Competition 10: Industrial Applications of Offline Reinforcement Learning

1. Competition Background

With the wave of intelligent manufacturing sweeping across the globe, traditional industrial control systems are facing increasingly severe challenges in improving production efficiency, reducing energy consumption, and ensuring safe operations. Conventional methods that rely on expert experience and manual parameter tuning are no longer sufficient to cope with the growing complexity of industrial environments and diverse dynamic disturbances. In recent years, reinforcement learning has emerged in the field of adaptive control, demonstrating remarkable potential. However, its deployment in real industrial scenarios is constrained by the high costs and safety risks associated with online experimentation. In contrast, offline reinforcement learning leverages historical operational data and condition records to train efficient and stable control policies in risk-free simulation environments, opening a new pathway for the automation and intelligent upgrading of industrial systems.

2. Competition Application Scenario

In process industries such as steel, chemical, and energy, control systems must balance multiple objectives, including production output, energy consumption, and equipment lifespan. Traditional control strategies based on expert experience and PID tuning often fall short in the following aspects:

(1) Insufficient robustness under complex couplings: As multivariable interconnections and strong noise disturbances increase, relying solely on experience and linear PID control cannot ensure system stability.

(2) Difficulty maintaining optimality under load and feedstock fluctuations: When production rhythms and raw material compositions change frequently, manually tuned control parameters cannot adapt quickly, resulting in suboptimal output and energy efficiency.

(3) Poor adaptability in cross-condition transfer: Each process switch requires significant tuning time, severely limiting flexible scheduling and rapid response of production lines.

Offline Reinforcement Learning (Offline RL), by deeply mining historical production data and learning control policies in risk-free simulation or replay buffer environments, offers the following advantages:

(1) Risk isolation: All experiments are conducted in simulated environments or replay buffers, eliminating safety hazards and downtime risks for real equipment.

(2) Efficient exploration: With deep networks, the system can identify the most valuable historical trajectories and automatically discover parameter combinations that achieve “minimal trial-and-error → maximum return.”

(3) Strong transferability: The learned policy embodies generalizable knowledge of the state–action mapping, enabling rapid adaptation to different conditions with only minimal online fine-tuning.

(4) Cost reduction and efficiency improvement: By avoiding repeated offline trial-and-error and manual tuning, Offline RL achieves plug-and-play intelligent control, significantly shortening switching cycles, reducing energy consumption, and extending equipment lifespan.

3. Problem-Setting Information

This competition problem is designed by an expert team organized by the competition committee. The organizers are Nanjing University and Polixir. Problem setters from the organizers: Yang Yu , Rongjun Qin, Songyi Gao, Zuolin Tu.

4. Competition Task

In this competition, participants will be provided with a high-fidelity industrial control simulator along with an accompanying offline dataset. The dataset is generated through interactions between the simulator and suboptimal control strategies, aiming to realistically reproduce complex control problems in industrial scenarios. Participants are required to design and train offline reinforcement learning models based on the provided dataset to achieve precise control of target variables. The control performance will be evaluated by the cumulative reward, with the ultimate goal of demonstrating strong control effectiveness and robustness under the following key challenges:

(1) Temporal delay: The effect of control commands does not appear immediately, leading to significant time lag.

(2) Complex noise: Observation data are mixed with multiple types of noise, simulating typical disturbances in industrial sensor readings.

(3) Partial observability (POMDP): The agent can only access partial information about the system state and must infer the global state and make reasonable decisions under incomplete observations.

Participant teams must strictly adhere to the offline dataset provided by the organizers for algorithm development and policy training. The learned control policy must achieve the highest possible cumulative reward in the simulation environment to demonstrate its control performance and robustness.

5. Dataset and Data Description

The offline dataset provided in this competition is generated by a high-fidelity industrial control simulator. The preliminary dataset is only for participants to validate and debug their algorithms. In the semifinal, a new dataset will be provided (with the same data format as the

preliminary dataset but with different simulator parameters). This design simulates the challenge in industry where “opportunities for online testing are limited,” thereby selecting algorithms and models with stronger practicality and deployability.

Details of the dataset are as follows:

5.1 Observation Space

(1) Dimension: 5

(2) Value range: each observation lies within the interval [-1, 1]

(3) Description: represents the observable information of the system

5.2 Action Space

(1) Dimension: 3

(2) Value range: each action lies within the interval [-1, 1]

5.3 Data Structure

(1) Number of trajectories: 100

(2) Time steps: each trajectory contains 1,000 time steps

(3) Total data points: 100,000 (100 trajectories × 1,000 time steps)

5.4 Data contained in each time step

(1) obs: observation data at the current time step

(2) action: the action taken by the suboptimal control strategy

(3) next_obs: observation data after state transition

(4) reward: reward value obtained at the current time step

(5) index: trajectory identifier; data points within the same trajectory share the same index

The dataset is provided in CSV format. Example data can be accessed at:

https://pan.baidu.com/s/1tDFjWFOhAJ8rOBl_HN79ww?pwd=48ux

6. Performance Metrics Requirements

This competition uses the cumulative reward value as the primary evaluation metric. Participants must submit their trained agent policy models. The testing server will interact with the simulator using the submitted model to generate multiple trajectories. The system will then compute the average cumulative reward across all trajectories, which will serve as the final evaluation score of the model.

7. Functional Requirements

Algorithm Design Requirements:

7.1 Training Method:

End-to-end training must be completed based on the offline dataset.

7.2 Robustness Handling:

1.Must exhibit robustness to time delays, ensuring control decisions remain reliable under varying delay conditions.

2.Must exhibit robustness to noise, effectively suppressing the impact of environmental noise on control performance.

3.Must address the issue of partial observability in the state space, ensuring decision-making capability under incomplete information.

7.3 Control Performance:

1.Must achieve high cumulative reward values in the simulation environment.

2.Must demonstrate stable control performance to ensure reliable system operation.

7.4 Model Characteristics:

1.Must have strong generalization ability to adapt to different environmental conditions.

2.Must be scalable to facilitate deployment in various application scenarios.

8. Development Environment

Participants must use Python for development and upload their models to the model validation service platform for scoring. It is recommended that participants ensure their development environment is consistent with the provided testing environment configuration to avoid runtime issues caused by environment differences.

9. Evaluation Criteria

This competition consists of three stages: the preliminary, the semifinal, and the final.

9.1 Preliminary Round:

The dataset provided in the preliminary round is only for participants to validate and debug their algorithms in the early stage. The submitted models will be automatically scored by the system, but the scores will not count toward the final total score.

9.2 Semifinal Round:

A new dataset will be provided in the semifinal round (with the same format as the preliminary dataset, but with changed simulator parameters). The submitted models will be automatically scored by the system, and the scores will count toward the final total score.

9.3 Offline Final:

The final comprehensive score consists of objective and subjective evaluations, with weights of 70% and 30%, respectively. (For the second and third prizes at the national level, the 30% subjective evaluation component is not included.)

(1) Objective evaluation: Based on the standardized semifinal machine evaluation score.

(2) Subjective evaluation: Based on the standardized defense score. The defense evaluation will comprehensively assess the participants' presentation performance, as well as the submitted technical solution and code documentation.

10. Problem-Solving Approach

10.1 Policy Training:

Adopt offline reinforcement learning algorithms to train policies using historical data samples and build an optimal control policy model.

10.2 Policy Offline Evaluation and Tuning:

Construct a high-fidelity virtual environment to conduct closed-loop testing and validation of policies. By monitoring and analyzing cumulative reward metrics in real time, establish a feedback optimization mechanism to continuously improve control strategies.

10.3 Key Technical Challenges:

(1) Design compensation algorithms to address control command delays.

(2) Apply noise filtering to mitigate sensor noise interference.

(3) Introduce state reconstruction methods based on deep neural networks to handle partial observability in the state space.

11. References and Resources

Textbooks, research papers, and open-source code repositories related to reinforcement learning and offline reinforcement learning.

12. Submission Requirements

In this competition, participants are required to submit their trained policy models (or inference code plus model weights). The testing server will use the submitted model to interact with the simulator and generate a batch of trajectories. The final score will be calculated as the mean cumulative reward across these trajectories.

Participants must package all files into a single .zip file, with the top-level directory structure strictly following the format below:

```
submission.zip
├── agent/
│   ├── agent.py              # policy implementation (inherit BaseAgent, implement PolicyAgent and get_action; a reference is provided)
│   ├── model.pth             # policy model weights
├── evaluator.py              # sample simulator for testing policy
├── test_agent.py             # policy test script (no need to modify; run with simulator to verify implementation)
```

The example code will be provided as a Git repository. Participants may download it, modify the policy implementation, and submit the package. The repository will be available after official registration.

Resources and Format Restrictions:

1.The maximum size of a submission package is 10 MB. If the model weights are too large, participants should design their own compression or pruning scheme.

2.Model testing time must not exceed 5 minutes; overtime will be treated as a failure.

3.Models will be tested in an offline environment and cannot access online services.

4.For complex models, it is recommended to export them in ONNX format.

13. Updates and Q&A

The competition problem may be updated. To address participants' questions during the contest, a dedicated Q&A group will be set up, which will be accessible after official registration.

14. Competition Process and Award Settings

14.1 Registration Stage

Participants complete registration on the official competition website, submit individual or team information, and obtain the download link for the preliminary dataset.

14.2 Preliminary Stage

Participants design algorithm models using the training dataset provided by the organizers. Each team is allowed only one submission per day during the preliminary stage.

14.3 Semifinal (Provincial) Stage

After the preliminary stage concludes, the semifinal stage begins, and the download link for the semifinal dataset is released. Only teams that submitted valid results in the preliminary stage are eligible to enter the semifinal. During the semifinal, participants debug their algorithms using the dataset provided by the organizers and submit inference results on the semifinal test data. The semifinal lasts for 3 days, and each team may submit only once per day.

14.4 Semifinal (Provincial) Results

Semifinal results will be published on the official competition website. The number of teams advancing to the semifinal serves as the award base. According to the award ratio set for the provincial stage, first, second, and third prizes will be selected (provincial award certificates will be issued). If a team's submitted algorithm performance falls below the baseline reference score provided by the organizers, it will be deemed invalid and will not be eligible for awards. Teams awarded first and second prizes in the semifinal will advance to the national final.

14.5 Final (National) Stage

(1) Online Evaluation: Based on the semifinal leaderboard results, the number of teams entering the final serves as the award base. According to the award ratio set for the national stage, a list of national first prize candidates and winners of the national second and third

prizes will be selected (certificates for national second and third prizes will be issued).

(2) Final Submission: National first prize candidate teams must submit technical documentation, algorithm code and model files, demonstration videos, and supplementary materials within the specified timeframe. No modifications or additions will be accepted after the submission deadline.

(3) Final Review: A professional review panel will reproduce and verify the results of the submissions from the national first prize candidate teams. If any issues arise during the review process, participants may be asked to provide explanations.

(4) Onsite Final Defense: National first prize candidate teams must submit updated technical documentation, algorithm code and model files, demonstration videos, and supplementary materials within the specified timeframe, and participate in the onsite final defense of the national competition. The final list and rankings of the national first prize winners will be determined based on both algorithm performance scores and onsite defense scores (failure to attend the onsite defense will be regarded as forfeiture of the award). National first prize winners will be awarded honor certificates.

15. Additional Notes

15.1 Fairness:

Any form of cheating is strictly prohibited, including but not limited to data leakage, overlap between pretraining data and test data, and plagiarism of others' code. Once discovered, participants will be immediately disqualified, and relevant responsibilities will be pursued.

15.2 Intellectual Property:

Submitted works must be original and must not have won awards in other competitions or been publicly published. The organizers reserve the right to display and promote the submitted works for related activities, but the intellectual property rights remain with the participants.

16. Contact Information

Competition Q&A group on QQ: 437743461

Email: songyi.gao@polixir.ai

Official registration website: www.aicomp.cn