Optimizing Reward Models with Proximal Policy Exploration in Preference-Based Reinforcement Learning

Yiwen Zhu[1,2], Jinyi Liu[3], Yifu Yuan[3], Wenya Wei[1], Zhenxing Ge[4], Qianyi Fu[1], Zhou Fang[1,*], Yujing Hu[5], Bo An[2,6]
[1]Zhejiang University, [2]Nanyang Technological University, [3]Tianjin University, [4]Nanjing University, [5]NetEase Fuxi AI Lab, [6]Skywork AI
{evanzhu, wwy_vivian, qyfu, zfang}@zju.edu.cn
{jyliu, yuanyf}@tju.edu.cn, zhenxingge@smail.nju.edu.cn
huyujing@corp.netease.com, boan@ntu.edu.sg

Abstract

Traditional reinforcement learning (RL) relies on carefully designed reward functions, which are challenging to implement for complex behaviors and may introduce biases in real-world applications. Preference-based RL (PbRL) offers a promising alternative by using human feedback, yet its extensive demand for human input constrains scalability. To address that, this paper proposes a proximal policy exploration algorithm (**PPE**), designed to enhance the efficiency of human feedback by concentrating on near-policy regions. By incorporating a policy-aligned query mechanism, our approach not only increases the accuracy of the reward model but also reduces the need for extensive human interaction. Our results demonstrate that improving the reward model's evaluative precision in near-policy regions enhances policy optimization reliability, ultimately boosting overall performance. Furthermore, our comprehensive experiments show that actively encouraging diversity in feedback substantially improves human feedback efficiency.

1 Introduction

In reinforcement learning (RL), the reward function is pivotal as it specifies the learning objectives and guides agents toward desired behaviors. Traditional RL has seen significant achievements in complex domains such as gaming and robotics, largely due to the use of well-designed reward functions [1, 2, 3]. Yet, constructing these functions presents significant challenges. The intricate process of designing suitable reward functions that accurately encapsulate complex behaviors like cooking or summarizing books is both time-consuming and prone to human cognitive biases [4, 5, 6, 7, 8]. Additionally, embedding social norms into these functions remains an unresolved issue [9].

An emerging alternative that addresses these challenges is preference-based reinforcement learning (PbRL), also known as RL from human feedback (RLHF). This approach bypasses the need for meticulously engineered rewards by leveraging (human) overseer preferences between pairs of agent behaviors [10, 11, 12, 13, 14, 15, 16, 17]. In PbRL, agents learn to optimize behaviors that align with the demonstrated human preferences, offering a more intuitive and flexible method for teaching desired outcomes. This not only enables a more natural communication of complex desired behaviors but also aligns the agents' actions more closely with human values.

*corresponding author

NeurIPS 2024 Workshop on Behavioral Machine Learning.

Despite its advantages, PbRL has notable limitations. The approach heavily depends on human input, and labeling a vast number of preference queries can be labor-intensive, potentially limiting its applicability in real-world settings where rapid adaptation is essential[13, 14, 15]. Furthermore, PbRL typically requires extensive human feedback, which can be time-consuming or sometimes infeasible to gather. To overcome these challenges, prior research has explored various strategies for improving feedback efficiency. These strategies include selecting the most informative queries to improve the quality of the learned reward function while minimizing the required teacher input [12, 18, 19, 20]. Also, techniques such as sampling based on ensemble disagreements, mutual information, or behavior entropy have been employed to target behaviors to refine the overall reward model more effectively [10, 13, 16, 18, 20]. Moreover, query-policy alignment (QPA) [21] method ensures that both queries and policy learning progress concurrently, significantly reducing feedback unrelated to the current policy, thereby enhancing feedback efficiency. However, these methods overlook the investigation of the relationship between the preference buffer and the effectiveness of the reward model. This oversight can lead the reward model to inaccurately evaluate data that is out of the preference buffer's distribution, potentially leading to misguided policy improvements.

To address this issue, we conducted a study focusing on enhancing the coverage of the preference buffer. We found that the learned reward model provides more accurate evaluations for trajectories that fall within the preference buffer's distribution. Based on this insight, we developed the Proximal Policy Exploration (PPE) algorithm. This approach encourages the agent to explore data that is out of the preference buffer's distribution but close to the current policy, thereby indirectly increasing the preference buffer's coverage and enhancing the reliability of the reward model's evaluations for the near-policy distribution.

2 Preliminaries

In PbRL, we consider an agent that interacts with an environment in discrete time steps. At each time step t, the agent receives a state s_t from the environment and selects an action a_t based on its policy. Unlike traditional RL, where the environment returns a reward $r(s_t, a_t)$ evaluating the agent's behavior, PbRL employs human feedback. Here, a teacher provides preferences between pairs of agent behaviors, which the agent uses to adjust its policy [10, 11, 12, 22, 23].

Formally, a behavior segment τ consists of a sequence of time-indexed observations and actions $\{(s_1, a_1), \ldots, (s_H, a_H)\}$. The teacher indicates their preferences among these segments, identifying preferred behaviors or marking segments as equally preferred or incomparable. The primary objective in PbRL is to train the agent to perform behaviors aligned with human with minimal feedback.

The PbRL learning process involves two main steps: (1) *Agent Learning*: The agent interacts with the environment to collect experiences and updates its policy using existing RL algorithms to maximize the sum of proxy rewards. (2) *Reward Learning*: The reward model $\hat{r}_\psi$ is optimized based on feedback received from the teacher, denoted as $(\tau^0, \tau^1, y_p) \sim \mathcal{D}_p$. This cyclical process continually refines both the policy and the reward model.

Using a preference dataset $\mathcal{D}_p$, the reward model $\hat{r}_\psi$ learns to assign higher proxy returns $\hat{G}_\psi = \sum_t \hat{r}_\psi(s_t, a_t)$ to preferred trajectories. Employing the Bradley-Terry model [24], the probability that one trajectory is preferred over another is computed as:

$$P_\psi(\tau^1 \succ \tau^0) = \frac{\exp\left(\sum_t \hat{r}_\psi(s_t^1, a_t^1)\right)}{\sum_{i\in\{0,1\}} \exp\left(\sum_t \hat{r}_\psi(s_t^i, a_t^i)\right)}. \tag{1}$$

The probability estimate P_ψ is used to minimize the cross-entropy between the predicted and true preference labels:

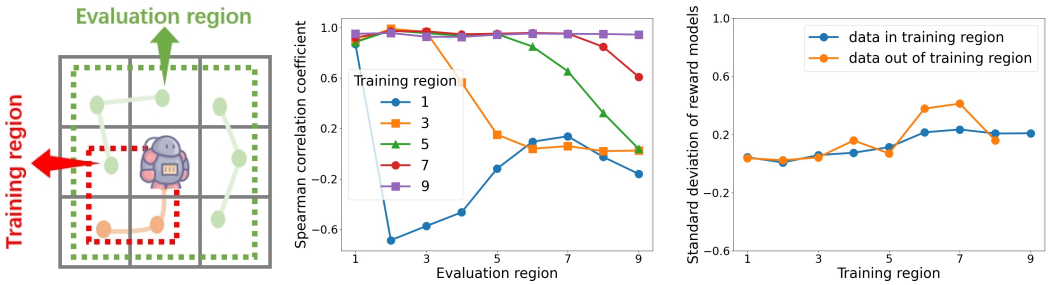
$$L_{CE} = -\mathbb{E}_{(\tau^0,\tau^1,y_p)\sim\mathcal{D}_p}\left[\mathbb{I}\{y_p = (\tau^0 \succ \tau^1)\} \log P_\psi(\tau^0 \succ \tau^1) + \mathbb{I}\{y_p = (\tau^1 \succ \tau^0)\} \log P_\psi(\tau^1 \succ \tau^0)\right]. \tag{2}$$

After optimizing the reward function $\hat{r}_\psi$ from human preferences, PbRL algorithms enable training of RL agents with standard RL algorithms, treating the proxy rewards from $\hat{r}_\psi$ as if they were direct rewards from the environment.

3 Method

3.1 Why Coverage Is Important? — A Motivating Example

We designed an experiment to observe the relationship between the effectiveness of the reward model and the coverage of transitions in the preference buffer used to train the reward model.



(a) (b) (c)

Figure 1: Observe the reward model's effectiveness in a random walk task with a grid world. **(a)**. Training the reward model with preference data generated from trajectory pairs within the training region marked by the red frame, and assessing the correlation between the proxy and ground truth returns across all trajectories in the evaluation region denoted by the green frame; **(b)**. The Spearman correlation coefficient between proxy returns and ground truth returns for all trajectories in various evaluation regions, using reward models trained with preference data from different training regions; **(c)**. The variance in the proxy rewards associated with transitions inside and outside of the training region changes in the size of the training region.

As shown in Figure 1, we set up an environment in a grid world where the robot can move in four directions: up, down, left, and right. Each cell in the grid world has an associated ground truth reward, which corresponds to a ground truth return for the robot's trajectory. It should be noted that Figure 1a serves as a schematic representation; in reality, the grid world is a 10x10 lattice. Additionally, the horizontal axes in Figures 1b and 1c represent the side lengths of the respective region.

To further experiment, we designated two areas within the grid world as the training region and the evaluation region, as illustrated in Figure 1a . First, we uniformly sampled 1,000 trajectory pairs of length 3 in the training region. Based on the relative sizes of their ground truth returns, we assigned preference labels to these trajectory pairs and stored them in a preference buffer. Next, we trained a reward model using the data from the preference buffer with a Bradley-Terry (BT) loss. Finally, we evaluated all trajectories of length 6 in the evaluation region using the learned reward model to determine their merit. The correlation between the proxy returns computed by the reward model and the ground truth returns were assessed using the Spearman correlation coefficient to further analyze the effectiveness of the reward model.

Results displayed in Figure 1b indicate that a larger training region enhances the ability of the reward model, learned from the corresponding preference buffer, to effectively evaluate the merits of trajectories. This phenomenon is intuitive yet underscores the critical importance of increasing the coverage of the preference buffer over the transition space. Consider the policy optimization process: if the preference buffer does not comprehensively cover the transition distribution associated with the current policy, the proxy rewards generated by the reward model may be unreliable, rendering the direction of policy optimization meaningless. It is only when the coverage of the preference buffer is extensive that the reward model, learned from it, can reliably evaluate a broader area. Based on this insight, it is essential to include the coverage of the preference buffer as an optimization target within the pipeline of PbRL.

Figure 1c demonstrates that the variance in outputs from ensemble reward models, given the same transition input, does not enable distinction of whether the transition belongs to the training region. Therefore, the method proposed by [15] cannot expand the preference buffer's coverage actively. Consequently, it is crucial to design an exploration method specifically aimed at actively enlarging the coverage of the preference buffer.

3.2 How to Improve Coverage of Preference Buffer? — Proximal Policy Exploration

3.2.1 Estimating the Uncertainty of Transitions via Random Network

To increase the coverage of the preference buffer, we propose a metric to measure the out-of-distribution (OOD) degree of current state-action pairs. Our approach is inspired by [25]. We utilize ensemble number n detection networks $d_{\theta_i}(s,a)$, for $i \in \{1,\ldots,n\}$, to distill a target random network $d_0(s,a)$ that does not learn during the process. This ensemble of networks is utilized to assess the transitions $(s,a) \in \tau$, wherein τ belongs to $\mathcal{D}_p$ and symbolizes the trajectory preserved in the preference buffer $\mathcal{D}_p$. The discrepancy $\sigma_\theta(s,a) = \max_i |d_{\theta_i}(s,a) - d_0(s,a)|$ serves as an indicator to determine whether a given state-action pair (s,a) is encompassed by $\mathcal{D}_p$.

3.2.2 Maximizing Preference Buffer Coverage via Proximal Policy Exploration

Algorithm 1 Proximal Policy Exploration

for Each interaction to the environment **do**
  $a_t \sim \pi_T(s_t)$ ▷ sample from the target policy
  if $\sigma(s_t,a_t) \leq 0.1$ **then** ▷ detect if (s_t,a_t) out of the preference buffer distribution
    pass
  else
    $a_t \sim \pi_E(s_t)$ ▷ resample from the behavior policy
  end if
  Using a_t to interact with the environment.
end for

We aim to develop a behavior policy π_E such that the state-action pairs (s,a) it generates when interacting with the environment can support the distribution produced by the current target policy π_T. This support is crucial as it enhances the transition distribution in the replay buffer, which in turn improves the distribution in the preference buffer used for training with respect to the current target policy π_T. Formally, the exploration policy $\pi_E = \mathcal{N}(\mu_E, \Sigma_E)$ is defined as follows:

$$\mu_E, \delta_E = \underset{\mu,\Sigma: Kl(\mathcal{N}(\mu,\Sigma)|\mathcal{N}(\mu_T,\Sigma_T)) \leq \epsilon}{argmax} \underset{a\sim\mathcal{N}(\mu,\Sigma)}{\mathbb{E}}[\sigma(s,a)]. \tag{3}$$

where the use of ϵ constrains the selection of the behavior policy to the vicinity of the current target policy. The closed-form solution for the parameters can be computed as:

$$\mu_E = \mu_T + \frac{\sqrt{2\epsilon} \cdot \Sigma_T[\nabla_a \sigma(s,a)]_{a=\mu_T}}{\sqrt{[\nabla_a \sigma(s,a)]^{\mathrm{T}}_{a=\mu_T} \Sigma_T [\nabla_a \sigma(s,a)]_{a=\mu_T}}}, \text{and } \Sigma_E = \Sigma_T. \tag{4}$$

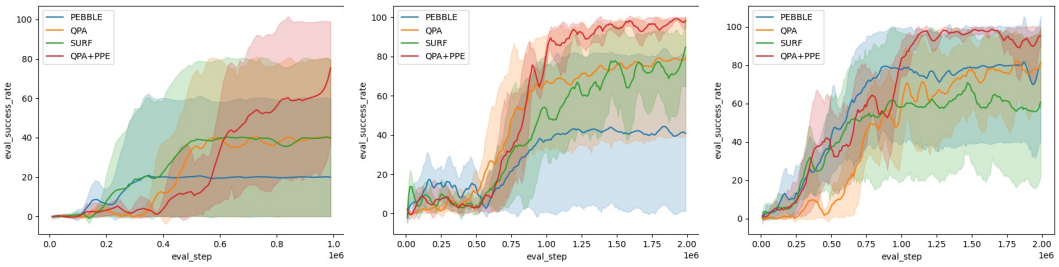
In practical applications, we can simply implement the proximal policy exploration as an algorithmic plugin within an existing algorithmic framework, as shown in Algorithm 1. This integration allows the enhancement of the policy exploration process without the need for extensive modifications to the current framework.

4 Experiments and Conclusion

	PEBBLE	SURF	QPA	QPA+PPE
Hammer	37.76 ± 51.59	49.29 ± 35.77	65.52 ± 40.38	$\mathbf{77.41 \pm 16.71}$
Drawer-open	20.00 ± 44.72	39.48 ± 54.15	39.67 ± 54.37	$\mathbf{79.52 \pm 44.53}$
Swap-Into	73.23 ± 34.66	59.64 ± 51.12	58.58 ± 31.54	$\mathbf{77.17 \pm 20.69}$
Door-Open	$\mathbf{99.11 \pm 2.60}$	79.88 ± 42.95	80.00 ± 44.72	96.47 ± 7.02

Table 1: The experimental results of Meta-World tasks are documented in the table, where we have recorded the mean and variance of the final 10-step evaluation outcomes for five seeds trained via different algorithms.

We conducted experiments on the tasks of Hammer, Drawer-Open, Sweep-Into, and Door-Open within the Meta-World environment. We set the maximum query feedback limit for Hammer and Sweep-Into at 10,000, while for Drawer-Open and Door-Open, the limit was set at 4,000.



(a) Drawer-open (b) Hammer (c) Sweep-into

Figure 1: The outperforming experiment results of our method compared with other methods are depicted in the graph. The x-axis represents the number of training steps during evaluation, while the y-axis indicates the success rate of the agent completing the task at the current training step.

We compared our method against well-established algorithms such as PEBBLE, SURF, and QPA, on these complex tasks. Our method demonstrated a significant advantage, underscoring the importance of enhancing the coverage of the preference buffer and its influence on the agent's learning process.

Our method actively optimizes the coverage of the preference buffer around the proximal policy distribution. As a result, the reward model learned by our method can provide a more reliable evaluation standard for policy optimization. This leads to our method outperforming the baseline methods.

In future work, we plan to explore a query method that is compatible with current algorithms to further improve feedback efficiency.

Acknowledgements

This project was accomplished while being a visiting PhD student at Nanyang Technological University, with the financial support from the China Scholarship Council. We express our heartfelt gratitude to Professor Bo An for his invaluable guidance and to the China Scholarship Council for their generous funding. Furthermore, this project received support from the Program B for Outstanding PhD Candidates of Nanjing University.

References

[1] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

[2] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359, 2017.

[3] Jonas Degrave, Federico Felici, Jonas Buchli, Michael Neunert, Brendan Tracey, Francesco Carpanese, Timo Ewalds, Roland Hafner, Abbas Abdolmaleki, Diego de Las Casas, et al. Magnetic control of tokamak plasmas through deep reinforcement learning. *Nature*, 602(7897):414–419, 2022.

[4] Jeff Wu, Long Ouyang, Daniel M Ziegler, Nisan Stiennon, Ryan Lowe, Jan Leike, and Paul Christiano. Recursively summarizing books with human feedback. *arXiv preprint arXiv:2109.10862*, 2021.

[5] Dylan Hadfield-Menell, Smitha Milli, Pieter Abbeel, Stuart J Russell, and Anca Dragan. Inverse reward design. *Advances in neural information processing systems*, 30, 2017.

[6] David Abel, Will Dabney, Anna Harutyunyan, Mark K Ho, Michael Littman, Doina Precup, and Satinder Singh. On the expressivity of markov reward. *Advances in Neural Information Processing Systems*, 34:7799–7812, 2021.

[7] Jianxiong Li, Xiao Hu, Haoran Xu, Jingjing Liu, Xianyuan Zhan, Qing-Shan Jia, and Ya-Qin Zhang. Mind the gap: Offline policy optimization for imperfect rewards. *arXiv preprint arXiv:2302.01667*, 2023.

[8] Jonathan Daniel Sorg. *The optimal reward problem: Designing effective reward for bounded agents*. PhD thesis, University of Michigan, 2011.

[9] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*, 2016.

[10] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

[11] Borja Ibarz, Jan Leike, Tobias Pohlen, Geoffrey Irving, Shane Legg, and Dario Amodei. Reward learning from human preferences and demonstrations in atari. *Advances in neural information processing systems*, 31, 2018.

[12] Kimin Lee, Laura Smith, Anca Dragan, and Pieter Abbeel. B-pref: Benchmarking preference-based reinforcement learning. *arXiv preprint arXiv:2111.03026*, 2021.

[13] Kimin Lee, Laura Smith, and Pieter Abbeel. Pebble: Feedback-efficient interactive reinforcement learning via relabeling experience and unsupervised pre-training. *arXiv preprint arXiv:2106.05091*, 2021.

[14] Jongjin Park, Younggyo Seo, Jinwoo Shin, Honglak Lee, Pieter Abbeel, and Kimin Lee. Surf: Semi-supervised reward learning with data augmentation for feedback-efficient preference-based reinforcement learning. *arXiv preprint arXiv:2203.10050*, 2022.

[15] Xinran Liang, Katherine Shu, Kimin Lee, and Pieter Abbeel. Reward uncertainty for exploration in preference-based reinforcement learning. *arXiv preprint arXiv:2205.12401*, 2022.

[16] Daniel Shin, Anca D Dragan, and Daniel S Brown. Benchmarks and algorithms for offline preference-based reward learning. *arXiv preprint arXiv:2301.01392*, 2023.

[17] Jeremy Tien, Jerry Zhi-Yang He, Zackory Erickson, Anca D Dragan, and Daniel S Brown. Causal confusion and reward misidentification in preference-based reward learning. *arXiv preprint arXiv:2204.06601*, 2022.

[18] Erdem Biyik and Dorsa Sadigh. Batch active preference-based learning of reward functions. In *Conference on robot learning*, pages 519–528. PMLR, 2018.

[19] Dorsa Sadigh, Anca Dragan, Shankar Sastry, and Sanjit Seshia. *Active preference-based learning of reward functions*. 2017.

[20] Erdem Bıyık, Nicolas Huynh, Mykel J Kochenderfer, and Dorsa Sadigh. Active preference-based gaussian process regression for reward learning. *arXiv preprint arXiv:2005.02575*, 2020.

[21] Xiao Hu, Jianxiong Li, Xianyuan Zhan, Qing-Shan Jia, and Ya-Qin Zhang. Query-policy misalignment in preference-based reinforcement learning. *arXiv preprint arXiv:2305.17400*, 2023.

[22] Richard S Sutton. Reinforcement learning: An introduction. *A Bradford Book*, 2018.

[23] Jan Leike, David Krueger, Tom Everitt, Miljan Martic, Vishal Maini, and Shane Legg. Scalable agent alignment via reward modeling: a research direction. *arXiv preprint arXiv:1811.07871*, 2018.

[24] Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

[25] Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation. *arXiv preprint arXiv:1810.12894*, 2018.

A Why Spearman Correlation Coefficient?

We chose the Spearman correlation coefficient over the Pearson correlation coefficient for several compelling reasons. Firstly, Spearman's method excels at assessing monotonic relationships, which is advantageous as it does not presuppose a linear relationship between variables. This attribute is crucial for analyzing data where linear assumptions may not hold. Secondly, Spearman's coefficient is less sensitive to outliers and non-normal distributions, thereby providing robustness in handling datasets that deviate from normal distribution patterns. Significantly, our study involves preference data that inherently rank pairs of items based on preference. Consequently, the Spearman correlation coefficient is particularly suited to evaluate the relationship between proxy return and ground truth return, as it effectively measures the strength and direction of association between ranked variables.

B Proof of Eq.(4)

Consider the formula for the KL divergence between two high-dimensional Gaussian distributions:

$$D_{KL}(\mathcal{N}(\mu,\Sigma),\mathcal{N}(\mu_T,\Sigma_T)) = \frac{1}{2}\left[(\mu-\mu_T)^{\mathrm{T}}\Sigma_T^{-1}(\mu-\mu_T) - \log\det(\Sigma_T^{-1}\Sigma) + tr(\Sigma_T^{-1}\Sigma) - n\right]. \tag{5}$$

When $D_{KL}(\mathcal{N}(\mu,\Sigma),\mathcal{N}(\mu_T,\Sigma_T)) \leq \epsilon$ is employed as a constraint, the solution to the optimization problem $\underset{\mu,\Sigma}{argmax}\,\mathbb{E}_{a\sim\mathcal{N}(\mu,\Sigma)}[\sigma_\theta(s,a)]$ is typically achieved through iterative means. However, considering our objective for the calculated μ, Σ to more effectively explore data from the out-of-preference buffer distribution within the proximal policy region, and the real-time requirement for problem-solving with each agent-environment interaction, we propose a more efficient closed-form approximation to the original problem by appropriately tightening the constraint, as shown in Eq.(4).

We introducing $\Sigma = \Sigma_T$, and the tightened constraint can be expressed as:

$$\begin{aligned} &D_{KL}(\mathcal{N}(\mu,\Sigma_T),\mathcal{N}(\mu_T,\Sigma_T)) \leq \epsilon. \\ \rightarrow &\frac{1}{2}\left[(\mu-\mu_T)^{\mathrm{T}}\Sigma_T^{-1}(\mu-\mu_T) - \log\det(\Sigma_T^{-1}\Sigma_T) + tr(\Sigma_T^{-1}\Sigma_T) - n\right] \leq \epsilon. \\ \rightarrow &\frac{1}{2}\left[(\mu-\mu_T)^{\mathrm{T}}\Sigma_T^{-1}(\mu-\mu_T)\right] \leq \epsilon. \end{aligned} \tag{6}$$

Substituting this into Eq.(3), we derive a simplified optimization problem:

$$\begin{aligned} &\max_{\mu} \underset{a\sim\mathcal{N}(\mu,\Sigma_T)}{\mathbb{E}}[\sigma_\theta(s,a)], \\ &\text{s.t.}(\mu-\mu_T)^{\mathrm{T}}\Sigma_T^{-1}(\mu-\mu_T) \leq 2\epsilon. \end{aligned} \tag{7}$$

To address the problem in Eq.(7), we construct the following Lagrangian function:

$$L = \sigma_\theta(s,a) - \xi((\mu-\mu_T)^{\mathrm{T}}\Sigma_T^{-1}(\mu-\mu_T) - 2\epsilon). \tag{8}$$

Deriving with respect to μ yields:

$$\nabla_\mu L = \nabla_a\sigma_\theta(s,a)|_{a=\mu} - \xi\Sigma_T^{-1}(\mu-\mu_T). \tag{9}$$

Setting $\nabla_\mu L = 0$, we find:

$$\mu = \mu_T + \frac{1}{\xi}\Sigma_T\,\nabla_a\sigma_\theta(s,a)|_{a=\mu}. \tag{10}$$

By applying the KKT conditions, we deduce:

$$\begin{aligned} &(\mu-\mu_T)^{\mathrm{T}}\Sigma_T^{-1}(\mu-\mu_T) - 2\epsilon = 0. \\ &\xi > 0. \end{aligned} \tag{11}$$

Further, via plugging Eq.(10) in Eq.(11), we can solve to obtain:

$$\begin{aligned}
&\frac{1}{\xi^2}\left(\Sigma_T\ \nabla_a\sigma_\theta(s,a)|_{a=\mu}\right)^{\mathrm{T}}\Sigma_T^{-1}\left(\Sigma_T\ \nabla_a\sigma_\theta(s,a)|_{a=\mu}\right)=2\epsilon,\ \xi>0.\\
\rightarrow&\xi^2=\frac{[\nabla_a\sigma_\theta(s,a)]_{a=\mu}^{\mathrm{T}}\,\Sigma_T\,[\nabla_a\sigma_\theta(s,a)]_{a=\mu}}{2\epsilon},\ \xi>0.\\
\rightarrow&\xi=\sqrt{\frac{[\nabla_a\sigma_\theta(s,a)]_{a=\mu}^{\mathrm{T}}\,\Sigma_T\,[\nabla_a\sigma_\theta(s,a)]_{a=\mu}}{2\epsilon}}.
\end{aligned}\tag{12}$$

Through Eq.(12), we find that ξ is a function of μ. However, Eq.(10) is a differential equation, which is challenging to solve directly for μ. Therefore, we perform a Taylor expansion on $[\nabla_a\sigma_\theta(s,a)]_{a=\mu}$:

$$\nabla_a\sigma_\theta(s,a)|_{a=\mu}\approx\ \nabla_a\sigma_\theta(s,a)|_{a=\mu_T}+\ \nabla_a^2\sigma_\theta(s,a)\big|_{a=\mu_T}(\mu-\mu_T).\tag{13}$$

This implies that when μ is sufficiently close to μ_T, we can approximate:

$$\nabla_a\sigma_\theta(s,a)|_{a=\mu}\approx\ \nabla_a\sigma_\theta(s,a)|_{a=\mu_T}.\tag{14}$$

Since our goal is to increase the density of proximal policy data in the preference buffer, thereby enhancing the reward model's evaluation capability under the current policy distribution, this approximation does not conflict with our objective and is indeed very fitting.

Thus, further, we can deduce:

$$\mu\approx\mu_T+\frac{\sqrt{2\epsilon}\cdot\Sigma_T[\nabla_a\sigma_\theta(s,a)]_{a=\mu_T}}{\sqrt{[\nabla_a\sigma_\theta(s,a)]_{a=\mu_T}^{\mathrm{T}}\Sigma_T[\nabla_a\sigma_\theta(s,a)]_{a=\mu_T}}}.\tag{15}$$

Therefore, the exploration behavior policy $\mathcal{N}(\mu_E,\Sigma_E)$ can be expressed as

$$\mu_E=\mu_T+\frac{\sqrt{2\epsilon}\cdot\Sigma_T[\nabla_a\sigma_\theta(s,a)]_{a=\mu_T}}{\sqrt{[\nabla_a\sigma_\theta(s,a)]_{a=\mu_T}^{\mathrm{T}}\Sigma_T[\nabla_a\sigma_\theta(s,a)]_{a=\mu_T}}},\text{and }\Sigma_E=\Sigma_T.\tag{16}$$

C Experimental Details

The details of the experiment setting are shown below:

Task	Total Feedback	Frequency of feedback	Reward batch
Hammer	10000	5000	50
Drawer-open	4000	5000	20
Swap-Into	10000	5000	50
Door-Open	4000	5000	20