experimental scenario, expanding upon the original “4-injector 5-producer” configuration.

In this experiment, the fundamental environmental parameters, including the model hyperparameters, were kept consistent with previous settings (refer to Tables 1 and 2), while the total water injection quota was adjusted to 400m^3/d. As illustrated in Figure 6, the nine production wells

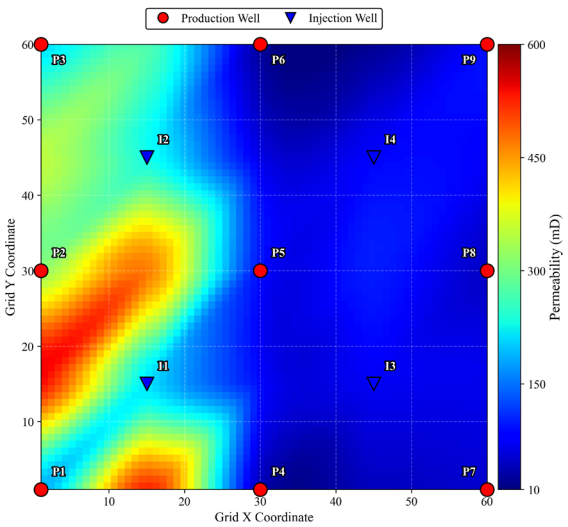

Figure 6. Layout of the 4-injector 9-producer complex well pattern within the reservoir simulation area

(P1–P9) and four injection wells (I1–I4) form a denser, staggered distribution within the simulated region. With the increased number of wells, the seepage dynamics of multiphase fluids within the subterranean porous media and the topological evolution of the waterflood front become significantly more intricate. Consequently, the flow-field interference effects are markedly intensified, presenting greater challenges to the dynamic optimization and anti-interference capabilities of the reinforcement learning agent.

5.2 Dynamic Evolution and Effect Analysis of the Intelligent Water Injection Strategy

After multiple iterations of trial-and-error learning, the SAC agent successfully mastered the evolutionary dynamics of the complex flow field. Following the application of this reinforcement learning strategy, the dynamic water injection allocation scheme generated by the agent is illustrated in Figure 7. Over 10 control steps, the SAC agent adaptively adjusted the allocation weights among the four injection wells. For instance, during the early to middle stages of production, Inj-3 and Inj-4 were allocated significantly higher injection volumes, whereas the rates for Inj-1 and Inj-2 were suppressed at markedly low levels. As production progressed, the agent executed a large-scale reallocation of injection weights among the wells between the 8th and 10th control steps. This significantly increased the injection volume for Inj-1, thereby maximizing the waterflood sweep efficiency and delaying water breakthrough in dominant channels.

The ultimate optimization efficacy of the waterflood development can be intuitively assessed through the reservoir's remaining oil saturation field. Figure 8 presents a comparison between the outcomes of the intelligent reinforcement learning optimization strategy (Figure 8a) and the traditional uniform water injection allocation strategy (Figure 8b) after 300 days of production. It is clearly observable that under the SAC reinforcement learning strategy, the overall remaining oil saturation of the reservoir is substantially lower, with blind spots and dead oil zones drastically reduced. Conversely, the uniform water injection strategy resulted in severe water channeling in certain high-permeability regions, leaving a significant amount of unswept residual

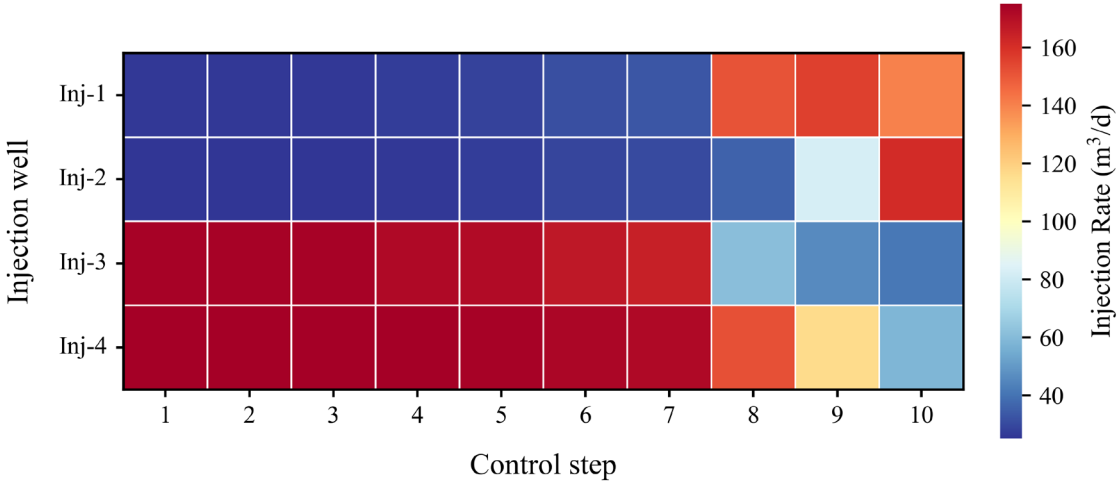

Figure 7. Dynamic evolution of water injection rates for individual injection wells controlled by the reinforcement learning strategy under the complex well pattern

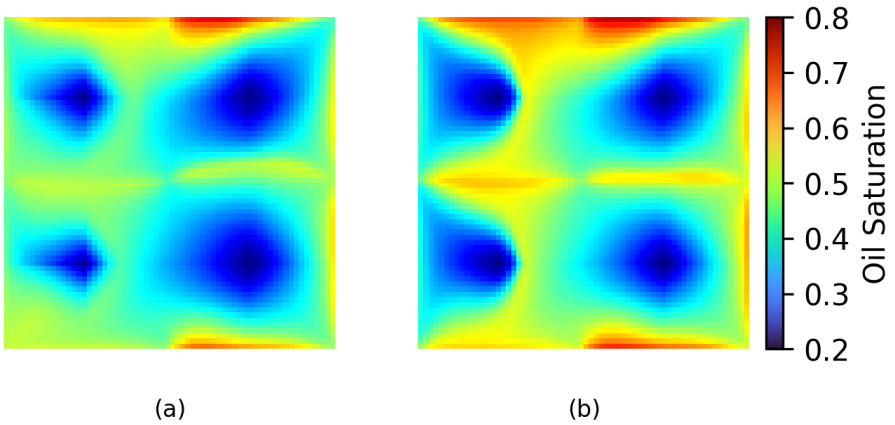

Figure 8. Comparison of the remaining oil saturation field distributions under different water injection strategies after 300 days of production: (a) Reinforcement learning optimization strategy; (b) Uniform allocation strategy

oil trapped in the surrounding matrix.

In conclusion, under the complex "4-injector 9-producer" well pattern and conditions of strong flow-field interference, the proposed SAC deep reinforcement learning model successfully overcomes the bottlenecks of traditional optimization methods. It accurately captures the evolutionary characteristics of subterranean multiphase fluids to achieve a more refined and globally optimal dynamic regulation of the water injection flow field. This comprehensively demonstrates the immense application potential of the proposed method under the complex engineering conditions of actual oilfields.

6 Conclusions

This paper proposes an intelligent water injection decision-making and production optimization method based on Soft Actor-Critic (SAC) deep reinforcement learning to address the challenges of high nonlinearity and spatial heterogeneity of multiphase flow in the closed-loop management of complex reservoirs. Distinguished from traditional models that solely rely on dynamic production data, this study deeply integrates a cutting-edge maximum entropy reinforcement learning algorithm with a finite volume method (FVM)-based reservoir numerical simulator, establishing a cross-language dynamic interaction environment. By directly utilizing high-dimensional oil saturation evolution images from environmental feedback as state inputs, the proposed model achieves life-cycle optimal control over the dynamic allocation of injection weights among multiple injection wells, while strictly adhering to physical and engineering constraints such as total injection quotas.

In a basic "four-injector, five-producer" well pattern experiment, the SAC agent demonstrated outstanding convergence stability and sample exploration efficiency. The results indicate that the intelligent decision-making model can autonomously learn and establish an adaptive strategy for water control and oil stabilization. It proactively suppresses injection intensity in high-permeability zones and redirects fluid kinetic energy toward medium- and low-permeability zones. Compared to traditional uniform allocation strategies, this method effectively delays the heterogeneous breakthrough of the fluid front, substantially expands the macroscopic sweep volume of the injected water within the reservoir,

and significantly reduces the overall remaining oil saturation, thereby maximizing cumulative oil production.

Furthermore, the robustness and generalization capabilities of the model were validated in a more complex "four-injector, nine-producer" well pattern with stronger interference. The results demonstrate that, even under complex flow field conditions, the SAC model accurately achieves the globally optimal allocation of injection weights and substantially diminishes unswept remaining oil zones. This effectively overcomes the bottlenecks of traditional algorithms, such as high computational costs and susceptibility to local optima.

In conclusion, this study provides a novel solution paradigm that integrates both "physics-informed cognition" and "efficient sequential decision-making" capabilities for intelligent water injection decision-making under complex, heterogeneous reservoir conditions. The proposed method fully demonstrates immense application potential in overcoming traditional production optimization bottlenecks and achieving global dynamic regulation under the complex engineering conditions of real-world oilfields.

Acknowledgement
None.

Funding Statement
None.

Author Contributions
All authors contributed to the study conception and design, data collection, analysis and interpretation of the results, and manuscript preparation, and take responsibility for the integrity of the work.

Availability of Data and Materials
None.

Conflicts of Interest
The authors declare that they have no conflicts of interest to report regarding the present study.

References
[1] Brouwer, D. R., & Jansen, J.-D. (2004). Dynamic Optimization of Waterflooding With Smart Wells Using Optimal Control Theory. *SPE Journal*, *9*(04), 391–402. https://doi.org/10.2118/78278-PA

[2] Brunton, S. L., Noack, B. R., & Koumoutsakos, P. (2020). Machine Learning for Fluid Mechanics. *Annual Review of Fluid Mechanics*, *52*(Volume 52, 2020), 477–508. https://doi.org/10.1146/annurev-fluid-010719-060214

[3] Cai, S., Mao, Z., Wang, Z., Yin, M., & Karniadakis, G. E. (2021). Physics-informed neural networks (PINNs) for fluid mechanics: A review. *Acta Mechanica Sinica*, *37*(12), 1727–1738. https://doi.org/10.1007/s10409-021-01148-1

[4] Chen, Z., Zhang, K., Liu, P., Xin, G., Sun, Z., Tao, Z., Zhang, Y., Ji, W., Lu, Y., Jia, L., & Meng, H. (2025). Worst-Case Soft Actor-Critic-Based Safe Reinforcement Learning Method for Nonlinear Constrained Waterflood Reservoir Production Optimization. *SPE Journal*, *30*(12), 7745–7766. https://doi.org/10.2118/230322-PA

[5] Ding, Y., Wang, X., Cao, X., Hu, H., & Bu, Y. (2023). A reinforcement learning method for optimal control of oil well production using cropped well group samples. *Heliyon*, *9*(7). https://doi.org/10.1016/j.heliyon.2023.e17919

[6] Foroud, T., Baradaran, A., & Seifi, A. (2018). A comparative evaluation of global search algorithms in black box optimization of oil production: A case study on Brugge field. *Journal of Petroleum Science and Engineering*, *167*, 131–151. https://doi.

org/10.1016/j.petrol.2018.03.028

[7] Fujimoto, S., Hoof, H., & Meger, D. (2018). Addressing Function Approximation Error in Actor-Critic Methods. *Proceedings of the 35th International Conference on Machine Learning*, 1587–1596. https://proceedings.mlr.press/v80/fujimoto18a.html

[8] Garnier, P., Viquerat, J., Rabault, J., Larcher, A., Kuhnle, A., & Hachem, E. (2021). A review on deep reinforcement learning for fluid mechanics. *Computers & Fluids*, *225*, 104973. https://doi.org/10.1016/j.compfluid.2021.104973

[9] Guo, Z., & Reynolds, A. C. (2018). Robust Life-Cycle Production Optimization With a Support-Vector-Regression Proxy. *SPE Journal*, *23*(06), 2409–2427. https://doi.org/10.2118/191378-PA

[10] Haarnoja, T., Zhou, A., Abbeel, P., & Levine, S. (2018). Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. *Proceedings of the 35th International Conference on Machine Learning*, 1861–1870. https://proceedings.mlr.press/v80/haarnoja18b.html

[11] Hourfar, F., Bidgoly, H. J., Moshiri, B., Salahshoor, K., & Elkamel, A. (2019). A reinforcement learning approach for waterflooding optimization in petroleum reservoirs. *Engineering Applications of Artificial Intelligence*, *77*, 98–116. https://doi.org/10.1016/j.engappai.2018.09.019

[12] Hu, J., Ren, F., Wang, Z., & Jia, D. (2024). Efficient Scheduling of Constant Pressure Stratified Water Injection Flow Rate: A Deep Reinforcement Learning Method. *IEEE Access*, *12*, 123856–123871. https://doi.org/10.1109/ACCESS.2024.3425837

[13] Jansen, J. D., Douma, S. D., Brouwer, D. R., Van den Hof, P. M. J., Bosgra, O. H., & Heemink, A. W. (2009, February 2). *Closed-Loop Reservoir Management*. SPE Reservoir Simulation Symposium. https://doi.org/10.2118/119098-MS

[14] Karniadakis, G. E., Kevrekidis, I. G., Lu, L., Perdikaris, P., Wang, S., & Yang, L. (2021). Physics-informed machine learning. *Nature Reviews Physics*, *3*(6), 422–440. https://doi.org/10.1038/s42254-021-00314-5

[15] Lou, Q., Meng, X., & Karniadakis, G. E. (2021). Physics-informed neural networks for solving forward and inverse flow problems via the Boltzmann-BGK formulation. *Journal of Computational Physics*, *447*, 110676. https://doi.org/10.1016/j.jcp.2021.110676

[16] Miftakhov, R., Al-Qasim, A., & Efremov, I. (2020, January 13). *Deep Reinforcement Learning: Reservoir Optimization from Pixels*. International Petroleum Technology Conference. https://doi.org/10.2523/IPTC-20151-MS

[17] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., & Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature*, *518*(7540), 529–533. https://doi.org/10.1038/nature14236

[18] Mo, S., Zhu, Y., Zabaras, N., Shi, X., & Wu, J. (2019). Deep Convolutional Encoder-Decoder Networks for Uncertainty Quantification of Dynamic Multiphase Flow in Heterogeneous Media. *Water Resources Research*, *55*(1), 703–728. https://doi.org/10.1029/2018WR023528

[19] Nasir, Y., & Durlofsky, L. J. (2023a). Deep reinforcement learning for optimal well control in subsurface systems with uncertain geology. *Journal of Computational Physics*, *477*, 111945. https://doi.org/10.1016/j.jcp.2023.111945

[20] Nasir, Y., & Durlofsky, L. J. (2023b). Practical Closed-Loop Reservoir Management Using Deep Reinforcement Learning. *SPE Journal*, *28*(03), 1135–1148. https://doi.org/10.2118/212237-PA

[21] Nwachukwu, A., Jeong, H., Pyrcz, M., & Lake, L. W. (2018). Fast evaluation of well placements in heterogeneous reservoir models using machine learning. *Journal of Petroleum Science and Engineering*, *163*, 463–475. https://doi.org/10.1016/j.petrol.2018.01.019

[22] Rabault, J., Kuchta, M., Jensen, A., Réglade, U., & Cerardi, N. (2019). Artificial neural networks trained through deep reinforcement learning discover control strategies for active flow control. *Journal of Fluid Mechanics*, *865*, 281–302. https://doi.org/10.1017/jfm.2019.62

[23] Rao, X. (2024a, November 4). *The First Application of Quantum Computing Algorithm in Streamline-Based Simulation of Water-Flooding Reservoirs*. ADIPEC. https://doi.org/10.2118/221850-MS

[24] Rao, X., Liu, Y., Fu, Q., He, X., Kwak, H., Zhao, H., & Hoteit, H. (2025, September 16). *Boundary-Integral Type Neural Network (BINN) for Flow Problems in Anisotropic Reservoirs*. Middle East Oil, Gas and Geosciences Show (MEOS GEO). https://doi.org/10.2118/227536-MS

[25] Rao, X., Liu, Y., He, X., & Hoteit, H. (2025). Physics-informed Kolmogorov–Arnold networks to model flow in heterogeneous porous media with a mixed pressure-velocity formulation. *Physics of Fluids*, *37*(7), 076654. https://doi.org/10.1063/5.0279122

[26] Rao, X., Liu, Y., & Shen, Y. (2026). *Quantum-Classical Physics-Informed Neural Networks for Solving Reservoir Seepage Equations* (arXiv:2512.03923). arXiv. https://doi.org/10.48550/arXiv.2512.03923

[27] Rao, X., Luo, C., He, X., & Hyung, K. (2024, November 4). *An Efficient Quantum Neural Network Model for Prediction of Carbon Dioxide CO2 Sequestration in Saline Aquifers*. ADIPEC. https://doi.org/10.2118/222257-MS

[28] Rao, X. (饶翔). (2024b). Performance study of variational quantum linear solver with an improved ansatz for reservoir flow equations. *Physics of Fluids*, *36*(4), 047104. https://doi.org/10.1063/5.0201739

[29] Ren, F., Rabault, J., & Tang, H. (2021). Applying deep reinforcement learning to active flow control in weakly turbulent conditions. *Physics of Fluids*, *33*(3), 037121. https://doi.org/10.1063/5.0037371

[30] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). *Proximal Policy Optimization Algorithms* (arXiv:1707.06347). arXiv. https://doi.org/10.48550/arXiv.1707.06347

[31] Tang, M., Liu, Y., & Durlofsky, L. J. (2020). A deep-learning-based surrogate model for data assimilation in dynamic subsurface flow problems. *Journal of Computational Physics*, *413*, 109456. https://doi.org/10.1016/j.jcp.2020.109456

[32] Wang, Z.-Z., Zhang, K., Chen, G.-D., Zhang, J.-D., Wang, W.-D., Wang, H.-C., Zhang, L.-M., Yan, X., & Yao, J. (2023). Evolutionary-assisted reinforcement learning for reservoir real-time production optimization under uncertainty. *Petroleum Science*, *20*(1), 261–276. https://doi.org/10.1016/j.petsci.2022.08.016

[33] Xin, G., Zhang, K., Wang, Z., Sun, Z., Zhang, L., Liu, P., Yang, Y., Sun, H., & Yao, J. (2024). Soft Actor-Critic Based Deep Reinforcement Learning Method for Production Optimization. In J. Lin (Ed.), *Proceedings of the International Field Exploration and Development Conference 2023* (pp. 353–366). Springer Nature. https://doi.org/10.1007/978-981-97-0272-5_31

[34] Yan, B., & Zhong, Z. (2025). Deep reinforcement learning for optimal hydraulic fracturing design in real-time production optimization. *Geoenergy Science and Engineering*, *250*, 213815. https://doi.org/10.1016/j.geoen.2025.213815

[35] Ye, M., & Elsheikh, A. H. (2025). Model-based reinforcement learning for active flow control. *Physics of Fluids*, *37*(9), 093363. https://doi.

org/10.1063/5.0287427

[36] Zhu, Y., & Zabaras, N. (2018). Bayesian deep convolutional encoder–decoder networks for surrogate modeling and uncertainty quantification. *Journal of Computational Physics*, *366*, 415–447. https://doi.org/10.1016/j.jcp.2018.04.018