PB²: Preference Space Exploration via Population-Based Methods in Preference-Based Reinforcement Learning

Anonymous Author(s)
Affiliation
Address
email

Abstract

Preference-based reinforcement learning (PbRL) has emerged as a promising approach for learning behaviors from human feedback without predefined reward functions. However, current PbRL methods face a critical challenge in effectively exploring the preference space, often converging prematurely to suboptimal policies that satisfy only a narrow subset of human preferences. In this work, we identify and address this preference exploration problem through population-based methods. We demonstrate that maintaining a diverse population of agents enables more comprehensive exploration of the preference landscape compared to single-agent approaches. Crucially, this diversity improves reward model learning by generating preference queries with clearly distinguishable behaviors, a key factor in real-world scenarios where humans must easily differentiate between options to provide meaningful feedback. Our experiments reveal that current methods may fail by getting stuck in local optima, requiring excessive feedback, or degrading significantly when human evaluators make errors on similar trajectories, a realistic scenario often overlooked by methods relying on perfect oracle teachers. Our population-based approach demonstrates robust performance when teachers mislabel similar trajectory segments and shows significantly enhanced preference exploration capabilities, particularly in environments with complex reward landscapes.

1 Introduction

Reinforcement learning (RL) has demonstrated remarkable success across a wide range of applications, from game playing to robotic control. However, the effectiveness of traditional RL methods remains heavily dependent on carefully designed reward functions, which are often challenging to specify for complex tasks involving subjective outcomes or intricate human preferences [Hadfield-Menell et al., 2017]. Preference-based reinforcement learning (PbRL) offers a promising alternative by enabling agents to learn directly from human feedback through preferences between pairs of behavior trajectories [Christiano et al., 2017, Lee et al., 2021a]. This approach eliminates the need for hand-crafted reward functions and provides a more intuitive interface for humans to express their intentions. Despite these advantages, PbRL faces a fundamental challenge: policies optimized for the current reward model often generate similar queries for preference elicitation, limiting the informativeness of human feedback needed to improve the model.

Current PbRL implementations typically use single-policy approaches that often converge too early to behaviors representing only a limited subset of human preferences. This happens because these methods increasingly generate similar trajectory segments pairs for human evaluation, limiting the range of behaviors presented. As a result, the reward model learns from less diverse examples over time, leading to poor exploration and suboptimal policies. The problem becomes worse when humans

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

need to evaluate trajectory segments pairs with only minor differences. In such cases, evaluators often give inconsistent feedback, adding noise that significantly degrades learning performance.

Existing approaches attempt to address preference illiciation challenges through various query selection strategies: uncertainty-based methods targeting uncertain preference regions [Marta et al., 2023], policy-aligned techniques focusing on current behavior [Hu et al., 2024], ensemble-based strategies leveraging model disagreement [Liang et al., 2022], and information-theoretic approaches maximizing information gain [Biyik et al., 2020, Biyik and Sadigh, 2018]. While improving sample efficiency, these methods operate within a single-agent framework, limiting behavioral diversity during evaluation. This becomes problematic when humans must compare similar trajectories, often leading to inconsistent feedback [Huang et al., 2025]. Without mechanisms for maintaining diversity, existing approaches remain vulnerable to local optima, requiring excessive feedback and degrading under realistic conditions of human evaluation inconsistency.

In this paper, we present PB²: a novel population-based approach to preference space exploration in PbRL. Our key insight is that simultaneously training multiple distinct policies facilitates more thorough preference landscape exploration than conventional single-policy methods. By collecting experiences across different policies to construct comparison pairs, we substantially enhance the variety of behaviors evaluated while preserving alignment with expressed human preferences. This behavioral diversity significantly improves reward model training by creating preference queries that are more easily distinguishable, an essential consideration often overlooked by existing methods that implicitly assume humans can reliably evaluate even subtly different behaviors. As illustrated in Figure 1, our approach implements this insight through a feedback loop where diverse policies generate distinct trajectories, human preferences on these trajectories train a reward model, and a discriminator maintains population diversity while encouraging behaviors that align with current preferences.

Our contributions are as follows:

1. We identify the preference exploration problem in PbRL, demonstrating how single-agent methods frequently fail by converging to suboptimal local minima in the preference space.
2. We propose a population-based framework for PbRL that maintains policy diversity while optimizing for human preferences, significantly enhancing exploration of the preference landscape.
3. We demonstrate through experiments that PB² produces more distinguishable queries that improve reward learning efficiency, achieving greater robustness when human feedback is inconsistent and improving performance with limited feedback.

We validate these claims through three complementary experimental evaluations: (1) a systematic evaluation across DMControl locomotion tasks with varying similarity thresholds ϵ to simulate human judgment inconsistency, and (2) a qualitative demonstration of how PB² escapes local optima in complex preference landscapes where single-agent methods remain trapped and (3) a comparative analysis in navigation tasks with extremely limited feedback, demonstrating PB²'s feedback efficiency in this limited setting.

The remainder of this paper is organized as follows: Section 2 reviews related work in preference-based RL and population-based methods. Section 3 establishes preliminaries and formalizes the preference exploration problem. Section 4 presents our population-based approach in detail. Section 5 describes our experimental setup, and Section 6 presents and analyzes our results. Finally, Section 7 discusses limitations, future directions, and broader implications of our work.

2 Related Work

Preference-based Reinforcement Learning Preference-based Reinforcement Learning (PbRL) enables agents to learn from human feedback through trajectory comparisons [Christiano et al., 2017, Lee et al., 2021a, Wirth et al., 2017]. Despite its intuitive interface, PbRL faces a fundamental exploration challenge, as policies optimized for the current reward model often generate similar queries for preference elicitation, limiting the informativeness of human feedback needed to improve the model. Recent methods address this through reward uncertainty [Liang et al., 2022], semi-supervised learning [Park et al., 2022], model-based approaches [Liu et al., 2023], bi-level optimization [Liu et al., 2022],

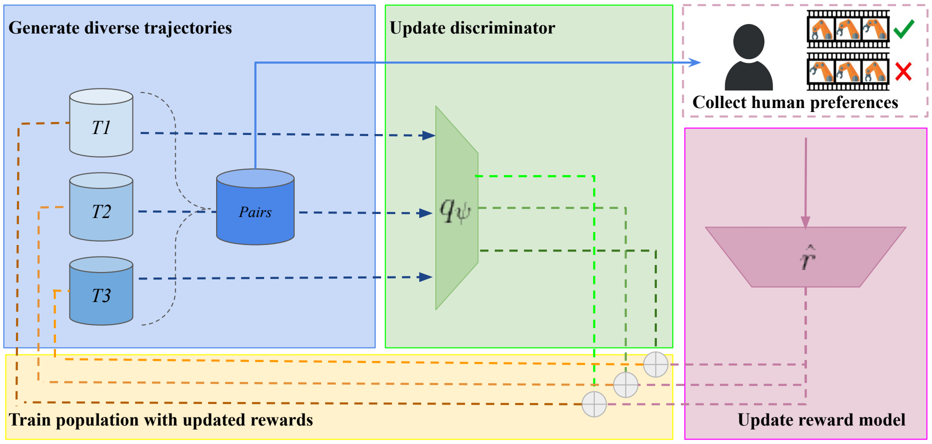
Generate diverse trajectories
T1
T2
T3
Pairs
Update discriminator
q_ψ
Collect human preferences
$\hat{r}$
Train population with updated rewards
Update reward model

Figure 1: **Overview of PB².** After an initial unsupervised exploration phase, we sample experience from different agents and form comparison pairs to train a reward model using human feedback. A discriminator is trained on the experience of the agents, with the goal of maintaining diversity in the population. The discriminator exploration bonus is added to the learned reward, in order to encourage the discovery of different behaviours, aligned with the current human preferences, thereby helping to elicit more informative and refined preferences.

dynamics encoding [Metcalf et al., 2022], and query-policy alignment (QPA) [Hu et al., 2024], though they remain primarily exploitative. Our approach extends QPA's insights while addressing exploration-exploitation tradeoffs through diversity that generates distinguishable behaviors crucial for meaningful human feedback.

Population-based Reinforcement Learning Population-based methods maintain multiple agents to enhance exploration, from Population-Based Training [Jaderberg et al., 2017] to extensions incorporating Bayesian optimization [Wan et al., 2022], long-term performance [Dalibard and Jaderberg, 2021], and evolutionary selection [Salimans et al., 2017, Alam et al., 2020]. Our approach shares core principles with these methods but targets preference landscape exploration rather than hyperparameter optimization, building on diversity-promoting concepts like DvD [Parker-Holder et al., 2020] while uniquely aligning diversity with human preferences.

Quality-Diversity Quality Diversity algorithms discover diverse high-performing solutions, evolving from Novelty Search [Lehman, 2012] through approaches like NSLC [Lehman and Stanley, 2011] and MAP-Elites [Mouret and Clune, 2015], with recent integration into RL [Nilsson and Cully, 2021, Cully, 2019]. Unlike QD methods requiring manually designed descriptors, our method automatically identifies meaningful behavioral patterns while adapting to evolving human preferences rather than pursuing general diversity.

Unsupervised Skill Discovery Skill discovery leverages intrinsic motivation through mutual information maximization [Eysenbach et al., 2019], dynamics-awareness [Sharma et al., 2020], and task-aligned methods [Kumar et al., 2020, Osa et al., 2022]. Our approach is inspired by SMERL [Kumar et al., 2020] but operates without expert access, encouraging distinctiveness between behaviors from different population members rather than between latent-conditioned policies of a single agent.

3 Preliminaries

Reinforcement Learning We consider the standard reinforcement learning framework with an environment modeled as a Markov Decision Process (MDP) $M := (S, A, T, r, \gamma)$ [Sutton et al., 1998], where S is the state space, A is the action space, T is the transition function, r is the reward function, and γ is the discount factor. The agent's goal is to find a policy π that maximizes the expected discounted cumulative reward, $\mathbb{E}_\pi[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)]$.

Preference-Based Reinforcement Learning In preference-based reinforcement learning (PbRL) [Christiano et al., 2017, Wirth et al., 2017], the agent lacks access to an explicit reward function. Instead, a human teacher provides preference feedback between pairs of trajectory segments (σ^0, σ^1), where a segment σ is a sequence of state-action pairs. Following the Bradley-Terry model [Bradley and Terry, 1952, Fürnkranz et al., 2012], we learn a reward function $\hat{r}_\phi$ that induces a preference predictor:

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

The reward function is trained by minimizing the cross-entropy loss:

$$L_{\text{reward}} = -\mathbb{E}_{(\sigma^0,\sigma^1,y)\sim D}\left[y^{(0)} \log P_\phi[\sigma^0 \succ \sigma^1] + y^{(1)} \log P_\phi[\sigma^1 \succ \sigma^0]\right] \tag{2}$$

This learned reward function guides policy training through standard RL methods. Like recent PbRL methods [Lee et al., 2021b, Hu et al., 2024, Liang et al., 2022], we also use Soft Actor-Critic (SAC) [Haarnoja et al., 2018] as our base RL algorithm.

4 Balancing Preference Space Exploration and Exploitation

In preference-based reinforcement learning (PbRL), human feedback serves as a crucial but costly resource. The strategic selection of queries presented to human evaluators significantly impacts the learning process. PbRL systems face a fundamental challenge: effectively exploring the preference space to discover human preferences while exploiting current knowledge to improve policies, requiring two competing objectives:

- **Exploitation:** Refining the reward model in areas most relevant to current policy improvement
- **Exploration:** Discovering previously unknown aspects of human preferences through diverse queries

Current approaches struggle with this balance. PEBBLE [Lee et al., 2021b] samples queries from outdated trajectories, creating temporal misalignment between feedback and current policies. Uncertainty-driven approaches like RUNE [Liang et al., 2022] often prioritize informationally rich regions that may be irrelevant to current capabilities. Even QPA [Hu et al., 2024], which aligns queries with current policies, becomes overly exploitative, creating blind spots in the preference model and potentially trapping agents in local optima.

To overcome these limitations, our approach employs a **population of agents**, where each agent maintains a slightly different policy. This population collectively generates trajectories that explore different preference regions while remaining aligned with known preferences. Our method actively encourages agents to develop distinct behavioral patterns, enhancing the natural diversity from maintaining multiple agents. The population-based approach offers several advantages: (1) Natural behavioral diversity that spans the preference landscape (2) More informative queries with clearly distinguishable behaviors (3) Targeted exploration relevant to learning human preferences (4) Maintained alignment with current reward model while discovering new preference information

4.1 Motivating Example

Figure 2 compares preference exploration strategies in a 2D navigation task, where the agent starts at the bottom left and the goal is at the top right. Trajectories from a single-agent, using QPA in this case, concentrate in high-reward regions but provide limited exploration, thus hindering future queries. In contrast, our population-based method generates trajectories from multiple agents that collectively cover a wider area of the state space, including both high-reward regions and informative boundary areas. Trajectories from different population members are more distinguishable, making preference queries more informative for human evaluators.

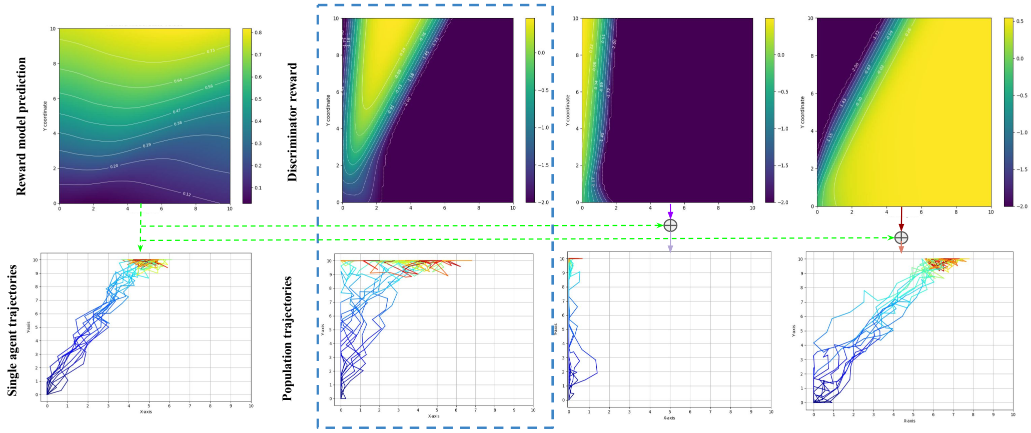

Figure 2: **Diverse population-based query strategy improves preference space coverage compared to single-agent approaches.** ***(Top row)*** (Column 1) Reward model predictions across the state space, showing higher rewards in the upper region. (Column 2-4) Diversity bonus (Sec. 5.1) for three different agents, with distinct spatial concentrations. ***(Bottom row)*** (Column 1) Single agent trajectories (QPA), (Column 2-4) Trajectories from 3 agents of PB². QPA maximizes only the reward model, while PB² agents maximizes the combination of reward model predictions and the diversity bonus. The agent in the blue box does not receive the diversity bonus (Sec. 5.1). Note how agents naturally converge to different strategies that satisfy both the learned reward model (reaching upper regions) and the diversity bonuses (exploring different areas), demonstrating comprehensive preference space exploration. PB² achieves better coverage of both high-reward regions and informative boundary areas compared to the single-agent approach. Comparing trajectories across agents is also easier (distinguishable) and more informative than in the single-agent approach.

5 PB²: Population-Based Preference-Based Reinforcement Learning

In this section, we present PB²: a **P**opulation-**B**ased approach for **P**reference-**B**ased Reinforcement Learning that effectively explores the preference landscape while maintaining alignment with human preferences.

As illustrated in Figure 1, PB² creates a feedback loop where population diversity enhances preference learning. Our approach maintains a population of policies that collectively explore the preference space while remaining individually aligned with current human feedback, ensuring comprehensive exploration without sacrificing performance on known preferences.

The central advantage of PB² lies in its information-theoretic formulation that maximizes the mutual information between policies and their state distributions. This encourages each agent to explore a distinct subset of states that can be readily identified by a discriminator, naturally generating clearly distinguishable behaviors for human evaluation. To maintain effective diversity in a continuously evolving preference landscape, we employ an adaptive discriminator that identifies policy-specific behavioral patterns while remaining aligned with the current reward model, as detailed in the following section.

5.1 Performance-Constrained Diversity for Preference Exploration

PB² addresses the preference exploration challenge through a population-based approach with two key mechanisms: (1) a reference policy that tracks achievable performance under current preferences, and (2) diverse policies that explore the preference landscape while remaining aligned with human intent.

Performance-Constrained Diversity. Building on the performance-constrained diversity principle introduced in SMERL [Kumar et al., 2020], we adapt this mechanism to the preference learning setting. Unlike SMERL, which requires access to an expert or optimal return value, PB² operates in settings where the reward function itself is being learned. Our approach maintains a reference policy that purely maximizes the current reward model, establishing a performance baseline. The remaining

Algorithm 1 PB²: Population-Based Preference-Based RL

```
Initialize reference policy $\pi_{\text{ref}}$, diverse policies $\{\pi_i\}_{i=2}^{N}$, discriminator $q_\psi$, reward model $r_\phi$ (Section 5)
Perform initial unsupervised exploration to collect diverse trajectory    [Lee et al., 2021b]
while feedback budget not exhausted do
    Sample trajectories from all policies (reference + diverse)    (Section 5)
    Collect human preferences on trajectory segments pairs across policies    (Section 5, Fig. 1)
    Update reward model $r_\phi$ to predict preferences    (Section 3)
    Update $\pi_{\text{ref}}$ to maximize $r_\phi(\tau)$ only    (Section 5.1)
    $R_{\text{ref}} = \mathbb{E}[R_\phi(\tau)]$ for trajectories from $\pi_{\text{ref}}$    (Section 5.1)
    for each diverse policy $\pi_i, i \geq 2$ do
        $R_i = \mathbb{E}[R_\phi(\tau)]$ for trajectories from $\pi_i$    (Section 5.1)
        if $R_i \geq \alpha \cdot R_{\text{ref}}$ then
            Update $\pi_i$ to maximize $r_\phi(\tau) + \lambda \cdot q_\psi(i)$    (Section 5.1)
        else
            Update $\pi_i$ to maximize $r_\phi(\tau)$ only    (Section 5.1)
        end if
    end for
    Train $q_\psi$ to distinguish between diverse policies    (Section 5.2, Eq. 4)
end while
```

policies are encouraged to develop distinct behaviors, but only when their performance is within a specified threshold of the reference policy:

$$\pi_i^* = \arg\max_{\pi_i} \mathbb{E}_{\tau\sim\pi_i} \left[R_\phi(\tau) + \lambda \cdot \mathbf{1}_{[R_\phi(\tau)\geq\alpha\cdot R_\phi(\tau_{\text{ref}})]} \cdot \log q_\psi(i|\tau)\right] \tag{3}$$

where $R_\phi(\tau) = \sum_{t=0}^{T} \gamma^t r_\phi(s_t, a_t)$ is the expected discounted return from the current reward model, $\mathbf{1}_{[R_\phi(\tau)\geq\alpha\cdot R_\phi(\tau_{\text{ref}})]}$ is an indicator function that equals 1 when the agent's expected return is at least α times the reference agent's return, and $q_\psi(i|\tau)$ is a learned discriminator that predicts which policy generated a given trajectory.

Adaptive Discriminator. To maintain diversity as preferences evolve, we employ a discriminator trained to maximize the mutual information between policies and their trajectories:

$$L_{\text{disc}}(\psi) = \mathbb{E}_{i\sim p(i),\tau\sim\pi_i}[\log q_\psi(i|\tau)] \tag{4}$$

Unlike approaches with fixed reward objectives, our discriminator adapts continuously to the changing landscape of behaviors. This adaptation is crucial when human preferences shift, as it encourages policies to discover new distinguishable behaviors that remain aligned with current preferences. When the reward model r_ϕ is updated with new preferences, we temporarily disable the diversity bonus, allowing policies to first adapt to the new reward landscape before reintroducing diversity incentives. This prevents diversity from interfering with the initial adaptation to updated preferences. Figure 2 illustrates how this mechanism results in each agent receiving distinct exploration bonuses, creating a diverse set of behaviors that collectively span the preference space.

Implementation details, including network architectures, hyperparameter values, and training procedures, are provided in appendix.

6 Experiments

We evaluate PB² across diverse environments to demonstrate its effectiveness in preference space exploration and learning from limited human feedback.

Environments and Baselines. We use two environment categories: (1) **Low-feedback environments** (2D Navigation and PointMaze) for testing exploration efficiency, and (2) **DMControl locomotion tasks** (cheetah_run, quadruped_walk, walker_run, walker_walk) [Tassa et al., 2018]

Table 1: Performance Comparison in 2D Navigation and PointMaze environments by Feedback Amount (N)

		Feedback Amount				
	Algorithm	**N=2***	**N=4**	**N=6**	**N=8**	**N=10**
2D Navig.	PB²	−503.7 ± 72.8	**−211.3 ± 58.7**	**−141.0 ± 48.5**	**−178.7 ± 70.7**	−126.5 ± 18.9
	QPA	−520.8 ± 67.3	−405.7 ± 164.6	−285.4 ± 160.3	−234.8 ± 154.9	**−112.1 ± 24.4**
	PEBBLE	−472.3 ± 62.0	−323.6 ± 66.0	−156.1 ± 41.2	−196.3 ± 95.9	−164.5 ± 71.1
	RUNE	**−450.2 ± 81.1**	−359.8 ± 78.5	−204.6 ± 109.6	−251.45 ± 81.5	−147.08 ± 62.6
		N=4*	**N=8**	**N=12**	**N=16**	**N=20**
PointMaze	PB²	64.1 ± 48.4	18.9 ± 3.63	**85.0 ± 45.3**	**132.2 ± 28.7**	**146.1 ± 29.3**
	QPA	35.9 ± 42.2	30.7 ± 37.0	63.6 ± 55.0	116.4 ± 18.5	110.6 ± 14.2
	PEBBLE	62.3 ± 49.2	**33.2 ± 30.5**	80.6 ± 63.1	91.4 ± 60.4	98.1 ± 52.5
	RUNE	**72.4 ± 45.6**	36.3 ± 36.3	57.9 ± 38.3	65.5 ± 38.6	90.2 ± 49.9

for evaluating distinguishability. We compare against three leading PbRL methods: *PEBBLE* [Lee et al., 2021b], which samples queries from previous policies; *RUNE* [Liang et al., 2022], which uses ensemble-based uncertainty for exploration; and *QPA* [Hu et al., 2024], which focuses on queries from recently generated trajectories. We selected these specific methods as they represent the primary query selection strategies in PbRL literature: historical sampling, uncertainty-driven exploration, and current-policy alignment. Additional methods are not included as they typically fall into these same categories without fundamentally altering the query diversity mechanism, which is the focus of our study.

Evaluation methodology. For all experiments, we use the ground truth reward (which is hidden from the learning algorithms) to evaluate performance. We report the mean and standard deviation across 5 random seeds. For simulating human evaluation inconsistency, we implement the Equal teacher from B-Pref [Lee et al., 2021a] with a similarity threshold mechanism. Specifically, for a query comparing trajectories with ground truth returns R_1 and R_2, we assign random preference labels when $|R_1 - R_2| < \epsilon \cdot \max(R_1, R_2)$, where ϵ is the similarity threshold parameter. This approach only introduces inconsistency when trajectories are genuinely similar and difficult to distinguish, better simulating human evaluation challenges. While $\epsilon = 0$ (no similarity threshold) provides a theoretical baseline, we emphasize results with $\epsilon > 0$ as more realistic, since human evaluators inevitably provide less consistent feedback when comparing similar behaviors.

Escaping Local Optima in Preference Landscapes Figure 4 illustrates a failure case for single-agent methods and how PB² overcomes it. After identical initial feedback (4 queries), both algorithms learn similar reward models favoring an upper-left region. After 20 feedback instances, QPA remains trapped in this suboptimal region due to its exploitative behavior, while PB² successfully discovers a path to the high-reward region through its diverse population. The discriminator encourages different agents to explore distinct regions while maintaining performance, allowing discovery of paths that eventually lead to optimal areas. This demonstrates a key advantage of population-based methods: when facing potentially suboptimal initial queries, PB² can escape local optima through diversity, while single-agent approaches often remain trapped. The complete progression of trajectories with increasing feedback iterations is provided in appendix, showing the step-by-step evolution of each method's exploration patterns.

Preference Exploration Improves Feedback Efficiency Table 1 presents performance with varying amounts of feedback in navigation tasks. In 2D Navigation, PB² achieves significantly better performance than QPA with limited feedback (N=4 to N=8), with improvements of up to 50% at some feedback levels. As feedback increases to N=10, the gap narrows, suggesting that with sufficient feedback, the exploration advantage becomes less critical.

In the more complex PointMaze environment, PB² consistently outperforms QPA at most feedback levels (N=12, N=16, N=20), with advantages of 20-30% in some cases. Overall, these results confirm that maintaining a diverse population enables more efficient exploration of complex preference landscapes when feedback is scarce, a crucial advantage in applications where minimizing human interaction is essential. Note that the N=2 for 2D Navigation and N=4 for PointMaze represents performance after initial unsupervised exploration and random query selection, before the distinguishing characteristics of each algorithm have meaningful impact on the learning process.

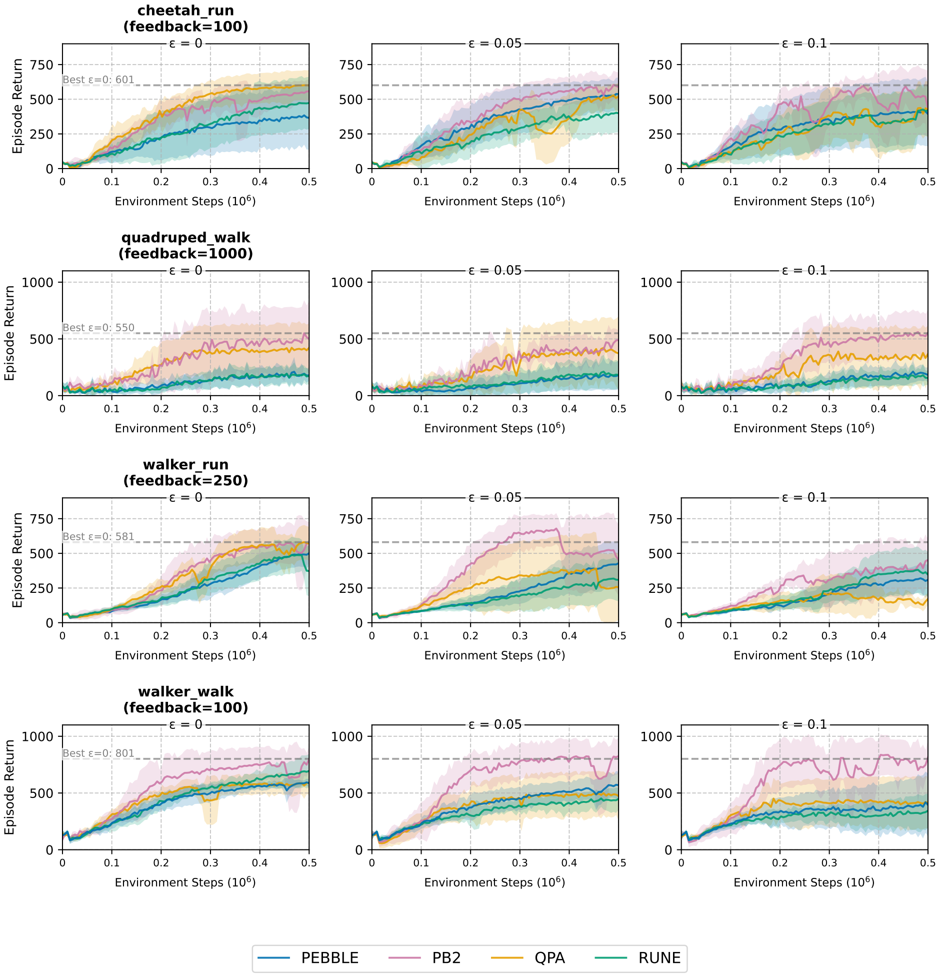

Figure 3: **Performance comparison on DMControl tasks with varying similarity thresholds.** Each column represents a different similarity threshold ϵ that determines when trajectory comparisons result in inconsistent feedback. As ϵ increases from 0 (perfect oracle) to 0.1 (significant inconsistency with similar trajectories), PB² (pink) maintains robust performance while single-agent methods degrade more substantially, particularly in the quadruped_walk and walker_walk tasks. This demonstrates PB²'s advantage in scenarios where humans must evaluate similar behaviors.

Robustness to Trajectory Distinguishability Challenges Figure 3 compares performance across DMControl tasks with three trajectory similarity thresholds ($\epsilon \in \{0, 0.05, 0.1\}$). With $\epsilon = 0$ (oracle teacher, no similarity consideration), PB² and QPA achieve comparable performance. However, as the similarity threshold increases, PB² maintains stronger performance, particularly in quadruped_walk and walker_walk. This advantage emerges because similar trajectories present a fundamental challenge for preference learning. When trajectories are nearly indistinguishable, humans may provide inconsistent feedback, potentially causing the reward model to learn incorrect preferences.

The superior performance of PB² is particularly evident in the walker_walk task with $\epsilon = 0.1$, where it achieves approximately 750 return compared to around 400 for QPA and 350 for PEBBLE. By presenting more diverse and distinguishable trajectory segments pairs for evaluation, PB² makes it easier for humans to provide consistent, reliable feedback, offering a significant advantage in real-world settings where trajectories may be similar and difficult to differentiate. Notably, we used the same diversity parameter $\lambda = 0.25$ across all DMControl environments for fair comparison, though task-specific tuning could potentially yield further improvements by better balancing the exploration-exploitation tradeoff for each environment.

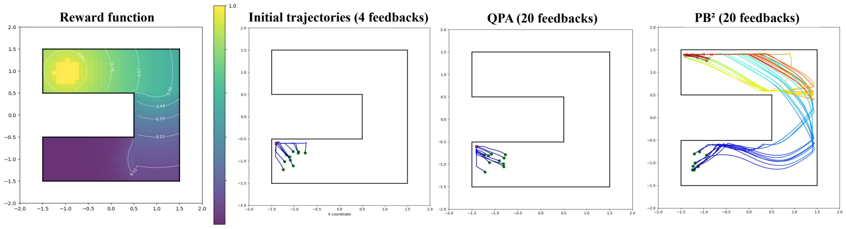

Figure 4: **Escaping local optima in preference landscapes.** After the same initial feedback following unsupervised exploration (4 queries), both algorithms learn similar reward models favoring the upper-left region. After 20 feedback instances, QPA (middle) remains trapped in this suboptimal region due to its exploitative behavior, while PB² (right) successfully discovers a path to the high-reward region (left) through its diverse population approach.

7 Limitations and Future Work

Computational Complexity PB² requires training multiple policies simultaneously, currently increasing computational demands compared to single-agent methods. While frameworks like JAX [Frey et al., 2023, Nikulin et al., 2024, Rutherford, 2022] enable efficient parallelization through just-in-time compilation and vectorized operations on GPUs/TPUs, implementing these optimizations remains as future work. Such improvements would allow population-based methods to scale with minimal overhead. Nevertheless, the computational bottleneck in preference-based RL typically remains human feedback rather than compute resources, making the additional computational cost justified by the improved robustness to human inconsistency that our method demonstrates.

Scalability with Population Size Our implementation uses a small population (3 agents) due to practical constraints in the preference learning setting. With feedback budgets typically limited to approximately 10 evaluations per iteration, larger populations would reduce the number of learning iterations possible. Future work could explore adaptive population sizing strategies that balance diversity benefits against feedback constraints in different preference landscapes.

Exploration-Exploitation Tradeoff hyperparameter PB² introduces a new hyperparameter, λ, that controls the balance between diversity and reward optimization. To ensure fair evaluation, we kept λ consistent across all environments within each class (locomotion tasks and navigation tasks), rather than tuning it per environment. Nevertheless, requiring different values for each environment classes remains a limitation, as individually tuning λ for each specific environment could potentially yield better performance. Future work could develop adaptive methods that automatically adjust this exploration-exploitation tradeoff based on the current state of preference learning, eliminating the need for domain-specific parameter tuning.

8 Conclusion

We presented PB², a population-based approach for preference-based reinforcement learning that addresses the challenge of preference space exploration. By maintaining diverse agents that collectively explore the preference landscape, PB² generates more distinguishable behaviors for human evaluation, improving reward model learning efficiency and robustness to evaluation inconsistencies.

Our experimental results demonstrate three key advantages: improved feedback efficiency with limited feedback, greater robustness to labeling inconsistency, and enhanced ability to escape local optima in complex preference landscapes. These benefits make PB² particularly well-suited for real-world applications where human feedback is costly and potentially inconsistent.

References

Mohammed Nabi Alam, Kok-Why Duong, Sepideh Hosseini, and Kathryn Kasmarik. Evolutionary reinforcement learning: A survey. *arXiv preprint arXiv:2008.12340*, 2020.

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

Erdem Biyik, Malayandi Palan, Nicholas C. Landolfi, Dylan P. Losey, and Dorsa Sadigh. Active preference-based gaussian process regression for reward learning. In *Robotics: Science and Systems*, 2020.

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.

Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in Neural Information Processing Systems*, 30:4299–4307, 2017.

Antoine Cully. Autonomous skill discovery with quality-diversity and unsupervised descriptors. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 81–89. ACM, 2019.

Valentin Dalibard and Max Jaderberg. Faster improvement rate population based training. *arXiv preprint arXiv:2109.13800*, 2021.

Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is all you need: Learning skills without a reward function. *International Conference on Learning Representations*, 2019.

Sascha Yves Frey, Kang Li, Peer Nagy, Silvia Sapora, Christopher Lu, Stefan Zohren, Jakob Foerster, and Anisoara Calinescu. Jax-lob: A gpu-accelerated limit order book simulator to unlock large scale reinforcement learning for trading. In *Proceedings of the Fourth ACM International Conference on AI in Finance*, pages 583–591, 2023.

Johannes Fürnkranz, Eyke Hüllermeier, Weiwei Cheng, and Sang-Hyeun Park. Preference-based reinforcement learning: a formal framework and a policy iteration algorithm. *Machine learning*, 89:123–156, 2012.

Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, et al. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2018.

Dylan Hadfield-Menell, Smitha Milli, Pieter Abbeel, Stuart J. Russell, and Anca D. Dragan. Inverse reward design. *Advances in Neural Information Processing Systems*, 30, 2017.

Xiao Hu, Jianxiong Li, Xianyuan Zhan, Qing-Shan Jia, and Ya-Qin Zhang. Query-policy misalignment in preference-based reinforcement learning. In *The Twelfth International Conference on Learning Representations*, 2024.

Shuaiyi Huang, Mara Levy, Anubhav Gupta, Daniel Ekpo, Ruijie Zheng, and Abhinav Shrivastava. Trend: Tri-teaching for robust preference-based reinforcement learning with demonstrations. *arXiv preprint arXiv:2505.06079*, 2025.

Max Jaderberg, Valentin Dalibard, Simon Osindero, Wojciech M. Czarnecki, Jeff Donahue, Ali Razavi, Oriol Vinyals, Tim Green, Iain Dunning, Karen Simonyan, Chrisantha Fernando, and Koray Kavukcuoglu. Population based training of neural networks. *arXiv preprint arXiv:1711.09846*, 2017.

Saurabh Kumar, Aviral Kumar, Sergey Levine, and Chelsea Finn. One solution is not all you need: Few-shot extrapolation via structured maxent rl. *Advances in Neural Information Processing Systems*, 33:8198–8210, 2020.

Kimin Lee, Laura Smith, Anca Dragan, and Pieter Abbeel. B-pref: Benchmarking preference-based reinforcement learning. *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, 2021a.

Kimin Lee, Laura M. Smith, and Pieter Abbeel. Pebble: Feedback-efficient interactive reinforcement learning via relabeling experience and unsupervised pre-training. *Proceedings of the 38th International Conference on Machine Learning*, pages 6152–6163, 2021b.

Joel Lehman. Evolution through the search for novelty. 2012.

Joel Lehman and Kenneth O Stanley. Evolving a diversity of virtual creatures through novelty search and local competition. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, pages 211–218, 2011.

Xinran Liang, Katherine Shu, Kimin Lee, and Pieter Abbeel. Reward uncertainty for exploration in preference-based reinforcement learning. In *International Conference on Learning Representations*, 2022.

Runze Liu, Fengshuo Bai, Yali Du, and Yaodong Yang. Meta-reward-net: Implicitly differentiable reward learning for preference-based reinforcement learning. *Advances in Neural Information Processing Systems*, 35:22270–22284, 2022.

Yi Liu, Gaurav Datta, Ellen Novoseller, and Daniel S Brown. Efficient preference-based reinforcement learning using learned dynamics models. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2921–2928. IEEE, 2023.

Daniel Marta, Simon Holk, Christian Pek, Jana Tumova, and Iolanda Leite. Variquery: Vae segment-based active learning for query selection in preference-based reinforcement learning. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7878–7885. IEEE, 2023.

Katherine Metcalf, Miguel Sarabia, and Barry-John Theobald. Rewards encoding environment dynamics improves preference-based reinforcement learning. *arXiv preprint arXiv:2211.06527*, 2022.

Jean-Baptiste Mouret and Jeff Clune. Illuminating search spaces by mapping elites. *arXiv preprint arXiv:1504.04909*, 2015.

Alexander Nikulin, Vladislav Kurenkov, Ilya Zisman, Artem Agarkov, Viacheslav Sinii, and Sergey Kolesnikov. Xland-minigrid: Scalable meta-reinforcement learning environments in jax. *Advances in Neural Information Processing Systems*, 37:43809–43835, 2024.

Olle Nilsson and Antoine Cully. Policy gradient assisted MAP-Elites. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 866–875. ACM, 2021.

Takayuki Osa, Voot Tangkaratt, and Masashi Sugiyama. Discovering diverse solutions in deep reinforcement learning by maximizing state–action-based mutual information. *Neural Networks*, 152:90–104, 2022.

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.

Jack Parker-Holder, Aldo Pacchiano, Krzysztof M Choromanski, and Stephen J Roberts. Effective diversity in population based reinforcement learning. *Advances in Neural Information Processing Systems*, 33:18050–18062, 2020.

Alexander Rutherford. Jaxmarl: Multi-agent rl environments in jax. *Decision-making*, page 1, 2022.

Tim Salimans, Jonathan Ho, Xi Chen, Szymon Sidor, and Ilya Sutskever. Evolution strategies as a scalable alternative to reinforcement learning. *arXiv preprint arXiv:1703.03864*, 2017.

Archit Sharma, Shixiang Gu, Sergey Levine, Vikash Kumar, and Karol Hausman. Dynamics-aware unsupervised discovery of skills. *International Conference on Learning Representations*, 2020.

Richard S Sutton, Andrew G Barto, et al. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, et al. Deepmind control suite. *arXiv preprint arXiv:1801.00690*, 2018.

Xingchen Wan, Cong Lu, Jack Parker-Holder, Philip J. Ball, Vu Nguyen, Binxin Ru, and Michael A. Osborne. Bayesian generational population-based training. In *Proceedings of the First International Conference on Automated Machine Learning*, pages 14/1–27, 2022.

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

A Implementation Details

A.1 PB² Algorithm

In this section, we provide the full procedure for PB², our population-based approach for preference-based reinforcement learning, in Algorithm 2.

A.2 Population Management

A.2.1 Population Design choice

Unlike methods such as DIAYN and SMERL that use a single policy network conditioned on a latent variable to learn multiple behaviors, PB² maintains separate policy networks for each agent in the population. This design choice offers key advantages: it allows for independent update of population members, enables different agents to fulfill distinct roles (exploration vs. exploitation) and provides a robust mechanism for recovery when individual agents fail. Having a dedicated reference agent that focuses solely on maximizing the learned reward function creates a stable anchor for the population, serving as a reliable fallback that can replace underperforming explorer agents when necessary.

A.2.2 Reference Agent Approach

One of the key innovations in PB² is our reference agent mechanism, which provides a stable performance benchmark without requiring access to an expert policy or ground-truth reward function. This section details the complete implementation of this approach.

The reference agent (indexed as agent 0 in our implementation) is trained to maximize only the current reward model without any diversity bonus:

$$\pi_0 = \arg\max_{\pi} \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{T} \gamma^t r_\phi(s_t, a_t)\right] \tag{5}$$

The reward model r_ϕ is continuously updated based on human preference feedback. After each reward model update, we track the reference agent's expected return under the new reward function:

$$R_{\text{ref}} = \mathbb{E}_{\tau\sim\pi_0}\left[\sum_{t=0}^{T} \gamma^t r_\phi(s_t, a_t)\right] \tag{6}$$

In practice, this expectation is approximated by maintaining a running average of episode returns for the reference agent over a window of recent episodes.

The other agents in the population (indexed 1 through $N-1$) are trained with the following objective:

$$\pi_i = \arg\max_{\pi} \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{T} \gamma^t r_\phi(s_t, a_t) + \lambda \cdot \mathbb{1}_{[R_\phi(\tau)\geq\alpha\cdot R_{\text{ref}}]} \cdot \log q_\psi(i|\tau)\right] \tag{7}$$

Where:

- λ is the diversity coefficient (typically 0.5)
- α is the performance threshold (approximately 0.9 in our implementation)
- $\mathbb{1}_{[R_\phi(\tau)\geq\alpha\cdot R_{\text{ref}}]}$ is an indicator function that equals 1 when the agent's expected return is at least α times the reference return
- $q_\psi(i|\tau)$ is the discriminator that predicts which policy generated a trajectory

In our implementation, we track the recent performance of each agent and apply the diversity bonus only when an agent's performance exceeds the threshold relative to the reference agent. This performance-constrained diversity mechanism ensures that exploration through diversity is only encouraged when it doesn't significantly compromise task performance.

A.2.3 Policy Inheritance in the Population

After each feedback session in PB², we transfer the critic and actor networks from the reference agent to all explorer agents in the population. This inheritance mechanism preserves the core task knowledge while still enabling agent specialization through the diversity bonuses. It serves multiple purposes:

It allows all agents to quickly benefit from updated reward models based on human feedback, ensuring the entire population adapts to new preference information simultaneously. It maintains a healthy balance between exploration and exploitation across the population, with the reference agent focusing on exploitation while explorer agents develop diverse behaviors from a common foundation. Most importantly, it prevents explorer agents from drifting too far from the task objective due to diversity pressures, which could otherwise lead to behaviors that are novel but ineffective.

This periodic knowledge sharing proved particularly beneficial in complex environments where unguided exploration can be inefficient, allowing the population to maintain both performance and diversity throughout training.

A.3 Stability Mechanisms

A.3.1 Transition Handling Between Preference Updates

A critical challenge in our population-based approach is managing the discriminator's behavior during preference transitions. When the reward model updates, we implement several specific techniques to ensure stable and effective diversity guidance:

1. **Temporary Diversity Suspension**: Immediately after a reward model update, we temporarily disable the diversity bonus by setting the discriminator coefficient to zero for a fixed period (typically 5000 environment steps). This suspension period allows all agents to first adapt to the new reward landscape before reintroducing diversity incentives, preventing potentially counterproductive exploration based on outdated preferences.
2. **On-Policy Discriminator Training**: During transition periods, the discriminator is trained exclusively on recent experiences rather than the full replay buffer. This is implemented through our on-policy sampling mechanism that selects state samples from only the most recent trajectories for each agent.
3. **Reward Normalization and Clipping**: To prevent extreme diversity bonuses during transition periods, we implement Exponential Moving Average (EMA) normalization with a decay rate of 0.99 for the discriminator's outputs. Additionally, we clip the normalized intrinsic rewards to the range [-2, 2], which prevents destabilizing spikes in the diversity bonus that could otherwise derail learning during preference transitions.

These implementation details are crucial for maintaining stable diversity during preference transitions, as they address the practical challenges of keeping a population-based system aligned with changing preference signals. By temporarily reducing diversity pressure, carefully normalizing rewards, and ensuring balanced training, we allow the discriminator to smoothly adapt to new preference landscapes without destabilizing the learning process.

B Technical Details

B.1 Implementation Framework

Our implementation builds upon established preference-based reinforcement learning frameworks. We extend the QPA codebase [Hu et al., 2024], which itself extends the B-Pref framework [Lee et al., 2021a] that provides core functionality for preference-based learning through the PEBBLE [Lee et al., 2021b] algorithm architecture.

To accommodate our population-based approach, we expanded this framework to manage multiple policy networks simultaneously, while incorporating the on-policy query selection mechanism introduced in QPA [Hu et al., 2024]. This combination allows us to leverage on-policy query selection across a diverse population of agents, rather than limiting it to a single policy.

Each agent in our population maintains its own replay buffer for experience collection and policy optimization. We introduced additional components for our discrimination-based diversity mechanism, and policy inheritance procedures.

For comparisons against baselines, we integrated implementations of competing methods including PEBBLE [Lee et al., 2021b], RUNE [Liang et al., 2022] and QPA [Hu et al., 2024] using their official codebases, ensuring fair evaluation across all approaches. Our implementation maintains compatibility with the underlying frameworks while introducing the population management and diversity mechanisms that define PB².

B.2 Network Architectures

For fair comparison, PB² uses the same network architectures as QPA for both the SAC algorithm and reward model, with our modifications focused exclusively on the population-based exploration mechanism.

The reward model consists of 3 hidden layers with 256 units each and LeakyReLU activations, taking state-action pairs as input and producing scalar reward predictions with tanh activation.

The policy networks follow the standard SAC architecture with 2 hidden layers of 1024 units and ReLU activations. The actor outputs action means and log standard deviations (bounded between -5 and 2), while the critic takes state-action pairs as input and outputs Q-value predictions.

Our discriminator network uses 2 hidden layers of 256 units with ReLU activations and layer normalization. The output dimension matches the population size, with softmax activation for classification. This discriminator is trained using cross-entropy loss on balanced batches of states from each policy in the population.

B.3 Hyperparameters

We maintain consistent hyperparameters across all comparison methods for fair evaluation, with the only differences being in the population-specific parameters introduced by PB². Tables 2-6 provide a comprehensive overview of the hyperparameters used in our experiments.

The general hyperparameters (Table 2) are common across all environments, while environment-specific parameters (Table 6) highlight the varying complexity and requirements of different tasks. DMControl tasks generally required more feedback than navigation tasks due to their higher-dimensional state and action spaces, while maintaining a consistent diversity coefficient of 0.25. Navigation tasks benefited from a higher diversity coefficient (0.5) to encourage more extensive exploration of the state space.

For the discriminator-related parameters (Table 5), we performed a limited hyperparameter search and found that a learning rate of 1e-5 and hidden size of 256 worked well across environments. The reward model (Table 4) and SAC parameters (Table 3) were selected to match those used in prior work for direct comparison.

Parameter	Description	Value
Discount factor (γ)	Reward discount factor	0.99
Replay buffer capacity	Maximum transitions stored	Training steps
Population size	Number of agents (including reference)	3
Activation function	For all networks	tanh
Gradient update frequency	Updates per environment step	1

Table 2: General hyperparameters used in the PB² algorithm

Parameter	Description	Value
Ensemble size	Number of networks in reward ensemble	1
Learning rate	Learning rate for reward model	3e-4
Number of hidden layers	Hidden layers in reward network	3
Hidden size	Units per hidden layer	256

Table 4: Reward model hyperparameters

Parameter	Description	Value
Actor learning rate	Learning rate for policy network	5e-4
Critic learning rate	Learning rate for Q-networks	5e-4
Alpha learning rate	Learning rate for temperature parameter	1e-4
Initial temperature	Initial value of entropy coefficient	0.1
Target update rate (τ)	Polyak averaging coefficient	0.005
Target update frequency	Steps between target network updates	2
Actor update frequency	Steps between policy updates	1

Table 3: SAC hyperparameters used in the PB² algorithm

Parameter	Description	Value
Batch size	States per gradient update	256
Learning rate	Learning rate for discriminator	1e-5
Hidden size	Units per hidden layer	256
On-policy ratio	Ratio of on-policy samples for training	0.5

Table 5: Discriminator hyperparameters

Environment	Feedback Budget	Total Steps	Diversity λ	Other Parameters
Cheetah_run	100	500,000	0.25	Segment length: 50
Walker_walk	100	500,000	0.25	Unsupervised steps: 9,000
Walker_run	250	500,000	0.25	Interact steps: 20,000
Quadruped_walk	1000	500,000	0.25	Queries per iteration: 10
2D Navigation	10	20,000	0.5	Segment length: 20-50
PointMaze	20	80,000	0.5	Unsupervised steps: 900-400
				Interact steps: 2,000-10,000
				Queries per iteration: 2-4

Table 6: Environment-specific hyperparameters

C Environment Details

To thoroughly evaluate our population-based approach for preference-based RL, we selected environments that specifically challenge the two key aspects of our method: efficient preference space exploration and robustness to human feedback inconsistency. Our environment selection targets two critical scenarios: (1) low-feedback settings where efficient exploration is crucial, and (2) complex environments where trajectory similarity makes human evaluation difficult and error-prone.

Navigation tasks provide intuitive visualization of exploration patterns and allow us to demonstrate how our method escapes local optima in preference landscapes with limited feedback. DMControl tasks, with their high-dimensional state-action spaces, create scenarios where trajectories can appear similar to human evaluators despite having different underlying rewards, testing our method's ability to generate distinguishable queries and handle inconsistent feedback.

While previous methods often include robotic manipulation tasks from Meta-World, these environments typically require substantially more feedback (often 2,000-3,000 queries) to achieve meaningful performance. Such high feedback requirements are unrealistic in practical human-in-the-loop scenarios. Nevertheless, we include one high-feedback experiment (1,000 queries) on the Quadruped_walk task to demonstrate our method's performance across the feedback spectrum.

C.1 Navigation Tasks

C.1.1 2D Navigation

The 2D navigation environment consists of a 10×10 continuous arena where the agent starts at the bottom left corner $(0, 0)$ and must navigate to a goal position at the top right corner $(10, 10)$. The state space is 2-dimensional, corresponding to the agent's (x, y) position. The action space is also

2-dimensional, where actions directly change the agent's position with values in the range $[-1, 1]$. If the agent attempts to move outside the boundaries of the arena, it is projected to the closest point inside. The reward function used for ground truth evaluation (not accessible to the agent) is the negative Euclidean distance to the goal position.

For human feedback simulation, we compare trajectory segments of length 50 timesteps. The oracle provides preferences based on the total progress made toward the goal during each segment. When the similarity threshold ϵ is applied, random labels are provided when the difference in progress between segments falls below the threshold.

C.1.2 PointMaze

In the PointMaze environment, a point mass agent navigates through a maze with walls to reach a designated goal location. The state space consists of the agent's position and velocity (4D). The action space is 2-dimensional, controlling the force applied in the x and y directions.

Instead of using the original reward function based on Euclidean distance to the goal, we replace it with a handcrafted reward function that better aligns with human preferences by guiding the agent through the maze as shown in Figure 4. This reward function provides higher values along the correct path through the maze corridors, creating a more structured reward landscape that captures the preference for following the intended route rather than attempting to move directly toward the goal (which would cause collisions with walls). This modification helps simulate realistic human preferences that incorporate domain knowledge about the maze's structure rather than simple distance metrics.

C.2 DMControl Tasks

We evaluate our method on four continuous control tasks from the DeepMind Control Suite (DMControl) [Tassa et al., 2018]: Cheetah_run, Walker_run, Walker_walk, and Quadruped_walk. These environments feature continuous state and action spaces with increasing complexity, from the 17-dimensional state space of Cheetah_run to the 78-dimensional state space of Quadruped_walk. All DMControl tasks have episode lengths of 1000 timesteps.

The ground truth rewards in these environments, which are used only for evaluation and not accessible to the learning algorithms, combine task-specific objectives (such as forward velocity above environment-specific thresholds) with control penalties. The Walker and Quadruped environments additionally reward upright posture maintenance.

D Additional Experimental Results

D.1 Component Ablation Study

To evaluate the contribution of individual components in PB², we conducted ablation studies examining the impact of on-policy sampling and policy inheritance mechanisms. Figure 5 shows results across three DMControl environments.

The results demonstrate that both on-policy sampling and inheritance contribute positively to performance. The full PB² method (On-policy , Inheritance) consistently achieves the highest performance across all environments. Removing policy inheritance (On-policy , Inheritance) leads to noticeable performance degradation, particularly in the later stages of training. This confirms the importance of knowledge sharing between agents in the population. Interestingly, removing on-policy sampling while keeping inheritance (On-policy , Inheritance) shows competitive performance in some environments, suggesting that the benefits of these components can be complementary depending on the task complexity.

D.2 Diversity Parameter Analysis

We investigated the effect of the diversity parameter λ on learning performance. Figure 6 shows results for the Walker_walk environment with different values of λ ranging from 0 to 1.

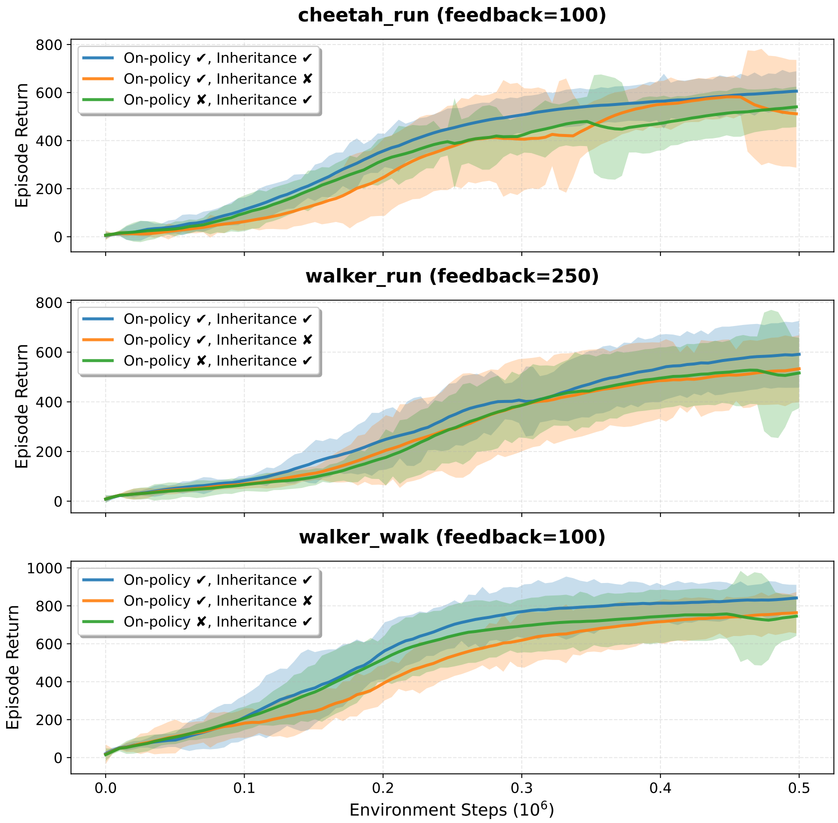

Figure 5: Ablation study on DMControl locomotion tasks showing the contribution of key components in PB². Results demonstrate that both on-policy query generation and agent inheritance mechanisms are essential for achieving optimal performance across different environments and feedback budgets.

The results reveal that moderate values of λ (0.1-0.25) achieve the best performance, with $\lambda = 0.25$ showing the strongest results. Setting $\lambda = 0$ (no diversity bonus) leads to reduced performance due to insufficient exploration and limited behavioral diversity across the population. Conversely, very high values ($\lambda = 1$) also underperform, suggesting that excessive emphasis on diversity can distract agents from optimizing the primary reward signal. The optimal range around $\lambda = 0.25$ (for the current environment) provides an effective balance between reward maximization and exploration diversity, enabling agents to discover distinct yet effective behaviors.

D.3 Population Size Sensitivity

Figure 7 examines the impact of different population sizes (2, 3, 4, 5 agents) on learning performance in the Walker_walk environment.

The results show that population sizes of 3-4 agents achieve the best performance, with diminishing returns as population size increases further. A population of size 2 shows competitive early performance but fails to maintain the same final performance level as larger populations. This is likely because with a fixed query budget of 10 per iteration, having only 2 agents results in trajectories that are too similar to each other, limiting the diversity of preference queries. Conversely, populations of size 5 do not provide significant benefits over size 4, potentially because the available queries become too sparse across agents, reducing the learning signal for each individual policy. The population size of 3, which we use in our main experiments, appears to be well-chosen based on this analysis, providing sufficient diversity while maintaining concentrated learning signals.

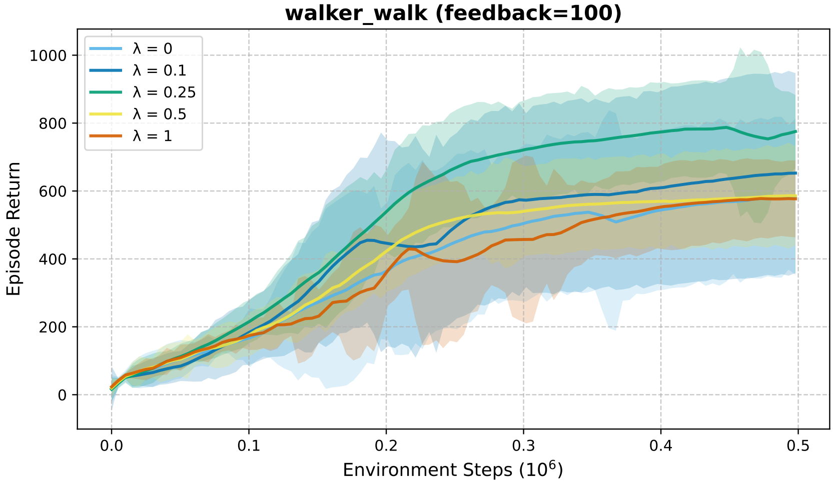

Figure 6: Sensitivity analysis of the diversity parameter λ in the walker_walk environment with 100 feedback queries. The results show that small values of λ (0.1-0.25) achieve the best balance between exploration and exploitation in this setup, while extreme values ($\lambda = 0$ or $\lambda = 1$) lead to suboptimal performance.

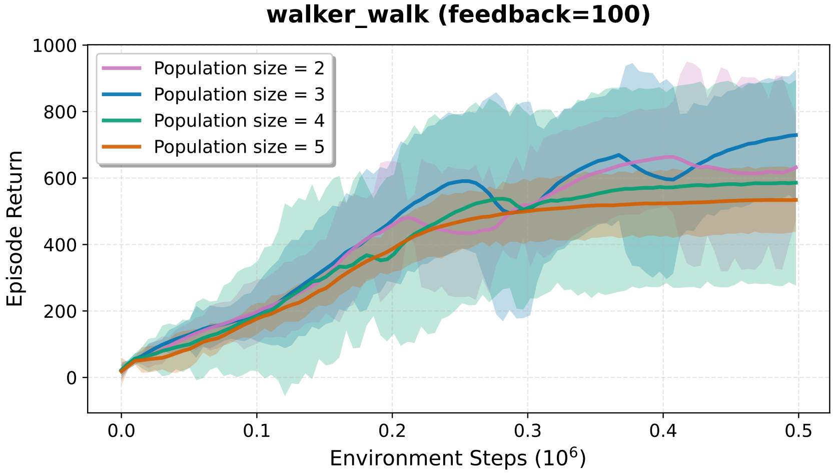

Figure 7: Impact of population size on learning performance in the walker_walk environment with 100 feedback queries. Results indicate that a population size of 3-4 agents provides the optimal trade-off between diversity benefits and computational efficiency, with diminishing returns for larger populations.

D.4 Detailed Trajectory Evolution in Point Maze Environment

This section provides the complete trajectory progression referenced in Figure 4 of the main text, showing the step-by-step evolution of exploration patterns as feedback increases from N=4 to N=16 queries in the Point Maze environment.

Figure 8 illustrates how PB² (red box) and QPA (blue box) evolve their exploration strategies with increasing feedback. At N=4, both methods show similar initial exploration patterns after receiving identical feedback. However, as feedback increases, the population-based approach in PB² enables the three agents to explore distinct regions of the maze, attempting to find high reward regions. In contrast, QPA's single-agent approach becomes increasingly concentrated in the initially promising but suboptimal upper-left region, demonstrating the local optima problem discussed in the main text.

Crucially, while not all agents in PB² necessarily discover the optimal path simultaneously, once one agent finds a better trajectory (Agent 3, N=12), the reward model update incorporates this improved knowledge by comparing it against the previous suboptimal behaviors from other agents. This allows the shared reward model to capture the superior strategy, subsequently guiding all agents toward the newly discovered high reward region (N=16, Agents 1,2 and 3). This progression clearly shows how PB²'s diverse population prevents premature convergence and enables discovery of multiple pathways that eventually lead to finding the optimal solution, while QPA remains trapped in its initial exploration pattern.

E Computational Resources and Reproducibility

E.1 Compute Resources

All experiments were conducted on NVIDIA V100. Training times varied significantly across environments and methods due to the population-based nature of our approach.

Training Times Navigation tasks (2D Navigation, PointMaze) required approximately 5 minutes per seed for baseline methods and 20 minutes for PB² due to population management overhead. DMControl tasks took 30 minutes to 2 hours for baselines depending on task complexity and feedback budget, while PB² required 7-21 hours for the same tasks. The computational overhead of PB² scales approximately 3-4× compared to single-agent baselines, primarily due to maintaining and training multiple agents simultaneously. While our current implementation trains agents sequentially, this overhead could be significantly reduced through parallel training in future work.

E.2 Code and Data Access

Code Repository Our implementation will be made publicly available upon paper acceptance. We provide the complete codebase in the supplementary materials as a zip archive for immediate access. The codebase includes all experimental configurations, training scripts, and evaluation utilities necessary for reproduction.

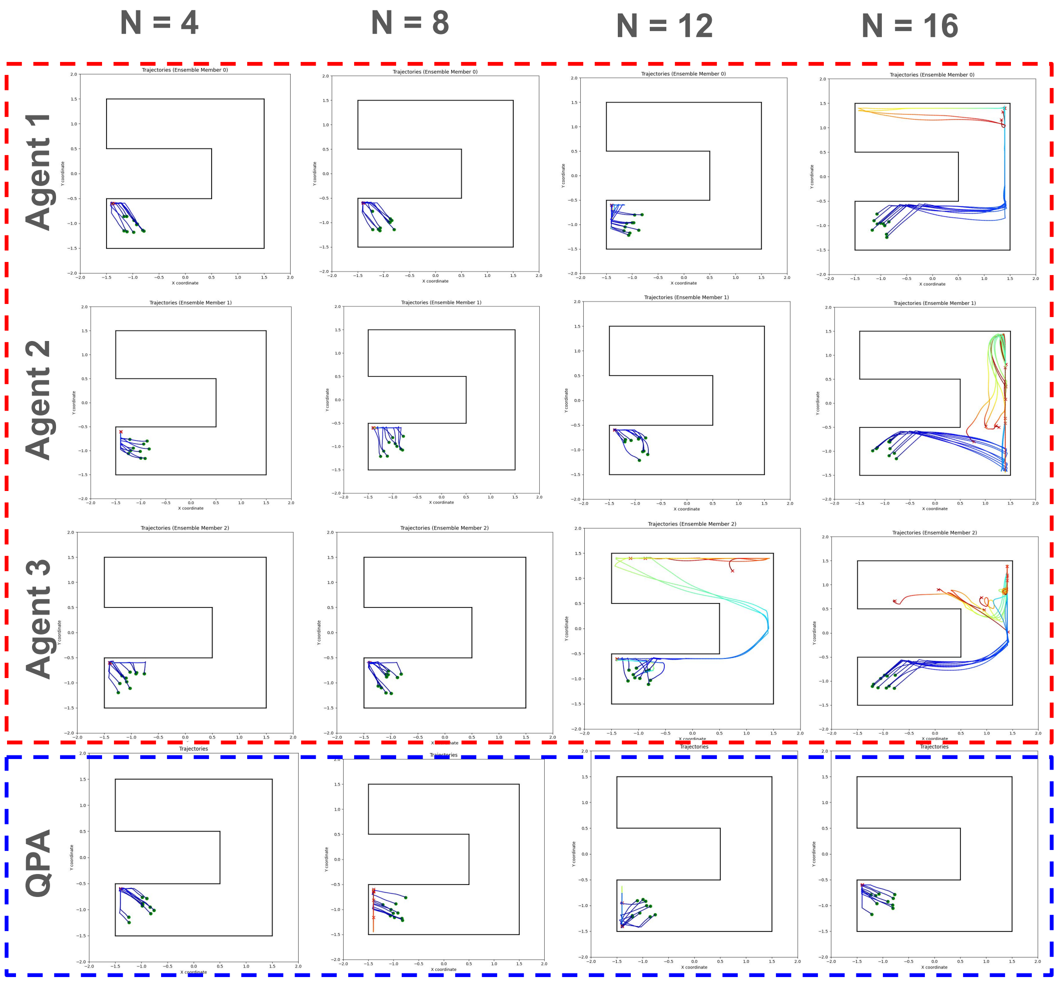

Figure 8: Complete trajectory evolution in Point Maze environment showing exploration patterns of PB² (red box, with Agent 1, 2, 3) versus QPA (blue box) as feedback increases from N=4 to N=16 queries. The progression demonstrates how PB²'s population-based approach maintains diverse exploration strategies across multiple agents, while QPA's single-agent method becomes trapped in suboptimal regions due to its exploitative nature.

Algorithm 2 PB²: Population-Based Preference-Based Reinforcement Learning (Detailed)

```
**Initialize:** Reference policy $\pi_{\text{ref}}$, diverse policies $\{\pi_i\}_{i=2}^N$, discriminator $q_\psi$, reward model $r_\phi$
Initialize replay buffers $\{B_i\}_{i=1}^N \leftarrow \emptyset$ for each policy
Initialize preference dataset $D \leftarrow \emptyset$
for each unsupervised pre-training step $t$ do
    for each policy $\pi_i$ in population do
        Collect $s_{t+1}^i$ by taking $a_t^i \sim \pi_i(a_t|s_t^i)$
        Compute state entropy reward $r_{\text{int}}^i \leftarrow -\log p(s_{t+1}^i)$
        Store transitions $B_i \leftarrow B_i \cup (s_t^i, a_t^i, s_{t+1}^i, r_{\text{int}}^i)$
    end for
    for each gradient step do
        for each policy $\pi_i$ in population do
            Sample minibatch $(s_j^i, a_j^i, s_{j+1}^i, r_{\text{int},j}^i)_{j=1}^B \sim B_i$
            Optimize policy $\pi_i$ using SAC with intrinsic reward
        end for
    end for
end for
while feedback budget not exhausted do
    // Experience Collection Phase
    for each environment step do
        for each policy $\pi_i$ in population do
            Collect $s_{t+1}^i$ by taking $a_t^i \sim \pi_i(a_t|s_t^i)$
            Compute reward $\hat{r}_t^i = r_\phi(s_t^i, a_t^i)$
            Compute discriminator reward $r_{\text{disc}}^i = \log q_\psi(i|s_t^i) - \log p(i)$
            Store transitions $B_i \leftarrow B_i \cup (s_t^i, a_t^i, s_{t+1}^i, \hat{r}_t^i, r_{\text{disc}}^i)$
        end for
    end for
    // Feedback Collection Phase
    if step to query preferences then
        Sample $K$ recent trajectories from each policy's replay buffer $\{B_i\}_{i=1}^N$
        Randomly select trajectory segments to form candidate query set $\mathcal{Q} = \{(\sigma_0, \sigma_1)\}$
        Collect human feedback $\{y_i\}$ for queries in $\mathcal{Q}$
        Store preferences $D \leftarrow D \cup \{(\sigma_0^i, \sigma_1^i, y_i)\}$
        // Reward Model Update
        Update reward model $r_\phi$ using dataset $D$ by minimizing loss in Equation (2)
        Relabel replay buffers $\{B_i\}_{i=1}^N$ using updated $r_\phi$
    end if
    // Policy Optimization Phase
    // Reference Agent Update
    Calculate reference performance $R_{\text{ref}} = \mathbb{E}_{\tau \sim \pi_{\text{ref}}}[R_\phi(\tau)]$
    Update reference policy $\pi_{\text{ref}}$ to maximize $\mathbb{E}[r_\phi(s, a)]$ using SAC
    // Diverse Agents Update
    for each diverse policy $\pi_i, i \geq 2$ do
        Calculate agent performance $R_i = \mathbb{E}_{\tau \sim \pi_i}[R_\phi(\tau)]$
        if $R_i \geq \alpha \cdot R_{\text{ref}}$ then
            Update $\pi_i$ to maximize $\mathbb{E}[r_\phi(s, a) + \lambda \cdot r_{\text{disc}}^i(s)]$ using SAC
        else
            Update $\pi_i$ to maximize $\mathbb{E}[r_\phi(s, a)]$ using SAC (no diversity bonus)
        end if
    end for
    // Discriminator Update
    Collect state samples $\{s_i^j\}$ from each policy $\pi_i$
    Update discriminator $q_\psi$ by maximizing $\mathbb{E}[\log q_\psi(i|s_i^j)]$
end while
```