QUERY-EFFICIENT PREFERENCE-BASED ROBUST REINFORCEMENT LEARNING FROM PAIRWISE COMPARISONS

Anonymous authors
Paper under double-blind review

ABSTRACT

We propose a preference-based robust reinforcement learning (PbRRL) framework that directly optimizes policies against the worst-case reward consistent with observed pairwise comparisons. The study focuses on settings in which only a very limited number of preference data are available and these data are highly trustworthy. Under the assumption of a linear reward model, the preference data with no response error induce a polyhedral ambiguity set over reward parameters, leading to a tractable max–min formulation of PbRRL. We propose a geometry-aware Vertical Cutting Method for a Cuboid, which adaptively generates preference queries to efficiently reduce the ambiguity set. We establish exponential convergence of the ambiguity set and asymptotic optimality of the resulting robust policy. Experiments on Ant-v5 show that our method substantially outperforms a traditional baseline when only a very limited number of preference queries are available. Our paper is among the first to bring the preference elicitation method that reduces the ambiguity set by pairwise comparisons directly into reinforcement learning.

1 INTRODUCTION

Reinforcement Learning (RL) is a powerful framework for sequential decision-making, where an agent interacts with an environment to learn a policy that maximizes cumulative reward over time. RL has achieved significant success in many applications, such as robot control (Kober et al., 2013), neuroeconomics (Lee et al., 2012) and natural language processing (Luketina et al., 2019). The effectiveness of RL critically depends on whether the reward function captures the objectives of the decision-maker(DM), especially in personalized settings, such as customized robot behavior and individualized financial recommendations (Sadigh et al., 2017; Wei & Liu, 2025). Nevertheless, in many complex real-world scenarios, explicitly formulating an appropriate reward function for a specific decision-maker(DM) is challenging.

A prominent approach is reinforcement learning from human feedback (RLHF), which uses human feedback to align the learned policy with human objectives (Christiano et al., 2017; Stiennon et al., 2020; Ouyang et al., 2022). When such human feedback conveys preference information, for example pairwise comparisons between trajectories, the resulting learning framework is commonly referred to as preference-based reinforcement learning (PbRL)(Wirth et al., 2017; Mu et al., 2025). Although PbRL approach reduces the need for manual reward specification, multiple reward functions may explain the observed preferences with limited feedback. Selecting a single estimate can conceal this uncertainty, and subsequent policy optimization may exploit inaccuracies in the estimated reward (Gao et al., 2022).

A natural response is to retain the remaining reward uncertainty during policy optimization rather than committing to a single estimated reward. This motivates preference-based robust reinforcement learning (PbRRL), in which a policy is optimized against the worst-case reward within an ambiguity set constructed from preference data. Existing PbRRL approaches, e.g., Zhu et al. (2023), rely on a two-step statistical estimation procedure: a parametric stochastic choice model (e.g., the Bradley-Terry-Luce model) is first fitted to noisy preference comparison data, and then a confidence region

Table 1: Representative work in large-sample and small-sample settings of preference elicitation.

	Large sample setting	Small sample setting
Data size	Large	Small
Data quality	With response error	Without response error
Preference-information representation	Bayesian or stochastic choice models	Linear or moment-type constraints (Armbruster & Delage, 2015; Guo et al., 2024)
Query design	Pre-designed queries (Kuhfeld et al., 1994; Arora & Huber, 2001)	Adaptive and personalized queries (Toubia et al., 2004; Zhang et al., 2025)
Decision making	Estimate-then-optimize (Shin et al., 2023; Ouyang et al., 2022; Christiano et al., 2017)	Preference robust optimization (Armbruster & Delage, 2015; Hu & Mehrotra, 2015; Guo et al., 2024; Liu et al., 2025)
RL extension	Zhu et al. (2023)	our paper

around the estimated reward is constructed. This paradigm inherently requires a large number of preference samples to control the estimation error.

In contrast, many practical preference-elicitation problems involve only a small number of samples. Examples arise in personalized healthcare (Danner et al., 2011), material selection (Yang et al., 2022), and engineering design (Elliott, 2010). In these applications, the preference information is often collected through structured pairwise comparisons and can be checked for consistency before being used in subsequent analysis. Motivated by such settings, we focus on a small-sample regime in which the number of interactions with the DM is limited and the elicited responses are sufficiently reliable to be modeled as error-free.

Small-sample preference elicitation and decision making have been studied extensively outside RL, e.g, the preference robust optimization (PRO), (Armbruster & Delage, 2015; Hu & Mehrotra, 2015; Guo et al., 2024; Liu et al., 2025). Table 1 summarizes representative methods for the two settings considered here. However, how to develop a reinforcement learning algorithm under a small sample setting remains much less explored. This is the central problem studied in this paper.

Developing PbRRL in this small-sample regime raises two fundamental challenges.

Challenge 1: How should the reward ambiguity set be constructed? Existing PbRRL methods typically construct reward ambiguity sets around statistical estimates, as in Zhu et al. (2023), which might be inaccurate when only a small number of comparisons are available. Motivated by this limitation, we directly encode each error-free pairwise comparison as a linear constraint on the reward parameter and construct a polyhedral ambiguity set containing all reward functions consistent with the observed preferences (Toubia et al., 2004; Armbruster & Delage, 2015; Guo et al., 2024). We then optimize the policy against the worst-case reward in this set.

Challenge 2: How should informative preference queries be designed? In the small-sample setting, each comparison is valuable, so the queries should be carefully chosen to reduce the reward ambiguity set as efficiently as possible. Existing approaches include utility-splitting methods such as RUS/RRUS/MUS (Armbruster & Delage, 2015; Chen et al., 2026) and the famous adaptive polyhedral methods (Toubia et al., 2004; Zhang et al., 2025), which generate subsequent queries according to the uncertainty remaining after previous responses. However, classical polyhedral query design does not provide a general quantitative guarantee that the feasible set contracts to the true parameter. To address this issue, we propose VCMC, which systematically cuts the ambiguity set along different directions and yields an explicit exponential convergence guarantee.

To address these two challenges, we develop a query-efficient PbRRL framework that directly constructs a reward ambiguity set from pairwise comparison constraints and adaptively refines it through VCMC. This framework links preference elicitation, ambiguity-set contraction, and robust policy optimization within a unified learning procedure. We further apply the framework to a continuous-control task on Ant-v5, where the agent must learn a policy from limited preference information over competing behavioral objectives.

Our main contributions are summarized as follows:

- We propose a PbRRL framework for small-sample settings without response error, which is among the first to bring the preference elicitation method that reduces the ambiguity set by pairwise comparisons directly into reinforcement learning.
- We introduce a new adaptive query generation method, Vertical Cutting Method for a Cuboid (VCMC), and establish exponential convergence of the resulting ambiguity set.
- We derive the asymptotic optimality of the optimized robust policy as the number of preference queries increases.
- We reformulate the nested max–min problem as an equivalent single-level problem via linear programming duality. Experiments on Ant-v5 show that both formulations perform nearly identically and substantially outperform the traditional PbRL baseline.

In the rest of this paper, Section 2 formulates the PbRRL problem with pairwise comparisons; Section 3 develops the VCMC method for adaptive preference query generation; Section 4 presents numerical experiments on Ant-v5.

2 Preference-based Robust Reinforcement Learning (PbRRL)

In this section, we develop a PbRRL framework under a linear reward model and pairwise trajectory comparisons without response error. We seek a robust policy that maximizes the worst-case expected discounted return over an ambiguity set of reward parameters consistent with observed comparisons.

2.1 Markov Decision Process with Preferences

We study PbRRL under the finite-horizon Markov Decision Process with Preferences (MDPP) framework (Wirth et al., 2017), where the agent does not know the exact reward function as in the standard MDP but only has limited preference data.

We define the MDPP by a seven-tuple $(\mathcal{S}, \mathcal{A}, T, P, \rho, \gamma, \mathcal{D})$, where $\mathcal{S}$ denotes the state space and $\mathcal{A}$ denotes the action space. The state and the action spaces can be continuous or discrete. $T < \infty$ is the horizon length. $P : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ is the state transition probability function, which maps each state-action pair (s, a) to a probability distribution over the next states, $\rho \in \Delta(\mathcal{S})$ denotes the initial state distribution, and $\gamma \in [0, 1]$ denotes the discount factor. A policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$ is a mapping from the current state $s \in \mathcal{S}$ to the probability of choosing action a in $\mathcal{A}$.

In a standard MDP, the agent has a stationary reward $r : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$, which is a function of state $S_t = s_t$, action $A_t = a_t$ and state $S_{t+1} = s_{t+1}$, for any stage $t = 0, \ldots, T-1$. We assume that the reward function belongs to a parametric family $\{r_\theta : \theta \in \mathbb{R}^d\}$, where θ is the reward parameter vector. We denote the unknown true reward function by $r_{\theta^\star}$. The agent aims to maximize the expected discounted cumulative reward

$$J(\pi) = \mathbb{E}_{S_0 \sim \rho, A_t \sim \pi(S_t), S_{t+1} \sim P(S_t, A_t)} \left[\sum_{t=0}^{T-1} \gamma^t r_{\theta^\star}(S_t, A_t, S_{t+1}) \right], \tag{1}$$

under the true reward function $r_{\theta^\star}(s_t, a_t, s_{t+1})$.

In contrast, under the MDPP framework, the true reward function is unknown and the agent only has some preference data $\mathcal{D}$ collected from the DM. The dataset consists of N pairwise trajectory comparisons, denoted by $\mathcal{D} := \{(\tau_n^A, \tau_n^B, \delta_n) | n = 1, \ldots, N\}$. $\tau = \{s_0, a_0, s_1, a_1, \ldots, s_{T-1}, a_{T-1}, s_T\}$ is a trajectory generated by the interaction between the agent and the environment. δ_n represents the DM's binary choice (preference information) between the two trajectories (τ_n^A, τ_n^B), i.e.

$$\delta_n = \begin{cases} 1, & \text{if the DM chooses } \tau_n^A, \\ -1, & \text{if the DM chooses } \tau_n^B. \end{cases}$$

2.2 Pairwise Comparison Ambiguity Set

In PbRRL, we use the pairwise comparison data from the DM to construct an ambiguity set of the unknown true reward function. For convenience, we assume that the true reward function follows

a parametric structure specified in Assumption 2.1, which allows us to construct the ambiguity set over the finite-dimensional parameter space.

Assumption 2.1. We assume that the true reward function of the DM is a linear function $r_{\theta^\star}(s_t, a_t, s_{t+1}) = (\theta^\star)^\top \phi(s_t, a_t, s_{t+1})$, where $\phi : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}^d$ is a known feature mapping that satisfies $\sup_{s_t \in \mathcal{S}_t, a_t \in \mathcal{A}_t, s_{t+1} \in \mathcal{S}_{t+1}} \|\phi(s_t, a_t, s_{t+1})\|_2 \le \overline{B}_\phi$, where $\overline{B}_\phi$ is a fixed constant, $\theta^\star$ is the parameter characterizing the DM's true preference (reward function).

Without loss of generality, we consider an initial ambiguity set of the parameter

$$\Theta_0 = \left\{\theta \in [0,1]^d \mid \langle \mathbf{1}, \theta \rangle = 1\right\}, \tag{2}$$

and we assume $\theta^\star \in \Theta_0$.

We follow the approach in Armbruster & Delage (2015), which constructs the static ambiguity set directly based on pairwise comparison data $\mathcal{D}$ collected from the DM. Specifically, we incorporate the pairwise trajectory comparison data in $\mathcal{D}$ as constraints that refine the initial ambiguity set in the MDP setting and thereby define the ambiguity set of the reward parameters. Focusing on the case study in which the queries are answered with high quality, we make the following assumption regarding the preference data:

Assumption 2.2. We assume that the pairwise trajectory comparison data in $\mathcal{D}$ are aligned with the discounted cumulative true reward of the DM. Specifically, the DM prefers trajectory τ^A over τ^B whenever $G^{\theta^\star}_{\tau^A} \ge G^{\theta^\star}_{\tau^B}$, where $G^{\theta^\star}_{\tau} = \sum_{t=0}^{T-1} \gamma^t r_{\theta^\star}(s_t, a_t, s_{t+1})$.

Accordingly, we define the pairwise-comparison-no-error (PCNE) ambiguity set of θ as

$$\begin{aligned}
\Theta_{\mathrm{PCNE}} &:= \left\{\theta \in \Theta_0 \mid \delta_n G^{\theta}_{\tau^A_n} \ge \delta_n G^{\theta}_{\tau^B_n}, \forall n = 1, \dots, N\right\}, \\
&= \left\{\theta \in \Theta_0 \mid \delta_n \sum_{t=0}^{T_n^A - 1} \gamma^t r_\theta(s_t^A, a_t^A, s_{t+1}^A) \ge \delta_n \sum_{t=0}^{T_n^B - 1} \gamma^t r_\theta(s_t^B, a_t^B, s_{t+1}^B), \forall n = 1, \dots, N\right\}, \\
&= \{\theta \in \mathbb{R}^d \mid A_N \theta \le b_N\}.
\end{aligned} \tag{3}$$

where $A_N = \Big[I_d, -I_d, \mathbf{1}_d^\top, -\mathbf{1}_d^\top, \delta_1\Big(\sum_{t=0}^{T_1^B-1} \gamma^t \phi_(s_t^B, a_t^B, s_{t+1}^B) - \sum_{t=0}^{T_1^A-1} \gamma^t \phi_(s_t^A, a_t^A, s_{t+1}^A)\Big)^\top, \cdots, \delta_N\Big(\sum_{t=0}^{T_N^B-1} \gamma^t \phi_(s_t^B, a_t^B, s_{t+1}^B) - \sum_{t=0}^{T_N^A-1} \gamma^t \phi_(s_t^A, a_t^A, s_{t+1}^A)\Big)^\top\Big]^\top$, $b_N = [\mathbf{1}_d, \mathbf{0}_d, 1, -1, \mathbf{0}_N]^\top$, I_d denotes the $d \times d$ identity matrix, and $\mathbf{1}_d$ and $\mathbf{0}_d$ denote the d-dimensional all-ones and all-zeros vectors, respectively. By Assumption 2.2, we have that $\theta^\star \in \Theta_{\mathrm{PCNE}}$. Thus, Θ_{PCNE} is a well-defined ambiguity set characterizing the preference of the DM.

Remark 2.3. Traditional robust RLHF approaches, e.g. the PbRRL method proposed by Zhu et al. (2023), first derive a maximum likelihood estimation $\hat{\theta}$ based on a preference model with response error, e.g. Bradley-Terry-Luce model, and then construct an ambiguity set by a confidence set centered at $\hat{\theta}$: $\Theta_{\mathrm{MLE}} := \left\{\theta \in \Theta_0 \;\middle|\; \|\theta - \hat{\theta}\|_{\Sigma_{\mathcal{D}} + \lambda I} \le C\sqrt{\frac{d}{\gamma^2 n} \log \frac{1}{\delta} + \lambda B^2}\right\}$.

The key difference between Zhu et al. (2023) and our approach lies in the construction of the ambiguity set. First, they assume that the responses from the DM (environment) may contain errors due to environmental factors. However, in our setting, the DM responds to queries with his/her true preference, which means that there are no response errors in the high-quality preference data $\mathcal{D}$. Our setting is more suitable for cases with high quality but a limited number of queries. Second, they first estimate a MLE estimation and then find a worst-case utility from the confidence set, which introduces additional error in the two-step estimation. In contrast, we construct the ambiguity set by directly imposing pairwise comparisons as constraints, thereby avoiding additional estimation error.

Algorithm 1 PbRRL-PCNE

1: **Input:** Number of queries N, initial ambiguity set Θ_0.
2: **Output:** Policy $\hat{\pi}_{\mathrm{PCNE}}$
3: **Step 1: Construct the ambiguity set:**
4: **for** $n = 1$ to N **do**
5: **Step 1.1:** Generate query (A_n, B_n) based on the current ambiguity set Θ_{n-1}.
6: **Step 1.2:** Provide the query to the DM and collect her/his response δ_n, where $\delta_n = 1$ if A_n is chosen and $\delta_n = -1$ if B_n is chosen.
7: **Step 1.3:** Update the ambiguity set by adding the new pairwise comparison data:

$$\Theta_n = \Theta_{n-1} \cap \{\theta \in \Theta_0 \mid \delta_n G^{\theta}_{\tau_n^A} \geq \delta_n G^{\theta}_{\tau_n^B}\}.$$

8: **end for**
9: Set $\Theta_{\mathrm{PCNE}} \leftarrow \Theta_N$.
10: **Step 2: Solve the following max-min problem:**

$$\hat{\pi}_{\mathrm{PCNE}} \in \arg\max_{\pi} \min_{\theta \in \Theta_{\mathrm{PCNE}}} \mathbb{E}_{S_0 \sim \rho, A_t \sim \pi(S_t), S_{t+1} \sim P(S_t, A_t)} \left[\sum_{t=0}^{T-1} \gamma^t r_\theta(S_t, A_t, S_{t+1}) \right]. \quad (5)$$

2.3 PbRRL Algorithm

Given the ambiguity set Θ_{PCNE} defined above, the PbRRL objective seeks a policy $\hat{\pi}_{\mathrm{PCNE}} = \arg\max_\pi \hat{J}(\pi)$ that maximizes the worst-case expected discounted return

$$\hat{J}(\pi) \coloneqq \min_{\theta \in \Theta_{\mathrm{PCNE}}} \mathbb{E}_{S_0 \sim \rho, A_t \sim \pi(S_t), S_{t+1} \sim P(S_t, A_t)} \left[\sum_{t=0}^{T-1} \gamma^t r_\theta(S_t, A_t, S_{t+1}) \right]. \quad (4)$$

Algorithm 1 summarizes the PbRRL-PCNE framework under the ambiguity set Θ_{PCNE}. The details of ambiguity set updating in Step 1 will be introduced later in Section 3 . The determination of the optimal policy $\hat{\pi}_{\mathrm{PCNE}}$ in equation 5 is a max-min optimization problem. We apply the duality method to reformulate the max-min problem.

Theorem 2.4. *Given Assumption 2.1, the max-min MDP problem equation 5 is equivalent to the following expected discounted return constrained MDP problem:*

$$\begin{aligned} \max_{\pi, y} \quad & b_N^\top y \\ \text{s.t.} \quad & A_N^\top y = \mathbb{E}_{S_0 \sim \rho, A_t \sim \pi(S_t), S_{t+1} \sim P(S_t, A_t)} \left[\sum_{t=0}^{T-1} \gamma^t \phi(S_t, A_t, S_{t+1}) \right], \\ & y \leq 0. \end{aligned} \quad (6)$$

The proof of all theorem in the paper is provided in Appendix D. If the transaction probability is known, problem equation 6 can be reformulated as a linear program by introducing the occupation measure (Puterman, 1994). If we do not know the exact transaction probability but have draw some samples of the trajectory, the sample average approximation or stochastic approximation methods can be applied to solve equation 6.

3 Vertical Cutting Method for a Cuboid

As the high-quality queries with limited numbers play an essential role in our PbRRL method, the generation of these queries largely affects the efficiency of the algorithm. We introduce the VCMC method, which generates queries via mutually orthogonal cuts on a fixed cuboid to iteratively refine the ambiguity set. The polyhedral method of Toubia et al. (2004) does not provide convergence guarantees for the resulting feasible set, whereas VCMC achieves exponential convergence of the ambiguity set toward the true reward parameter. To our knowledge, this is the first polyhedral preference elicitation method with an exponential convergence guarantee in an MDP setting.

Algorithm 2 VCMC

Input: Maximal number of cutting rounds I, initial ambiguity set $\mathcal{P}_0 \subset \mathbb{R}^{d-1}$, edge direction $\alpha = (\alpha_1, \ldots, \alpha_{d-1})$.
Output: $\Theta_{\mathrm{PCNE}} = R(\mathcal{P}_0 \cap \mathcal{C}_{I(d-1)})$
Find the minimal cuboid $\mathcal{C}_0$ covering $\mathcal{P}_0$.
for $i = 0$ to $I - 1$ **do**
  Find the geometric center $c_{i(d-1)}$ of $\mathcal{C}_{i(d-1)}$.
  for $k = 1$ to $d - 1$ **do**
    Step 1: Find the axis $l_{i,k}$ that passes through the center $c_{i(d-1)}$ and is parallel to the edge direction α_k. Find the two intersections $(\mu^A_{i,k}, \mu^B_{i,k})$ between the axis $l_{i,k}$ and the boundary of $\mathcal{C}_{i(d-1)}$.
    Step 2: Find the cutting hyperplane $\mathcal{H}_{i,k} = \left\{\mu \in \mathbb{R}^{d-1} \mid \alpha_k^\top(\mu - c_{i(d-1)}) = 0\right\}$.
    Step 3: Obtain a pairwise comparison query $(A_{i,k}, B_{i,k})$ constructed from the trajectory feature vectors $z^A_{i,k}$ and $z^B_{i,k}$ by solving the optimization problem equation 8.
    Step 4: Provide the query $(A_{i,k}, B_{i,k})$ to the DM and collect her/his response $\delta_{i,k}$, where $\delta_{i,k} = 1$ if the DM chooses $A_{i,k}$ and $\delta_{i,k} = -1$ if the DM chooses $B_{i,k}$.
    Step 5: Update the cuboid
    $\mathcal{C}_{i(d-1)+k} = \mathcal{C}_{i(d-1)+k-1} \cap \left\{\mu \in \mathbb{R}^{d-1} \mid \delta_{i,k} \left(R(\mu)\right)^\top \left(z^B_{i,k} - z^A_{i,k}\right) \leq 0\right\}$.
  end for
end for

3.1 Generation of the approximating cuboid

We first describe how VCMC constructs an initial cuboid approximation of the projected ambiguity set. Notice that the initial ambiguity set Θ_0 lies in the $(d-1)$ dimensional affine hyperplane $\mathcal{H} := \left\{\theta \in \mathbb{R}^d \mid \langle \mathbf{1}, \theta \rangle = 1\right\}$. For computational convenience, we project Θ_0 from $\mathbb{R}^d$ onto $\mathbb{R}^{d-1}$. We denote the projected initial ambiguity set as

$$\begin{aligned}\mathcal{P}_0 \;:=\; R^{-1}(\Theta_0) &= \left\{\mu \in \mathbb{R}^{d-1} \;\mid 0 \leq \mu_j \leq 1,\; j = 1, \ldots, d-1, \sum_{j=1}^{d-1} \mu_j \leq 1\right\} \\ &= \left\{\mu \in \mathbb{R}^{d-1} \mid A_0^{\mathcal{P}} \mu \leq b_0^{\mathcal{P}}\right\},\end{aligned}$$

where $A_0^{\mathcal{P}} = [I_{d-1}, -I_{d-1}, \mathbf{1}_{d-1}^\top]^\top$, $b_0^{\mathcal{P}} = [\mathbf{1}_{d-1}, \mathbf{0}_{d-1}, 1]^\top$, and the affine map $R^{-1} : \mathbb{R}^d \to \mathbb{R}^{d-1}$ and $R : \mathbb{R}^{d-1} \to \mathbb{R}^d$ are defined as: $R^{-1}(x) = (x_1, x_2, \ldots, x_{d-1}), R(x) = (x_1, x_2, \ldots, x_{d-1}, 1 - \sum_{j=1}^{d-1} x_j)$. To simplify the query generation, we approximate the initial ambiguity set $\mathcal{P}_0$ by an outer-covering cuboid $\mathcal{C}_0$ (details in Appendix B).

3.2 Algorithm: Vertical Cutting Method for a Cuboid

In traditional polyhedral cut algorithms, e.g. Toubia et al. (2004), the generated hyperplane in the next step might be parallel or very close to the hyperplane generated at the previous step. In this case, the maximal Kolmogorov (sup-norm) distance between any two points in the ambiguity set might not be reduced in this step, which prevents the convergence of the cutting algorithm theoretically.

To address this limitation, we propose a new cutting method, VCMC (see its detailed procedure in Algorithm 2), which starts from the approximating cuboid $\mathcal{C}_0$ and generates $d-1$ mutually orthogonal cutting hyperplanes passing through the geometric center of the cuboid at each iteration $i = 0, \ldots, I-1$. In the cutting procedure at each iteration i, the cutting hyperplanes are imposed sequentially along each cuboid edge direction , ensuring a continuous reduction of the size of the cuboid across all edge directions and maintaining the structure of the initial approximation cuboid (See Figure 1 for a graphic illustration at an iteration). This generates a series of cuboid $\{\mathcal{C}_{i(d-1)+k}\}_{k=1,\ldots,d-1}^{i=0,\ldots,I-1}$, denoted for short by $\{\mathcal{C}_n\}_{n=1,\ldots,N}$, where $n = i(d-1) + k$. Finaly, intersecting $\mathcal{C}_N$ with $\mathcal{P}_0$ and mapping back to the original parameter space yields the PCNE ambiguity set $\Theta_{\mathrm{PCNE}} = R(\mathcal{P}_0 \cap \mathcal{C}_{I(d-1)})$.

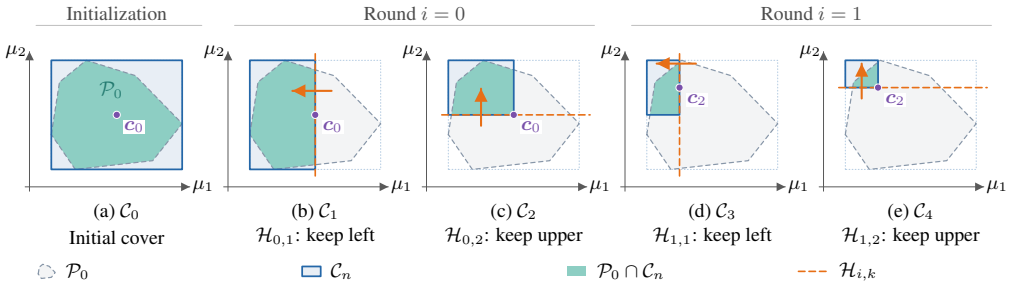

Figure 1: Illustration of VCMC in the two-dimensional projected space ($d = 3$).

3.3 Trajectory Query Generation

A challenge of the VCMC algorithm is to find a pairwise comparison query $(A_{i,k}, B_{i,k})$, which corresponds to the cutting plane in the preference parameter space. If the DM chooses A, one side of the hyperplane is retained within the current ambiguity set; otherwise, the opposite side is preserved. Therefore, all points lying on the hyperplane $\mathcal{H}_{i,k}$ attain the same preference value for the two options. In PbRRL, such a query is constructed from feature trajectories, denoted by $(z^A_{i,k}, z^B_{i,k})$, where the feature trajectory is defined as

$$z = \sum_{t=0}^{T-1} \gamma^t \phi_(s_t, a_t, s_{t+1}) \in \mathcal{Z}_{\text{traj}} := \left\{ \sum_{t=0}^{T-1} \gamma^t \phi_(s^\tau_t, a^\tau_t, s^\tau_{t+1}) \mid \tau \in \mathcal{T} \right\},$$

where,$\mathcal{T}$ denotes the set of all feasible trajectories. Based on the observation between the query and the hyperplane, we propose the following optimization method to determine the query in the trajectories feature space

$$\begin{aligned} \max_{z^A_{i,k}, z^B_{i,k} \in \mathcal{Z}_{traj}} \quad & \left| \left(R(\mu^A_{i,k})\right)^\top z^A_{i,k} - \left(R(\mu^B_{i,k})\right)^\top z^B_{i,k} \right| \\ \text{s.t.} \quad & (R(\mu))^\top (z^A_{i,k} - z^B_{i,k}) = 0, \ \forall \mu \in \mathcal{H}_{i,k}, \end{aligned} \tag{7}$$

where $\mathcal{H}_{i,k} = \left\{ \mu \in \mathbb{R}^{d-1} \mid \alpha_k^\top (\mu - c_{i(d-1)}) = 0 \right\}$.

Problem equation 7 maximizes the gap between the cumulative discounted rewards of trajectory A and that of trajectory B, which promotes the responses more consistent with the DM's true preference. Problem equation 7 is a nonlinear programming problem and we can reformulate it as a mixed-integer linear optimization problem equation 8 which can be solved efficiently by state-of-art solvers, e.g. Gurobi.

Theorem 3.1. *Denote e_k as the k-th coordinate direction (standard basis vector) in $\mathbb{R}^{d-1}$. Problem equation 7 can be reformulated as the following mixed-integer linear problem:*

$$\begin{aligned} \max_{\substack{z^A_{i,k}, z^B_{i,k} \in \mathcal{Z}_{traj}, \\ b \in \{0,1\}, U \in \mathbb{R}, \lambda \in \mathbb{R}}} \quad & U \\ \text{s.t.} \quad & z^A_{i,k,j} - z^B_{i,k,j} - z^A_{i,k,d} + z^B_{i,k,d} = \lambda \alpha_{k,j}, \quad j = 1, \ldots, d-1, \\ & z^A_{i,k,d} - z^B_{i,k,d} = -\lambda \sum_{j=1}^{d-1} \alpha^\top_{k,j} c_{i(d-1),j}, \\ & U \le \Delta_{i,k} + M(1-b), \quad U \le -\Delta_{i,k} + Mb \\ & U \ge 0, \quad \lambda > 0 \end{aligned} \tag{8}$$

where, $\Delta_{i,k} = \left(R(\mu^A_{i,k})\right)^\top z^A_{i,k} - \left(R(\mu^B_{i,k})\right)^\top z^B_{i,k}$ and M is a sufficiently large constant.

3.4 Convergence of VCMC

The vertical cutting procedure generates a nested sequence of cuboids in the projected parameter space. Under noiseless preference feedback, the true parameter remains feasible at every iteration, and the resulting sequence admits an explicit exponential contraction, as formalized below.

Theorem 3.2. *Let $\mathcal{P}_0 = R^{-1}(\Theta_0) \subset \mathbb{R}^{d-1}$ be the initial polyhedron ambiguity set, and $\mathcal{C}_0$ be the minimal axis-aligned cuboid containing $\mathcal{P}_0$, with edge length $(L_{0,1}, L_{0,2}, \ldots, L_{0,(d-1)})$. Let*

$\{\mathcal{C}_n\}_{n\geq 0}$ be the sequence of cuboids generated by the Algorithm 2. Let $\theta^\star \in \Theta_0$ be the true reward parameter and $\mu^* = R^{-1}(\theta^\star)$ be the true parameter in the projected space. Under Assumptions 2.1 and 2.2, and B.1 , we have 1). $\mu^* \in \mathcal{C}_n$ for all $n \geq 0$. 2). Write $n = i(d-1)+k$ with $i = 0, 1, \ldots, I-1$, $k = 1, \ldots, d-1$. Then for every $\mu \in \mathcal{C}_n$, $\|\mu - \mu^*\|_2 \leq \sqrt{\sum_{j=1}^{k}\left(\frac{L_{0,j}}{2^{i+1}}\right)^2 + \sum_{j=k+1}^{d-1}\left(\frac{L_{0,j}}{2^i}\right)^2}$, 3). $\lim_{n\to\infty} \mathcal{C}_n = \{\mu^*\}$.

The proof of Theorem 3.2 is provided in Appendix D.3. We have a similar convergence result and error bound estimation on the unprojected parameter θ, introduced in Appendix C.

3.5 Convergence of the Learned Robust Policy

We now analyze the policy performance of PbRRL-PCNE algorithm (Algorithm 1) whose ambiguity set Θ_{PCNE} is constructed via VCMC algorithm (Algorithm 2).

Let $\pi^\star \in \arg\max_\pi J(\pi)$ denote the optimal policy maximizing the true reward function as in equation 1. The PbRRL-PCNE algorithm outputs a policy $\hat{\pi}_{\mathrm{PCNE}} \in \arg\max_\pi \hat{J}(\pi)$, optimal to the worst-case utility in equation 4. We define the suboptimality gap as $\mathrm{SubOpt}(\pi) := J(\pi^\star) - J(\pi)$, and have the following conclusion in terms of the ambiguity sets produced by VCMC.

Theorem 3.3. *Let $\hat{\pi}_{\mathrm{PCNE}}$ be the output of Algorithm 1. Suppose Assumption 2.1 and 2.2 hold. Then, for any $n = i(d-1)+k$ with $i = 0, 1, \ldots, I-1$ and $k = 1, \ldots, d-1$, $\mathrm{SubOpt}(\hat{\pi}_{\mathrm{PCNE}}) \leq T\overline{B}_n \cdot \overline{B}_\phi$, where $\overline{B}_n$ is the parameter error bound established from Theorem 3.2 and Proposition C.1, i.e.,*

$$\overline{B}_n := (1+\sqrt{d-1})\sqrt{\sum_{j=1}^{k}\left(\frac{L_{0,j}}{2^{i+1}}\right)^2 + \sum_{j=k+1}^{d-1}\left(\frac{L_{0,j}}{2^i}\right)^2}.$$

In particular, as the number of queries grows, $\overline{B}_n \to 0$ and therefore $\mathrm{SubOpt}(\hat{\pi}_{\mathrm{PCNE}}) \to 0$.

4 Experiments

We evaluate the proposed PbRRL method on the continuous-control benchmark **Ant-v5**. Following Assumption 2.1, the reward model is defined as $r_{\theta^\star}(s_t, a_t, s_{t+1}) = \theta^{\star\top}\phi(s_t, a_t, s_{t+1}), \theta^\star \in \Theta_0$, where $\phi = \left(\phi_1, \ldots, \phi_4\right)^\top$ is a fixed 4-dimensional feature map. We use the expected discounted return $J(\pi)$ defined in equation 1 as the performance metric, with a discount factor $\gamma = 0.99$. Further details of the experimental settings are provided in Appendix A.1.

We compare four methods that use Proximal Policy Optimization (PPO)(Schulman et al., 2017) as the underlying RL optimizer and differ only in how the reward is constructed. Detailed implementations of all four methods are provided in Appendix A.

- **PbRRL-PCNE**: Constructed a fixed ambiguity set Θ_{PCNE} by VCMC, and then we optimize a robust policy via an alternating (inner-outer) procedure.
- **PbRRL-PCNE-Dual**: We use the dual formulation in Theorem 2.4 over the ambiguity set Θ_{PCNE} constructed by VCMC, and jointly optimize the policy and dual variable y.
- **PbRL-BTL-MLE**: Fit a point estimate $\hat{\theta}_{\mathrm{MLE}}$ from random preference queries using the BTL model with MLE, and then train PPO with the fixed estimated reward $r_{\hat{\theta}_{\mathrm{MLE}}}$.
- **PbRL-True**: Train PPO directly on the true reward $r_{\theta^\star}$ as a ground-truth benchmark.

Our experiments are designed to answer two questions. First, how does the number of preference queries affect the policy performance of PbRRL-PCNE? Second, under the very limited queries, how do PbRRL-PCNE and dual implementations compare with each other and with baselines?

Effect of the Number of Preference Queries. We first investigate how the number of preference queries affects policy performance in the proposed PbRRL-PCNE model. Figure 2 shows that increasing the number of queries generally improves policy learning by refining the reward ambiguity

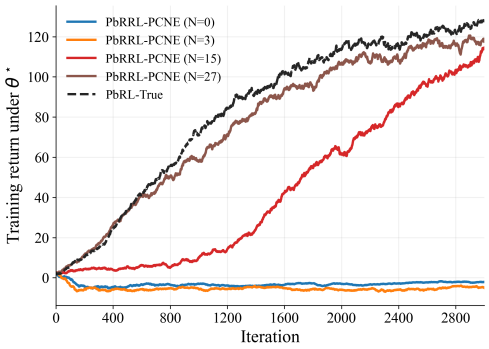

Figure 2: Training performance of PbRRL-PCNE for different numbers of queries.

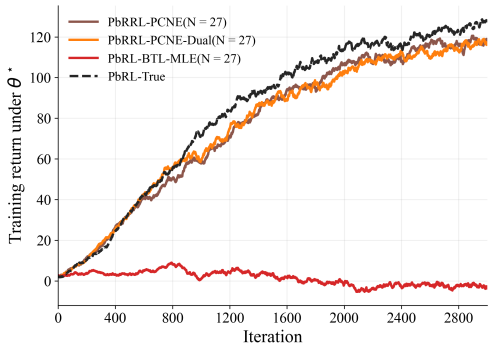

Figure 3: Training performance of different methods.

Table 2: Out-of-sample performance of different methods (mean $\pm$ 95% confidence interval).

Method	$J(\pi)$	Method	$J(\pi)$
PbRRL-PCNE ($N = 0$)	-0.88 ± 0.18	PbRRL-PCNE-Dual ($N = 27$)	159.40 ± 2.45
PbRRL-PCNE ($N = 3$)	-2.25 ± 0.63	PbRL-BTL-MLE ($N = 27$)	7.75 ± 2.20
PbRRL-PCNE ($N = 15$)	149.64 ± 1.64	PbRL-True	$\mathbf{167.90 \pm 1.51}$
PbRRL-PCNE ($N = 27$)	159.49 ± 4.47		

set. With only $N = 0$ or $N = 3$ queries, the learned policies remain at a low-return level, whereas $N = 15$ and $N = 27$ lead to substantially better performance. Notably, with $N = 15$, policy learning progresses relatively slowly during the early training stage, but the performance improves substantially with sufficient optimization and gradually approaches that of the PbRL-True benchmark. Table 2 shows that increasing the number of queries from $N = 3$ to $N = 15$ yields a substantial improvement in out-of-sample performance, whereas the additional gain from $N = 15$ to $N = 27$ becomes considerably smaller. This suggests that, once enough queries have been collected, the marginal benefit of additional queries becomes limited, as the learned policy already approaches the performance of the ground-truth benchmark. Thus, effective preference elicitation should balance policy performance against the cost of acquiring further comparisons.

Comparison across Methods. We then compare PbRRL-PCNE, PbRRL-PCNE-Dual, and PbRL-BTL-MLE under the same query budget $N = 27$, while PbRL-True serves as a ground-truth benchmark. In Figure 3 and Table 2, PbRRL-PCNE and its dual implementation achieve almost identical performance, empirically validating the exact dual reformulation in Theorem 2.4. Both substantially outperform PbRL-BTL-MLE and approach the PbRL-True benchmark. These results highlight the limitations of statistical point estimation with random queries when preference data are very limited, and demonstrate the effectiveness of VCMC in designing informative preference queries that rapidly shrink the reward ambiguity set and lead to better policy performance.

5 Conclusion

We proposed a PbRRL framework in target of settings where only a very limited number of adaptive interactions with the decision-maker are available, while the responses are highly trustworthy. We proposed the Vertical Cutting Method for a Cuboid to generate queries which is proved to guarantee the exponential convergence of the ambiguity set and asymptotic optimality of the learned robust policy. Future work includes extending the framework to handle inconsistent responses from the DM with response errors, considering state- or context-dependent preferences, incorporating richer reward representations beyond linear models, and applying VCMC-based preference elicitation to larger-scale and real-world decision-making problems.

AI Use Statement

In this work, we used generative AI tools to assist with language polishing and revision of the manuscript, as well as with the implementation, debugging, and correction of empirical code. All

AI-assisted manuscript edits and code modifications were reviewed and verified by the authors, and the empirical code was tested for correctness. We take full responsibility for the final content of this work, including all text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Benjamin Armbruster and Erick Delage. Decision making under uncertainty when preference information is incomplete. *Management science*, 61(1):111–128, 2015.

Neeraj Arora and Joel Huber. Improving parameter estimates and model prediction by aggregate customization in choice experiments. *Journal of Consumer Research*, 28(2):273–283, 2001.

Bo Chen, Jia Liu, and Huifu Xu. Maximum utility split method for utility preference elicitation. *arXiv preprint arXiv:2606.12868*, 2026.

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.

Marion Danner, J Marjan Hummel, Fabian Volz, Jeannette G Van Manen, Beate Wiegard, Charalabos-Markos Dintsios, Hilda Bastian, Andreas Gerber, and Maarten J IJzerman. Integrating patients' views into health technology assessment: Analytic hierarchy process (ahp) as a method to elicit patient preferences. *International journal of technology assessment in health care*, 27(4):369–375, 2011.

Michael A Elliott. Selecting numerical scales for pairwise comparisons. *Reliability Engineering & System Safety*, 95(7):750–763, 2010.

Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*, 2022.

Shaoyan Guo, Huifu Xu, and Sainan Zhang. Utility preference robust optimization with moment-type information structure. *Operations Research*, 72(5):2241–2261, 2024.

Jian Hu and Sanjay Mehrotra. Robust decision making over a set of random targets or risk-averse utilities with an application to portfolio optimization. *IIE Transactions*, 47(4):358–372, 2015.

Jens Kober, J Andrew Bagnell, and Jan Peters. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 32(11):1238–1274, 2013.

Warren F Kuhfeld, Randall D Tobias, and Mark Garratt. Efficient experimental design with marketing research applications. *Journal of Marketing Research*, 31(4):545–557, 1994.

Daeyeol Lee, Hyojung Seo, and Min Whan Jung. Neural basis of reinforcement learning and decision making. *Annual review of neuroscience*, 35(1):287–308, 2012.

Jia Liu, Zhiping Chen, and Huifu Xu. Preference ambiguity and robustness in multistage decision making. *Mathematical Programming*, 214(1–2):847–939, 2025.

Jelena Luketina, Nantas Nardelli, Gregory Farquhar, Jakob Foerster, Jacob Andreas, Edward Grefenstette, Shimon Whiteson, and Tim Rocktaschel. A survey of reinforcement learning informed by natural language. In *IJCAI 2019: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, pp. 6309–6317, August 2019.

Ni Mu, Yao Luan, and Qing-Shan Jia. Preference-based multi-objective reinforcement learning. *IEEE Transactions on Automation Science and Engineering*, 22:18737–18749, 2025. doi: 10.1109/TASE.2025.3589271.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35: 27730–27744, 2022.

Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 1994.

Dorsa Sadigh, Anca D. Dragan, Shankar Sastry, and Sanjit A. Seshia. Active preference-based learning of reward functions. In *Robotics: Science and Systems Conference*, 2017.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *ArXiv*, abs/1707.06347, 2017.

Daniel Shin, Anca D. Dragan, and Daniel S. Brown. Benchmarks and algorithms for offline preference-based reward learning. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. *Advances in neural information processing systems*, 33:3008–3021, 2020.

Olivier Toubia, John R Hauser, and Duncan I Simester. Polyhedral methods for adaptive choice-based conjoint analysis. *Journal of Marketing Research*, 41(1):116–131, 2004.

Jiaxin Wei and Jia Liu. Personalized fund recommendation with dynamic utility learning. *Financial Innovation*, 11(1):48, 2025.

Christian Wirth, Riad Akrour, Gerhard Neumann, and Johannes Fürnkranz. A survey of preference-based reinforcement learning methods. *Journal of Machine Learning Research*, 18(136):1–46, 2017.

Won-Chol Yang, Jae-Bok Ri, Ji-Yon Yang, and Ju-Song Kim. Materials selection criteria weighting method using analytic hierarchy process (ahp) with simplest questionnaire and modifying method of inconsistent pairwise comparison matrix. *Proceedings of the Institution of Mechanical Engineers, Part L: Journal of Materials: Design and Applications*, 236(1):69–85, 2022.

Sainan Zhang, Shaoyan Guo, Melvyn Sim, and Huifu Xu. Modified polyhedral method for elicitation of shape-free utility and conservatism reduction in robust optimization. *arXiv preprint arXiv:2503.23269*, 2025.

Banghua Zhu, Michael Jordan, and Jiantao Jiao. Principled reinforcement learning with human feedback from pairwise or k-wise comparisons. In *International Conference on Machine Learning*, pp. 43037–43067. PMLR, 2023.

A Experimental Setup and Implementation Details

This appendix describes the Ant-v5 experimental setup, the shared PPO training protocol, and the implementation details of the four methods evaluated in Section 4.

A.1 Ant-v5 Experimental Setup

Ant-v5 controls a quadruped through an 8-dimensional continuous action $a_t \in [-1,1]^8$ and 105-dimensional observation s_t. An episode terminates when the quadruped becomes unhealthy or reaches the maximum horizon of 1000 steps.

The official Ant-v5 reward for the transition (s_t, a_t, s_{t+1}) is

$$r_t^{\mathrm{Ant}} = v_{x,t} + h_{t+1} - \frac{1}{2}\|a_t\|_2^2 - 5\times 10^{-4}\|\widetilde{F}_{t+1}\|_F^2,$$

where $v_{x,t}$ represents the forward velocity of the torso, h_{t+1} represents healthy survival, $\|a_t\|_2^2$ and $\|\widetilde{F}_{t+1}\|_F^2$ represents control cost and contact cost. Specifically,

$$v_{x,t} = \frac{x_{t+1}^{\mathrm{torso}} - x_t^{\mathrm{torso}}}{\Delta t},\ \Delta t = 0.05, \quad \widetilde{F}_{t+1} = \mathrm{clip}(F_{t+1}, -1, 1),$$
$$h_{t+1} = \mathbf{1}\big\{s_{t+1} \text{ is finite and } 0.2 \le z_{t+1}^{\mathrm{torso}} \le 1.0\big\}.$$

Here, x_t^{torso} and z_t^{torso} are components of the state s_t , and F_{t+1} and z_{t+1}^{torso} are components of the state s_{t+1}. Accordingly, we define the 4-dimensional feature mapping as

$$\phi(s_t, a_t, s_{t+1}) = \Big(v_{x,t}, h_{t+1}, -\|a_t\|_2^2, -\|\widetilde{F}_{t+1}\|_F^2\Big)^\top.$$

The official Ant-v5 reward corresponds to the coefficient vector $w = (1, 1, 0.5, 5\times 10^{-4})^\top$. To satisfy the assumption on the reward parameter ($\theta^\star \in \Theta_0 = \big\{\theta \in [0,1]^4 : \mathbf{1}^\top\theta = 1\big\}$), we normalize these coefficients and define$\theta^\star = \frac{w}{\mathbf{1}^\top w}$. The true reward model used throughout the experiments is therefore $r_{\theta^\star}(s_t, a_t, s_{t+1}) = \theta^{\star\top}\phi(s_t, a_t, s_{t+1})$.

A.2 Shared PPO Training Protocol

All methods use the same proximal policy optimization (PPO) implementation for policy optimization. Let $\hat{\pi}_m$ denote the policy at iteration m. At each iteration, we collect $G = 10$ complete trajectories under $\hat{\pi}_m$. The training reward of each transition is determined according to the corresponding method, as described in the following subsections. Based on these rewards, we compute the discounted returns and generalized advantage estimates (GAE), and then perform four PPO epochs on the same batch to update the policy from $\hat{\pi}_m$ to $\hat{\pi}_{m+1}$. Each method is trained for 3000 PPO iterations. For evaluation, the learned policy is executed for 100 independent episodes. All methods are evaluated under the same performance metric $J(\pi)$ regardless of the reward used during policy training. For visualization, we apply an exponential moving average (EMA) to the training curves to reduce short-term fluctuations.

A.3 PbRRL-PCNE

We first construct the reward ambiguity set $\Theta_{\mathrm{PCNE}} = \{\theta : A_N\theta \le b_N\}$ from the pairwise preference queries generated by VCMC. Following Algorithm 1, after constructing the reward ambiguity set, PbRRL-PCNE requires solving the following max-min optimization problem:

$$\max_{\pi} \min_{\theta\in\Theta_{\mathrm{PCNE}}} \theta^\top H(\pi), \tag{9}$$

where

$$H(\pi) := \mathbb{E}_{S_0\sim\rho, A_t\sim\pi(S_t), S_{t+1}\sim P(S_t,A_t)}\left[\sum_{t=0}^{T-1}\gamma^t\phi(S_t, A_t, S_{t+1})\right].$$

We solve Problem equation 9 using a bilevel optimization procedure. For a given policy, the inner problem identifies the worst-case reward parameter in Θ_{PCNE}, while the outer problem updates the policy to maximize the return under this worst-case reward. In our implementation, the outer policy problem is optimized using PPO, and the inner reward problem is a linear program solved by Gurobi.

Specifically, at iteration m, we collect a batch $\mathcal{B}_m = \{\tau_{m,g}\}_{g=1}^{G}$ of G trajectories under the current policy $\hat{\pi}_m$. The corresponding discounted feature expectation is estimated by$\widehat{H}_m = \frac{1}{G}\sum_{g=1}^{G}\sum_{t=0}^{T_{m,g}-1}\gamma^t\phi_{m,g,t}$. Holding $\widehat{H}_m$ fixed, we solve the inner linear program

$$\widehat{\theta}^m_{\mathrm{PCNE}} \in \arg\min_{\theta}\left\{\theta^\top \widehat{H}_m \mid A_N\theta \le b_N\right\}. \tag{10}$$

The reward for each transition in the sampled trajectories is then defined as

$$r^{\mathrm{PCNE}}_{m,g,t} = (\widehat{\theta}^m_{\mathrm{PCNE}})^\top\phi_{m,g,t}.$$

Using these rewards, the shared PPO procedure updates the policy from $\hat{\pi}_m$ to $\hat{\pi}_{m+1}$. At the next iteration, new trajectories are sampled under $\hat{\pi}_{m+1}$, a new $\widehat{H}_{m+1}$ is estimated, and the inner linear program is solved again to obtain the corresponding worst-case reward parameter.

A.4 PbRRL-PCNE-Dual

PbRRL-PCNE-Dual uses the same ambiguity set Θ_{PCNE} as PbRRL-PCNE, which is constructed from the preference queries generated by VCMC and remains fixed during policy optimization. Instead of repeatedly solving the inner linear problem, Theorem 2.4 reformulates equation 9 as the following equivalent single-level constrained problem:

$$\begin{aligned}\max_{\pi,y}\quad & b_N^\top y\\ \text{s.t.}\quad & H(\pi) - A_N^\top y = 0,\\ & y \le 0.\end{aligned}$$

To handle the equality constraint, we define the residual $e(\pi,y) = H(\pi) - A_N^\top y$ and construct the augmented Lagrangian

$$\mathcal{L}_\beta(\pi,y,\lambda) = b_N^\top y + \lambda^\top e(\pi,y) - \frac{\beta}{2}\|e(\pi,y)\|_2^2,$$

where $\lambda \in \mathbb{R}^d$ is a free equality multiplier and $\beta > 0$ is the penalty coefficient. The equality constraint is therefore incorporated into the augmented objective, so that the PPO update of the policy does not contain an explicit equality constraint. The constraint $y \le 0$ is enforced by projection. PbRRL-PCNE-Dual jointly solves this single-level problem by alternating updates of the dual variable, the policy, and the equality multiplier, as described below.

Sampling and updating the dual variable. At iteration m, we collect G trajectories under $\hat{\pi}_m$ and compute $\widehat{H}_m$. Holding $\widehat{H}_m$ and λ_m fixed, we update y by projected gradient ascent. Starting from $y_m^{(0)} = y_m$, the inner updates are

$$\begin{aligned} e_m^{(r)} &= \widehat{H}_m - A_N^\top y_m^{(r)},\\ y_m^{(r+1)} &= \min\left\{y_m^{(r)} + \eta_y\left(b_N - A_N\lambda_m + \beta A_N e_m^{(r)}\right), 0\right\},\end{aligned} \tag{11}$$

where the minimum is applied componentwise and $\eta_y > 0$ is the step size. After the prescribed number of inner updates, the resulting vector is denoted by y_{m+1}.

Constructing the reward and updating the policy. Using the updated dual vector, we recompute and cache

$$e_m = \widehat{H}_m - A_N^\top y_{m+1}, \qquad W_m = \lambda_m - \beta e_m. \tag{12}$$

The reason for using W_m as the training coefficient follows from differentiating the augmented Lagrangian with respect to the feature expectation:

$$\nabla_H \mathcal{L}_\beta = \lambda - \beta\big(H - A_N^\top y\big) =: W. \tag{13}$$

Since $W = \nabla_H \mathcal{L}_\beta$, the policy gradient of the augmented Lagrangian satisfies $\nabla_\pi \mathcal{L}_\beta = (\nabla_\pi H(\pi))^\top W$. When W is fixed within the current policy-update block, we have $(\nabla_\pi H(\pi))^\top W = \nabla_\pi\big(W^\top H(\pi)\big)$. Therefore, optimizing $W^\top H(\pi)$ gives the same local policy update direction as the augmented Lagrangian. And, we define the training reward as $r_t^{\mathrm{Dual}} = W^\top \phi_t$ and use it to update the policy, yielding $\hat{\pi}_{m+1}$.

Updating the equality multiplier. After the policy update, the multiplier is updated using the cached residual:

$$\lambda_{m+1} = \lambda_m - \eta_\lambda e_m, \tag{14}$$

where $\eta_\lambda > 0$ is the multiplier step size. Fresh samples from $\hat{\pi}_{m+1}$ are collected at the next iteration to update the feature-expectation estimate. The auxiliary variables are initialized at $y_0 = 0$ and $\lambda_0 = 0$.

A.5 PbRL-BTL-MLE

Different from PbRRL-PCNE and PbRRL-PCNE-Dual, PbRL-BTL-MLE does not construct a reward ambiguity set by VCMC. Instead, it fit a point estimate $\hat{\theta}_{\mathrm{MLE}}$ from random preference queries using the BTL model with MLE, and then train PPO with the fixed estimated reward $r_{\hat{\theta}_{\mathrm{MLE}}}$. We next describe the random-query generation, BTL-MLE estimation, and the corresponding parameter-convergence comparison.

Random preference query generation. Random preference data are obtained by comparing trajectory pairs sampled from a trajectory pool. We construct this pool by collecting complete Ant-v5 trajectories using three pretrained behavior policies with low, intermediate, and high performance. For each query, two distinct trajectories are randomly sampled from the pool and presented to the DM, who returns a preference based on his/her true expected discounted cumulative returns. For BTL-MLE, we represent the preference label as

$$\delta_{\mathrm{MLE}}^n = \begin{cases} 1, & \text{if the DM chooses } \tau_n^A, \\ 0, & \text{if the DM chooses } \tau_n^B. \end{cases} \tag{15}$$

BTL-MLE reward estimation. The BTL model represents a pairwise preference as a probabilistic choice determined by the difference in expected discounted return between the two trajectories. For the n-th query, the probability of choosing τ_n^A is

$$\begin{aligned} p_n(\theta) &:= \mathbb{P}_\theta\left(\delta_{\mathrm{MLE}}^n = 1 \mid \tau_n^A, \tau_n^B\right) \\ &= \frac{\exp\big(\theta^\top z_n^A/\kappa\big)}{\exp(\theta^\top z_n^A/\kappa) + \exp(\theta^\top z_n^B/\kappa)}, \end{aligned} \tag{16}$$

where $\kappa > 0$ is a temperature parameter. For a fixed nonzero return difference, decreasing κ increases the probability of selecting the higher-return alternative. Motivated by the Assumption 2.2, we set $\kappa = 0.01$.

Given N pairwise comparisons, the maximum-likelihood estimate is obtained by solving

$$\widehat{\theta}_{\mathrm{MLE}}^N \in \arg\min_{\theta \in \Theta_0} \mathcal{L}_{\mathrm{MLE}}^N(\theta),$$

where

$$\mathcal{L}_{\mathrm{MLE}}^N(\theta) = -\frac{1}{N}\sum_{n=1}^N \left[\delta_{\mathrm{MLE}}^n \log p_n(\theta) + (1 - \delta_{\mathrm{MLE}}^n)\log(1 - p_n(\theta))\right].$$

We solve this constrained optimization problem using SLSQP. After obtaining $\widehat{\theta}_{\mathrm{MLE}}^N$, the parameter is fixed throughout policy training, and the reward supplied to PPO is$r_{m,g,t}^{\mathrm{MLE}} = (\widehat{\theta}_{\mathrm{MLE}}^N)^\top \phi_{m,g,t}$.

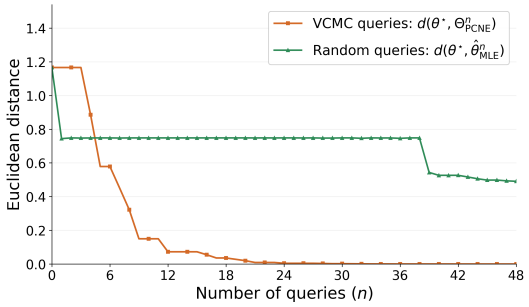

Figure 4: Distance between the point estimation $\hat{\theta}^n_{\text{MLE}}$ and the true reward parameter $\theta^\star$, and distance between the PCNE ambiguity set Θ^n_{PCNE} and the true reward parameter $\theta^\star$, as the number of pairwise queries n increases.

Thus, unlike PbRRL-PCNE, which repeatedly identifies a worst-case reward parameter from an ambiguity set, PbRL-BTL-MLE performs reward estimation once and subsequently applies standard PPO under the estimated reward.

Parameter convergence comparison. We compare the effectiveness of the traditional BTL-MLE method with random preference queries and VCMC in reducing uncertainty about preference.

For an ambiguity set Θ^n_{PCNE} obtained after n cutting steps, we report two quantitative measures: $d(\theta^\star, \Theta^n_{\text{PCNE}}) = \max_{\theta \in \Theta^n_{\text{PCNE}}} \|\theta - \theta^\star\|_2$ measures the farthest distance between the PCNE ambiguity set and the true reward, which is bounded by Proposition C.1; $d(\hat{\theta}^n_{\text{MLE}}, \theta^\star) = \|\hat{\theta}^n_{\text{MLE}} - \theta^\star\|_2$ directly measures the distance between the point estimate and the true reward parameter.

Figure 4 shows that $d(\theta^\star, \Theta^n_{\text{PCNE}})$ decreases rapidly as the number of pairwise queries increases and becomes close to zero after approximately 24 queries. This behavior is consistent with the *exponential convergence rate* guaranteed by Theorem 3.2 In contrast, BTL-MLE based on random trajectory queries retains a substantial parameter error despite receiving additional comparisons. These results highlight the informativeness of VCMC-generated queries and support their effectiveness in reducing reward uncertainty when only a limited number of preference comparisons are available.

A.6 PbRL-True

Train PPO directly on the true reward $r_{\theta^\star}$ as a ground-truth benchmark.

B Initial Cuboid Approximation

We consider two constructions of $\mathcal{C}_0$, which contains the projected initial ambiguity set $\mathcal{P}_0$. The first uses an axis-aligned cuboid defined in assumption B.1. However, the initial cuboid $\mathcal{C}_0$ defined in Assumption B.1 might be overly loose in some certain cases, for example, when the initial polyhedron is sharply shaped but tilted. To mitigate this case and reduce the size of the initial cuboid, the second allows the cuboid to rotate and can provide a tighter approximation. Unless otherwise stated, we use the first construction because of its lower computational cost.

Assumption B.1. The initial approximating cuboid is in the form of

$$\begin{aligned} \mathcal{C}_0 &= \left\{ \mu \in \mathbb{R}^{d-1} \mid |\mu_j - c_{0,j}| \le \tfrac{L_{0,j}}{2}, j = 1, \dots, d-1 \right\} \\ &= \left\{ \mu \in \mathbb{R}^{d-1} \mid A_0^{\mathcal{C}} \mu \le b_0^{\mathcal{C}} \right\}, \end{aligned}$$

where $A_0^{\mathcal{C}} = [I_{d-1}, -I_{d-1}]^\top$, $b_0^{\mathcal{C}} = [\frac{L_0}{2} + c_0, \frac{L_0}{2} - c_0]^\top$, $L_0 = (L_{0,1}, \dots, L_{0,d-1})$ is the edge length, and $c_0 = (c_{0,1}, \dots, c_{0,d-1})$ is the geometric center of the cuboid. Here, the edges are all parallel to the edge directions.

The first $\mathcal{C}_0$, which is a minimal axis-aligned cuboid that fully contains the initial ambiguity set $\mathcal{P}_0$, can be determined by its edge lengths and geometric center:

$$L_{0,j} = \overline{\mu}_j - \underline{\mu}_j, \quad c_{0,j} = \frac{\overline{\mu}_j + \underline{\mu}_j}{2}, \quad j = 1, \ldots, d-1.$$

Here, $\overline{\mu}_j$ and $\underline{\mu}_j$ are the maximum and minimum values of $\mathcal{P}_0$ along the j-th coordinate axis, determined by solving the following linear programs:

$$\overline{\mu}_j = \max_{\mu} \ \mu_j \quad \text{s.t.} \quad A_0^{\mathcal{P}}\mu \le b_0^{\mathcal{P}}, \qquad \underline{\mu}_j = \min_{\mu} \ \mu_j \quad \text{s.t.} \quad A_0^{\mathcal{P}}\mu \le b_0^{\mathcal{P}}.$$

The second cuboid is denoted as

$$\mathcal{C}_0 = \left\{ \mu \in \mathbb{R}^{d-1} \mid |\alpha_j^\top(\mu - c_0)| \le \frac{L_{0,j}}{2}, j = 1, 2, \ldots, d-1 \right\}.$$

In detail, this method involves solving the following optimization problem:

$$\min_{\alpha, L_0} \quad \max(L_{0,1}, L_{0,2}, \ldots, L_{0,d-1}) \tag{17a}$$

$$\text{s.t.} \quad \alpha_j^\top(\mu - c_0) \le \frac{L_{0,j}}{2}, \quad \forall \mu \in \mathcal{P}_0, \quad j = 1, 2, \ldots, d-1 \tag{17b}$$

$$-\alpha_j^\top(\mu - c_0) \le \frac{L_{0,j}}{2}, \quad \forall \mu \in \mathcal{P}_0, \quad j = 1, 2, \ldots, d-1 \tag{17c}$$

$$\alpha_i^\top \alpha_j = 0, \quad i \ne j \tag{17d}$$

$$\alpha_j^\top \alpha_j = 1, \quad j = 1, 2, \ldots, d-1 \tag{17e}$$

Here, c_0 is the analytic center of the polyhedron, $\alpha = (\alpha_1, \ldots, \alpha_{d-1})$ represents the orientation of the cuboid, and $L_0 = (L_{0,1}, \ldots, L_{0,(d-1)})$ denotes the length of its edge.

As $\mathcal{P}_0 := R^{-1}(\Theta_0) = \left\{\mu \in \mathbb{R}^{d-1} \mid A_0^{\mathcal{P}}\mu \le b_0^{\mathcal{P}}\right\}$, we can reformulate equation 17 as the following optimization problem

$$\begin{aligned}
\min_{\alpha, L_0, \lambda, \eta} \quad & L_{\max} \\
\text{s.t.} \quad & L_{0,j} \le L_{\max}, \quad j = 1, 2, \ldots, d-1, \\
& -\alpha_j^\top - \eta_j^\top A_0^{\mathcal{P}} = 0, \quad j = 1, 2, \ldots, d-1, \\
& \alpha_j^\top - \lambda_j^\top A_0^{\mathcal{P}} = 0, \quad j = 1, 2, \ldots, d-1, \\
& \eta_j^\top b_0^{\mathcal{P}} + \alpha_j^\top c_0 \le \frac{L_{0,j}}{2}, \quad j = 1, 2, \ldots, d-1, \\
& \lambda_j^\top b_0^{\mathcal{P}} - \alpha_j^\top c_0 \le \frac{L_{0,j}}{2}, \quad j = 1, 2, \ldots, d-1, \\
& \alpha_i^\top \alpha_j = 0, \quad i \ne j, \\
& \alpha_j^\top \alpha_j = 1, \quad j = 1, 2, \ldots, d-1, \\
& \lambda_j, \eta_j \in \mathbb{R}_+^{2d-1}, \ j = 1, 2, \ldots, d-1.
\end{aligned} \tag{18}$$

C Convergence of VCMC on the un-projected parameter

Proposition C.1. *Let $\theta^\star \in \Theta_0$ be the true reward parameter and define $\mu^* := R^{-1}(\theta^\star) \in \mathbb{R}^{d-1}$. Then for any $\mu \in \mathbb{R}^{d-1}$,*

$$\|R(\mu) - R(\mu^*)\|_2 \le (1 + \sqrt{d-1})\|\mu - \mu^*\|_2,$$

and

$$\theta^\star = R(\mu^*) \in \Theta_{\mathrm{PCNE}}.$$

Proof. By the definition of R, we have

$$\|R(\mu) - R(\mu^*)\|_2 = \sqrt{\sum_{j=1}^{d-1}(\mu_j - \mu_j^*)^2 + \Big(\sum_{j=1}^{d-1}(\mu_j - \mu_j^*)\Big)^2} \tag{19}$$

$$\leq \sqrt{\sum_{j=1}^{d-1}(\mu_j - \mu_j^*)^2} + \sqrt{(d-1)\sum_{j=1}^{d-1}(\mu_j - \mu_j^*)^2} \tag{20}$$

$$= (1 + \sqrt{d-1})\|\mu - \mu^*\|_2. \tag{21}$$

From equation 19 to equation 20, we used $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$ with $a = \sum_{j=1}^{d-1}(\mu_j - \mu_j^*)^2$ and $b = \Big(\sum_{j=1}^{d-1}(\mu_j - \mu_j^*)\Big)^2$, and Cauchy–Schwarz: $\Big(\sum_{j=1}^{d-1} a_j\Big)^2 \leq (d-1)\sum_{j=1}^{d-1} a_j^2$ with $a_j = \mu_j - \mu_j^*$.

It remains to show that $R(\mu^*) = \theta^\star \in \Theta_{\mathrm{PCNE}}$. By the definition $\mu^* := R^{-1}(\theta^\star)$ and $\langle \mathbf{1}, \theta^\star \rangle = 1$ (since $\theta^\star \in \Theta_0$), we have $R(\mu^*) = \theta^\star$. Moreover, by Assumption 2.2 and the definition of Θ_{PCNE} in equation 3, we conclude that $\theta^\star \in \Theta_{\mathrm{PCNE}}$. □

D PROOFS

D.1 PROOF OF THEOREM 2.4

Proof. Fix any policy π and we can define

$$H(\pi) \coloneqq \mathbb{E}_{S_0 \sim \rho, A_t \sim \pi(S_t), S_{t+1} \sim P(S_t, A_t)} \left[\sum_{t=0}^{T-1} \gamma^t \phi(S_t, A_t, S_{t+1}) \right].$$

By Assumption 2.1 and the linearity of expectation, we can reformulate $\hat{J}(\pi)$ as

$$\min_{\theta \in \Theta_{\mathrm{PCNE}}} \theta^\top H(\pi). \tag{22}$$

According to equation 3, $\Theta_{\mathrm{PCNE}} = \{\theta : A_N \theta \leq b_N\}$ is a nonempty bounded polyhedron. Hence, the inner LP is feasible and attains a finite optimum, and strong duality applies. Therefore,

$$\begin{aligned} \min_{\theta} \quad & \theta^\top H(\pi) \\ \text{s.t.} \quad & A_N \theta \leq b_N \end{aligned} \iff \begin{aligned} \max_{y} \quad & b_N^\top y \\ \text{s.t.} \quad & A_N^\top y = H(\pi), \\ & y \leq 0. \end{aligned}$$

Substituting into $\max_\pi \hat{J}(\pi)$ and merging the two maximizations yields the claimed single-level formulation.

$$\begin{aligned} \max_{\pi, y} \quad & b_N^\top y \\ \text{s.t.} \quad & A_N^\top y = H(\pi), \\ & y \leq 0. \end{aligned} \tag{23}$$

□

D.2 PROOF OF THEOREM 3.1

Proof. We first prove a more general statement where the cuboid edge direction α_k is not necessarily aligned with the edge directions. Next, we will prove that the following two constraint conditions are equivalent.

1. $(R(\mu))^\top (z^A_{i,k} - z^B_{i,k}) = 0, \quad \forall \mu \in \mathcal{H}_{i,k} = \left\{\mu \in \mathbb{R}^{d-1} \mid \alpha_k^\top(\mu - c_{i(d-1)}) = 0\right\}.$

2. $$\begin{cases} z^A_{i,k,j} - z^B_{i,k,j} - z^A_{i,k,d} + z^B_{i,k,d} = \lambda\alpha_{k,j}, \quad j = 1, \dots, d-1, \\ z^A_{i,k,d} - z^B_{i,k,d} = -\lambda \sum_{j=1}^{d-1} \alpha_{k,j} c_{i(d-1),j}. \end{cases}$$

To establish the equivalence, we start by expanding the first condition.

$$(R(\mu))^\top (z^A_{i,k} - z^B_{i,k}) = 0 \tag{24a}$$

$$\Rightarrow (\mu_1, \mu_2, \dots, \mu_{d-1}, 1 - \sum_{j=1}^{d-1} \mu_j)^\top (z^A_{i,k1} - z^B_{i,k1}, z^A_{i,k2} - z^B_{i,k2}, \dots, z^A_{i,k,d} - z^B_{i,k,d}) = 0 \tag{24b}$$

$$\Rightarrow \sum_{j=1}^{d-1} \mu_j (z^A_{i,k,j} - z^B_{i,k,j}) + (1 - \sum_{j=1}^{d-1} \mu_j)(z^A_{i,k,d} - z^B_{i,k,d}) = 0 \tag{24c}$$

$$\Rightarrow \sum_{j=1}^{d-1} \mu_j (z^A_{i,k,j} - z^B_{i,k,j} - z^A_{i,k,d} + z^B_{i,k,d}) + (z^A_{i,k,d} - z^B_{i,k,d}) = 0 \tag{24d}$$

Next, we expand the hyperplane condition that defines $\mu \in \mathcal{H}_{i,k}$.

$$\alpha_k^\top(\mu - c_{i(d-1)}) = 0 \tag{25a}$$

$$\Rightarrow (\alpha_{k,1}, \alpha_{k,2}, \dots, \alpha_{k,d-1})^\top (\mu_1 - c_{i(d-1),1}, \mu_2 - c_{i(d-1),2}, \dots, \mu_{d-1} - c_{i(d-1),(d-1)}) = 0 \tag{25b}$$

$$\Rightarrow \sum_{j=1}^{d-1} \alpha_{k,j} (\mu_j - c_{i(d-1),j}) = 0 \tag{25c}$$

$$\Rightarrow \sum_{j=1}^{d-1} \alpha_{k,j} \mu_j - \sum_{j=1}^{d-1} \alpha_{k,j} c_{i(d-1),j} = 0 \tag{25d}$$

For $\forall \mu \in \mathcal{H}_{i,k} = \left\{\mu \in \mathbb{R}^{d-1} \mid \alpha_k^\top(\mu - c_{i(d-1)}) = 0\right\}$, the equality $(R(\mu))^\top (z^A_{i,k} - z^B_{i,k}) = 0$ holds. Therefore, we obtain that $\mu \in \mathcal{H}_{i,k} \cap \left\{\mu \in \mathbb{R}^{d-1} \mid (R(\mu))^\top (z^A_{i,k} - z^B_{i,k}) = 0\right\}$. It follows that $(24d) \iff (25d)$, such that

$$z^A_{i,k,j} - z^B_{i,k,j} - z^A_{i,k,d} + z^B_{i,k,d} = \lambda\alpha_{k,j}, \quad j = 1, \dots, d-1, \tag{26a}$$

$$z^A_{i,k,d} - z^B_{i,k,d} = -\lambda \sum_{j=1}^{d-1} \alpha_j^\top c_{i(d-1),j}, \tag{26b}$$

$$\lambda > 0, \tag{26c}$$

Without loss of generality, we impose $\lambda > 0$, since for $\lambda < 0$ we can simply exchange $A_{i,k}$ and $B_{i,k}$, which changes the sign of λ while leaving the induced hyperplane unchanged. The reverse derivation still holds, thus the two constraints are equivalent.

In particular, when $\alpha_k = e_k, \lambda = 1$, we have

1. $(R(\mu))^\top (z^A_{i,k} - z^B_{i,k}) = 0, \quad \forall \mu \in \mathcal{H}_{i,k} = \left\{\mu \in \mathbb{R}^{d-1} \mid e_k^\top(\mu - c_{i(d-1)}) = 0\right\}.$

2. $$\begin{cases} z^A_{i,k,j} - z^B_{i,k,j} - z^A_{i,k,d} + z^B_{i,k,d} = 0, \quad j = 1, 2, \dots, k-1, k+1, \dots, d-1, \\ z^A_{i,k,k} - z^B_{i,k,k} - z^A_{i,k,d} + z^B_{i,k,d} = 1, \\ z^A_{i,k,d} - z^B_{i,k,d} = -c_{i(d-1),k}. \end{cases}$$

Then we consider the nonlinear objective

$$\max_{z_{i,k}^A, z_{i,k}^B \in \mathbb{R}^d} |\left(R(\mu_{i,k}^A)\right)^\top z_{i,k}^A - \left(R(\mu_{i,k}^B)\right)^\top z_{i,k}^B| \tag{27}$$

To linearize the absolute-value objective in equation 27, we introduce an auxiliary nonnegative variable $U \in \mathbb{R}$ representing the objective value and impose $U \leq |\Delta|$, where

$$\Delta := \left(R(\mu_{i,k}^A)\right)^\top z_{i,k}^A - \left(R(\mu_{i,k}^B)\right)^\top z_{i,k}^B.$$

The nonlinear constraint $U \leq |\Delta|$ is equivalent to the disjunction $U \leq \Delta$ or $U \leq -\Delta$. This disjunction can be encoded via a standard big-M formulation by introducing a binary variable $b \in \{0, 1\}$, yielding the following mixed-integer linear program:

$$\max_{z_{i,k}^A, z_{i,k}^B \in \mathcal{Z}_{traj}, b \in \{0,1\}, U \in \mathbb{R}} \quad U \tag{28a}$$

$$\text{s.t.} \quad U \leq \left(R(\mu_{i,k}^A)\right)^\top z_{i,k}^A - \left(R(\mu_{i,k}^B)\right)^\top z_{i,k}^B + M(1-b), \tag{28b}$$

$$U \leq \left(R(\mu_{i,k}^B)\right)^\top z_{i,k}^B - \left(R(\mu_{i,k}^A)\right)^\top z_{i,k}^A + Mb, \tag{28c}$$

$$U \geq 0. \tag{28d}$$

Here M is a sufficiently large constant such that $2|\Delta| \leq M$ for all feasible $(z_{i,k}^A, z_{i,k}^B)$. Since the objective maximizes U, the optimal solution satisfies $U = |\Delta|$, and hence equation 28 is equivalent to the original problem equation 27.

□

D.3 Proof of Theorem 3.2

Proof. (i) By construction, $\mathcal{P}_0 = R^{-1}(\Theta_0)$ and $\theta^\star \in \Theta_0$, hence $\mu^* = R^{-1}(\theta^\star) \in \mathcal{P}_0$. Since $\mathcal{C}_0$ is chosen as the minimal axis-aligned cuboid containing $\mathcal{P}_0$, we have $\mu^* \in \mathcal{C}_0$.

Consider an arbitrary iteration $n \geq 1$. Suppose $\mathcal{C}_{n-1}$ is the current cuboid, and a pairwise comparison query (A_n, B_n) with trajectory feature vectors (z_n^A, z_n^B) is constructed by Algorithm 2.

By Assumption 2.2, the DM prefers the alternative with the larger discounted cumulative reward under the true parameter $\theta^\star$. Therefore, its response selects the halfspace

$$\left\{\mu \in \mathbb{R}^{d-1} \;\middle|\; \delta_n \, R(\mu)^\top (z_n^A - z_n^B) \geq 0\right\},$$

which contains μ^*.

Consequently, according to the VCMC algorithm, the updated cuboid is

$$\mathcal{C}_n = \mathcal{C}_{n-1} \cap \left\{\mu \in \mathbb{R}^{d-1} \mid \delta_n \, R(\mu)^\top (z_n^A - z_n^B) \geq 0\right\},$$

and hence $\mu^* \in \mathcal{C}_n$ whenever $\mu^* \in \mathcal{C}_{n-1}$.

Since we have already established $\mu^* \in \mathcal{C}_0$, an induction on n implies that $\mu^* \in \mathcal{C}_n, \; \forall \, n \geq 0$.

(ii) Fix an arbitrary round $i = 0, 1, \ldots, I-1$ and $k = 1, \ldots, d-1$. By Algorithm 2, the cutting hyperplane is

$$\mathcal{H}_{i,k} = \{\mu \in \mathbb{R}^{d-1} \mid e_k^\top (\mu - c_{i(d-1)}) = 0\}.$$

By Theorem 3.1 and equation 24, it holds that

$$\delta_{i,k} \left(R(\mu)\right)^\top (z_{i,k}^A - z_{i,k}^B) = \delta_{i,k} \, \lambda(\mu_k - c_{i(d-1)}), \; \forall k = 1, \ldots, d-1$$

and hence the indifference set of the comparison coincides with the hyperplane $H_{i,k}$. Therefore, the update in Algorithm 2 keeps one of the two halfspaces separated by $H_{i,k}$. Since $H_{i,k}$ passes through the geometric center $c_{i(d-1)}$ and is orthogonal to the k-th edge of the cuboid $\mathcal{C}_{i(d-1)+(k-1)}$, it bisects the k-th edge of the current cuboid, i.e., it halves the edge length along e_k while leaving all other edge lengths unchanged.

Consequently, writing $n = i(d-1)+k$ with $1 \le k \le d-1$, the edge length of $\mathcal{C}_n$ along coordinate j equals $\frac{L_{0,j}}{2^{i+1}}$ for $1 \le j \le k$ and $\frac{L_{0,j}}{2^i}$ for $k < j \le d-1$. Therefore, for any $\mu \in \mathcal{C}_n$ and using $\mu^* \in \mathcal{C}_n$,

$$|\mu_j - \mu_j^*| \le \begin{cases} L_{0,j}/2^{i+1}, & j \le k, \\ L_{0,j}/2^i, & j > k, \end{cases}$$

which implies

$$\|\mu - \mu^*\|_2 \le \sqrt{\sum_{j=1}^{k} \left(\frac{L_{0,j}}{2^{i+1}}\right)^2 + \sum_{j=k+1}^{d-1} \left(\frac{L_{0,j}}{2^i}\right)^2}.$$

(iii) Note that $\{\mathcal{C}_n\}_{n\ge 0}$ is nested and nonempty with $\mu^* \in \mathcal{C}_n$ for all n. Moreover, the length of the longest diagonal of $\mathcal{C}_n$ is bounded by $\frac{1}{2^i}\sqrt{\sum_{j=1}^{d-1} L_{0,j}^2} \to 0$, as $n \to \infty$ (since $i = \frac{n-k}{d-1} \to \infty$). Hence for any $\varepsilon > 0$ there exists $\bar{N}$ such that $\mathcal{C}_n \subseteq B(\mu^*, \varepsilon)$ for all $n \ge \bar{N}$, which yields $\lim_{n\to\infty} \mathcal{C}_n = \{\mu^*\}$. □

D.4 Proof of Theorem 3.3

Proof. We start from the definition of the suboptimality gap:

$$\mathrm{SubOpt}(\hat{\pi}_{\mathrm{PCNE}}) = J(\pi^\star) - J(\hat{\pi}_{\mathrm{PCNE}}).$$

We insert and subtract $\hat{J}(\pi^\star)$ and $\hat{J}(\hat{\pi}_{\mathrm{PCNE}})$ to obtain

$$\mathrm{SubOpt}(\hat{\pi}_{\mathrm{PCNE}}) = \big(J(\pi^\star) - \hat{J}(\pi^\star)\big) + \big(\hat{J}(\pi^\star) - \hat{J}(\hat{\pi}_{\mathrm{PCNE}})\big) + \big(\hat{J}(\hat{\pi}_{\mathrm{PCNE}}) - J(\hat{\pi}_{\mathrm{PCNE}})\big).$$

By definition of $\hat{\pi}_{\mathrm{PCNE}}$ as a maximizer of $\hat{J}(\cdot)$,

$$\hat{J}(\pi^\star) - \hat{J}(\hat{\pi}_{\mathrm{PCNE}}) \le 0.$$

Moreover, by Proposition C.1, $\theta^\star \in \Theta_{\mathrm{PCNE}}$, hence for any policy π,

$$\hat{J}(\pi) = \min_{\theta \in \Theta_{\mathrm{PCNE}}} \mathbb{E}\left[\sum_{t=0}^{T-1} \gamma^t \theta^\top \phi_(s_t, a_t, s_{t+1})\right] \le \mathbb{E}\left[\sum_{t=0}^{T-1} \gamma^t \theta^{*\top} \phi_(s_t, a_t, s_{t+1})\right] = J(\pi),$$

which implies $\hat{J}(\pi) - J(\pi) \le 0$ for all π, and in particular

$$\hat{J}(\hat{\pi}_{\mathrm{PCNE}}) - J(\hat{\pi}_{\mathrm{PCNE}}) \le 0.$$

Combining these inequalities yields

$$\mathrm{SubOpt}(\hat{\pi}_{\mathrm{PCNE}}) \le J(\pi^\star) - \hat{J}(\pi^\star).$$

We now bound the right-hand side. By definition,

$$J(\pi^\star) - \hat{J}(\pi^\star) = (\theta^*)^\top \mathbb{E}\left[\sum_{t=0}^{T-1} \gamma^t \phi_{A_t \sim \pi^\star(S_t)}(S_t, A_t, S_{t+1})\right] - \min_{\theta \in \Theta_{\mathrm{PCNE}}} \theta^\top \mathbb{E}\left[\sum_{t=0}^{T-1} \gamma^t \phi_{A_t \sim \pi^\star(S_t)}(S_t, A_t, S_{t+1})\right]$$

$$= \max_{\theta \in \Theta_{\mathrm{PCNE}}} (\theta^\star - \theta)^\top \mathbb{E}\left[\sum_{t=0}^{T-1} \gamma^t \phi_{A_t \sim \pi^\star(S_t)}(S_t, A_t, S_{t+1})\right].$$

Using the Cauchy-Schwarz inequality,

$$J(\pi^\star) - \hat{J}(\pi^\star) \le \max_{\theta \in \Theta_{\mathrm{PCNE}}} \|\theta^\star - \theta\|_2 \cdot \left\| \mathbb{E}\left[\sum_{t=0}^{T-1} \gamma^t \phi_{A_t \sim \pi^\star(S_t)}(S_t, A_t, S_{t+1})\right] \right\|_2.$$

By Theorems 3.2 and Proposition C.1, for every parameter $\theta \in \Theta_{\mathrm{PCNE}}$ induced by some $\mu \in \mathcal{C}_n$, we have

$$\|\theta^\star - \theta\|_2 \le (1 + \sqrt{d-1})\|\mu^* - \mu\|_2 \le \overline{B}_n.$$

Therefore, we can write

$$\max_{\theta \in \Theta_{\mathrm{PCNE}}} \|\theta^\star - \theta\|_2 \leq \overline{B}_n.$$

According to the linear of expectation, the Cauchy-Schwarz inequality and $\gamma \in [0, 1]$, we get

$$\left\| \mathbb{E}\left[\sum_{t=0}^{T-1} \gamma^t \phi_{A_t \sim \pi^\star(S_t)}(S_t, A_t, S_{t+1}) \right] \right\|_2 \leq \sum_{t=0}^{T-1} \left\| \mathbb{E}\left[\phi_{A_t \sim \pi^\star(S_t)}(S_t, A_t, S_{t+1}) \right] \right\|_2.$$

Next, by assumption of the feature mapping that satisfies $\sup_{s_t \in \mathcal{S}_t, a_t \in \mathcal{A}_t, s_{t+1} \in \mathcal{S}_{t+1}} \|\phi(s_t, a_t, s_{t+1})\|_2 \leq \overline{B}_\phi$, where $\overline{B}_\phi$ is a fixed constant, we have

$$\sum_{t=0}^{T-1} \left\| \mathbb{E}\left[\phi_{A_t \sim \pi^\star(S_t)}(S_t, A_t, S_{t+1}) \right] \right\|_2 \leq \sum_{t=0}^{T-1} \mathbb{E}\left[\left\| \phi_{A_t \sim \pi^\star(S_t)}(S_t, A_t, S_{t+1}) \right\|_2 \right] \leq T\overline{B}_\phi$$

Combining the above inequalities, we obtain

$$\mathrm{SubOpt}(\hat{\pi}_{\mathrm{PCNE}}) \leq T\overline{B}_n \cdot \overline{B}_\phi,$$

which is the desired bound. Since $\overline{B}_n \to 0$ as $n \to \infty$ (Theorem 3.2), we obtain $\mathrm{SubOpt}(\hat{\pi}_{\mathrm{PCNE}}) \to 0$.

□