Offline Reinforcement Learning for Bioprocess Optimization with Historical Data

Haiting Wang * **Cleo Kontoravdi** * **Antonio Del Rio Chanona** *

* *Sargent Centre for Process Systems Engineering, Imperial College London, SW7 2AZ, UK.*

Abstract: The development of optimal control strategies for bioprocesses has become essential in pharmaceutical and industrial applications due to the growing demand for sustainable bioproducts. Traditional model-based methods rely heavily on the accuracy of the system model, requiring frequent recalibration and experimentation to sustain performance. In contrast, advanced model-free control techniques, such as Reinforcement Learning (RL), are widely researched. However, training RL controllers online is constrained by the need for extensive online interactions with the biosystem environment, which can be costly and present safety risks. To overcome these limitations, we propose leveraging offline Reinforcement Learning algorithms to train control agents using historical data collected from previous bioprocess operations. These agents can subsequently be fine-tuned, improving current control strategies by utilizing past data without extensive real-time interactions with the system. The effectiveness of offline RL for policy training was demonstrated through an in-silico semi-batch bioprocess case study, where it achieved superior performance compared to alternative machine learning methods. Additionally, the proposed fine-tuning approach successfully transitioned the offline RL-trained policy into the online operational setting, highlighting the practical advantages of combining offline training with targeted online adaptation to improve real-time performance.

Keywords: Biosystems and Bioprocesses, Non-Linear Control Systems, Reinforcement Learning, Batch Processes

1. INTRODUCTION

The control and optimization of bioprocesses have been extensively researched due to their potential to produce valuable bioproducts. These processes are valued for producing competitive alternatives to fossil fuel-derived chemicals and high-value biopharmaceuticals (Brennan and Owende, 2010). Traditional control and optimization strategies for bioprocesses are derived by formulating optimization problem with respect to the target, represented by a mathematic model that can precisely describe the process dynamics is essential. However, biological processes often exhibit highly stochastic and nonlinear behaviors and involve complex bioreaction kinetics, making the optimization problem not only computationally intensive but also vulnerable to model-plant mismatches, which can compromise the effectiveness of the resulting optimization strategies (Del Rio-Chanona et al., 2019; Narayanan et al., 2020). Consequently, traditional model-based methods require regular experimentation for model recalibration, which leads to significant laboratory cost for bioprocess industry. On the other hand, model-free optimization schemes are widely researched to provide alternative solution to this problem, where model-free Reinforcement Learning (RL) can be considered as an effective method. Unlike model-based RL, model-free RL estimates the value or action-value function directly from data without depending on a physical model. Value-based functions predict the total achievable cost based on the current state and input values. As a result, the control policy can be adjusted to optimize the overall cost, all without the need for a model.

1.1 Model-free Reinforcement Learning for bioprocess optimization

Implementation of model-free RL algorithms to train the control agent for process control and optimization has been proved as an effective method in both biochemical and chemical engineering fields. A deep RL controller was implemented by Spielberg et al. (2019) for set-point tracking in distillation column control. For bioprocess, Petsagkourakis et al. (2020) utilized the policy gradient algorithm to train a control policy for batch-to-batch optimization, demonstrating that RL is capable of achieving near-optimal policies for stochastic biosystems. Building on the successful application of the policy gradient algorithm for optimal control policy identification, Sachio et al. (2021) further incorporated RL-based optimal policy identification into an integrated chemical process design and optimization framework. Similarly, Ma et al. (2021) successfully applied Proximal Policy Optimization (PPO) algorithm on the fed-batch optimization task. On the other hand, instead of applying on-policy algorithm like the policy gradient algorithm, Oh et al. (2022) implemented the double deep Q-network (DDQN) algorithm for off-policy learning to train the control policy for semi-batch bioreactor optimization. Compared with the on-policy training,

Q-learning based off-policy learning shows advantages in terms of the data efficiency, since the policy can learn from past trajectories that stored in the data buffer. To leverage this advantage, Monteiro and Kontoravdi (2024) applied off-policy learning algorithm for the optimization of monoclonal antibodies production process. Besides using RL for upstream tasks, Nikita et al. (2021) employed deep Q-learning to optimize process chromatography in continuous biopharmaceutical production.

Despite significant advancements, several challenges persist in the implementation of RL for bioprocess optimization. One of the primary obstacles is the data-intensive nature of online RL training, which requires a substantial amount of process data or the development of a model, thereby returning to the original problem. This limitation arises from two key factors: firstly, in many RL frameworks, both the value function approximation and the control policy are parameterized using machine learning models, such as artificial neural networks (ANNs). The performance of these models is heavily dependent on the quantity and quality of training data; insufficient data can lead to overfitting, thereby reducing the predictive accuracy of the ANN. Furthermore, RL training necessitates continuous interaction between the control agent and the operational environment, during which the agent learns through a trial-and-error process, thus making the learning procedure highly data-demanding. However, real-time interactions are often impractical in real-world applications due to potential high operational costs and safety risks. Additionally, it is important to note that in many studies, the control policy is trained using an offline process model that simulates system behavior. Thus, the effectiveness of the trained policy may be constrained by model accuracy, and valuable process information is often lost as training data is not reused.

To address these challenges, RL algorithms that enable policies to learn from historical data have been researched. One such approach is behavior cloning, a supervised learning technique where the control policy is trained by replicating previously collected optimal control behaviors (Levine et al., 2020). Apprenticeship learning, which incorporates modified RL techniques, has also been applied to chemical process optimization (Mowbray et al., 2021), demonstrating the advantages of RL over supervised learning for policy training based on historical data. Additionally, cutting-edge offline RL methods have been proposed and continue to be a subject of active research. As a variant of RL, the control policy learns from fixed, pre-collected process datasets without interacting with bioprocesses during training. Unlike traditional RL, which requires real-time exploration and data collection, offline RL works by training policies on historical data. The goal is to optimize a policy that can be deployed in real environments without exploration, making it particularly useful when interacting with the environment is impractical. Furthermore, leveraging historical process data helps address data scarcity in policy training, particularly for limited bioprocess datasets.

2. METHOD AND INTEGRATION

2.1 Offline Reinforcement Learning

Due to the focus on policy learning from static datasets, off-policy learning algorithms like Q-learning are particularly well-suited for offline RL compared to on-policy algorithms. This is because off-policy methods decouple the process of policy learning from the data collection phase. Unlike on-policy methods, which require real-time interaction between the agent and the environment, Q-learning allows the agent to learn from previously collected data, a critical feature in offline RL where real-time interaction is not feasible (Levine et al., 2020). In a standard Q-learning framework, the objective is to maximize the expected cumulative reward in a Markov decision process (MDP). MDPs can be defined by a tuple (S, A, P, R, γ), where S, A are the state and action spaces respectively, P is the state transition probability function, defined as $P(s' \mid s, a)$, represents the probability of transitioning to a new state s' when action a is taken in state s. R is the reward function which provides the expected immediate reward received after taking action a in state s and transitioning to state s', and γ is the discount factor. Based on the defined MDP, a Q-function can be approximated by a ANN and the predicted Q-value, denoted as Q_θ, can learned by minimizing the Bellman error as shown in Eq. 1:

$$L_{\mathrm{Q}}(\theta) = \mathbb{E}_{(s,a,r,s')\sim\mathcal{D}}\left[\left(Q_\theta(s,a) - \left(r + \gamma \max_{a'} Q_\theta(s',a')\right)\right)^2\right] \tag{1}$$

However, directly applying Q-learning in an offline RL setting can result in suboptimal training performance due to the presence of out-of-distribution (OOD) actions (Kumar et al., 2020). In online Q-learning, the control policy frequently interacts with the environment, allowing it to receive accurate reward feedback for each action taken. Conversely, in offline RL, the correct reward information cannot be obtained for actions if the action-reward pair is absent in the training dataset, leading to Q-values that are poorly estimated. This limitation may lead the policy to overestimate Q-values for these OOD actions, and the resulting inaccurate Q-value estimates can lead to incorrect control decisions by the policy.

Therefore, one of the state-of-art offline RL algorithms such as the Implicit Q-Learning (IQL) is designed to penalize the overestimation of the Q-values (Kostrikov et al., 2021). IQL penalize the OOD action by adding the penalty term into the objective function which discourages high Q-values for unsupported actions:

$$\mathcal{L}_{\mathrm{IQL}} = \mathbb{E}_{(s,a,r,s')\sim D}\left[\left(Q(s,a) - (r + \gamma V(s'))\right)^2 + \alpha \max\left(Q(s,a) - \tau, 0\right)^2\right] \tag{2}$$

where $V(s')$ represents the value function for the state s', α is the hyperparameter that controls the weight of the penalty term, and τ is the threshold that defines the conservative level in Q-values. In this equation, the penalty term is $\max\left(Q(s,a) - \tau, 0\right)^2$, which penalize the Q-value that exceed the value of τ. In this way, the OOD actions which may lead to over optimistic prediction of Q-values are penalized, thus the agent is encouraged to

rely more on actions well-supported in the data, reducing the risk of overestimating values for actions that are not representative of the actual system dynamics.

Instead of updating the control policy by directly maximizing the Q-value, an implicit policy improvement step using advantage-weighted actions is applied in the IQL algorithm. The advantage function $A(s, a)$ estimates the difference between Q-value and the value function, which can show how much better (or worse) action a is compared to the average action the policy would take in state s. IQL algorithm embeds a soft advantage-weighted policy update as shown in Eq. 3:

$$\pi^*(a|s) \propto \exp\left(\frac{A(s,a)}{\beta}\right) \pi_{\text{BC}}(a|s) \tag{3}$$

where $A(s, a) = Q(s, a) - V(s)$ is the advantage function, β controls how sharply the policy favors high-advantage actions, and $\pi_{\text{BC}}(a|s)$ is the empirical behavior cloning prior. Frequent, high-reward dataset actions yield higher advantages, thus penalizing out-of-distribution actions. The algorithm table detailing the implementation of Q-Learning (IQL) is provided in Algorithm. 1:

Algorithm 1 Implicit Q-Learning Algorithm

1: **Input:** Dataset $\mathcal{D}$ of transitions, discount factor γ, penalty weight α, temperature parameter β.
2: **Initialize:** Q-function $Q(s, a)$ and value function $V(s)$.
3: **for** each iteration **do**
4: Sample a mini-batch (s, a, r, s') from the dataset $\mathcal{D}$.
5: **Value function update:** Estimate the value function $V(s)$ using the Q-values:
$$V(s) = \mathbb{E}_{a\sim\pi(\cdot|s)}[Q(s, a)]$$
6: **Q-function update:** Compute the Implicit Q-learning loss with a penalty to discourage overestimation:
$$L_{\text{IQL}} = \big(Q(s,a) - \big(r + \gamma V(s')\big)\big)^2 + \alpha \max\left(Q(s,a) - \tau, 0\right)^2$$
7: Update Q-function parameters by minimizing L_{IQL}.
8: **Advantage-weighted policy improvement:** Compute the advantage function $A(s, a)$:
$$A(s, a) = Q(s, a) - V(s)$$
9: Update the policy with advantage-weighted softmax:
$$\pi^*(a \mid s) \propto \exp\left(\frac{A(s,a)}{\beta}\right) \pi_{\text{BC}}(a \mid s)$$
10: **end for**

2.2 Online Finetuning

To enhance adaptability and robustness, the IQL-trained agent undergoes online fine-tuning with Twin Delayed Deep Deterministic Policy Gradient (TD3), using an experience replay buffer that mixes offline historical data and new real-time interactions. The replay buffer employs a dynamically updated FIFO structure, prioritizing recent experiences. Exploration during fine-tuning is encouraged by adding Ornstein-Uhlenbeck (OU) noise to actions, enabling smooth and continuous action exploration:

$$a' = \pi_\theta(s) + x_t \tag{4}$$

with OU noise updated as:

$$x_{t+1} = \theta(\mu - x_t) + \sigma \cdot \mathcal{N}(0, 1) \tag{5}$$

where denotes the mean reversion rate, the mean, the volatility, and Gaussian noise.

The TD3 objective combines a Q-learning loss for the critic networks:

$$L_Q(\theta) = \left(Q_\theta(s,a) - \left(r + \gamma \min_{i=1,2} Q_{\theta_i}(s', \pi_\theta(s'))\right)\right)^2 \tag{6}$$

with delayed policy updates to maximize the critic's estimated Q-value:

$$\nabla_{\theta^\pi} J \approx \frac{1}{N} \sum \nabla_a Q_{\theta_1}(s,a) \nabla_{\theta^\pi} \pi_\theta(s) \tag{7}$$

This integration of TD3 improves stability and robustness during online fine-tuning by introducing controlled action noise, promoting exploration without generating extreme or unseen actions that could destabilize learning. The algorithm for the online fine-tuning procedure is summarized in Algorithm. 2.

Algorithm 2 Mixed Experience Replay with Twin Delayed Deep Deterministic Policy Gradient algorithm

1: **Input**: Initialize dataset D with offline transitions and set up FIFO replay buffer B with capacity limit L. Define discount factor γ.
2: **Initialize**: Initialize Q-function $Q_\theta(s, a)$ (critic) and policy $\pi_\theta(s)$ (actor).
3: **For each iteration do**
Sample a mini-batch (s, a, r, s') from B.
Critic Update: Compute the Q-learning loss for the critic:
$$L_Q(\theta) = \big(Q_\theta(s,a) - \big(r + \gamma Q_\theta(s', \pi_\theta(s'))\big)\big)^2$$
Experience Mixture Sampling: Sample a mixture of offline and online experiences from B based on sampling ratio α:
$$\text{Mini-batch} = \alpha \cdot \text{Online Experiences} + (1-\alpha) \cdot \text{Offline Experiences}$$
Exploration Strategy:
For each action a chosen by the policy, add exploratory action noise. The action is calculated as:
$$a' = \pi_\theta(s) + x_t$$
Update the Ornstein-Uhlenbeck process noise x_t:
$$x_{t+1} = \theta(\mu - x_t) + \sigma \cdot \mathcal{N}(0, 1)$$
Actor Update: Update the policy (actor) by taking a gradient step to maximize the Q-value estimated by the critic:
$$\nabla_{\theta^\pi} J \approx \frac{1}{N} \sum \nabla_a Q_\theta(s,a) \nabla_{\theta^\pi} \pi_\theta(s)$$
Buffer Update: Add new online experience (s, a, r, s') to B, ensuring the buffer size limit L is maintained.
4: **Repeat until convergence**: Continue the process, gradually increasing the ratio of online experiences in B as more data is collected in the online environment.

In this work an offline RL framework is developed by applying Implicit Q-learning (IQL) algorithm to develop an optimal control strategy for bioprocess optimization from historical data that previously collected from bioprocess operation. The methodology consists of three steps: First, historical bioprocess data is prepared for IQL training. This dataset, collected across diverse operating conditions and policy variations, provides a comprehensive foundation of states, actions, and rewards, allowing the RL agent to learn system dynamics without online interaction. Second, a control policy is trained on this offline dataset using IQL, which applies a penalty to reduce Q-values for actions outside the dataset distribution, thus maintaining a conservative approach.

Finally, the offline-trained policy is fine-tuned online through the integration of TD3 algorithm with Experience Replay, allowing it to adapt to real-time bioprocess conditions. By learning from a blend of offline and online data, the policy transitions smoothly to the real-time environment while retaining knowledge from the offline phase. This integrated approach enhances the effectiveness and stability of the policy in optimizing bioprocess operations. The full methodology is summarized in the Algorithm 3.

Algorithm 3 IQL-based Offline Policy Optimization with TD3 Online Fine-tuning

1: **Input**: Historical bioprocess operational data from past operations across diverse conditions.
2: **Output**: Robust optimized control policy for real-time bioprocess operation
3: **Step 1: Offline Dataset Preparation**
4: Collect historical bioprocess operation data.
5: Preprocess data into state-action-reward tuples.
6: **Step 2: Offline Training using IQL**
7: Initialize IQL agent.
8: Train agent using the offline dataset.
9: Apply advantage-weighted penalties to limit out-of-distribution actions.
10: Optimize conservative Q-function to remain within dataset distribution.
11: **Step 3: Online Fine-Tuning with TD3**
12: Initialize Experience Replay Buffer (FIFO) with offline data.
13: Continuously collect new online bioprocess data.
14: Dynamically update replay buffer, retaining recent experiences.
15: Fine-tune agent using TD3 algorithm:
16: Update critic networks using TD3 targets.
17: Update actor with delayed policy updates for stability.
18: Apply Ornstein-Uhlenbeck noise for controlled exploration.

It is worth mentioning that the proposed framework is flexible to incorporate other offline RL algorithms such as the Conservative Q-Learning (Kumar et al., 2020), Batch-Constrained Q-Learning (BCQ) (Fujimoto et al., 2019) and Decision Transformer (Chen et al., 2021) instead of the IQL algorithm.

The proposed methodology is tested by optimizing a fed-batch photobioreactor. This biosystem models the dynamic changes in biomass growth, nitrate consumption, and bioproduct formation, represented by a system of ordinary differential equations (ODEs) based on Monod kinetics. The reactor volume is assumed to remain constant throughout the process. It is assumed that the volume of the fed-batch reactor remains constant throughout the operational process. This bioprocess is controlled by regulating the feeding rate of nitrate F_N ($\mathrm{mg \cdot L^{-1} \cdot h^{-1}}$) and the light intensity I ($\mu\mathrm{mol \cdot m^{-2} \cdot s^{-1}}$).The effects of light intensity on microalgae growth and bioproduct formation, incorporating the phenomena of photolimitation, photosaturation, and photoinhibition are described the Aiba model (Bradford et al., 2020). The biosystem is represented from Eq. 8.a to Eq. 8.c.

$$\frac{dC_X}{dt} = u_m \cdot \frac{I}{1 + k_s + \frac{I^2}{k_i}} \cdot C_X \cdot \frac{C_N}{C_N + K_N} - u_d \cdot C_X \tag{8.a}$$

$$\frac{dC_N}{dt} = -Y_{NX} \cdot u_m \cdot \frac{I}{1 + k_s + \frac{I^2}{k_i}} \cdot C_X \cdot \frac{C_N}{C_N + K_N} + F_N \tag{8.b}$$

$$\frac{dC_{qc}}{dt} = k_m \cdot \frac{I}{1 + k_{sq} + \frac{I^2}{k_{iq}}} \cdot C_X - \frac{k_d C_{qc}}{C_N + K_{Np}} \tag{8.c}$$

where C_X is the biomass concentration (g/L), C_N is the nitrate concentration (mg/L), and C_{qc} is the concentration of phycocyanin (bioproduct) in the photobioreactor (mg/L). The control actions are implemented as piecewise constant values over fixed time intervals, with each interval set to 10 hours in this case. During the bioreactor operation, a total of 20 control actions are executed. The objective of the bioprocess optimization is to find the optimal control actions that can maximize the concentration of the bioproduct throughout the entire process while simultaneously penalizing the control actions to account for economic considerations. Consequently, the optimal control problem for this bioprocess can be defined as follows:

$$\begin{aligned} \max_{\mathbf{u}} J_{OCP} = \quad & \sum_{t=0}^{\mathcal{T}} C_{qc}(t) - \mathrm{penalty}_I \cdot \sum_{i \in \mathcal{T}} \left(\frac{I(i)}{400} \right)^2 \\ & - \mathrm{penalty}_{F_N} \cdot \sum_{i \in \mathcal{T}} \left(\frac{F_N(i)}{40} \right)^2 \end{aligned} \tag{9.a}$$

$$\text{s.t.} \quad \dot{\mathbf{x}} = f(\mathbf{x}(t), \mathbf{u}(t), t) \tag{9.b}$$

$$\mathbf{x}(0) = \mathbf{x}_0 \tag{9.c}$$

$$0 \leq F_N(t) \leq 40 \tag{9.d}$$

$$0 \leq I(t) \leq 400 \tag{9.e}$$

where the J_{OCP} is the objective function, $\mathrm{penalty}_I$ and $\mathrm{penalty}_{F_N}$ are assigned values of 0.0001 and 0.008 respectively, to penalize the corresponding control actions. The state vector and control vector for this optimal control problem are defined as $\mathbf{x} = [C_X, C_N, C_{qc}]^T$ and $\mathbf{u} = [I, F_N]^T$, the initial condition is represented as $\mathbf{x}_0 = [C_{X0}, C_{N0}, C_{qc0}]^T$, and both control actions are limited

to be below 40 $\text{mg} \cdot \text{L}^{-1} \cdot \text{h}^{-1}$ and 400 $\mu\text{mol} \cdot \text{m}^{-2} \cdot \text{s}^{-1}$ respectively during the whole operating horizon.

In this case, 1,000 trajectories, each containing 20 data points, are generated as in-silico historical data to train the control agent. These trajectories are produced by varying the initial operating conditions of the biosystem to simulate industrial conditions, the total number of data points in each trajectory is kept relatively low to reflect the limited data availability typically encountered in bioprocess engineering. The variations are introduced by sampling values within ±30% of the baseline conditions for the initial biomass concentration C_{X0} and initial nitrate concentration C_{N0}, set at 1.0 g/L and 150 mg/L, respectively. In this case, the optimal control solutions are solved in the Python optimization environment Pyomo (Hart et al., 2017), by firstly discretise the NLP problem through collocation and then solving the problem using the Interior Point Optimization (IPOPT) as nonlinear solver. The objective function of this optimal control problem is also used for the estimation of the stage cost for the IQL training.

4. RESULT AND DISCUSSION

The training of control agents using the Implicit Q-Learning (IQL) and Behavioral Cloning (BC) algorithms was conducted using an open-source offline deep reinforcement learning library provided by Seno and Imai (2022). To enhance the training efficiency, all states and rewards were normalized using standard normalization techniques, and actions are scaled to the range [-1, 1] to align with the tanh activation function in the actor neural network. The offline-trained IQL agent was subsequently fine-tuned online using the TD3 algorithm, following the proposed offline-to-online fine-tuning methodology.

The online fine-tuning was conducted using the same mechanistic bioprocess model described by Eqs. 8.a to 8.c. For the online evaluation phase, this mechanistic model was again utilized, by assuming the mechanistic model represents the ground truth of the bioreaction kinetics. All three agents were tested over 50 episodes under varying system disturbance to assess robustness and operational effectiveness. Comparative performance results of these agents are illustrated in Fig. 1.

As shown in Fig. 1, both the IQL-trained and BC-trained offline policies effectively manage nitrate flowrate, exhibiting overlapping mean trajectories and confidence intervals despite measurement noise, indicating a consistent approach to nitrate management. However, for controlling light intensity, the BC-trained policy displays greater fluctuations and broader uncertainty bands, reflecting more variability in response, likely due to stronger penalization associated with deviations in light intensity.

In contrast, the TD3-fine-tuned agent demonstrates significantly reduced uncertainty and achieves higher cumulative rewards (Fig. 2), outperforming both directly applied offline-trained policies. This improvement is expected and reasonable, given that offline reinforcement learning inherently produces conservative policies that avoid unfamiliar, potentially risky actions. Thus, the introduction of TD3 for offline-to-online fine-tuning effectively addresses the

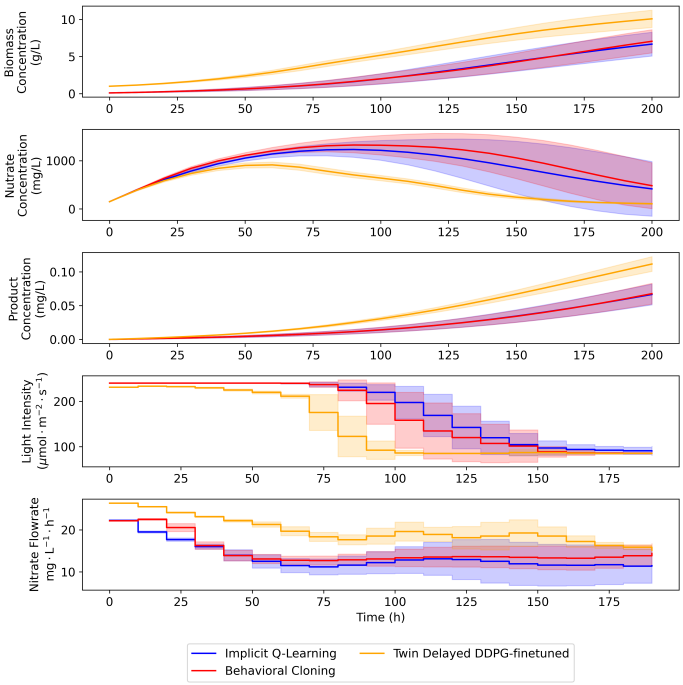

Fig. 1. Average states and actions during the online implementation of control agents trained with Implicit Q-Learning, Behavioral Cloning and TD3 finetuned policy. Shaded areas indicate the standard deviation caused by measurement disturbances in the states.

overly cautious nature of offline RL, enabling the agent to better adapt and robustly perform in real-time bioprocess operations. This highlights the advantage of leveraging TD3 integration to bridge the gap between conservative offline learning and dynamic online application.

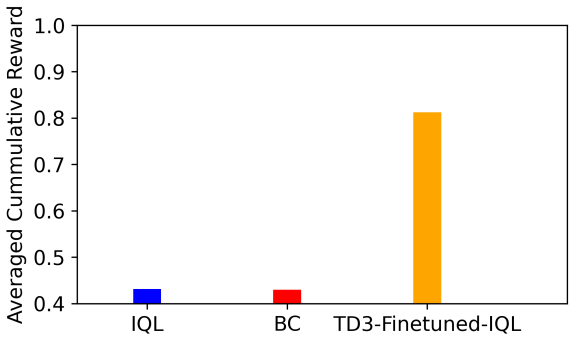

Fig. 2. Average cumulative reward.

5. CONCLUSION

This work proposes an offline Reinforcement Learning (RL) framework for bioprocess optimization, leveraging historical process data to train control agents without direct interaction with the real biosystem. The results demonstrate that offline RL algorithms, particularly Implicit Q-Learning (IQL), effectively learn optimal control policies from historical datasets. Through a case study, the IQL-trained agent demonstrated robust performance in maximizing bioproduct concentration even under system

disturbances, outperforming the Behavioral Cloning (BC)-trained agent in cumulative reward.

Moreover, integrating the Twin Delayed Deep Deterministic Policy Gradient (TD3) algorithm for online fine-tuning on the same mechanistic model substantially enhanced the performance of agents, reducing uncertainty and further increasing cumulative reward. This integration addresses the conservatism inherent in offline RL, enabling effective adaptation to real-time operational conditions.

Future directions for this framework include exploring safer online fine-tuning methodologies, such as Safe RL techniques, to further minimize risks during real-time adaptation. Additionally, leveraging transfer learning for online fine-tuning will be considered to improve efficiency and scalability. Ultimately, the framework will be extended to test more complex bioprocess systems subject to higher degrees of disturbance, validating the robustness and adaptability of the offline-trained RL agents.

REFERENCES

Bradford, E., Imsland, L., Zhang, D., and del Rio Chanona, E.A. (2020). Stochastic data-driven model predictive control using gaussian processes. *Computers Chemical Engineering*, 139, 106844. doi:https://doi.org/10.1016/j.compchemeng.2020.106844.

Brennan, L. and Owende, P. (2010). Biofuels from microalgae—a review of technologies for production, processing, and extractions of biofuels and co-products. *Renewable and Sustainable Energy Reviews*, 14(2), 557–577. doi:https://doi.org/10.1016/j.rser.2009.10.009.

Chen, L., Lu, K., Rajeswaran, A., Lee, K., Grover, A., Laskin, M., Abbeel, P., Srinivas, A., and Mordatch, I. (2021). Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems*, 34, 15084–15097.

Del Rio-Chanona, E.A., Cong, X., Bradford, E., Zhang, D., and Jing, K. (2019). Review of advanced physical and data-driven models for dynamic bioprocess simulation: Case study of algae–bacteria consortium wastewater treatment. *Biotechnology and Bioengineering*, 116(2), 342–353. doi:https://doi.org/10.1002/bit.26881.

Fujimoto, S., Meger, D., and Precup, D. (2019). Off-policy deep reinforcement learning without exploration. In *International conference on machine learning*, 2052–2062. PMLR.

Hart, W.E., Laird, C.D., Watson, J.P., Woodruff, D.L., Hackebeil, G.A., Nicholson, B.L., Siirola, J.D., et al. (2017). *Pyomo-optimization modeling in python*, volume 67. Springer.

Kostrikov, I., Nair, A., and Levine, S. (2021). Offline reinforcement learning with implicit q-learning. *arXiv preprint arXiv:2110.06169*.

Kumar, A., Zhou, A., Tucker, G., and Levine, S. (2020). Conservative q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 33, 1179–1191.

Levine, S., Kumar, A., Tucker, G., and Fu, J. (2020). Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*.

Ma, Y., Wang, Z., Castillo, I., Rendall, R., Bindlish, R., Ashcraft, B., Bentley, D., Benton, M.G., Romagnoli, J.A., and Chiang, L.H. (2021). Reinforcement learning-based fed-batch optimization with reaction surrogate model. In *2021 American Control Conference (ACC)*, 2581–2586. doi:10.23919/ACC50511.2021.9482807.

Monteiro, M. and Kontoravdi, C. (2024). Bioprocess control: A shift in methodology towards reinforcement learning. In F. Manenti and G.V. Reklaitis (eds.), *34th European Symposium on Computer Aided Process Engineering / 15th International Symposium on Process Systems Engineering*, volume 53 of *Computer Aided Chemical Engineering*, 2851–2856. Elsevier. doi:https://doi.org/10.1016/B978-0-443-28824-1.50476-2.

Mowbray, M., Smith, R., Del Rio-Chanona, E.A., and Zhang, D. (2021). Using process data to generate an optimal control policy via apprenticeship and reinforcement learning. *AIChE Journal*, 67(9), e17306. doi:https://doi.org/10.1002/aic.17306.

Narayanan, H., Luna, M.F., von Stosch, M., Cruz Bournazou, M.N., Polotti, G., Morbidelli, M., Butté, A., and Sokolov, M. (2020). Bioprocessing in the digital age: the role of process models. *Biotechnology journal*, 15(1), 1900172.

Nikita, S., Tiwari, A., Sonawat, D., Kodamana, H., and Rathore, A.S. (2021). Reinforcement learning based optimization of process chromatography for continuous processing of biopharmaceuticals. *Chemical Engineering Science*, 230, 116171. doi:https://doi.org/10.1016/j.ces.2020.116171.

Oh, T.H., Park, H.M., Kim, J.W., and Lee, J.M. (2022). Integration of reinforcement learning and model predictive control to optimize semi-batch bioreactor. *AIChE Journal*, 68(6), e17658. doi:https://doi.org/10.1002/aic.17658.

Petsagkourakis, P., Sandoval, I.O., Bradford, E., Zhang, D., and del Rio-Chanona, E.A. (2020). Reinforcement learning for batch bioprocess optimization. *Computers & Chemical Engineering*, 133, 106649.

Sachio, S., del Rio Chanona, A.E., and Petsagkourakis, P. (2021). Simultaneous process design and control optimization using reinforcement learning. *IFAC-PapersOnLine*, 54(3), 510–515. doi:https://doi.org/10.1016/j.ifacol.2021.08.293. 16th IFAC Symposium on Advanced Control of Chemical Processes ADCHEM 2021.

Seno, T. and Imai, M. (2022). d3rlpy: An offline deep reinforcement learning library. *Journal of Machine Learning Research*, 23(315), 1–20.

Spielberg, S., Tulsyan, A., Lawrence, N.P., Loewen, P.D., and Bhushan Gopaluni, R. (2019). Toward self-driving processes: A deep reinforcement learning approach to control. *AIChE Journal*, 65(10), e16689. doi:https://doi.org/10.1002/aic.16689.