Limited Preference Aided Imitation Learning from Imperfect Demonstrations

Xingchen Cao[1] Fan-Ming Luo[1] Junyin Ye[1] Tian Xu[1] Zhilong Zhang[1] Yang Yu[1]

Abstract

Imitation learning mimics high-quality policies from expert data for sequential decision-making tasks. However, its efficacy is hindered in scenarios where optimal demonstrations are unavailable, and only imperfect demonstrations are present. To address this issue, introducing additional limited human preferences is a suitable approach as it can be obtained in a human-friendly manner, offering a promising way to learn the policy that exceeds the performance of imperfect demonstrations. In this paper, we propose a novel imitation learning (IL) algorithm, **P**reference **A**ided **I**mitation **L**earning from imperfect demonstrations (PAIL). Specifically, PAIL learns a preference reward by querying experts for limited preferences from imperfect demonstrations. This serves two purposes during training: 1) Reweighting imperfect demonstrations with the preference reward for higher quality. 2) Selecting explored trajectories with high cumulative preference rewards to augment imperfect demonstrations. The dataset with continuously improving quality empowers the performance of PAIL to transcend the initial demonstrations. Comprehensive empirical results across a synthetic task and two locomotion benchmarks show that PAIL surpasses baselines by **73.2%** and breaks through the performance bottleneck of imperfect demonstrations.

1. Introduction

Imitation learning (IL) (Osa et al., 2018; Liu et al., 2023) eliminates the need for manually crafting sophisticated reward functions by utilizing a handful of expert demonstrations, offering a significant advantage in a variety of real-world applications (Peng et al., 2020; Zhao et al., 2023). Nevertheless, the applicability of IL is primarily seen in bio-mimetic tasks or those already resolved by humans, constrained by its dependency on optimal expert demonstrations. This limitation becomes pronounced in scenarios involving novel or highly specialized tasks, where obtaining optimal demonstrations is either impractical or prohibitively expensive. Examples of such challenges include the control of robots with highly versatile morphologies (Thor & Manoonpong, 2022) and the management of Tokamak fusion devices (Degrave et al., 2022). In these instances, reliance on suboptimal demonstrations is often inevitable.

Traditional methods of learning from imperfections typically involve weighting the demonstrations with a discriminator, aiming to assign higher weights to better demonstrations (Wang et al., 2021a;b). The policy is then trained from these reweighted demonstrations using weighted variants of IL. Nevertheless, the policies learned by these methods are constrained by the quality of the dataset, as they only imitate trajectories from imperfect demonstrations, preventing them from achieving better performance beyond the demonstrations. To overcome these limitations, Wu et al. (2019) and Brown et al. (2019a) proposed incorporating additional information, such as rankings and confidence scores of the demonstrations, enabling policies to reach or even surpass the performance of the best demonstrations. Limited data with scores over diverse tasks (Zhou et al., 2024) and an extra misspecified simulator (Jiang et al., 2020) are also considered helpful for policy learning. However, generating such information demands significant human effort or poses difficulties in providing precise and consistent confidence scores (Sasaki & Yamashina, 2021). In contrast, preference-based data, which only requires humans to make relative judgments, emerges as a more cost-effective and human-friendly approach and has demonstrated strong applicability in real-world tasks (Lee et al., 2021a; OpenAI, 2023), offering a promising way to learn the policy that surpass the performance of imperfect demonstrations.

In this paper, we propose to imitate policies from an unlabeled mixture of demonstrations with various qualities while leveraging limited preference queries to learn policies outperforming demonstrations with minimal human effort. Our method, termed **P**reference **A**ided **I**mitation **L**earning from imperfect demonstrations (PAIL), begins by utilizing the Bradley-Terry model (Bradley & Terry, 1952) to learn a

[1] National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, Jiangsu, China; School of Artificial Intelligence, Nanjing University, China; Polixir Technologies, Nanjing, Jiangsu, China. Correspondence to: Yang Yu <yuy@nju.edu.cn>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

preference reward model from suboptimal demonstrations supplemented by preference queries. As the potentially poor generalization of the preference reward learned with few queries might mislead the policy (Azar et al., 2023; Hejna & Sadigh, 2023), PAIL does not directly utilize the preference reward as the RL reward function. Instead, it employs the reward to reweight the demonstrations, resulting in a dataset consistent with the preference reward. Subsequently, PAIL adopts IL to train a policy using the reweighted demonstrations. To continuously enhance policy performance and exceed the initial dataset performance, PAIL periodically selects additional samples from its past experiences, based on their freshness and the preference reward evaluations. These samples are then incorporated into the demonstration dataset, expanding the training dataset for refining the preference rewards.

We empirically evaluate PAIL against existing state-of-the-art (SOTA) methods across various benchmarks, including a synthesized grid-world task and two locomotion benchmarks. In the grid-world task, PAIL demonstrated a significant improvement in learning accurate reward functions from a minimal set of preference queries, thus also showing a significant policy performance improvement. In the locomotion benchmarks, PAIL surpassed the baselines in all tasks with limited preference queries, showing a 73.2% averaged performance improvement. When preference queries were sufficiently available, PAIL can still achieve the best performance in most tasks. Further experimental analyses underscore the contribution of each component within the PAIL framework, affirming its effectiveness in leveraging preference information for policy optimization.

2. Related Work

Imitation learning from imperfection. Imitation learning from imperfection seeks to replicate optimal policy by learning from a dataset of imperfect (suboptimal) demonstrations, which may be noisy (Zheng et al., 2014; Choi et al., 2019; Sasaki & Yamashina, 2021; Li et al., 2024), supplemented with additional unlabeled suboptimal demonstrations (Zolna et al., 2020; Valko et al., 2012; Shiarlis et al., 2016; Yang et al., 2023), or unlabeled mixture of data with various quality (Wang et al., 2021a; Sasaki & Yamashina, 2021) considered in this paper. Existing methods typically require weighting the demonstrations with a discriminator, aiming to emphasize and discard the high-quality and low-quality demonstrations, respectively (Wang et al., 2021a;b). The policy is then trained from these reweighted demonstrations using weighted variants of Behavioral Cloning (BC) (Pomerleau, 1991) or Inverse Reinforcement Learning (IRL) (Ziebart et al., 2010). However, the quality of the demonstrations inherently limits the performance of policies, hindering the achievement of optimal performance. To address the limitations imposed by data constraints, Wu et al. (2019) and Brown et al. (2019a) introduced methods that augment the learning process with additional information, such as confidence scores and rankings of demonstrations. This enhancement enables policies not only to achieve but potentially exceed the performance of the highest-quality demonstrations. However, generating such information requires substantial human effort, as humans face challenges in providing precise and consistent confidence scores or rankings. PAIL distinguishes itself from these methods by utilizing human preferences, a more human-friendly information, as the additional information.

Imitation learning with preferences supplementary. Recent research has also explored the integration of preferences into IL, taking the advantages of Preference-based Reinforcement Learning (PBRL) (Bradley & Terry, 1952; Christiano et al., 2017; Wirth et al., 2017). A typical paradigm within PBRL first learns a preference reward by fitting the preference data and then utilizes the preference reward as the RL reward function to learn a policy (Christiano et al., 2017; Ibarz et al., 2018; Lee et al., 2021a). As a theoretical analysis of IL with preferences supplementary, Sekhari et al. (2023) studied the learning regret within the IL framework when actively soliciting preferences. On the algorithmic front, Brown et al. (2019b) and Chen et al. (2020) employed BC and AIL, respectively, for policy pre-training to accumulate preference data. Zhang et al. (2021) learns confidence score for state-action pairs from rankings between trajectories. Other advancements by Taranovic et al. (2023) integrated preference loss into the AIL discriminator loss, enhancing the efficiency of AIL and demonstrating superior performance compared to pure IL or PBRL. To address vague preferences information, Cai et al. (2023) explores imitation learning under conditions of less explicit guidance. PAIL differs from these methods by taking advantage of individual approaches, rather than separately applying each approach or directly loss summing. We additionally summarize the contributions of PAIL over previous methods in Appendix A.

3. Preliminaries

Markov Decision Process. An infinite-horizon Markov decision process (MDP) (Sutton & Barto, 1998) can be described by a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, r, \gamma, \rho_0 \rangle$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{P}(s' \mid s, a)$ represents the dynamics, $r : \mathcal{S} \times \mathcal{A} \to [0, 1]$ represents the oracle reward function of the environment, $\gamma \in [0, 1)$ is the discount factor and ρ_0 denotes the initial state distribution. A trajectory denoted by $\tau = \{(s_t^\tau, a_t^\tau)\}_{t=0}^{\infty}$ is a sequence of state-action pairs, where t denotes the time step. A stochastic policy $\pi(a \mid s) \in \Pi$ is an action distribution conditioned on state s, where Π is the space of policies. The quality of policy π is

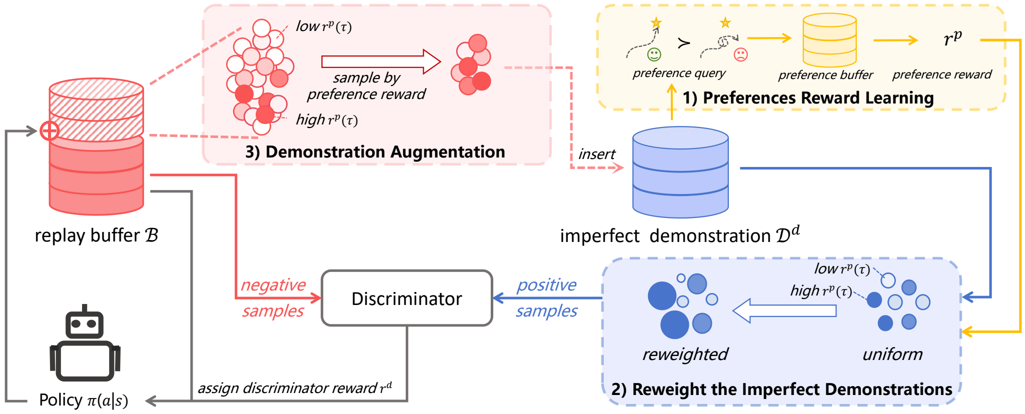
low $r^p(\tau)$
sample by preference reward
high $r^p(\tau)$
3) Demonstration Augmentation
preference query
preference buffer
preference reward
r^p
1) Preferences Reward Learning
insert
imperfect demonstration $\mathcal{D}^d$
replay buffer $\mathcal{B}$
negative samples
Discriminator
positive samples
low $r^p(\tau)$
high $r^p(\tau)$
reweighted
uniform
2) Reweight the Imperfect Demonstrations
Policy $\pi(a|s)$
assign discriminator reward r^d

Figure 1. Overview of PAIL. There are three key aspects: (1) learning the preference reward using samples from a preference buffer; (2) prioritizing demonstrations with higher preference rewards by assigning them larger weights; (3) selecting additional samples from its past experiences, based on their freshness and the preference reward evaluations.

evaluated by its policy value, i.e. discounted accumulative rewards, $V^\pi = \mathbb{E}_\pi\left[\sum_{t=0}^{\infty}\gamma^t r(s_t, a_t)\right]$, $s_0 \sim \rho_0(\cdot)$, $a_t \sim \pi(\cdot \mid s_t)$, $s_{t+1} \sim \mathcal{P}(\cdot \mid s_t, a_t)$. Reinforcement learning (RL) aims to learn a optimal policy π^* which maximize its policy value, i.e. $\pi^* = \arg\max_\pi V^\pi$.

To facilitate analysis, we define the discounted state-action distribution of policy π as $d^\pi(s,a) = (1-\gamma)\pi(a \mid s)\sum_{t=0}^{\infty}\gamma^t \Pr(s_t = s \mid \pi)$, where $\Pr(s_t = s|\pi)$ is the probability that the agent is in state s at time step t when following the policy π. The state-action distribution of trajectory τ is defined as $d^\tau(s,a) = (1-\gamma)\sum_{t=0}^{\infty}\gamma^t \mathbb{I}(s_t = s_t^\tau, a_t = a_t^\tau)$. For dataset $\mathcal{D}$ consisting of trajectories, we define the number of trajectories in $\mathcal{D}$ as $|\mathcal{D}|$ and the state-action distribution of dataset $\mathcal{D}$ as $d^\tau(s,a) = \frac{1}{|\mathcal{D}|}\sum_{\tau\in\mathcal{D}} d^\tau(s,a)$.

Inverse Reinforcement Learning. The goal of IRL (Ng & Russell, 2000) is to recover a reward function and learn a policy from the expert demonstrations. We use $\mathcal{D}^e$ to denote the expert demonstrations. The optimization problem of maximum causal entropy IRL (Ziebart et al., 2008; 2010) is defined as:

$$\begin{aligned}\max_{r^d}\min_{\pi}\mathcal{L}(\pi, r^d) = &-\left(\mathbb{E}_{(s,a)\sim\pi}\left[r^d(s,a)\right] + H(\pi)\right)\\ &+\mathbb{E}_{(s,a)\sim\mathcal{D}^e}\left[r^d(s,a)\right] - \psi(r^d),\end{aligned} \tag{1}$$

where $H(\pi) = \mathbb{E}_\pi\left[-\log\pi(a \mid s)\right]$ denotes the causal entropy of policy π (Bloem & Bambos, 2014) and $\psi(r^d)$ is a convex reward function regularizer. Garg et al. (2021) reveals that the problem (1) is equivalent to a state-action distribution matching problem

$$\min_\pi \psi^*(d^{\mathcal{D}^e} - d^\pi) - H(\pi), \tag{2}$$

where ψ^* is the convex conjugate of ψ.

Reward learning from preferences. A segment of the trajectory τ is defined as $\sigma = \{(s_t^\tau, a_t^\tau)\}_{t=k}^{k+(H-1)}$, where $k \geq 0, H \geq 1$. Preference is defined as ordering of trajectory segment pairs which can be described as a triple (σ^0, σ^1, y), where σ^0, σ^1 are trajectory segments and $y \in \{0, 1\}$ are the ordering ($y = 0$ and $y = 1$ respectively represent $\sigma^0 \prec \sigma^1$ and $\sigma^1 \prec \sigma^0$).

Common methods for reward learning from preferences aim to obtain the reward function estimate $r_\varphi^p(s,a)$ consistent with the preferences by supervised learning (Wilson et al., 2012; Christiano et al., 2017; Lee et al., 2021a). Assuming that the probability of preferring a trajectory segment depends exponentially on the sum of the reward function estimate for that segment (Bradley & Terry, 1952), a preference predictor is modeled by the reward function estimate $r_\varphi^p(s,a)$ as below:

$$P_\varphi\left[\sigma^0 \prec \sigma^1\right] = \frac{\exp\sum_{(s,a)\in\sigma^1} r_\varphi^p(s,a)}{\sum_{i\in\{0,1\}}\exp\sum_{(s,a)\in\sigma^i} r_\varphi^p(s,a)}, \tag{3}$$

Given a preference buffer $\mathcal{R}$, the optimization objective for learning a reward function r_φ^p with parameters φ, aiming to satisfy all preference orderings, is as follows:

$$\begin{aligned}\mathcal{L}^p = -\mathbb{E}_{(\sigma^0,\sigma^1,y)\sim\mathcal{R}}\Big[&\mathbb{I}(y=0)P_\varphi\left[\sigma^0 \prec \sigma^1\right]\\ &+\mathbb{I}(y=1)P_\varphi\left[\sigma^1 \prec \sigma^0\right]\Big],\end{aligned} \tag{4}$$

where $\mathbb{I}(y = a) = 1$ if $y = a$ and $\mathbb{I}(y = a) = 0$ if $y \neq a$.

Preferences are sampled from trajectories, and previous work invest significant efforts to explore efficient sampling schemes, like disagreement sampling and entropy sampling, to obtain informative preferences (Lee et al., 2021a;b).

4. Method

In this section, we propose PAIL: limited human **P**reference **A**ided **I**mitation **L**earning from imperfect demonstrations. Given imperfect demonstrations and allowed for querying human preferences when training, PAIL aims to learn good policies consistent with human preferences.

PAIL works by applying IRL to imitate from imperfect demonstrations reweighted by limited human preferences, and then augmenting the demonstrations with the trajectories newly collected for learning better policies. In Section 4.1, we show how we reweight the trajectories from the imperfect demonstrations by using human preferences. In Section 4.2, we describe the process of imitating the reweighted dataset by maximum causal entropy IRL and provide a theoretical analysis. In Section 4.3, we introduce how to augment the demonstration dataset.

4.1. Reweighting of the Imperfect Demonstrations

With the aim of obtaining a higher-quality demonstration dataset for imitation, we consider reweighting the dataset consistent with human preferences. Assume that we have imperfect demonstrations $\mathcal{D}^d = \{\tau_i^d\}_{i=1}^{|\mathcal{D}^d|}$ (τ_i denotes trajectories) and a parameterized preference reward r^p_φ consistent with human preferences.

Definition 4.1. Assume that $\mathcal{D}$ is a demonstration dataset, $\mathcal{D}_\rho$ denotes a dataset consisting of trajectories resampled from $\mathcal{D}$ by the trajectory distribution ρ, $\sum_{\tau \in \mathcal{D}} \rho(\tau) = 1$.

To identify an appropriate trajectory distribution ρ for reweighting, we try to solve the following optimization problem:

$$\max_{\rho} \mathbb{E}_{\rho(\tau)}\left[r^p_\varphi(\tau)\right] + \beta H(\rho), \ \ s.t. \sum_{\tau \in \mathcal{D}^d} \rho(\tau) = 1, \quad (5)$$

where $H(\rho) = \sum_{\tau \in \mathcal{D}^d} -\rho(\tau)\log(\rho(\tau))$ denotes the entropy of the reweighted trajectory distribution, and $r^p_\varphi(\tau) = \sum_{t=0}^{\infty} \gamma^t r^p_\varphi(s^\tau_t, a^\tau_t)$ is the accumulative preference reward (preference return) of the trajectory τ. Intuitively, the above optimization problem aims to find a distribution that can maximize the preference return of the reweighted dataset $\mathcal{D}^d_\rho$ for human preference consistency and entropy for the diversity of trajectories simultaneously. The following Theorem presents the solution of the optimization problem (5).

Theorem 4.2. *The closed-form solution to the optimization problem (5) is formulated as*

$$\tilde{\rho}_\varphi(\tau) = \left(\exp \frac{r^p_\varphi(\tau)}{\beta}\right)/Z, \quad (6)$$

where $Z = \sum_{\tau \in \mathcal{D}^d} \exp \frac{r^p_\varphi(\tau)}{\beta}$.

Also, we can easily find that $\tilde{\rho}_\varphi(\tau) \propto \exp \frac{r^p_\varphi(\tau)}{\beta}$. That is, trajectories with higher preference returns should have higher weights and a larger β implies greater diversity of the reweighted dataset.

Algorithm 1 PAIL

```
Input: Imperfect demonstrations $\mathcal{D}^d$; entropy coefficient $\beta$; augmentation entropy cofficient $\beta^{\text{aug}}$;
Parameterize policy as $\pi_\phi$, critic as $Q_\theta$, discriminator reward as $r^d_\xi$ and ensemble preference reward as $r^p_\varphi = \{r^p_{\varphi_i} \mid i = 1, \cdots, N^p\}$;
Initialize a replay bufffer $\mathcal{B}$ and a preference buffer $\mathcal{R}$;
Sample $K$ human preferences with uniform sampling from $\mathcal{D}^d$ and insert them to $\mathcal{R}$;
Update $r^p_\varphi$ with $\mathcal{R}$ by loss (4), calculate $\tilde{\rho}_\varphi$ by Eq. (6);
for $i = 1$ to $N^{\text{iter}}$ do
    Sample $n^{\text{step}}$ transitions by $\pi_\phi$ from the environment and insert them to $\mathcal{B}$;
    if $i \ \% \ N^{\text{aug}} = 0$ then
        Sample $m$ trajectories $\mathcal{D}^{\text{aug}}$ from latest $M$ trajectories by Eq. (10), $\mathcal{D}^d \leftarrow \mathcal{D}^d \bigcup \mathcal{D}^{\text{aug}}$;
        Sample $K$ human preferences with disagreement sampling from $\mathcal{D}^d$ and insert them to $\mathcal{R}$;
        Update $r^p_\varphi$ with $\mathcal{R}$ by loss (4), calculate $\tilde{\rho}_\varphi$ by Eq. (6);
    end if
    for $step\_d = 1$ to $n^{\text{disr}}$ do
        Update $r^d_\xi$ by loss $\mathbb{E}_{(s,a)\sim\mathcal{B}}\left[r^d_\xi(s,a)\right] - \mathbb{E}_{(s,a)\sim\mathcal{D}^d_{\tilde{\rho}_\varphi}}\left[r^d_\xi(s,a)\right] + \psi(r^d_\xi)$;
    end for
    for $step\_p = 1$ to $n^{\text{policy}}$ do
        Update $\pi_\phi$ and $Q_\theta$ by SAC with $\mathcal{B}$;
    end for
end for
```

4.2. Imitation from the Reweighted Demonstrations

In order to train the preference reward, we randomly sample K trajectory segments from the imperfect demonstrations D^d, query preferences, and add them to the preference buffer $\mathcal{R}$. Following the common methods for reward learning from preferences (Wilson et al., 2012; Christiano et al., 2017; Lee et al., 2021a), we parameterize the ensemble preference reward as $r^p_\varphi = \frac{1}{N^p}\sum_{i=1}^{N^p} r^p_{\varphi_i}$ and, based on Bradley-Terry model (Bradley & Terry, 1952), update the preference reward function by minimizing the loss (4). By reweighting the imperfect demonstration dataset D^d with the trajectory distribution $\tilde{\rho}_\varphi$ of Eq. (6), we obtain a reweighted dataset $D^d_{\tilde{\rho}_\varphi}$ consistent with human preferences.

With the intension to learn from trajectories perceived as good and disregard those considered unfavorable, we apply maximum causal entropy IRL (Ziebart et al., 2008; 2010) to imitate the reweighted dataset $D^d_{\tilde{\rho}_\varphi}$. By parameterizing the policy as π_ϕ, the critic as Q_θ, and the discriminator as r^d_ξ, we optimize the following objective:

$$\max_{\xi}\min_{\phi} -\left(\mathbb{E}_{(s,a)\sim\pi_\phi}\left[r^d_\xi(s,a)\right]+H(\pi_\phi)\right) + \mathbb{E}_{(s,a)\sim\mathcal{D}^d_{\tilde{\rho}_\varphi}}\left[r^d_\xi(s,a)\right]-\psi(r^d_\xi), \tag{7}$$

where $H(\pi_\phi)=\mathbb{E}_{\pi_\phi}\left[-\log\pi_\phi(a\mid s)\right]$. In practical implementation, we employ SAC (Haarnoja et al., 2018a;b) to update π_ϕ and Q_θ. Furthermore, following DAC (Kostrikov et al., 2019), we update the discriminator r^d_ξ using samples from the replay buffer of SAC, denoted as $\mathcal{B}$, for sample efficiency.

Next, we provide a theoretical analysis of imitation learning from the reweighted dataset $\mathcal{D}^d_{\tilde{\rho}_\varphi}$. The problem (7) is equivalent to the state-action distribution matching problem $\min_\phi \psi^*(d^{\mathcal{D}^d}_{\tilde{\rho}_\varphi}-d^{\pi_\phi})-H(\pi_\phi)$, where ψ^* is the convex conjugate of ψ. For simplicity, by letting '$\psi=0$ if $|r|<1/2$ and $+\infty$ otherwise', we consider the optimization problem (Xu et al., 2023)

$$\min_{\phi}\|d^{\pi}-d^{\mathcal{D}^d}_{\tilde{\rho}_\varphi}\|_1. \tag{8}$$

Assume that the trajectory $\tau^d_i\in\mathcal{D}^d$ is sampled by π^d_i, i.e. $\tau^d_i\sim\pi^d_i, i=1,\cdots,|\mathcal{D}^d|$. We define the best policy of $\mathcal{D}^d$ as $\pi^h\in\arg\max_{\pi\in\{\pi^d_1,\cdots,\pi^d_{|\mathcal{D}^d|}\}}V^\pi$. Let $\mathcal{D}^h=\{\tau\mid\tau\in\mathcal{D}^d,\tau\sim\pi^h\}$ and $\mathcal{D}^l=\mathcal{D}^d\backslash\mathcal{D}^h$.

Theorem 4.3. *If* $\|d^{\pi^h}-d^{\mathcal{D}^h}\|_1\le\epsilon_{\mathit{EST}}$*, and the learned policy* $\hat{\pi}$ *satisfies* $\|d^{\hat{\pi}}-d^{\mathcal{D}^d}_{\tilde{\rho}_\varphi}\|_1\le\min_{\pi\in\Pi}\|d^\pi-d^{\mathcal{D}^d}_{\tilde{\rho}_\varphi}\|_1+\epsilon_{\mathit{OPT}}$*, then*

$$\begin{aligned}V^{\pi^h}-V^{\hat{\pi}}\le&\underbrace{\frac{4|\mathcal{D}^l|}{1-\gamma}\exp\frac{-\left(r^p_\varphi(\tau^h_{\mathit{max}})-r^p_\varphi(\tau^l_{\mathit{max}})\right)}{\beta}}_{\textit{quality error}}\\&+\underbrace{\frac{4}{1-\gamma}\left(\exp\frac{r^p_\varphi(\tau^h_{\mathit{max}})-r^p_\varphi(\tau^h_{\mathit{min}})}{\beta}-1\right)}_{\textit{consistency error}}\\&+\frac{2\epsilon_{\mathit{EST}}+\epsilon_{\mathit{OPT}}}{1-\gamma}\end{aligned} \tag{9}$$

where $\tau^l_{\mathit{max}}=\arg\max_{\tau\in\mathcal{D}^l}\rho^p_\varphi(\tau)$, $\tau^h_{\mathit{max}}=\arg\max_{\tau\in\mathcal{D}^h}\rho^p_\varphi(\tau)$ *and* $\tau^h_{\mathit{min}}=\arg\min_{\tau\in\mathcal{D}^h}\rho^p_\varphi(\tau)$.

Theorem 4.3 analyses the value gap of the best policy π^h in $\mathcal{D}^d$ and the policy $\hat{\pi}$ learned by optimization objective (8). The quality error arises from the trajectories with low quality $\mathcal{D}^l$. If any trajectory in $\mathcal{D}^h$ has the highest preference return, i.e. $r^p_\varphi(\tau^h_{\max})>r^p_\varphi(\tau^l_{\max})$, we can mitigate the quality error by decreasing β. The consistency error stems from the inconsistent preference returns of trajectories in $\mathcal{D}^h$. A reduction in the consistency error can be achieved by either obtaining more consistent preference returns in $\mathcal{D}^h$ or by increasing the value of β. Therefore, a trade-off exists in the selection of β.

4.3. Demonstration Augmentation

In pursuit of imitating trajectories that outperform the origin demonstrations, we consider augmenting the demonstration dataset. During the policy learning process of IRL, the policy π_ϕ continuously improves and explores trajectories that may perform well from the environment. We augment the demonstration dataset by incorporating trajectories that align with human preferences among the trajectories explored from the environment.

Each N^{aug} iterations, we sample m trajectories $\mathcal{D}^{\text{aug}}$ from the latest M trajectories in replay buffer $\mathcal{B}$ using the following probability distribution:

$$P^{\text{aug}}_\varphi(\tau)=\left(\exp\tfrac{r^p_\varphi(\tau)}{\beta^{\text{aug}}}\right)/\left(\textstyle\sum_{\tau\in\mathcal{B}_{[-M:]}}\exp\tfrac{r^p_\varphi(\tau)}{\beta^{\text{aug}}}\right), \tag{10}$$

where r^p_φ is the preference reward, β^{aug} is the hyperparameter controlling entropy and $\mathcal{B}_{[-M:]}$ denotes the latest M trajectories from replay buffer $\mathcal{B}$. Smaller β^{aug} implies greater trust in the predictive capability of preference rewards r^p_φ on the newly sampled M trajectories, while conversely, larger β^{aug} indicates a higher degree of randomness in sampling. Subsequently, $\mathcal{D}^{\text{aug}}$ is added to the demonstration dataset, i.e. $\mathcal{D}^d\leftarrow\mathcal{D}^d\bigcup\mathcal{D}^{\text{aug}}$.

Since the preference buffer $\mathcal{R}$ are not sampled in the augmented trajectories $\mathcal{D}^{\text{aug}}$, the scoring for augmented trajectories by preference reward r^p_φ relies on the extrapolation ability of neural networks (Brown et al., 2019b). To enhance the accuracy on augmented data, we employ disagreement sampling (Lee et al., 2021a) to select pairs of trajectory segments with high variance across ensemble preference reward std $\left(\{r^p_{\varphi_i}\mid i=1,\cdots N^p\}\right)$ on the dataset $\mathcal{D}^d\bigcup\mathcal{D}^{\text{aug}}$, query for human preferences and refine the preference reward after each demonstration augmentation. An overview of PAIL is presented in Fig. 1, and the corresponding pseudocode is provided in Alg. 1.

5. Experiments

We perform a series of experiments aimed at answering the following questions: **Q1:** Can PAIL significantly outperform the primary baselines? (Table 1,2,3) **Q2:** How do limited preferences facilitate PAIL success? (Fig. 2,4) **Q3:** What are the effects of reweighting demonstrations, demonstration augmentation and the hyper-parameters? (Fig. 5)

5.1. Experimental Setup

Benchmark. We consider a synthetic task, i.e. GridWorld, along with 5 locomotion tasks of Mujoco benchmark (Todorov et al., 2012), and 3 locomotion tasks of DMControl (DMC) benchmark (Tassa et al., 2018; Tunyasuvunakool et al., 2020).

Imperfect demonstrations and human preferences. For

Table 1. Arrival rate and out rate (rate of leaving the boundary) in GridWorld averaged over 5 seeds. 'Demo (imp)' represents the imperfect demonstration dataset 'M' for GridWorld.

	Arrival Rate (%)	Out Rate (%)
Demo (imp)	58.40	0.00
PAIL	$\mathbf{87.24 \pm 4.15}$	$\mathbf{1.66 \pm 0.49}$
MCE-IRL	45.82 ± 2.27	2.66 ± 0.63
PEBBLE	0.39 ± 0.17	75.39 ± 12.63
BC-PEBBLE	0.49 ± 0.25	40.97 ± 20.54

the 5 Mujoco tasks, we use imperfect demonstrations of three distinct quality, denoted as 'L', 'M', and 'H'. Additionally, for GridWorld and 3 DMC tasks, we employ imperfect demonstrations of quality 'M'. For human preferences, akin to Christiano et al. (2017) and Ibarz et al. (2018), the agent queries feedback from a scripted teacher which provides preferences between trajectory segments based on the underlying task reward (oracle reward).

Primary baselines. We compare PAIL with 6 state-of-art baselines, i.e. MCE-IRL (Ziebart et al., 2008; 2010), SAIL-TRPO (Wang et al., 2021a), SAIL-SAC, AILP (Taranovic et al., 2023), PEBBLE (Lee et al., 2021a;b), BC-PBBLE (Lee et al., 2021a;b; Ibarz et al., 2018). Details on primary baselines can be seen in Appendix D.4.

5.2. Evaluation on a Synthetic Task

We first focus on the synthetic task GridWorld, enabling a thorough analysis of algorithms through visualizations. In GridWorld task, an agent tries to consistently remain within a designated circular boundary with the aim of reaching a target which is positioned randomly each episode. Deviating outside the boundary incurs a substantial penalty, emphasizing that the agent should prioritize staying within the boundary even when the target is positioned outside it. The agent's observations comprise its current position and the target position.

To answer **Q1**, we employ PAIL, MCE-IRL, PEBBLE and BC-PEBBLE to learn policies for GridWorld. We test the arrival rate (rate of reaching the target for different targets inside the boundary) and the out rate (rate of leaving the boundary) for the learned policies in Table 1. PAIL exhibits a significantly higher arrival rate compared to the other three methods, with the lowest rate of leaving the boundary. The arrival rate of MCE-IRL is slightly lower than that of the imperfect dataset 'M' by imitating the dataset 'M' directly, while PAIL achieves a significantly higher arrival rate compared to imperfect dataset 'M' with the assistance of limited preferences. PEBBLE and BC-PEBBLE, as PBRL algorithms, almost fail to reach the targets inside the boundary, with a notable rate of leaving the boundary.

To answer **Q2**, we visualize the learned policies and the discriminator rewards of PAIL and MCE-IRL. Note that we learn a state-only reward like AIRL (Fu et al., 2017) to facilitate the visualization of the reward function. From Fig. 2, we observe that in the case of 'target 2', the discriminator reward of MCE-IRL exhibits high values not only at the target position but also at another relatively conservative position. This leads to a conservative policy that does not reach 'target 2'. By reweighting the imperfect demonstration dataset with limited preferences, PAIL emphasizes demonstrations that reach the target while disregarding conservative demonstrations that do not, which enables the learning of an accurate discriminator reward. With such reward, PAIL can reach 'target 2' which is positioned close to the boundary.

Furthermore, we visualize the learned polices and the preference rewards of two PBRL methods, i.e. PEBBLE and BC-PEBBLE, in Fig. 3. With limited preferences, PEBBLE and BC-PEBBLE learn rough preference rewards which predict high rewards across a broad spectrum. Learned by such rewards, the policies of PEBBLE and BC-PEBBLE fails to accurately reach the target and leaves the boundary at times.

5.3. Performance in Mujoco and DMC tasks

To evaluate the applicability of PAIL in addressing more intricate tasks, we compare PAIL with the primary baselines in 5 locomotion tasks of Mujoco benchmark (Todorov et al., 2012) and 3 locomotion tasks of DMC benchmark (Tassa et al., 2018; Tunyasuvunakool et al., 2020). We consider 'L', 'M', 'H' demonstration dataset for Mujoco and 'M' demonstration dataset for DMC. 10 trajectories of the demonstration dataset are used for learning.

Overall performance. We use 60 preference queries for learning, and the normalized average returns of PAIL and the primary baselines are reported in Table 2. On all the tasks from the Mujoco and DMC benchmarks, PAIL exhibits a substantial performance improvement in comparison to all the baseline methods which provides a strong response to **Q1**. PAIL achieves an average performance improvement of 73.2% over prior SOTAs across all the tasks. (Respectively, 75.5% for Mujoco tasks and 61.8% for DMC tasks). Moreover, PAIL outperforms the best trajectory in the initial imperfect demonstration dataset in 16 out of 18 'Task & Dataset'.

We additionally compare PAIL with other baseline methods using preferences, i.e. PEBBLE, BC-PEBBLE and AILP, under the setting that 1400 preference queries (the most number of queries tested in Lee et al. (2021a)) are used. The normalized average returns of this setting are reported in Table 3. With much more queries for preferences, the performance of PEBBLE exhibits an improvement in 3 DMC tasks compared to the setting of 60 preference queries and

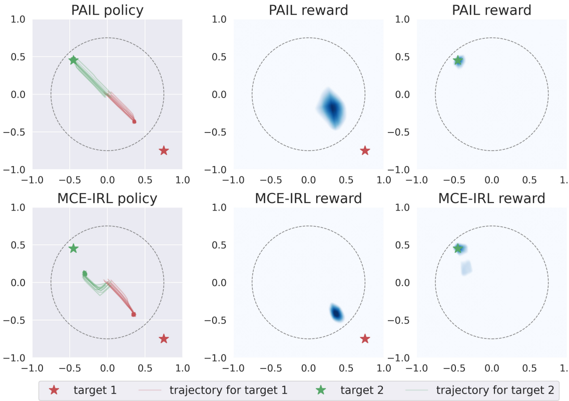

Figure 2. The learned policies and the discriminator rewards of PAIL and MCE-IRL in GridWorld.

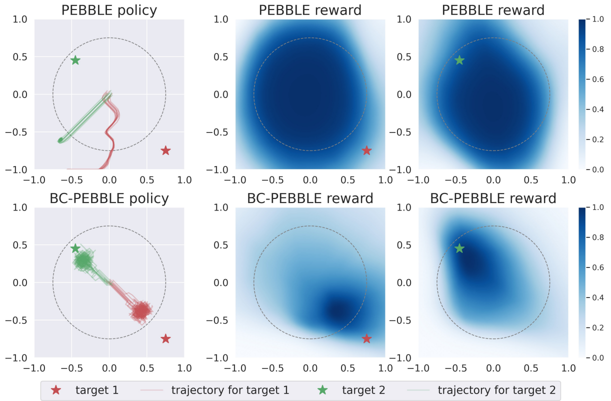

Figure 3. The learned policies and the preference rewards of PEBBLE and BC-PEBBLE in GridWorld.

Table 2. Normalized average returns in Mujoco and DMC tasks averaged over 5 seeds with 60 preference queries. 'Demo. (Avg.)' and 'Demo. (Best)' respectively represent the average and the best return within trajectories of the demonstration dataset. The underscore denotes that the average performance of PAIL's policy exceeds the performance of the best trajectory of the initial demonstration dataset.

Task & Dataset	Demo. (Avg.)	Demo. (Best)	PEBBLE	BC-PEBBLE	MCE-IRL	SAIL-TRPO	SAIL-SAC	AILP	PAIL (ours)
Ant-v2, L	0.46	0.61	0.29 ± 0.01	0.31 ± 0.00	0.37 ± 0.03	0.35 ± 0.01	0.39 ± 0.01	0.39 ± 0.04	$\underline{\mathbf{0.79}} \pm \mathbf{0.01}$
Ant-v2, M	0.63	0.77	0.29 ± 0.01	0.32 ± 0.00	0.58 ± 0.01	0.30 ± 0.01	0.55 ± 0.02	0.52 ± 0.06	$\underline{\mathbf{0.87}} \pm \mathbf{0.00}$
Ant-v2, H	0.80	0.93	0.29 ± 0.01	0.32 ± 0.00	0.19 ± 0.03	0.16 ± 0.02	0.67 ± 0.01	0.32 ± 0.00	$\underline{\mathbf{0.93}} \pm \mathbf{0.01}$
HalfCheetah-v2, L	0.20	0.34	0.13 ± 0.05	0.58 ± 0.02	0.28 ± 0.04	0.06 ± 0.00	0.18 ± 0.01	0.22 ± 0.04	$\underline{\mathbf{0.62}} \pm \mathbf{0.04}$
HalfCheetah-v2, M	0.39	0.57	0.13 ± 0.05	0.72 ± 0.03	0.38 ± 0.01	0.05 ± 0.02	0.35 ± 0.00	0.39 ± 0.04	$\underline{\mathbf{0.75}} \pm \mathbf{0.02}$
HalfCheetah-v2, H	0.61	0.81	0.13 ± 0.05	0.70 ± 0.15	0.53 ± 0.13	0.02 ± 0.02	0.48 ± 0.01	0.45 ± 0.10	$\underline{\mathbf{0.88}} \pm \mathbf{0.01}$
Hopper-v2, L	0.16	0.35	0.78 ± 0.05	0.69 ± 0.14	0.17 ± 0.03	0.25 ± 0.03	0.31 ± 0.00	0.22 ± 0.03	$\underline{\mathbf{0.93}} \pm \mathbf{0.01}$
Hopper-v2, M	0.41	0.61	0.78 ± 0.05	0.64 ± 0.11	0.45 ± 0.06	0.32 ± 0.05	0.34 ± 0.05	0.38 ± 0.12	$\underline{\mathbf{0.92}} \pm \mathbf{0.02}$
Hopper-v2, H	0.67	0.85	0.78 ± 0.05	0.77 ± 0.08	0.87 ± 0.07	0.54 ± 0.10	0.57 ± 0.09	0.47 ± 0.15	$\underline{\mathbf{0.96}} \pm \mathbf{0.01}$
Humanoid-v2, L	0.36	0.53	0.04 ± 0.00	0.04 ± 0.01	0.74 ± 0.06	0.06 ± 0.00	0.71 ± 0.06	0.05 ± 0.03	$\underline{\mathbf{0.95}} \pm \mathbf{0.02}$
Humanoid-v2, M	0.59	0.82	0.04 ± 0.00	0.03 ± 0.00	0.65 ± 0.14	0.05 ± 0.01	0.75 ± 0.06	0.05 ± 0.03	$\underline{\mathbf{0.98}} \pm \mathbf{0.02}$
Humanoid-v2, H	0.85	0.98	0.04 ± 0.00	0.03 ± 0.01	0.76 ± 0.11	0.05 ± 0.01	0.85 ± 0.05	0.24 ± 0.15	$\underline{\mathbf{1.04}} \pm \mathbf{0.02}$
Walker2d-v2, L	0.26	0.39	0.05 ± 0.01	0.09 ± 0.02	0.30 ± 0.03	0.21 ± 0.02	0.22 ± 0.03	0.28 ± 0.03	$\underline{\mathbf{0.48}} \pm \mathbf{0.02}$
Walker2d-v2, M	0.47	0.65	0.05 ± 0.01	0.05 ± 0.05	0.48 ± 0.02	0.22 ± 0.02	0.43 ± 0.01	0.39 ± 0.04	$\underline{\mathbf{0.88}} \pm \mathbf{0.01}$
Walker2d-v2, H	0.70	0.92	0.05 ± 0.01	0.06 ± 0.06	0.49 ± 0.14	0.37 ± 0.02	0.52 ± 0.05	0.46 ± 0.06	$\mathbf{0.90} \pm \mathbf{0.03}$
Average	0.5	0.68	0.26	0.36	0.48	0.2	0.49	0.32	**0.86**
cheetah_run, M	0.60	0.81	0.38 ± 0.06	0.63 ± 0.13	0.53 ± 0.04	0.12 ± 0.01	0.45 ± 0.01	0.52 ± 0.05	$\underline{\mathbf{0.87}} \pm \mathbf{0.00}$
quadruped_walk, M	0.70	0.91	0.13 ± 0.02	0.13 ± 0.06	0.25 ± 0.03	0.08 ± 0.01	0.09 ± 0.01	0.63 ± 0.03	$\mathbf{0.83} \pm \mathbf{0.04}$
walker_walk, M	0.68	0.93	0.21 ± 0.03	0.05 ± 0.01	0.45 ± 0.13	0.16 ± 0.04	0.61 ± 0.03	0.50 ± 0.09	$\underline{\mathbf{0.96}} \pm \mathbf{0.01}$
Average	0.66	0.88	0.24	0.27	0.41	0.12	0.38	0.55	**0.89**

BC-PEBBLE's performance shows enhancement in both *HalfCheetah-v2* and 3 DMC tasks, which indicates that PBRL requires a large amount of preferences. Under the setting of 1400 preference queries, PAIL still achieves the highest score in 16 out of 18 'Task & Dataset', although it exhibits a relatively smaller advantage on DMC tasks. In comparison, PAIL demonstrates a more pronounced advantage with fewer preference queries.

Visualization of preference returns and weights. To answer **Q2**, we visualize the preference returns and weights of 'Walker2d-v2, M' under the setting of 60 preference queries in Fig. 4. From the left figure, it can be observed that the preference reward learned with limited preferences accurately estimates the preference return. From the right figure, it can be seen that trajectories exhibiting higher oracle returns are assigned larger preference weights. In particular, trajectories with oracle returns greater than 5000 have weights greater than 0.025, while the remaining has weights close to 0. With such weights, PAIL mimics the trajectories with oracle returns over 5000 while ignoring other trajectories with lower performance. Furthermore, the augmented demonstrations outperform the initial demonstrations, allowing PAIL to break through the performance bottleneck of the imperfect demonstration dataset.

Preference reward analysis. To gain a deeper understanding of preference reward, the Pearson correlation coefficients between oracle rewards and preference rewards learned by 60 preference queries are listed in Table 4. In

Table 3. Normalized average returns in Mujoco and DMC tasks averaged over 5 seeds with 1400 preference queries.

Task & Dataset	PEBBLE	BC-PEBBLE	AILP	PAIL
Ant-v2, L	0.31 ± 0.00	0.31 ± 0.00	0.33 ± 0.01	$\mathbf{0.80 \pm 0.01}$
Ant-v2, M	0.31 ± 0.00	0.32 ± 0.00	0.38 ± 0.03	$\mathbf{0.88 \pm 0.01}$
Ant-v2, H	0.31 ± 0.00	0.32 ± 0.00	0.39 ± 0.08	$\mathbf{0.94 \pm 0.00}$
HalfCheetah-v2, L	0.55 ± 0.08	$\mathbf{0.75 \pm 0.03}$	0.30 ± 0.01	0.67 ± 0.03
HalfCheetah-v2, M	0.55 ± 0.08	$\mathbf{0.79 \pm 0.02}$	0.43 ± 0.05	0.71 ± 0.04
HalfCheetah-v2, H	0.55 ± 0.08	$\mathbf{0.89 \pm 0.01}$	0.44 ± 0.10	$\mathbf{0.89 \pm 0.01}$
Hopper-v2, L	0.29 ± 0.02	0.22 ± 0.05	0.29 ± 0.07	$\mathbf{0.91 \pm 0.02}$
Hopper-v2, M	0.29 ± 0.02	0.26 ± 0.02	0.27 ± 0.03	$\mathbf{0.94 \pm 0.07}$
Hopper-v2, H	0.29 ± 0.02	0.28 ± 0.05	0.37 ± 0.12	$\mathbf{0.94 \pm 0.01}$
Humanoid-v2, L	0.05 ± 0.00	0.04 ± 0.01	0.03 ± 0.02	$\mathbf{0.97 \pm 0.01}$
Humanoid-v2, M	0.05 ± 0.00	0.06 ± 0.01	0.01 ± 0.01	$\mathbf{1.01 \pm 0.01}$
Humanoid-v2, H	0.05 ± 0.00	0.06 ± 0.01	0.02 ± 0.02	$\mathbf{1.08 \pm 0.01}$
Walker2d-v2, L	0.08 ± 0.02	0.08 ± 0.03	0.26 ± 0.03	$\mathbf{0.56 \pm 0.12}$
Walker2d-v2, M	0.08 ± 0.02	0.05 ± 0.02	0.32 ± 0.06	$\mathbf{0.80 \pm 0.02}$
Walker2d-v2, H	0.08 ± 0.02	0.10 ± 0.02	0.38 ± 0.06	$\mathbf{0.74 \pm 0.16}$
Average	0.26	0.3	0.28	**0.86**
cheetah_run, M	0.64 ± 0.13	$\mathbf{0.86 \pm 0.15}$	0.37 ± 0.08	$\mathbf{0.86 \pm 0.00}$
quadruped_walk, M	0.48 ± 0.08	0.64 ± 0.07	0.67 ± 0.04	$\mathbf{0.90 \pm 0.01}$
walker_walk, M	$\mathbf{0.96 \pm 0.00}$	$\mathbf{0.96 \pm 0.02}$	0.51 ± 0.12	$\mathbf{0.96 \pm 0.00}$
Average	0.69	0.82	0.52	**0.91**

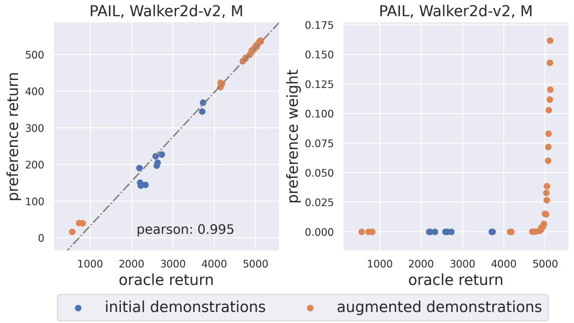

Figure 4. Visualization of preference returns and weights with 60 preference queries.

both 'Task & Dataset', preference reward of PAIL and oracle reward within demonstrations are highly correlated (Pearson correlation ≥ 0.8), which enables PAIL to reweight the demonstrations consistent with preferences. The Pearson correlation coefficient between oracle rewards and preference rewards for PEBBLE and BC-PEBBLE within their respective replay buffers is less than 0.7, which implies that with limited preferences, the learned preference reward struggles to generalize within all samples of the entire training process and misleads the RL process. Furthermore, we observe a significant decrease in the correlation between PAIL's preference reward and oracle reward when the evaluation dataset transitions from demonstrations to two replay buffers, which suggests that the extrapolation ability of PAIL's preference reward is also limited across the sample space throughout a RL process.

5.4. Ablation Studies

In this subsection, we explore the impact of demonstrations reweighting, demonstration augmentation, the entropy coefficient β in Eq. (6), the number of preference queries and the number of initial demonstrations in response to **Q3**. The experiments are conducted in 'Walker2d-v2, M' with 60 preference queries and the results are recorded in Fig. 5.

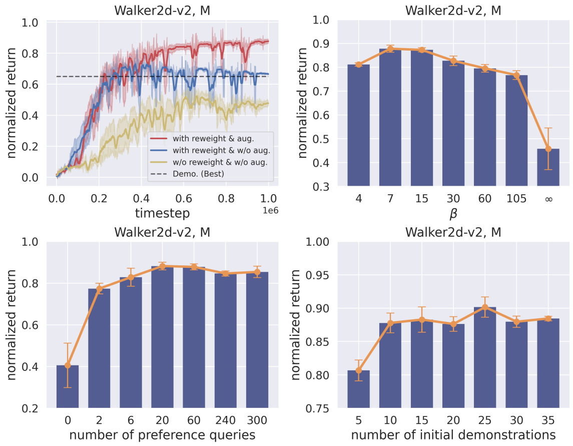

Figure 5. Ablation studies. Upper left: ablation studies on demonstration reweighting and augmentation ('w/o' denotes 'without'; 'aug.' means 'augmentation'). Others: sensitivity studies on the entropy coefficient (upper right), preference queries number (bottom left), and initial demonstrations number (bottom right).

Reweighting demonstrations & demonstration augmentation. In the upper-left figure of Fig. 5, a comparison between the yellow and blue curves reveals that through reweighting demonstrations, the policy's performance exhibits improvement and converges towards the optimal trajectory within the initial demonstrations. In comparison of the blue and red curves, the introduction of demonstration augmentation enables PAIL to break through the performance bottleneck of the initial demonstrations, resulting in a substantial improvement in its overall performance.

The entropy coefficient β. The upper-right figure of Fig. 5 indicates that performance is diminished when β is small, reaches its peak when β is set to 7 and 15, and gradually declines as β continues to increase. An excessively small beta may cause PAIL to imitate from few specific trajectories, resulting in instability and unsatisfactory performance, while an excessively large beta prevents PAIL from disregarding the poorly performing demonstrations. $\beta = \infty$ means not reweighting the demonstration dataset, resulting in significantly lower performance compared to cases where the demonstration dataset is reweighted. Hence, a trade-off should be considered when choosing β. Moreover, PAIL is robust to changes in the hyper-parameter β.

The number of preference queries and initial demonstrations. By observing the two figures at the bottom of Fig. 5, it becomes apparent that as the number of preference queries increases, the performance of PAIL shows an upward trend, and stabilizes without significant changes once the number of preference queries reaches 20. Even with only 2 prefer-

Table 4. The Pearson correlation coefficient between oracle rewards and preference rewards with 60 preference queries. 'R.B.' denotes 'Replay Buffer'. The preference reward of PAIL is evaluated within demonstrations (including initial the augmented demonstrations), replay buffer of PEBBLE and replay buffer of BC-PEBBLE.

Preference Reward	PAIL			PEBBLE	BC-PEBBLE
Evaluation Dataset	Demonstrations	R.B. of PEBBLE	R.B. of BC-PEBBLE	R.B. of PEBBLE	R.B. of BC-PEBBLE
Walker2d-v2, M	**0.914**	0.652	0.703	0.099	0.328
cheetah_run, M	**0.864**	0.121	0.603	0.174	0.654

ence queries, PAIL is capable of achieving an improvement of almost 0.4 in normalized return compared to the case with no query. A similar pattern is observed for the number of initial demonstrations. The performance stabilizes once the number of initial demonstrations reaches 10.

6. Discussion and Future Work

In this work, we proposed PAIL, an imitation learning method aided by preferences, to learn a policy from imperfect demonstrations and limited human preferences. To utilize the limited preferences, PAIL extract a preference reward from the demonstrations and employ it to reweight the demonstrations. By applying maximum causal entropy IRL to imitate the reweighted demonstrations with demonstration augmentation, PAIL learns from a flow-to-better dataset and break through the performance bottleneck of the initial dataset. Empirical studies on a synthetic task and two locomotion benchmarks show that PAIL requires much less preferences than PBRL methods and significantly outperforms various existing methods.

In practical applications, it can be difficult for human experts to provide perfect preferences for PAIL to learn from. Regarding this issue, we will consider enabling PAIL to learn from noisy preferences in future work, for example integrating COMPILER (Cai et al., 2023). Also, the learning efficiency of PAIL also faces challenges. For a trajectory consisting of both good parts and bad parts, PAIL fails to exclusively consider those good parts. In subsequent research, we aim to explore potential solutions like mimicking the scored trajectory segments.

Acknowledgements

This work was supported by the National Science Foundation of China (61921006).

Impact Statement

This research promotes the incorporation of less-than-ideal demonstrations alongside human preferences, readily available in real-world scenarios, into the framework of imitation learning. This integration plays a significant role in enhancing the practicality of imitation learning applications. While there are numerous potential societal implications stemming from our study, none which we feel must be specifically highlighted here.

References

Azar, M. G., Rowland, M., Piot, B., Guo, D., Calandriello, D., Valko, M., and Munos, R. A general theoretical paradigm to understand learning from human preferences. *CoRR*, abs/2310.12036, 2023.

Bloem, M. and Bambos, N. Infinite time horizon maximum causal entropy inverse reinforcement learning. In *Proceedings of the 53rd IEEE Conference on Decision and Control*, pp. 4911–4916, Los Angeles, CA, 2014.

Bradley, R. A. and Terry, M. E. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

Brown, D. S., Goo, W., Nagarajan, P., and Niekum, S. Extrapolating beyond suboptimal demonstrations via inverse reinforcement learning from observations. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 783–792, Long Beach, CA, 2019a.

Brown, D. S., Goo, W., and Niekum, S. Better-than-demonstrator imitation learning via automatically-ranked demonstrations. In *Proceedings of the 3rd Annual Conference on Robot Learning*, pp. 330–359, Osaka, Japan, 2019b.

Cai, X.-Q., Zhang, Y.-J., Chiang, C.-K., and Sugiyama, M. Imitation learning from vague feedback. *Advances in Neural Information Processing Systems*, 36:48275–48292, 2023.

Chen, L., Paleja, R. R., and Gombolay, M. C. Learning from suboptimal demonstration via self-supervised reward regression. In *Proceedings of the 4th Conference on Robot Learning*, pp. 1262–1277, virtual/Cambridge, MA, 2020.

Choi, S., Lee, K., and Oh, S. Robust learning from demonstrations with mixed qualities using leveraged gaussian processes. *IEEE Transactions on Robotics*, 35(3):564–576, 2019.

Christiano, P. F., Leike, J., Brown, T., Martic, M., Legg, S., and Amodei, D. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems 30*, pp. 4299–4307, Long Beach, CA, 2017.

Degrave, J., Felici, F., Buchli, J., Neunert, M., Tracey, B. D., Carpanese, F., Ewalds, T., Hafner, R., Abdolmaleki, A., de Las Casas, D., Donner, C., Fritz, L., Galperti, C., Huber, A., Keeling, J., Tsimpoukelli, M., Kay, J., Merle, A., Moret, J., Noury, S., Pesamosca, F., Pfau, D., Sauter, O., Sommariva, C., Coda, S., Duval, B., Fasoli, A., Kohli, P., Kavukcuoglu, K., Hassabis, D., and Riedmiller, M. A. Magnetic control of tokamak plasmas through deep reinforcement learning. *Nature*, 602(7897):414–419, 2022.

Fu, J., Luo, K., and Levine, S. Learning robust rewards with adversarial inverse reinforcement learning. *CoRR*, abs/1710.11248, 2017.

Garg, D., Chakraborty, S., Cundy, C., Song, J., and Ermon, S. IQ-Learn: Inverse soft-Q learning for imitation. In *Advances in Neural Information Processing Systems 34*, pp. 4028–4039, virtual, 2021.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of the 35th International Conference on Machine Learning*, pp. 1861–1870, Stockholm, Sweden, 2018a.

Haarnoja, T., Zhou, A., Hartikainen, K., Tucker, G., Ha, S., Tan, J., Kumar, V., Zhu, H., Gupta, A., Abbeel, P., and Levine, S. Soft actor-critic algorithms and applications. *CoRR*, abs/1812.05905, 2018b.

Hejna, J. and Sadigh, D. Inverse preference learning: Preference-based RL without a reward function. *CoRR*, abs/2305.15363, 2023.

Ho, J., Gupta, J. K., and Ermon, S. Model-free imitation learning with policy optimization. In *Proceedings of the 33rd International Conference on Machine Learning*, pp. 2760–2769, New York City, NY, 2016.

Ibarz, B., Leike, J., Pohlen, T., Irving, G., Legg, S., and Amodei, D. Reward learning from human preferences and demonstrations in atari. In *Advances in Neural Information Processing Systems 31*, pp. 8022–8034, Montréal, Canada, 2018.

Jiang, S., Pang, J., and Yu, Y. Offline imitation learning with a misspecified simulator. *Advances in neural information processing systems*, 33:8510–8520, 2020.

Kostrikov, I., Agrawal, K. K., Dwibedi, D., Levine, S., and Tompson, J. Discriminator-actor-critic: Addressing sample inefficiency and reward bias in adversarial imitation learning. In *Proceedings of the 7th International Conference on Learning Representations*, New Orleans, LA, 2019.

Lee, K., Smith, L. M., and Abbeel, P. PEBBLE: feedback-efficient interactive reinforcement learning via relabeling experience and unsupervised pre-training. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 6152–6163, virtual, 2021a.

Lee, K., Smith, L. M., Dragan, A. D., and Abbeel, P. B-pref: Benchmarking preference-based reinforcement learning. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1*, virtual, 2021b.

Li, Z., Xu, T., Qin, Z., Yu, Y., and Luo, Z.-Q. Imitation learning from imperfection: Theoretical justifications and algorithms. *Advances in Neural Information Processing Systems*, 36, 2024.

Liu, X.-H., Xu, F., Zhang, X., Liu, T., Jiang, S., Chen, R., Zhang, Z., and Yu, Y. How to guide your learner: Imitation learning with active adaptive expert involvement. *arXiv preprint arXiv:2303.02073*, 2023.

Mohri, M., Rostamizadeh, A., and Talwalkar, A. *Foundations of Machine Learning*. MIT Press, 2018.

Ng, A. Y. and Russell, S. Algorithms for inverse reinforcement learning. In *Proceedings of the 17th International Conference on Machine Learning*, pp. 663–670, Stanford, CA, 2000.

OpenAI. GPT-4 technical report. *CoRR*, abs/2303.08774, 2023.

Osa, T., Pajarinen, J., Neumann, G., Bagnell, J. A., Abbeel, P., and Peters, J. An algorithmic perspective on imitation learning. *Foundations and Trends in Robotics*, 7(1-2): 1–179, 2018.

Peng, X. B., Coumans, E., Zhang, T., Lee, T. E., Tan, J., and Levine, S. Learning agile robotic locomotion skills by imitating animals. In *Proceedings of Robotics: Science and Systems XVI*, virtual / Corvalis, OR, 2020.

Pomerleau, D. A. Efficient training of artificial neural networks for autonomous navigation. *Neural computation*, 3(1):88–97, 1991.

Sasaki, F. and Yamashina, R. Behavioral cloning from noisy demonstrations. In *Proceedings of the 9th International Conference on Learning Representations*, virtual, 2021.

Sekhari, A., Sridharan, K., Sun, W., and Wu, R. Contextual bandits and imitation learning via preference-based active queries. *CoRR*, abs/2307.12926, 2023.

Shiarlis, K., Messias, J. V., and Whiteson, S. Inverse reinforcement learning from failure. In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*, pp. 1060–1068, Singapore, 2016.

Sutton, R. S. and Barto, A. G. *Reinforcement learning: An introduction*. MIT Press, 1998.

Taranovic, A., Kupcsik, A. G., Freymuth, N., and Neumann, G. Adversarial imitation learning with preferences. In *Proceedings of the 11th International Conference on Learning Representations*, Kigali, Rwanda, 2023.

Tassa, Y., Doron, Y., Muldal, A., Erez, T., Li, Y., de Las Casas, D., Budden, D., Abdolmaleki, A., Merel, J., Lefrancq, A., Lillicrap, T. P., and Riedmiller, M. A. Deepmind control suite. *CoRR*, abs/1801.00690, 2018.

Thor, M. and Manoonpong, P. Versatile modular neural locomotion control with fast learning. *Nature Machine Intelligence*, 4(2):169–179, 2022.

Todorov, E., Erez, T., and Tassa, Y. MuJoCo: A physics engine for model-based control. In *Proceedings of the 25th International Conference on Intelligent Robots and Systems*, pp. 5026–5033, Algarve, Portugal, 2012.

Tunyasuvunakool, S., Muldal, A., Doron, Y., Liu, S., Bohez, S., Merel, J., Erez, T., Lillicrap, T., Heess, N., and Tassa, Y. dm_control: Software and tasks for continuous control. *Software Impacts*, 6:100022, 2020.

Valko, M., Ghavamzadeh, M., and Lazaric, A. Semi-supervised apprenticeship learning. In *Proceedings of the 10th European Workshop on Reinforcement Learning*, pp. 131–142, Scotland, UK, 2012.

Wang, Y., Xu, C., and Du, B. Robust adversarial imitation learning via adaptively-selected demonstrations. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence*, pp. 3155–3161, virtual / Montreal, Canada, 2021a.

Wang, Y., Xu, C., Du, B., and Lee, H. Learning to weight imperfect demonstrations. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 10961–10970, virtual, 2021b.

Wilson, A., Fern, A., and Tadepalli, P. A bayesian approach for policy learning from trajectory preference queries. In *Advances in Neural Information Processing Systems 25*, pp. 1142–1150, Lake Tahoe, NV, 2012.

Wirth, C., Akrour, R., Neumann, G., and Fürnkranz, J. A survey of preference-based reinforcement learning methods. *Journal of Machine Learning Research*, 18:136:1–136:46, 2017.

Wu, Y., Charoenphakdee, N., Bao, H., Tangkaratt, V., and Sugiyama, M. Imitation learning from imperfect demonstration. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 6818–6827, Long Beach, CA, 2019.

Xu, T., Li, Z., Yu, Y., and Luo, Z.-Q. Provably efficient adversarial imitation learning with unknown transitions. In *Uncertainty in Artificial Intelligence*, pp. 2367–2378. PMLR, 2023.

Yang, H., Yu, C., Chen, S., et al. Hybrid policy optimization from imperfect demonstrations. In *Advances in Neural Information Processing Systems 36*, New Orleans, LA, 2023.

Zhang, S., Cao, Z., Sadigh, D., and Sui, Y. Confidence-aware imitation learning from demonstrations with varying optimality. In *Advances in Neural Information Processing Systems 34*, pp. 12340–12350, virtual, 2021.

Zhang, Z., Sun, Y., Ye, J., Liu, T.-S., Zhang, J., and Yu, Y. Flow to better: Offline preference-based reinforcement learning via preferred trajectory generation. In *The Twelfth International Conference on Learning Representations*, 2023.

Zhao, T. Z., Kumar, V., Levine, S., and Finn, C. Learning fine-grained bimanual manipulation with low-cost hardware. In *Robotics: Science and Systems XIX*, Daegu, Republic of Korea, 2023.

Zheng, J., Liu, S., and Ni, L. M. Robust bayesian inverse reinforcement learning with sparse behavior noise. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence*, pp. 2198–2205, Québec, Canada, 2014.

Zhou, R., Gao, C.-X., Zhang, Z., and Yu, Y. Generalizable task representation learning for offline meta-reinforcement learning with data limitations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 17132–17140, 2024.

Ziebart, B. D., Maas, A. L., Bagnell, J. A., and Dey, A. K. Maximum entropy inverse reinforcement learning. In *Proceedings of the 23rd AAAI Conference on Artificial Intelligence*, pp. 1433–1438, Chicago, IL, 2008.

Ziebart, B. D., Bagnell, J. A., and Dey, A. K. Modeling interaction via the principle of maximum causal entropy. In *Proceedings of the 27th International Conference on Machine Learning*, pp. 1255–1262, Haifa, Israel, 2010.

Zolna, K., Novikov, A., Konyushkova, K., Gülçehre, Ç., Wang, Z., Aytar, Y., Denil, M., de Freitas, N., and Reed, S. E. Offline learning from demonstrations and unlabeled experience. *CoRR*, abs/2011.13885, 2020.

A. Contributions over Existing Work

The contributions of PAIL over existing work are summarized in this section.

A.1. Compared to PBRL Methods

Compared to PEBBLE(Lee et al., 2021a;b) and BC-PEBBLE (Lee et al., 2021a;b; Ibarz et al., 2018), PAIL provides a novel preference sampling technique and a novel usage for preference reward. In particular, PAIL queries for preferences from the continuously expanding demonstration dataset, and use the preference reward to weight the imperfect demonstration trajectories for imitation. PBRL methods sample preference from the whole training replay buffer and use the preference reward to guide the policy learning process. The hypothesis space of demonstration dataset is simpler than that of the training replay buffer, so with limited preference queries, the learned preference reward by PAIL can weight the demonstration trajectories precisely and thereby improving the performance of the imitated policy. However, with limited preference queries, PBRL methods learn poor preference reward over the training replay buffer, which mislead the policy learning progress.

A.2. Compared to IL from Imperfection Methods

Compared to SAIL(Wang et al., 2021a): 1. PAIL additionally introduces human-friendly preference-based data and reweights the imperfect demonstration trajectories by preferences for better imitation performance. In comparison, SAIL reweights the state-action pairs from the demonstrations by the IRL discriminator. However, the IRL discriminator is trained to distinguish from the demonstrations and the state-action pairs sampled by the learning agent, and to our best knowledge there is no evidence that the discriminator can reweights the state-action pairs from the demonstrations precisely, which leads methods like SAIL doesn't perform that well. 2. PAIL novelly augments the demonstration dataset. The performance of IL from imperfection methods is limited by the demonstrations, while PAIL outperforms the demonstrations by demonstration augmentation.

A.3. Compared to IL with preferences supplementary methods

Compared to AILP(Taranovic et al., 2023), PAIL addtionally trains a preference reward which is used to weight the imperfect demonstration trajectories and to choose which trajectory to augment in the process of AIL, while AILP integrates preference loss directly into the AIL discriminator loss.

Compared to CAIL(Zhang et al., 2021): In the aspect of problem setting, there are two main differences:

1. The extra information introduced is different. For PAIL, the agent is allowed to query a small amount of preferences during the training process. A preference is the ordering of a trajectory segment pair. In the setting of CAIL, the agent is provided with an extra evaluation trajectory dataset which is fully ranked. Certainly, CAIL can also be extended to handle the input of trajectory segment preferences.

2. The assumptions for demonstrations are different. For PAIL, the demonstration dataset does not necessarily contain optimal demonstrations. In the setting of CAIL, the demonstration dataset must include optimal demonstrations and may contain non-expert demonstrations. CAIL cannot handle the situation when all demonstrations are suboptimal.

In the aspect of algorithms, there are three main differences:

1. PAIL learns a preference reward by utilizing the Bradley-Terry model and calculate the weight by solving the optimization problem 5. CAIL directly solves the optimization problem to maximize the IL performance.

2. PAIL reweights the demonstration trajectories, while CAIL reweights the state-action pairs of demonstrations. Experimental results in Section 4.2 of Zhang et al. (2023) reveal that scoring state-action pairs is much more harder than scoring trajectories. The inaccurate weights of state-action pairs by CAIL make it difficult to improve the performance of imitation learning. However, for PAIL, even for two trajectories that are good in some parts and bad in others, PAIL may predict the according weights and mimics both trajectories simultaneously. Moreover, PAIL has the potential to explore trajectories consisting of the good parts from the two trajectories during the learning process, and incorporate them into the demonstration dataset through demonstration augmentation.

3. Moreover, PAIL novelly augment the demonstration dataset in the training process. When demonstrations are all suboptimal, demonstration augmentation enables PAIL to outperform the best demonstration. However, the performance of

CAIL is limited by the demonstration dataset in this case.

B. Proof

B.1. Proof of Theorem 4.2

Proof. Recall the optimization problem (5):

$$\begin{aligned} &\max_{\rho} \ \mathbb{E}_{\rho(\tau)} \left[r_{\varphi}^{p}(\tau) \right] - \beta \sum_{\tau \in \mathcal{D}^d} \rho(\tau) \log \rho(\tau), \\ &s.t. \ \sum_{\tau \in \mathcal{D}^d} \rho(\tau) = 1. \end{aligned} \tag{11}$$

The Lagrangian function of the problem follows by

$$\mathcal{L}(\rho, \lambda) = \sum_{\tau \in \mathcal{D}^d} \left[\rho(\tau) r_{\varphi}^{p}(\tau) - \beta \cdot \rho(\tau) \log \rho(\tau) \right] - \lambda \left[\sum_{\tau \in \mathcal{D}^d} \rho(\tau) - 1 \right]. \tag{12}$$

The partial derivative of L with respect to $\rho(\tau)$ is

$$\frac{\partial \mathcal{L}(\rho, \lambda)}{\partial \rho(\tau)} = r_{\varphi}^{p}(\tau) - \beta \cdot \log \rho(\tau) - \beta - \lambda. \tag{13}$$

Setting the partial derivative to zero yields that the closed-form solution $\tilde{\rho}_{\varphi}(\tau)$ satisfies

$$\tilde{\rho}_{\varphi}(\tau) = \exp\left(\frac{r_{\varphi}^{p}(\tau)}{\beta} \right) / \exp\left(\frac{\lambda + \beta}{\beta} \right). \tag{14}$$

Substitute the condition $\sum_{\tau \in \mathcal{D}^d} \tilde{\rho}_{\varphi}(\tau) = 1$, we obtain that $\exp\left((\lambda + \beta)/\beta\right) = \sum_{\tau \in \mathcal{D}^d} \exp\left(r_{\varphi}^{p}(\tau)/\beta \right)$. Finally, the result is as follows

$$\tilde{\rho}_{\varphi}(\tau) = \left(\exp \frac{r_{\varphi}^{p}(\tau)}{\beta} \right) / \left(\sum_{\tau \in \mathcal{D}^d} \exp \frac{r_{\varphi}^{p}(\tau)}{\beta} \right). \tag{15}$$

□

B.2. Proof of Theorem 4.3

Proof. For occupancy measure of the reweighted demonstration dataset $d_{\tilde{\rho}_{\varphi}}^{\mathcal{D}^d}$, we have

$$\begin{aligned} d_{\tilde{\rho}_{\varphi}}^{\mathcal{D}^d}(s,a) &= \sum_{\tau \in \mathcal{D}^d} \tilde{\rho}_{\varphi}(\tau) d^{\tau}(s,a) && (16) \\ &= \sum_{\tau \in \mathcal{D}^l} \tilde{\rho}_{\varphi}(\tau) d^{\tau}(s,a) + \sum_{\tau \in \mathcal{D}^h} \tilde{\rho}_{\varphi}(\tau) d^{\tau}(s,a) && (17) \\ &= \sum_{\tau \in \mathcal{D}^l} \left(\exp \frac{r_{\varphi}^{p}(\tau)}{\beta} \right) / Z \cdot d^{\tau}(s,a) + \sum_{\tau \in \mathcal{D}^h} \left(\exp \frac{r_{\varphi}^{p}(\tau)}{\beta} \right) / Z \cdot d^{\tau}(s,a), && (18) \end{aligned}$$

where $Z = \sum_{\tau \in \mathcal{D}^d} \exp \frac{r_{\varphi}^{p}(\tau)}{\beta}$. By defining that $\mathcal{Z}^l = \sum_{\tau \in \mathcal{D}^l} \exp \frac{r_{\varphi}^{p}(\tau)}{\beta}$ and $\mathcal{Z}^h = \sum_{\tau \in \mathcal{D}^h} \exp \frac{r_{\varphi}^{p}(\tau)}{\beta}$, we have

$$\begin{aligned} d_{\tilde{\rho}_{\varphi}}^{\mathcal{D}^d}(s,a) &= \sum_{\tau \in \mathcal{D}^l} \left(\exp \frac{r_{\varphi}^{p}(\tau)}{\beta} \right) / Z \cdot d^{\tau}(s,a) + \sum_{\tau \in \mathcal{D}^h} \left(\exp \frac{r_{\varphi}^{p}(\tau)}{\beta} \right) / Z \cdot d^{\tau}(s,a) \left(\frac{Z - Z^l}{Z^h} \right) && (19) \\ &= \sum_{\tau \in \mathcal{D}^l} \left(\exp \frac{r_{\varphi}^{p}(\tau)}{\beta} \right) / Z^l \cdot d^{\tau}(s,a) \cdot \frac{Z^l}{Z} + \sum_{\tau \in \mathcal{D}^h} \left(\exp \frac{r_{\varphi}^{p}(\tau)}{\beta} \right) / Z^h \cdot d^{\tau}(s,a) \\ &\quad - \sum_{\tau \in \mathcal{D}^h} \left(\exp \frac{r_{\varphi}^{p}(\tau)}{\beta} \right) / Z^h \cdot d^{\tau}(s,a) \cdot \frac{Z^l}{Z}. && (20) \end{aligned}$$

Define that $d_1(s,a) = \sum_{\tau \in \mathcal{D}^l} \left(\exp \frac{r^p_\varphi(\tau)}{\beta} \right) / Z^l$ and $d_2(s,a) = \sum_{\tau \in \mathcal{D}^h} \left(\exp \frac{r^p_\varphi(\tau)}{\beta} \right) / Z^h$. Obviously, d_1 and d_2 are two state-action distributions. By substituting d_1 and d_2 and let $w^h_{\max} = \max_{\tau \in \mathcal{D}^h} \exp \frac{r^p_\varphi(\tau)}{\beta}$, we obtain

$$d^{\mathcal{D}^d}_{\tilde{\rho}_\varphi}(s,a) = (d_1(s,a) - d_2(s,a)) \cdot \frac{Z^l}{Z} + \sum_{\tau \in \mathcal{D}^h} \left(\exp \frac{r^p_\varphi(\tau)}{\beta} \right) / Z^h \cdot d^\tau(s,a). \tag{21}$$

Consider the second term in Eq. (21)

$$\begin{aligned}
&\sum_{\tau \in \mathcal{D}^h} \left(\exp \frac{r^p_\varphi(\tau)}{\beta} \right) / Z^h \cdot d^\tau(s,a) \\
=& \sum_{\tau \in \mathcal{D}^h} w^h_{\max} / Z^h \cdot d^\tau(s,a) - \sum_{\tau \in \mathcal{D}^h} \left(w^h_{\max} - \exp \frac{r^p_\varphi(\tau)}{\beta} \right) / Z^h \cdot d^\tau(s,a) \qquad (22) \\
=& \sum_{\tau \in \mathcal{D}^h} w^h_{\max} / Z^h \cdot d^\tau(s,a) \cdot \frac{Z^h + (w^h_{\max}|\mathcal{D}^h| - Z^h)}{w^h_{\max}|\mathcal{D}^h|} \\
&- \sum_{\tau \in \mathcal{D}^h} \left(w^h_{\max} - \exp \frac{r^p_\varphi(\tau)}{\beta} \right) / (w^h_{\max}|\mathcal{D}^h| - Z^h) \cdot d^\tau(s,a) \cdot \frac{w^h_{\max}|\mathcal{D}^h| - Z^h}{Z^h} \qquad (23) \\
=& \frac{1}{|\mathcal{D}^h|} \sum_{\tau \in \mathcal{D}^h} d^\tau(s,a) + \frac{1}{|\mathcal{D}^h|} \sum_{\tau \in \mathcal{D}^h} d^\tau(s,a) \cdot \frac{w^h_{\max}|\mathcal{D}^h| - Z^h}{Z^h} \\
&- \sum_{\tau \in \mathcal{D}^h} \left(w^h_{\max} - \exp \frac{r^p_\varphi(\tau)}{\beta} \right) / (w^h_{\max}|\mathcal{D}^h| - Z^h) \cdot d^\tau(s,a) \cdot \frac{w^h_{\max}|\mathcal{D}^h| - Z^h}{Z^h}. \qquad (24)
\end{aligned}$$

Define that $d_3(s,a) = \sum_{\tau \in \mathcal{D}^h} \left(w^h_{\max} - \exp \frac{r^p_\varphi(\tau)}{\beta} \right) / (w^h_{\max}|\mathcal{D}^h| - Z^h) \cdot d^\tau(s,a)$, and d_3 is a state-action distribution. Then

$$\sum_{\tau \in \mathcal{D}^h} \left(\exp \frac{r^p_\varphi(\tau)}{\beta} \right) / Z^h \cdot d^\tau(s,a) = d^{\mathcal{D}^h}(s,a) + \left(d^{\mathcal{D}^h}(s,a) - d_3(s,a) \right) \cdot \frac{w^h_{\max}|\mathcal{D}^h| - Z^h}{Z^h}. \tag{25}$$

Combine Eq. 21 and Eq. 25, we obtain

$$d^{\mathcal{D}^d}_{\tilde{\rho}_\varphi}(s,a) = d^{\mathcal{D}^h}(s,a) + (d_1(s,a) - d_2(s,a)) \cdot \frac{Z^l}{Z} + \left(d^{\mathcal{D}^h}(s,a) - d_3(s,a) \right) \cdot \frac{w^h_{\max}|\mathcal{D}^h| - Z^h}{Z^h}. \tag{26}$$

The imitation gap follows by

$$\begin{aligned}
(1-\gamma)\left(V^{\pi^h} - V^{\hat{\pi}} \right) &= \sum_{(s,a) \in \mathcal{S} \times \mathcal{A}} \left(d^{\pi^h}(s,a) - d^{\hat{\pi}}(s,a) r(s,a) \right) && (27) \\
&\leq \| d^{\pi^h} - d^{\hat{\pi}} \|_1 && (28) \\
&\leq \| d^{\pi^h} - d^{\mathcal{D}^d}_{\tilde{\rho}_\varphi} \|_1 + \| d^{\mathcal{D}^d}_{\tilde{\rho}_\varphi} - d^{\hat{\pi}} \|_1 && (29) \\
&\leq \| d^{\mathcal{D}^d}_{\tilde{\rho}_\varphi} - d^{\pi^h} \|_1 + \min_{\pi \in \Pi} \| d^{\mathcal{D}^d}_{\tilde{\rho}_\varphi} - d^{\pi} \|_1 + \epsilon_{\text{OPT}} && (30) \\
&\leq 2 \| d^{\mathcal{D}^d}_{\tilde{\rho}_\varphi} - d^{\pi^h} \|_1 + \epsilon_{\text{OPT}} && (31) \\
&\leq 2 \| d^{\mathcal{D}^h} + (d_1 - d_2) \cdot \frac{Z^l}{Z} + \left(d^{\mathcal{D}^h} - d_3 \right) \cdot \frac{w^h_{\max}|\mathcal{D}^h| - Z^h}{Z^h} - d^{\pi^h} \| + \epsilon_{\text{OPT}} && (32) \\
&\leq 2 \| d_1 - d_2 \|_1 \cdot \frac{Z^l}{Z} + 2 \| d^{\mathcal{D}^h} - d_3 \|_1 \cdot \frac{w^h_{\max}|\mathcal{D}^h| - Z^h}{Z^h} + 2 \| d^{\mathcal{D}^h} - d^{\pi^h} \| + \epsilon_{\text{OPT}} && (33) \\
&\leq 4 \frac{Z^l}{Z} + 4 \frac{w^h_{\max}|\mathcal{D}^h| - Z^h}{Z^h} + 2\epsilon_{\text{EST}} + \epsilon_{\text{OPT}}. && (34)
\end{aligned}$$

Define that

$$\begin{aligned}
\tau^l_{\text{max}} &= \arg\max_{\tau \in \mathcal{D}^l} \exp \frac{r^p_\varphi(\tau)}{\beta}, \\
\tau^h_{\text{max}} &= \arg\max_{\tau \in \mathcal{D}^h} \exp \frac{r^p_\varphi(\tau)}{\beta}, \\
\tau^h_{\text{min}} &= \arg\min_{\tau \in \mathcal{D}^h} \exp \frac{r^p_\varphi(\tau)}{\beta}.
\end{aligned} \tag{35}$$

Finally we get that

$$\begin{aligned}
V^{\pi^h} - V^{\hat{\pi}} &\leq \frac{1}{1-\gamma} \left(4\frac{Z^l}{Z} + 4\frac{w^h_{\text{max}}|\mathcal{D}^h| - Z^h}{Z^h} + 2\epsilon_{\text{EST}} + \epsilon_{\text{OPT}} \right) && (36) \\
&\leq \frac{1}{1-\gamma} \left(4\frac{\sum_{\tau \in \mathcal{D}^l} \exp \frac{r^p_\varphi(\tau)}{\beta}}{\sum_{\tau \in \mathcal{D}^d} \exp \frac{r^p_\varphi(\tau)}{\beta}} + 4\frac{\sum_{\tau \in \mathcal{D}^h} \exp \frac{r^p_\varphi(\tau^h_{\text{max}})}{\beta} - \exp \frac{r^p_\varphi(\tau)}{\beta}}{\sum_{\tau \in \mathcal{D}^h} \exp \frac{r^p_\varphi(\tau)}{\beta}} + 2\epsilon_{\text{EST}} + \epsilon_{\text{OPT}} \right) && (37) \\
&\leq \frac{1}{1-\gamma} \left(4\frac{|\mathcal{D}^l| \exp \frac{r^p_\varphi(\tau^l_{\text{max}})}{\beta}}{\exp \frac{r^p_\varphi(\tau^h_{\text{max}})}{\beta}} + 4\frac{|\mathcal{D}^h| \left(\exp \frac{r^p_\varphi(\tau^h_{\text{max}})}{\beta} - \exp \frac{r^p_\varphi(\tau^h_{\text{min}})}{\beta} \right)}{|\mathcal{D}^h| \exp \frac{r^p_\varphi(\tau^h_{\text{min}})}{\beta}} + 2\epsilon_{\text{EST}} + \epsilon_{\text{OPT}} \right) && (38) \\
&= \frac{1}{1-\gamma} \left(4|\mathcal{D}^l| \exp \frac{-\left(r^p_\varphi(\tau^h_{\text{max}}) - r^p_\varphi(\tau^l_{\text{max}})\right)}{\beta} + 4\left(\exp \frac{r^p_\varphi(\tau^h_{\text{max}}) - r^p_\varphi(\tau^h_{\text{min}})}{\beta} - 1 \right) + 2\epsilon_{\text{EST}} + \epsilon_{\text{OPT}} \right) && (39)
\end{aligned}$$

□

C. Detailed Implementation of PAIL

We developed maximum entropy IRL part of PAIL by modifying the `f-IRL` codebase[1]. From this base, we have made the following primary modifications:

1. Train the discriminator from samples in the replay buffer instead of interacting with the environment for sample efficiency like DAC (Kostrikov et al., 2019);
2. Implement the reweighting of demonstrations;
3. Implement demonstration augmentation.

The preference reward part of PAIL was developed by the `BPref` codebase[2] (Lee et al., 2021b). While preserving the original functionality, we have modified the code to support trajectories with different lengths.

D. Experiment Details

D.1. GridWorld Task Construction

In the context of the grid-world task, the environment is designed to challenge an agent's ability to navigate towards a target location while staying within a predefined circular boundary. The environment is formalized as follows:

The state space $\mathcal{S} \subseteq \mathbb{R}^4$ is defined by the Cartesian coordinates of the agent and the target. A state $s \in \mathcal{S}$ is represented as a vector $(x, y, x_{\text{target}}, y_{\text{target}})$, where (x, y) denotes the agent's current position and $(x_{\text{target}}, y_{\text{target}})$ denotes the target's position.

The action space $\mathcal{A} \subseteq \mathbb{R}^2$, consists of two continuous dimensions representing the agent's intended movement along the horizontal (Δx) and vertical (Δy) axes. Each action $a \in \mathcal{A}$ is a vector $(\Delta x, \Delta y)$, constrained by a maximum step size Δ_{max}.

[1] https://github.com/twni2016/f-IRL

[2] https://github.com/rll-research/BPref

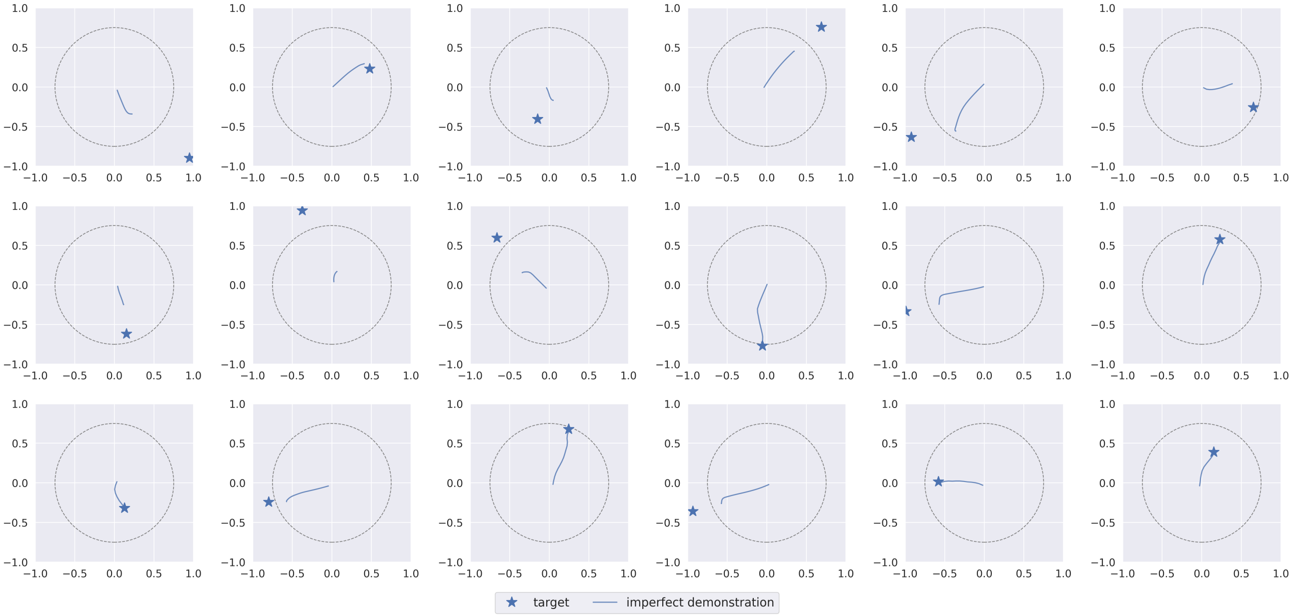

Figure 6. Part of the imperfect demonstration dataset ‘M’ for GridWorld.

The transition function $\mathcal{T} : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ dictates the state transition dynamics. Given a current state $s = (x, y, x_{\text{target}}, y_{\text{target}})$ and an action $a = (\Delta x, \Delta y)$, the next state $s' = (x', y', x_{\text{target}}, y_{\text{target}})$ is computed as follows:

$$x' = \text{clip}(x + \Delta x, -\frac{W}{2}, \frac{W}{2}),$$
$$y' = \text{clip}(y + \Delta y, -\frac{H}{2}, \frac{H}{2}),$$

where $\text{clip}(\cdot, \text{min}, \text{max})$ ensures the agent’s position remains within the environmental boundaries of width W and height H. The target’s position remains unchanged during the transition.

The reward function $R(s, a, s')$ is designed to incentivize the agent to minimize its distance to the target while staying within the circular boundary. Specifically, the reward for a transition is defined based on the agent’s position relative to the boundary and the target, encouraging strategic navigation and penalizing boundary violations. The reward function is defined as:

$$R(s, a, s') = \begin{cases} -P \cdot D_{\text{max}}, & \text{if } \sqrt{x'^2 + y'^2} > r, \\ D_{\text{max}} \cdot B - \sqrt{(x' - x_{\text{target}})^2 + (y' - y_{\text{target}})^2}, & \text{otherwise,} \end{cases}$$

where P is a penalty ratio for leaving the boundary, D_{max} represents the maximum possible distance within the environment, B is a baseline reward ratio, r denotes the radius of the circular boundary, and (x', y') and $(x_{\text{target}}, y_{\text{target}})$ are the agent’s and target’s positions in the next state s', respectively. This reward structure ensures that the agent is penalized for exiting the boundary and rewarded for proximity to the target, with the ultimate goal of reaching the target location within the circular boundary.

D.2. Imperfect Demonstrations and Human Preferences

Imperfect demonstration dataset for GridWorld. The imperfect demonstration dataset ‘M’ for GridWorld is collected by both policies saved at various training stages of RL and the optimal policy designed by hand. Some of the 600 trajectories in dataset ‘M’ for GridWorld is shown in Fig. 6. All the trajectories of the dataset ‘M’ for GridWorld remain within the boundary and 58.4% of the trajectories whose targets are positioned inside the boundary reach the target. 600 trajectories of the demonstration dataset ‘M’ for GridWorld are used for learning.

Imperfect demonstration dataset for Mujoco and DMC tasks. The imperfect demonstration datasets for Mujoco and DMC tasks are collected by policies saved at different training stages of RL. Quality of different datasets for Mujoco and DMC is as follows:

1. 'L' for Mujoco: trajectories have the normalized return within $[0.1, 0.6)$ (within $[0.3, 0.7)$ for *Ant-v2*);
2. 'M' for Mujoco: trajectories have the normalized return within $[0.35, 0.85)$ (within $[0.5, 0.9)$ for *Ant-v2*);
3. 'H' for Mujoco: trajectories have the normalized return within $[0.6, 1.0]$ (within $[0.7, 1.0]$ for *Ant-v2*);
4. 'M' for DMC: trajectories have the normalized return within $[0.45, 0.95)$.

Demonstration datasets for *Ant-v2* are sampled with higher normalized return to exclude bad trajectories with negative oracle returns. Each experiment of Mujoco and DMC tasks uses 10 trajectories of a certain dataset, except for the ablation study on the number of initial demonstrations.

Human preferences. We employ segments of length 100 for GridWorld and Mujoco tasks. For DMC tasks, we use segments of length 50 by following Lee et al. (2021a). 500 preference queries are used for GridWorld. For Mujoco and DMC tasks, 60 and 1400 preference queries are used excepted for the ablation study on preference queries number.

D.3. Return Normalization of Experiments

Table 5. The minimum and maximum returns of Mujoco and DMC tasks for return normalization in experiments.

	Ant-v2	HalfCheetah-v2	Hopper-v2	Humanoid-v2	Walker2d-v2	cheetah_run	quadruped_walk	walker_walk
Min. Return	-2934.14	-51.90	5.00	113.06	-1.21	0.00	0.00	0.00
Max. Return	6308.78	13043.84	3767.15	5770.77	5729.49	1000.00	1000.00	1000.00

Min-max normalization is applied to the returns across the Mujoco and DMC tasks for brevity. For Mujoco tasks, the returns of the worst trajectory and the best trajectory during the RL training process are utilized as the minimum and maximum returns for normalization, and the minimum and maximum returns for DMC tasks are defined by Tassa et al. (2018). We record the minimum and maximum returns for normalization in Table 5.

D.4. Introduction & Implementation of Primary Baselines

We compare PAIL with 2 kinds of baselines. The first kind is based on AIL:

* *MCE-IRL* (Ziebart et al., 2008; 2010), i.e. maximum casual entropy IRL, a typical IRL method. Moreover, inspired by DAC(Kostrikov et al., 2019), we sample from replay buffer for discriminator update in MCE-IRL for sample efficiency. For implement, we modify the `f-IRL` codebase[3];
* *SAIL-TRPO* (Wang et al., 2021a), a state-of-art method to learn from imperfect demonstrations based on IRL. SAIL-TRPO outperforms other methods learning from imperfect demonstrations like WGAIL (Wang et al., 2021b), D-REX(Brown et al., 2019b). Corresponding results are obtained by running the official source code[4] on our tasks;
* *SAIL-SAC*, a variant of SAIL-SAC, which use SAC instead of TRPO for policy update in the IRL process. For this algorithm, we extended the original SAIL-TRPO source code by replace the TRPO by SAC;
* *AILP* (Taranovic et al., 2023), a method based on IRL paradigm which learns from both demonstrations and human preferences. We implemented by ourselves due to the lack of open-source code. The hyperparameters are based on those detailed in the original paper and its appendix.

The second kind is PBRL, which learns policy by RL from the preference reward obtained by human preferences:

[3] https://github.com/twni2016/f-IRL

[4] https://github.com/yunke-wang/SAIL

- *PEBBLE* (Lee et al., 2021a), a feedback-efficient PBRL algorithm with unsupervised pre-training. Corresponding results are obtained by running the official source code[5] (Lee et al., 2021b) on our tasks;
- *BC-PEBBLE*, a variant of PEBBLE (Lee et al., 2021a) which utilizes behavior cloning (BC) to learn the initial policy from demonstrations and samples preferences from both demonstrations and replay buffer of trajectories inspired by Ibarz et al. (2018). For this algorithm, we extended the original Pebble source code by incorporating BC on given demonstrations and add demonstrations into replay buffer for preference sampling and policy learning.

All the baselines including PAIL update policy by SAC (Haarnoja et al., 2018a;b), except for SAIL-TRPO which trains policy by TRPO (Ho et al., 2016). Among them, PEBBEL learns from human preferences; MCE-IRL, SAIL-TRPO and SAIL-SAC learn from the imperfect demonstration dataset; BC-PEBBLE, AILP along with PAIL learn from both human preferences and the imperfect demonstration dataset.

D.5. Hyper-parameters

PAIL is based on general AIL framework. For demonstration augmentation mentioned in Section 4.3, we add 3 trajectories with the highest preference rewards $r^p(\tau)$ every $1e5$ steps. Before updating discriminator reward and policy, PAIL reweights the imperfect demonstration $\mathcal{D}^d$, with the entropy coefficient β as shown in Table 6.

For discriminator learning, PAIL's algorithm regularly updates the discriminator reward with weighted demonstrations from $\mathcal{D}^d$ and agent samples from replay buffer $\mathcal{B}$, with specific hyperparameters detailed in Table 8. For policy learning, the PAIL algorithm updates the policy based on SAC, with parameters referenced in Table 9. For all tasks, we largely follow this set of SAC parameters. Specifically, for the quadruped_walk task in the DMC benchmark using the Medium dataset, we set the SAC learning rate to $5e-4$ and enable auto alpha. To ensure stable learning between the discriminator and the policy, the updating of the policy and the discriminator is repeated 1 times, i.e. $n^{\text{disr}} = n^{\text{policy}} = 1$, except for 'HalfCheetah-v2, M' and 'HalfCheetah-v2, H' ($n^{\text{disr}} = 1, n^{\text{policy}} = 2$). For the part of learning preference reward in PAIL, we use the hyperparameters as shown in Table 7.

Additionally, We run GridWorld for $1e5$ time steps, and Mujoco and DMC tasks for $1e6$ time steps.

Table 6. Entropy Coefficient β for reweighting the imperfect demonstrations in all environments.

Task & Dataset	β
Ant-v2, L,M,H	50
HalfCheetah-v2, L	50
HalfCheetah-v2, M	20
HalfCheetah-v2, H	15
Hopper-v2, L	50
Hopper-v2, M	50
Hopper-v2, H	45
Humanoid-v2, L,M,H	50
Walker2d-v2, L	45
Walker2d-v2, M	7
Walker2d-v2, H	50
cheetah_run, M	50
quadruped_walk, M	25
walker_walk, M	50

E. Additional Experiment Results

E.1. PAIL with Imperfect preferences

To evaluate PAIL with more realistic and practical preference models, we consider the cases described in the Section 3.3 of Lee et al. (2021b), i.e. stochastic preference model, myopic behavior, skipping queries, equally preferable and making a mistake. For the each case above, we do experiments in the 'HalfCheetah-v2, M' and the 'Walker2d-v2, M' task with 60 preference queries and record the normalized return. The training curve averaged over 5 seeds are as in Fig. 7. By observing the curves, it becomes apparent that PAIL performs stably and robustly over 5 different preference models along with the

[5] https://github.com/rll-research/BPref

Table 7. Hyperparameters of Preference Reward r^p.

Hyperparameter	Value
Ensemble size	3
Number of layers	3
Hidden dimension	256
Activation	tanh
Batch size	128
Learning rate	$3e-4$
Optimizer	Adam

Table 8. Hyperparameters of Discriminator Reward r^d.

Hyperparameter	Value
Number of layers	2
Hidden dimension	128 for mujoco, 256 for dmc
Weight decay	$1e-3$
Activation	relu
Batch size	5000
Learning rate	$1e-4$
Optimizer	Adam

Table 9. Hyperparameters of SAC.

Hyperparameter	Value
Number of layers	2
Hidden dimension	128 for mujoco, 1024 for dmc
Auto Alpha	False
Activation	relu
Batch size	256
Learning rate	$1e-3$
Optimizer	Adam

’Oracle’ model. In particular, the ’Equal’ model even outperforms the ’Oracle’ model in the ’HalfCheetah-v2, M’ task. Therefore, judging from the experimental results, PAIL is expected to perform robustly and well in practical applications.

We also evaluated the performance of PAIL under different error rates of teacher’s judgements. We conducted experiments in the ’HalfCheetah-v2, M’ and the ’Walker2d-v2, M’ task with 60 preference queries. The training curve averaged over 5 seeds are as in Fig. 8. The results show that the perfermance of PAIL remains stable when error rate is less than 0.1, and gradually declines as error rate continues to increase. In real scenarios, the quality of preference queries should be ensured as much as possible. When the error rate is less than 10%, PAIL can be robust and still performs well.

E.2. Performance Compared with Automated Preference Generation Method

We additionally compare PAIL with SSRR (Chen et al., 2020). The results, summarized in Table 10, were derived from experiments conducted with five distinct seeds. It can be consistently observed that PAIL outperformed SSRR across all tasks. Notably, we find SSRR exhibited negative improvement in several tasks where SSRR performs worse than the scores of the demonstrations. In Walker2d, SSRR is even inferior to the minimal demonstration return, likely due to inaccuracies in its reward function.

E.3. Learning Curves of PAIL

The learning curves of PAIL, along with the average and the best normalized return of the initial demonstrations, for experiments in Table 2 are depicted in Fig. 9. These curves illustrate the stable learning process of PAIL and the small shaded areas indicate the robustness of PAIL across different seeds. Moreover, PAIL ourperforms the average normalized return in all the tasks and breaks through the performance bottleneck of the best demonstration in 16 out of 18 ‘Task & Dataset’.

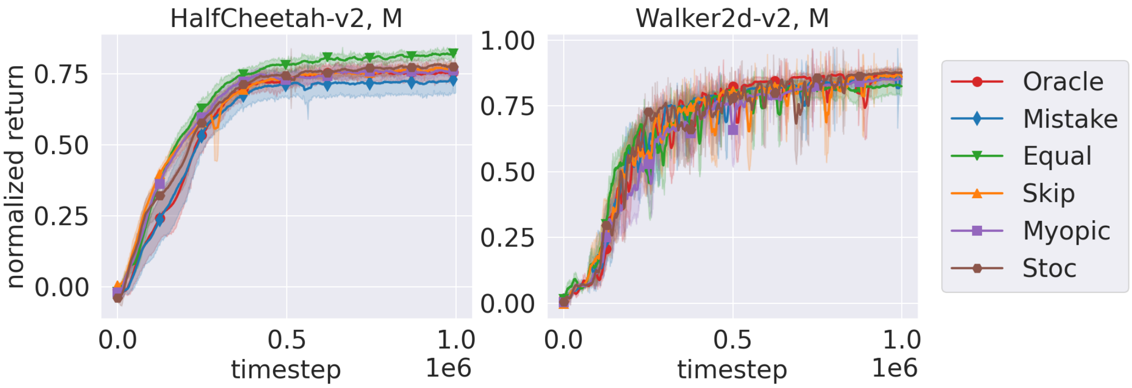

Figure 7. Learning curves of PAIL with preferences from 6 different scripted teachers. The curves are shaded with 1 standard error over 5 seeds.

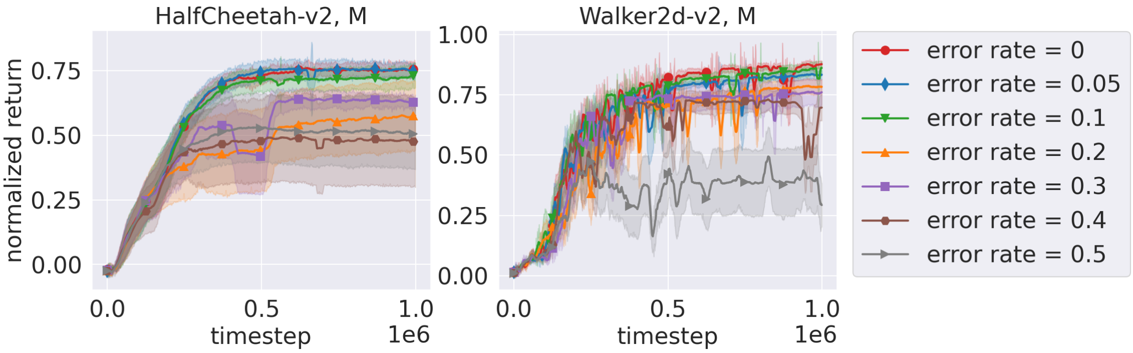

Figure 8. Learning curves of PAIL under different error rates of preferences. The curves are shaded with 1 standard error over 5 seeds.

E.4. Additional Results for Preference Returns and Weights

The preference returns and weights for all the experiments in Table 2 are visualized in Fig. 10. It is noted that in all the tasks, preference returns and oracle returns are almost positively correlated. The preference weights for trajectories with high oracle returns exhibit significantly elevated values, whereas those for trajectories with low oracle returns approach nearly zero, which enables PAIL to imitate the trajectories with high oracle returns and to disregard trajectories with poor performance. Moreover, augmented demonstrations outperforms initial demonstrations in 17 out of 18 'Task & Dataset', which helps PAIL to break the performance bottleneck of the initial demonstration dataset.

E.5. Additional Results for Preference Reward

PAIL trains the preference reward r_φ^p with the demonstration dataset $\mathcal{D}^d$ and subsequently update it through retraining with new queries for human preferences after each dataset augmentation. Policy improvement of PAIL is achieved through imitating the dataset reweighted by r_φ^p. PAIL works by utilizing r_φ^p to make precise predictions within the dataset $\mathcal{D}^d$ and reweighting the dataset consistent with human preferences.

PBRL (Lee et al., 2021a; Christiano et al., 2017) trains the preference reward using the human preferences sampled from all trajectories acquired through interactions with the environment, denoted as $\mathcal{B}^{\text{PBRL}}$. The policy is then improved based on the preference reward signal of $\mathcal{B}^{\text{PBRL}}$. Therefore, the preference reward needs to accurately predict all samples in $\mathcal{B}^{\text{PBRL}}$, including a variety of state-action pairs ranging from bad to good, to avoid misleading the learning policy.

With the same number of human preferences, preference reward is more likely to generalize to a simpler hypothesis

Table 10. Normalized average returns in Mujoco tasks averaged over 5 seeds with 60 preference queries. 'Demo. (Avg.)' and 'Demo. (Best)' respectively represent the average and the best return within trajectories of the demonstration dataset.

Task & Dataset	Demo. (Avg.)	Demo. (Best)	SSRR	PAIL (ours)
Ant-v2, L	0.46	0.61	0.22 ± 0.35	$\mathbf{0.79 \pm 0.01}$
Ant-v2, M	0.63	0.77	0.19 ± 0.34	$\mathbf{0.87 \pm 0.00}$
Ant-v2, H	0.80	0.93	0.19 ± 0.32	$\mathbf{0.93 \pm 0.01}$
HalfCheetah-v2, L	0.20	0.01	0.05 ± 0.05	$\mathbf{0.62 \pm 0.04}$
HalfCheetah-v2, M	0.39	0.10	0.04 ± 0.05	$\mathbf{0.75 \pm 0.02}$
HalfCheetah-v2, H	0.61	0.01	0.03 ± 0.05	$\mathbf{0.88 \pm 0.01}$
Hopper-v2, L	0.16	0.35	0.26 ± 0.04	$\mathbf{0.93 \pm 0.01}$
Hopper-v2, M	0.41	0.61	0.32 ± 0.05	$\mathbf{0.92 \pm 0.02}$
Hopper-v2, H	0.67	0.85	0.31 ± 0.06	$\mathbf{0.96 \pm 0.01}$
Humanoid-v2, L	0.36	0.53	0.54 ± 0.04	$\mathbf{0.95 \pm 0.02}$
Humanoid-v2, M	0.59	0.82	0.49 ± 0.11	$\mathbf{0.98 \pm 0.02}$
Humanoid-v2, H	0.85	0.98	0.45 ± 0.08	$\mathbf{1.04 \pm 0.02}$
Walker2d-v2, L	0.26	0.39	-0.03 ± 0.01	$\mathbf{0.48 \pm 0.02}$
Walker2d-v2, M	0.47	0.65	-0.06 ± 0.01	$\mathbf{0.88 \pm 0.01}$
Walker2d-v2, H	0.70	0.92	-0.02 ± 0.00	$\mathbf{0.90 \pm 0.03}$

space (Mohri et al., 2018). Clearly, the complexity of the hypothesis space for the demonstration dataset $\mathcal{D}^d$ is significantly smaller than that for all samples in the entire PBRL training process $\mathcal{B}^{\text{PBRL}}$, which potentially distributed across the whole state-action space. With preciser predictions within a smaller dataset, PAIL works with limited human preferences.

We visualize the preference rewards for experiments of Table 4 in Fig. 11,12. It is visually evident that PAIL demonstrates the highest correlation between preference reward and oracle reward within the demonstration dataset, whereas all preference rewards within the replay buffers in the entire learning process exhibit poor performance. This observation elucidates the rationale behind PAIL's need for fewer preference queries compared to PBRL.

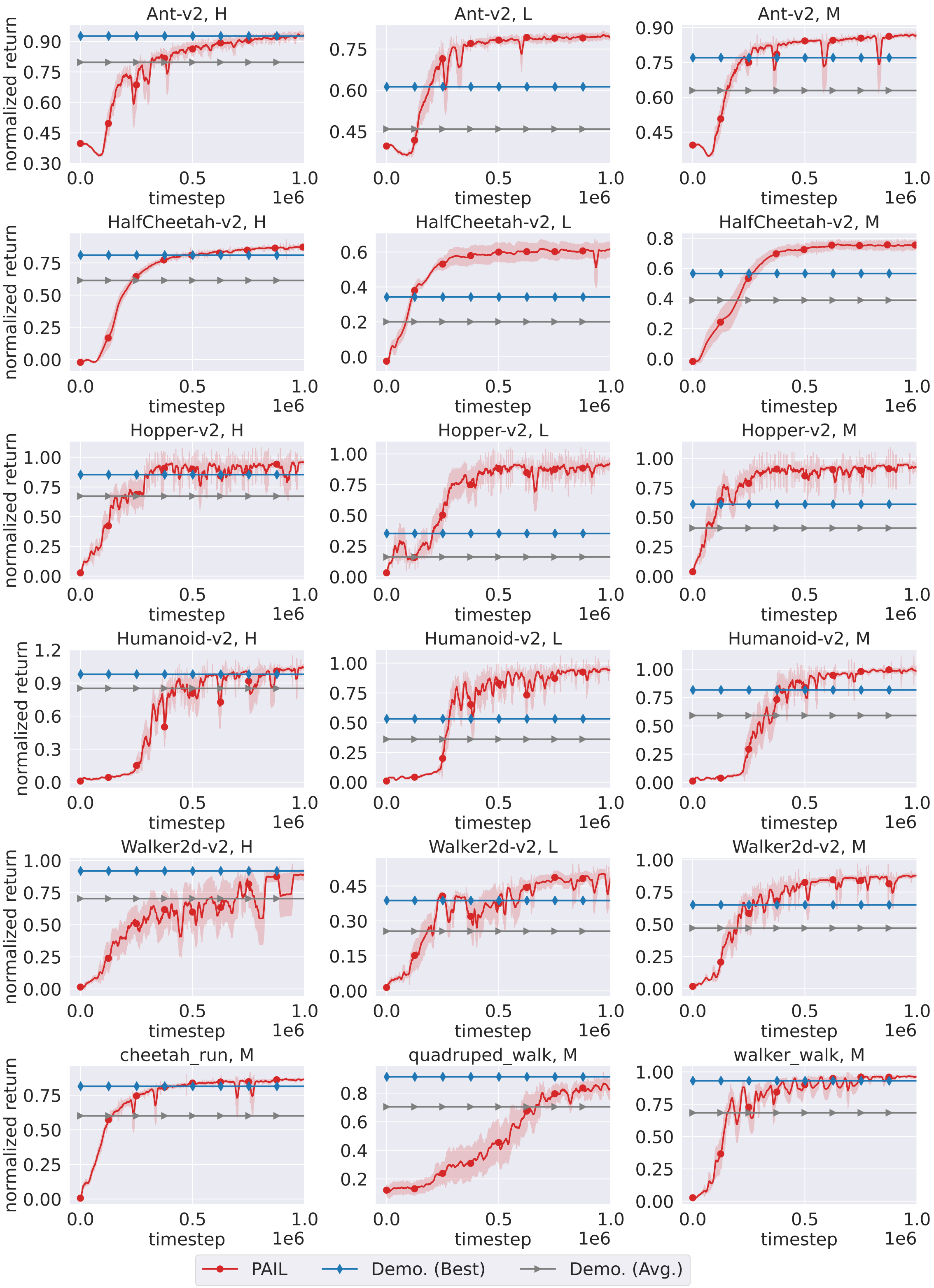

Figure 9. Learning curves of PAIL, along with the average and the best normalized return of the initial demonstrations, for experiments in Table 2. The curves are shaded with 1 standard error over 5 seeds.

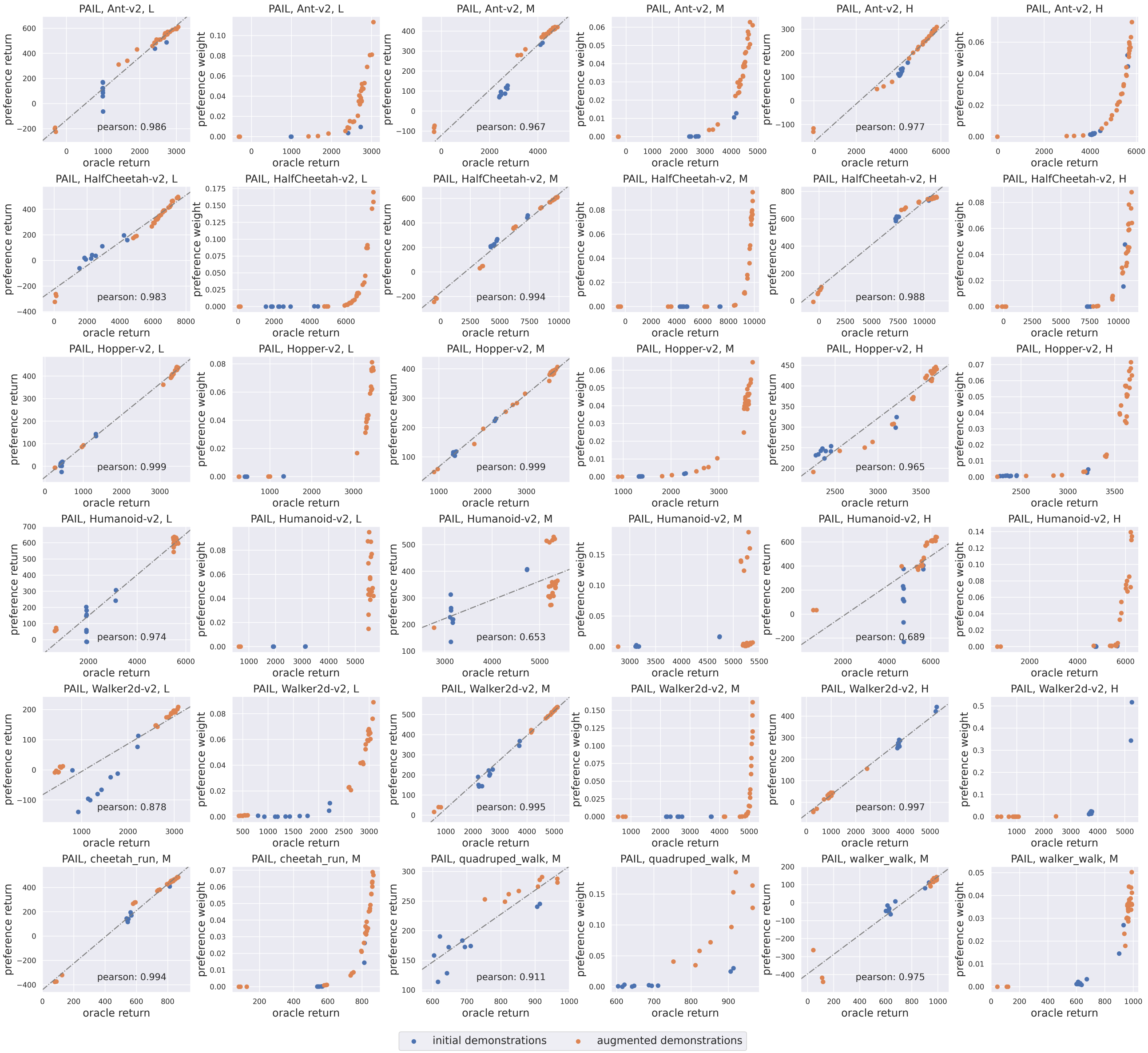

Figure 10. Visualization of preference returns and weights for all the experiments in Table 2.

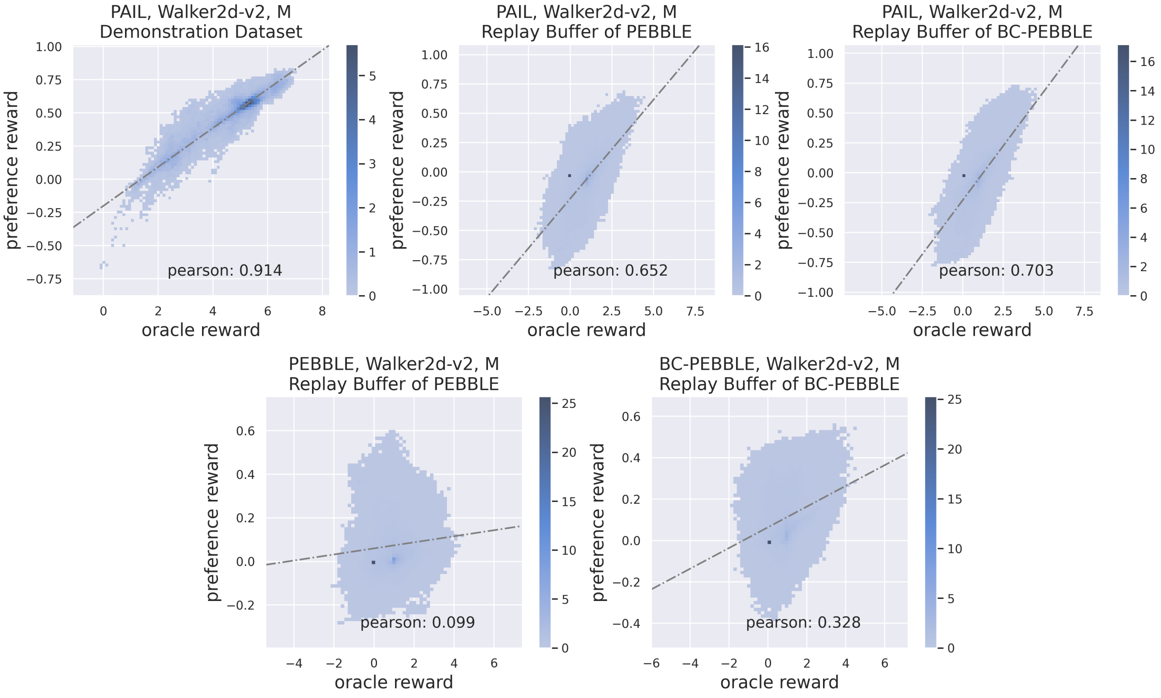

Figure 11. Visualization of preference rewards for ‘Walker2d-v2, M’ in Table 4 experiments.

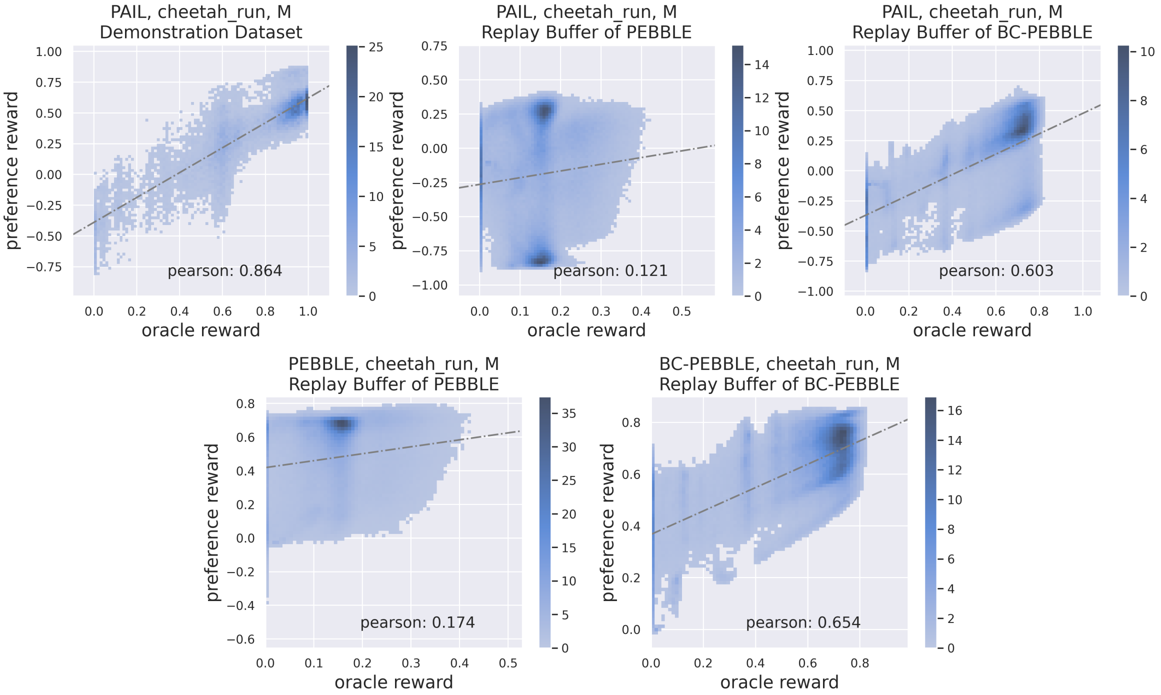

Figure 12. Visualization of preference rewards for ‘cheetah_run, M’ in Table 4 experiments.