Sample-Efficient Preference-Based Reinforcement Learning Using Diffusion Models

Jingjing Feng
Duke University
NC, United States
jingjing.feng@duke.edu

Lucheng Wang
University of Liverpool
Liverpool, United Kingdom
sglwan19@liverpool.ac.uk

Alona Tenytska
Sumy State University
Sumy, Ukraine
alona.tenytska@liverpool.ac.uk

Bei Peng
University of Liverpool
Liverpool, United Kingdom
bei.peng@liverpool.ac.uk

ABSTRACT

Preference-based reinforcement learning (RL) has shown great potential for scaling human feedback to deep RL settings, enabling agents to solve complex tasks without access to a pre-defined reward function. Many state-of-the-art preference-based RL methods use off-policy learning to allow the agent to reuse previously collected experiences to improve learning efficiency. However, passively reusing prior data can limit the generality since the dataset of recent experiences is typically limited and not sufficiently diverse. This data limitation problem, on the other hand, has recently been addressed by using diffusion generative models to upsample agent experiences in the context of offline and online RL. Inspired by this success, we introduce PRIDE: **P**reference-based **R**einforcement learning using d**I**ffusion mo**DE**l, a novel approach that integrates diffusion models into preference-based RL to improve both sample and feedback efficiency. PRIDE continually trains a diffusion model to approximate the RL agent's online behavioral distribution. The trained diffusion model then generates a large quantity of novel and diverse synthetic experiences, which are used to augment limited real data, enabling better generalization while reducing reliance on real data. We evaluate PRIDE on a variety of locomotion and robotic manipulation tasks. Empirical results demonstrate that PRIDE outperforms state-of-the-art preference-based RL method in most tasks tested and achieves comparable or superior performance with a 50% reduction in human feedback. The novel use of diffusion models in our approach presents a promising direction for improving sample and feedback efficiency in preference-based RL.

KEYWORDS

Reinforcement Learning; Preference-based Reinforcement Learning; Diffusion Models

1 INTRODUCTION

Deep reinforcement learning (RL) has made significant strides in recent years, enabling computers and robots to tackle increasingly complex and challenging tasks [1, 18, 22–24, 34, 40, 49, 52]. Despite the progress, deep RL has not been widely deployed to real-world systems. One central obstacle is that many real-world tasks involve goals that are complex, poorly defined, or unspecified. This makes it challenging to design a good reward function that closely aligns with the long-term goals of the system, which is critical for the success of any RL system.

To solve complex RL tasks without access to a pre-defined reward function, reinforcement learning from human feedback (RLHF), also known as human-in-the-loop reinforcement learning [6, 15, 21, 30, 33, 46, 48], has emerged as a compelling approach. Compared to reward engineering or other reward learning methods [36], RLHF allows us to learn a reward model from human feedback, which can more closely capture the complex behaviors preferred by humans. By optimizing against this reward model using deep RL, we can closely align the agent's behaviors with complex human values and preferences. With RLHF, the agent's objectives can be defined and dynamically refined by a human in the loop, overcoming the limitations of classical RL methods.

Much of the early work on RLHF has focused on solving tabular RL problems and investigating how to better interpret and model human feedback to extract more useful information and improve learning efficiency [21, 31, 33, 46]. One of the early work that scales human feedback up to deep RL is conducted by Christiano et al. [6], which enables agents to solve complex deep RL tasks with high-dimensional state and/or action spaces using human feedback. In their approach, humans are asked to provide online feedback in the form of binary preferences between pairs of agent trajectory segments (i.e., sequences of state-action pairs). This preference-based online feedback has been shown to be easy and practical for humans to provide, scalable to deep RL settings, and effective in mitigating reward exploitation.

Following their work, significant progress has been made in preference-based reinforcement learning. Many state-of-the-art preference-based RL algorithms [5, 14, 25, 27, 38] use off-policy learning to enable the agent to reuse previously collected experiences during training to improve sample and feedback efficiency. While reusing prior data in preference-based RL has shown strong results, a common issue with these approaches is that the dataset of recent agent experiences is typically limited–suitable data for specific agent behaviors may not be available and thus limits the generality. Additionally, it is still expensive for humans to supervise the RL learning process, as it requires a large number of samples for the agent to learn a good policy. Therefore, it is crucial to improve

Proc. of the Adaptive and Learning Agents Workshop (ALA 2025), Avalos, Aydeniz, Müller, Mohammedalamen (eds.), May 19 – 20, 2025, Detroit, Michigan, USA, ala-workshop.github.io. 2025.

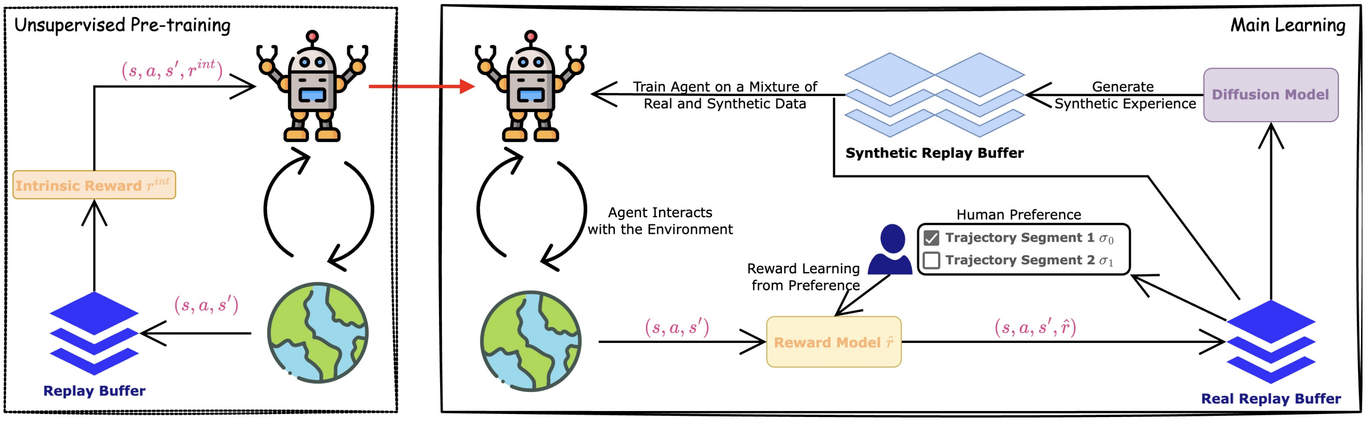

Unsupervised Pre-training
(s, a, s', r^{int})
Intrinsic Reward r^{int}
(s, a, s')
Replay Buffer
Main Learning
Train Agent on a Mixture of Real and Synthetic Data
Synthetic Replay Buffer
Generate Synthetic Experience
Diffusion Model
Agent Interacts with the Environment
Human Preference
☑ Trajectory Segment 1 σ_0
□ Trajectory Segment 2 σ_1
Reward Learning from Preference
(s, a, s')
Reward Model $\hat{r}$
$(s, a, s', \hat{r})$
Real Replay Buffer

Figure 1: Illustration of our approach, which consists of two phases: unsupervised pre-training and main learning. In the first phase, the agent is encouraged to explore the environment and learn diverse behaviors through maximising intrinsic rewards (the state entropy), without extrinsic rewards. In the main learning phase, humans are queried to provide binary preferences between pairs of agent trajectory segments, which are then used to train a reward model to predict human preferences. A diffusion model is periodically trained to approximate the agent's online behavioral distribution, using experiences sampled from the real replay buffer. Once trained, the diffusion model generates a large number of novel and diverse synthetic experiences, which are stored in a separate synthetic replay buffer. We then train the RL agent using experiences sampled from both the real and synthetic replay buffers, leading to improved sample and feedback efficiency.

the sample efficiency of preference-based RL methods and reduce the amount of human effort required for efficient learning.

Meanwhile, rather than passively reusing prior data during RL training, recent work [32] has shown that diffusion generative models [13, 42] can be used to upsample agent experiences, leading to significant improvements in sample efficiency in both offline and online RL settings. It is shown that, compared to other generative models such as VAEs [19] and GANs [10], diffusion models can generate synthetic data that is more dynamically accurate and diverse [32]. The large quantity of new, diverse synthetic experiences generated by the diffusion model can be used to augment limited real data as if they were real experiences. This can significantly broaden the training data available to the RL agent and thus improve sample efficiency and final performance.

Inspired by the success of diffusion models in augmenting traditional datasets in RL [32], our work aims to integrate diffusion generative models into online preference-based RL methods to improve both sample and feedback efficiency. To this end, we present PRIDE: **P**reference-based **R**einforcement learning using d**I**ffusion mo**DE**l. Like PEBBLE [25], a popular state-of-the-art preference-based RL algorithm, our approach uses unsupervised pre-training and off-policy learning to learn from human feedback. However, during off-policy learning, PRIDE continually trains a diffusion model to approximate the RL agent's behavioral distribution as it interacts with the environment and collects new experiences. The diffusion model learns this distribution based on the real experiences stored in the replay buffer. Once trained, the diffusion model generates a large quantity of new synthetic experiences that mimic real agent-environment interactions. These synthetic experiences are then stored in a separate synthetic replay buffer. When training the RL agent, PRIDE samples from both real and synthetic replay buffer to get a mixture of real and synthetic data, leveraging the diverse and novel synthetic experiences to boost learning efficiency. The diverse synthetic data also allows our approach to achieve better generalization while reducing reliance on real data. See Figure 1 for an illustration of our approach.

We remark that the novelty of our work lies in the novel integration of diffusion models into online preference-based RL to improve both sample and feedback efficiency. Our approach unlocks new, efficient, and general training strategies for preference-based RL methods. To summarize, the contributions of this paper are:

- We propose PRIDE, a new preference-based RL algorithm that leverages a diffusion model to generate a large number of novel and diverse synthetic experiences, augmenting real data to boost learning efficiency.
- We show that PRIDE outperforms PEBBLE in various complex locomotion and robotic manipulation tasks from DeepMind Control Suite (DMControl) [43, 44] and Meta-world [50].
- We also show that PRIDE achieves comparable or superior performance to PEBBLE with significantly fewer human feedback queries, demonstrating improved feedback efficiency through the use of novel and diverse synthetic experiences.

2 RELATED WORK

Learning from Human Feedback. There exists a large body of work on learning from human feedback. In the context of RL, the goal is typically to solve a learning problem in which an agent is situated in an environment described by a Markov Decision Process (MDP), with rewards generated by a human teacher instead of from a stationary MDP reward function. Much work in the past has focused on solving tabular RL problems and investigating how to better interpret and model human feedback to extract more useful information and learn more efficiently. Human feedback has been interpreted in many different ways, such as numerical rewards

[16, 21], shaping rewards [45], discrete positive or negative signals [29, 31], and advantage functions [33].

More recently, human feedback has been scaled up to deep RL to solve more complex tasks with high-dimensional state and/or action spaces [6, 15, 48]. Christiano et al. [6] demonstrates the effectiveness of using preference-based human feedback to help solve complex deep RL tasks, where humans provide online feedback in the form of binary preferences between pairs of agent trajectory segments. Following their work, significant progress has been made in preference-based RL. Most state-of-the-art preference-based RL methods [5, 14, 25, 27, 38] use off-policy learning to reuse previously collected experiences during training to improve sample and feedback efficiency. However, passively reusing prior data can limit the generality since the dataset of recent experiences is typically limited and not sufficiently diverse. Different from these methods, our approach makes use of a diffusion model to generate a large quantity of novel and diverse synthetic experiences to augment limited real data to improve learning in preference-based RL.

Generative Models in RL. Generative models, such as generative adversarial networks (GANs) [9] and variational autoencoders (VAEs) [20], have been widely used in RL to improve agent exploration capability and sample efficiency. For example, Gao et al. [8] propose to integrate Deep Q-Networks [35] with GANs to improve interactive recommendation systems. Their approach addresses the issue of sparse positive feedback in recommendation systems by using GANs to generate synthetic negative feedback. Baucum et al. [2] apply RL in healthcare by utilizing VAEs to generate additional patient trajectories, improving the model's ability to explore and learn from diverse patient data. More recent work has seen diffusion models emerge as a powerful generative tool in RL [17, 47, 53]. For instance, Ni et al.[37] introduce MetaDiffuser, a task-oriented conditioned diffusion planner for offline meta-RL, which enables the generation of task-oriented trajectories to improve generalization across diverse tasks.

Despite the widespread use of generative models in offline and online RL settings, very little work has applied them to online preference-based RL. One notable example is the recent work by Zhan et al. [51], who train a GAN to predict human preferences in preference-based RL. In their approach, a low-dimensional GAN is trained on human feedback data and then used to label trajectory segments in place of humans. Unlike their work, our approach integrates diffusion models into preference-based RL to generate novel and diverse synthetic experiences, augmenting the limited real data collected by the agent to improve both sample and feedback efficiency. This is inspired by the recent work of Lu et al.[32], who propose to train diffusion models to upsample agent experiences, leading to significant improvements in sample efficiency in both offline and online RL settings. They show that the quality of the synthetic samples generated by the diffusion model is significantly better than that of previous generative models such as VAEs [19] and GANs [10], exhibiting higher diversity and better alignment with the real data distribution. Inspired by their success, our approach trains diffusion models to generate synthetic experiences in a similar manner, but with a focus on integrating diffusion models into online preference-based RL, which, to the best of our knowledge, has not been explored before.

3 BACKGROUND

3.1 Reinforcement Learning

In standard reinforcement learning settings, an agent interacts with an uncertain environment and tries to maximize its long-term expected cumulative reward. The underlying decision-making problem can be modelled as a Markov decision process (MDP), which can be represented by a tuple $< S, A, P, R, \gamma >$. At each timestep t, the agent is in an environment state $s_t \in S$ and selects an action $a_t \in A$ to take to influence the environment. After taking action a_t, the agent moves to the next environment state s_{t+1} with some probability defined by the transition function $P(s_{t+1}|s_t, a_t) : S \times A \times S \to [0, 1]$. $R(s, a) : S \times A \to \mathbb{R}$ is the reward function specifying the numeric reward r_t the agent receives for taking action a_t in state s_t and transitioning to state s_{t+1}. $\gamma \in [0, 1]$ is a discount factor specifying how much immediate rewards are preferred to future rewards. The goal of the agent is to find the optimal policy $\pi^*(a|s)$ that maximizes the expected cumulative discounted reward $R_t = \sum_{i=0}^{\infty} \gamma^i r_{t+i}$.

3.2 Reward Learning from Preferences

In this work, we consider RL tasks without access to a pre-defined reward function. This means that, in our problem setting–unlike in standard RL settings–the environment does not provide a reward signal to the agent as it interacts with the environment. Instead, we assume that there is a human teacher who can provide feedback as the agent acts in the environment. We can then learn the reward function from human feedback. We adopt a reward learning from preference framework, where the reward function is learned from human feedback in the form of binary preferences between pairs of *trajectory segments* [6]. A trajectory segment is a sequence of states and actions, $\sigma = ((s_0, a_0), (s_1, a_1), ..., (s_{k-1}, a_{k-1})) \in (S \times A)^k$. Given two trajectory segments σ^0 and σ^1, $\sigma^0 \succ \sigma^1$ indicates that the human prefers trajectory segment σ^0 over trajectory segment σ^1. We assume that $\sigma^0 \succ \sigma^1$ if the sum of rewards of trajectory segments satisfies $\sum_{i=0}^{k-1} r(s_i^0, a_i^0) > \sum_{i=0}^{k-1} r(s_i^1, a_i^1)$. The human preference is stored as a triplet $(\sigma^0, \sigma^1, \mu)$ in a dataset $\mathcal{D}$, where $\mu \in \{0, 1\}$, with $\mu = 0$ indicating σ^0 is preferred and $\mu = 1$ indicating σ^1 is preferred. Following the Bradley-Terry model [4], the reward function estimator $\hat{r}$ predicts the reward value for each state-action pair in both σ^0 and σ^1, and the probability of $\sigma^0 \succ \sigma^1$ is modeled as depending exponentially on the total reward over each trajectory segment. This can be formalized as:

$$\hat{P}[\sigma^0 \succ \sigma^1] = \frac{\exp \sum_{i=0}^{k-1} \hat{r}(s_i^0, a_i^0)}{\exp \sum_{i=0}^{k-1} \hat{r}(s_i^0, a_i^0) + \exp \sum_{i=0}^{k-1} \hat{r}(s_i^1, a_i^1)} \quad (1)$$

This formulation reflects the intuition that the probability of preferring a trajectory segment increases exponentially with the cumulative reward over the segment, as in [6]. The higher the total reward for a segment, the more likely it is to be preferred over another segment with a lower total reward. The reward model is trained using a binary classification loss, where the objective is to match real human preferences with the model's predicted preferences.

The loss function is defined as:

$$\mathcal{L}_{pref} = - \mathop{\mathbb{E}}_{(\sigma^0,\sigma^1,\mu)\in\mathcal{D}} [\mu(0) \log \hat{P}[\sigma^0 \succ \sigma^1] + \mu(1) \log \hat{P}[\sigma^1 \succ \sigma^0]] \tag{2}$$

3.3 PEBBLE

PEBBLE [25] is a state-of-the-art preference-based RL algorithm that uses human preferences to train an agent to solve complex RL tasks without access to the reward function. It leverages unsupervised pre-training and off-policy learning to improve both sample and feedback efficiency for preference-based RL.

Unsupervised Pre-training. PEBBLE employs unsupervised pre-training to improve the efficiency of the human teacher's initial feedback. Through pre-training with unsupervised exploration, the agent is encouraged to explore a wide range of states and learn diverse behaviors. Compared to the random and limited trajectory samples generated by a random policy, the diverse trajectories generated through unsupervised pre-training can enable humans to provide more meaningful and informative initial feedback.

During unsupervised pre-training, PEBBLE uses state entropy as the intrinsic reward to motivate the agent to explore and collect diverse experiences. The entropy $\mathcal{H}(\mathbf{s}) = -\mathbb{E}_{\mathbf{s}\sim p(\mathbf{s})}[\log p(\mathbf{s})]$ encourages the agent to explore less-visited states by rewarding it for discovering states that it has not frequently encountered [12, 25, 26, 28, 39]. However, calculating state entropy directly is often intractable, PEBBLE thus approximates entropy using a particle-based estimator based on k-nearest neighbors (k-NN) [3, 41]. The intrinsic reward is defined as [25]:

$$r^{int}(\mathbf{s}_t) = \log(||\mathbf{s}_t - \mathbf{s}_t^k||), \tag{3}$$

where the reward is based on the distance between the current state $\mathbf{s}_t$ and its k-th nearest neighbor $\mathbf{s}_t^k$. By maximizing the distance between states and their nearest neighbors, the agent is encouraged to explore less-visited regions of the environment.

Off-Policy Learning and Reward Relabeling. After unsupervised pre-training, PEBBLE uses an off-policy RL algorithm to enable the agent to reuse previously collected experiences to achieve more efficient learning. In particular, it adopts Soft Actor-Critic (SAC)[11] as the underlying RL algorithm. During training, the agent interacts with the environment and collects new experiences, which are stored in the replay buffer. Human feedback is then provided in the form of binary preferences between pairs of trajectory segments, which are sampled from the replay buffer. These human preferences are used to train a reward model, which then guides the agent's learning process. An essential aspect of PEBBLE is its reward relabeling mechanism: each time the reward model is updated with new human feedback, all experiences in the replay buffer are relabeled using the updated reward model. This ensures that the agent learns stably from data that more accurately reflects human preferences, leading to improved performance.

3.4 Diffusion Model

Diffusion models [7, 13, 32, 42] are a class of generative models that learn to generate complex, realistic data by gradually transforming simple structured noise into more intricate patterns. These models consist of two key processes: *forward process* and *reverse process.*

In the *forward process*, given a data distribution $q(\mathbf{x}^{(0)})$ (representing the original, uncorrupted data), the data is progressively corrupted by adding Gaussian noise over multiple steps. At each time step t, a noised version $\mathbf{x}^{(t)}$ is generated from the previous step $\mathbf{x}^{(t-1)}$ by applying a small amount of noise. The corrupted data at each step becomes increasingly noisy until it approximates a simple noise distribution. This forward process is modeled as:

$$q(\mathbf{x}^{(t)}|\mathbf{x}^{(t-1)}) = \mathcal{N}(\mathbf{x}^{(t)}; \sqrt{1-\beta_t}\mathbf{x}^{(t-1)}, \beta_t\mathbf{I}), t \in \{1, ..., T\} \tag{4}$$

$$q(\mathbf{x}^{(0:T)}) = q(\mathbf{x}^{(0)}) \prod_{t=1}^{T} q(\mathbf{x}^{(t)}|\mathbf{x}^{(t-1)}) \tag{5}$$

where β_t is a variance schedule controlling the amount of noise added at each step, and $\mathcal{N}$ denotes a normal distribution with mean $\sqrt{1-\beta_t}\mathbf{x}^{(t-1)}$ and variance $\beta_t\mathbf{I}$ that produces $\mathbf{x}^{(t)}$. The forward process results in a sequence of increasingly noisy versions of the data, $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, ..., \mathbf{x}^{(T)}$, with the final step $\mathbf{x}^{(T)}$ being almost indistinguishable from pure noise.

In the *reverse process*, the model is able to generate new samples from the data distribution $q(\mathbf{x}^{(0)})$ by starting with a sample $x^{(T)} \sim \mathcal{N}(0, \mathbf{I})$ (a standard normal distribution). The reverse process progressively removes noise from this initial noisy sample, step by step, following a learned distribution

$$p_\theta(\mathbf{x}^{(t-1)}|\mathbf{x}^{(t)}) = \mathcal{N}(\mathbf{x}^{(t-1)}; \boldsymbol{\mu}_\theta(\mathbf{x}^{(t)}, t), \Sigma_\theta(\mathbf{x}^{(t),t})) \tag{6}$$

$$p_\theta(\mathbf{x}^{(0:T)}) = p(\mathbf{x}^{(T)}) \prod_{t=1}^{T} p_\theta(\mathbf{x}^{(t-1)}|\mathbf{x}^{(t)}) \tag{7}$$

where θ represents the parameters of the model, which predict the mean $\boldsymbol{\mu}_\theta(\mathbf{x}^{(t)}, t)$ and the covariance $\Sigma_\theta(\mathbf{x}^{(t),t})$. This reverse process gradually denoises $\mathbf{x}^{(t)}$, reconstructing the original data as it progresses through the time steps from T to 0.

The neural network of the model is trained by minimizing the following loss function:

$$\begin{aligned}\mathcal{L}_{diff} = &-\log p_\theta(\mathbf{x}^{(0)}|\mathbf{x}^{(1)}) + KL(q(\mathbf{x}^{(T)}|\mathbf{x}^{(0)})||p(\mathbf{x}^{(T)}) \\ &+ \sum_{t>1} KL(q(\mathbf{x}^{(t-1)}|\mathbf{x}^{(t)}, \mathbf{x}^{(0)})||p_\theta(\mathbf{x}^{(t-1)}|\mathbf{x}^{(t)})\end{aligned} \tag{8}$$

where KL represents the Kullback-Leibler divergence between two probability distributions.The first term maximizes the likelihood of reconstructing the original data, while the KL divergence terms ensure that the generated data matches the true data distribution at each time step.

4 METHODOLOGY

In this section, we propose a new preference-based RL approach called ***P**reference-based **R**einforcement learning using d**I**ffusion mo**DE**l* (PRIDE). Our method builds upon PEBBLE [25], which uses unsupervised pre-training and off-policy learning to learn from human feedback. However, by integrating a diffusion generative model into preference-based RL training, our approach achieves more efficient learning with significantly less human feedback.

Algorithm 1 PRIDE

```
Require: the ratio $u \in [0,1]$ of synthetic to real data, diffusion model training interval $R$, human feedback interval $M$, number of queries $K$ per feedback session, maximum feedback queries $Q$
Initialize real replay buffer $\mathcal{B}_{real} \leftarrow \emptyset$, synthetic replay buffer $\mathcal{B}_{syn} \leftarrow \emptyset$, a dataset of preferences $\mathcal{D} \leftarrow \emptyset$, agent policy $\pi$, reward model $\hat{r}$, diffusion model $F$, and the number of feedback collected $T \leftarrow 0$
// Unsupervised pre-training phase
pre-train policy $\pi$ by using intrinsic reward $r^{int}$
for each timestep do
    // Reward learning phase
    if $timestep \ \% \ M == 0$ and $T < Q$ then
        // Determine the number of feedback queries to collect
        $queries\_to\_collect \leftarrow \min(K, Q - T)$
        for $k = 1$ to $queries\_to\_collect$ do
            Sample $(\sigma^0, \sigma^1)$ from $\mathcal{B}_{real}$
            Query human for a preference label $\mu$
            $\mathcal{D} \leftarrow \mathcal{D} \cup \{(\sigma^0, \sigma^1, \mu)\}$
        end for
        $T \leftarrow T + queries\_to\_collect$
        for each gradient step do
            Sample a minibatch from $\mathcal{D}$
            Update reward model $\hat{r}$ by optimizing $\mathcal{L}_{pref}$
        end for
        // Replay buffer relabeling phase
        Relabel all data in $\mathcal{B}_{real}$ with reward values using $\hat{r}$
    end if
    // Diffusion model training phase
    if timestep % R == 0 then
        for each gradient step do
            Sample a minibatch from $\mathcal{B}_{real}$
            Update diffusion model $F$ by optimizing $\mathcal{L}_{diff}$
        end for
        // Synthetic data generation phase
        Generate synthetic samples from diffusion model $F$ and add them to $\mathcal{B}_{syn}$
    end if
    Gather data $d$ using policy $\pi$ in the environment
    $\mathcal{B}_{real} = \mathcal{B}_{real} \cup d$
    // Agent learning phase
    for each gradient step do
        Sample data from $\mathcal{B}_{real}$ and $\mathcal{B}_{syn}$ according to ratio $u$
        Train policy $\pi$ on the sampled data
    end for
end for
```

While PEBBLE has shown promising results in learning complex behaviors that are difficult to specify with standard reward functions, its reliance on reusing prior agent experiences stored in the replay buffer limits its generalization capability. The data collected through agent-environment interactions during training may not always cover a sufficiently diverse set of agent behaviors. Suitable data for specific agent behaviors might simply be unavailable, which can hinder exploration and reduce learning efficiency.

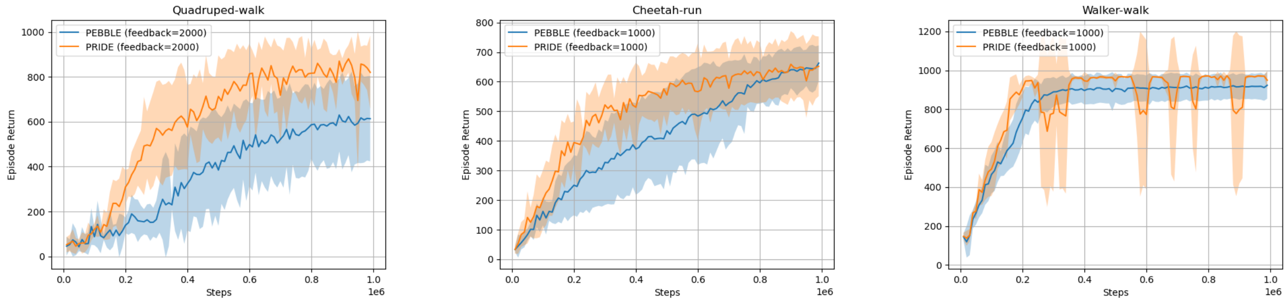
To address this issue, inspired by the recent success of using diffusion models to augment traditional datasets in RL [32], we propose to integrate diffusion models into the training process of preference-based RL to improve feedback and sample efficiency. In our approach, unsupervised pre-training enables the agent to accumulate valuable experiences in the replay buffer, which can later be used to train the diffusion model and provide diverse synthetic experiences for the agent during policy learning. Specifically, in our approach PRIDE, the diffusion model operates in two phases:

- **Model Training Phase**: Periodically, the diffusion model is trained to approximate online behavioral distribution as the agent interacts with the environment and gathers new experiences, using batches of experiences sampled from the real replay buffer. The model is optimized using the loss function shown in Eq. (8) to learn a distribution over behavioral experiences of agent, capturing the underlying dynamics of the environment and the agent's behavior.
- **Synthetic Data Generation Phase:** Following each training phase, the diffusion model generates a large set of synthetic experience data, which is diverse, novel, and dynamically accurate. The generated synthetic experiences are then stored in a separate synthetic replay buffer. These synthetic experiences simulate interactions between the agent and the environment, and are used to augment limited real data as if they were real experiences during policy training.

When training the RL agent, PRIDE samples from both the real and synthetic replay buffers to create a mixture of real and synthetic data. A predefined ratio of synthetic to real data is maintained throughout training to ensure that the agent benefits from both authentic and diverse synthetic experiences, which allows the agent to explore more efficiently. Compared to relying solely on limited real data, using a mixture of real and synthetic data increases the diversity of training data available to the agent, leading to improved sample efficiency and reducing the need for excessive human feedback. The novel and diverse synthetic data also enables our approach to achieve better generalization while minimizing dependence on real data.

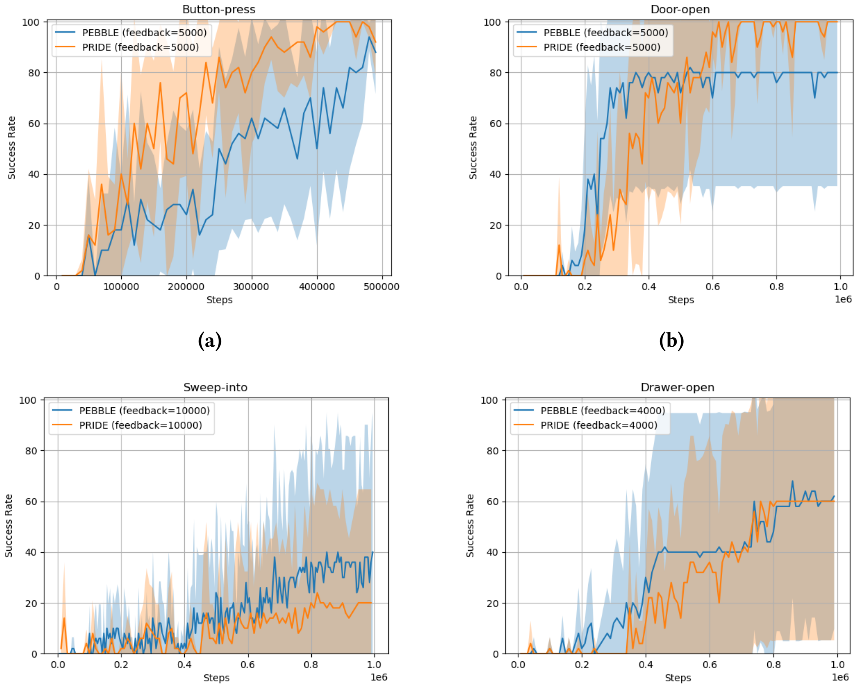
Our approach PRIDE consists of the following key steps:

- *Step 0 (unsupervised pre-training)*: We pre-train the policy π using intrinsic rewards to encourage the agent to explore a wide range of states within the environment.
- *Step 1 (reward learning)*: Human feedback, in the form of preferences between pairs of trajectory segments, is collected and used to train a reward model $\hat{r}$, which predicts human preferencees for different agent behaviors.
- *Step 2 (replay buffer relabeling)*: The current reward model $\hat{r}$ is used to relabel the rewards of all agent experiences in the real replay buffer $\mathcal{B}_{real}$.
- *Step 3 (diffusion model training)*: The diffusion model is trained using batches of experiences sampled from the real replay buffer $\mathcal{B}_{real}$ to approximate the agent's online behavioral distribution.
- *Step 4 (synthetic data generation)*: The trained diffusion model generates a large set of new and diverse synthetic experiences, which are then stored in a separate synthetic replay buffer $\mathcal{B}_{syn}$.



(a) **(b)** **(c)**

Figure 2: Mean episode return for PRIDE and PEBBLE on different locomotion tasks from the DMControl Suite. The mean across 5 random seeds is plotted and the standard deviation is shown shaded.



(a) **(b)**

(c) **(d)**

Figure 3: Success rate for PRIDE and PEBBLE on different robotic manipulation tasks from the Meta-world. The mean across 5 random seeds is plotted and the standard deviation is shown shaded.

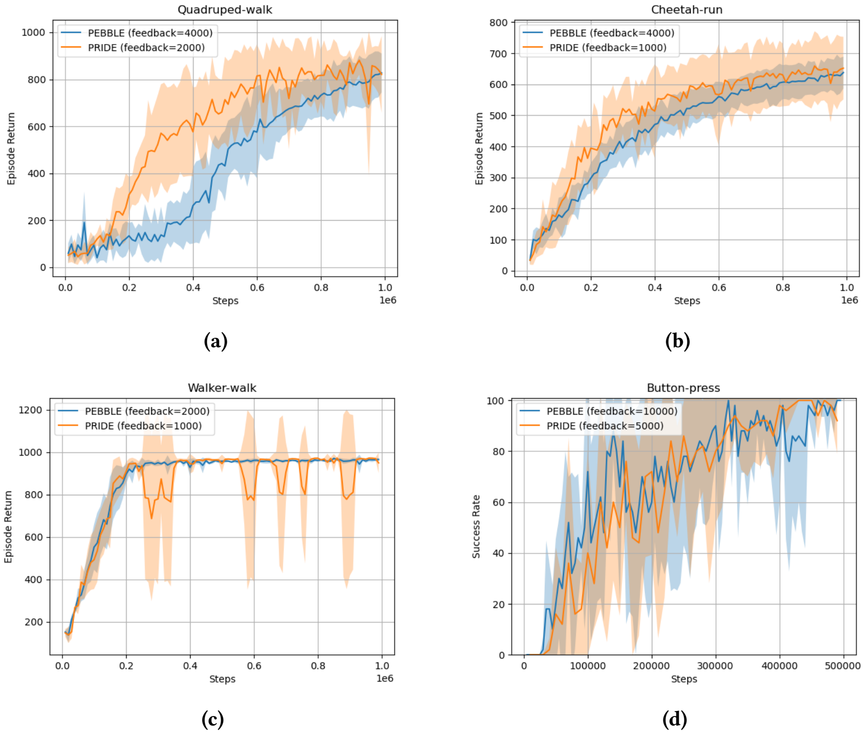
- *Step 5 (agent learning)*: The agent's policy π is updated using an off-policy RL algorithm, which samples experiences from both the real replay buffer $\mathcal{B}_{real}$ and the synthetic replay buffer $\mathcal{B}_{syn}$.
- Repeat *Step 1* to *Step 5*.

By training a diffusion model, PRIDE generates synthetic experiences that mimic online agent-environment interactions, but with added variability. The large number of new and diverse synthetic experiences enriches the replay buffer, broadening the training data available to the agent. This allows the agent to be trained with a much higher update-to-data ratio and thus learn more efficiently. The full procedure of PRIDE is summarized in Algorithm 1.

5 EXPERIMENTS

We design our experiments to investigate the following questions:

- Does the use of a diffusion model in preference-based RL improve sample efficiency?
- How does PRIDE compare to PEBBLE, after integrating diffusion models into the training process, in terms of overall performance?



(a) (b) (c) (d)

Figure 4: Mean episode return for PRIDE and PEBBLE under conditions where PEBBLE uses significantly more human feedback. The mean across 5 random seeds is plotted and the standard deviation is shown shaded.

- What is the impact of using synthetic replay experiences in preference-based RL on feedback efficiency?

We would like to highlight that, in our experiments, we only compare PRIDE against PEBBLE since our method is built upon PEBBLE and our aim is to demonstrate that integrating diffusion models into the preference-based RL training process can yield superior performance compared to PEBBLE.

5.1 Experimental Setups

We conduct experiments on a variety of complex locomotion and robotic manipulation tasks from two benchmarks: the DeepMind Control Suite (DMControl) [43, 44] and Meta-world [50]. Both benchmarks are commonly used for evaluating preference-based RL algorithms.

DeepMind Control Suite: DMControl provides a collection of continuous control tasks designed to simulate real-world physics, focusing on both locomotion and robotic manipulation. These tasks serve as a benchmark for evaluating the learning efficiency of RL algorithms. Each task involves controlling a physical system, such as making a quadruped walk or manipulating a robotic arm to perform precise tasks like item placement. The tasks in DMControl can be used to test an agent's ability to learn and adapt to complex, dynamic environments, making it a crucial tool for developing and evaluating RL methods.

Meta-world: Meta-World is a widely-used benchmark for evaluating robotic manipulation tasks in RL. It consists of a variety of challenging tasks that require precise control and coordination of a robotic arm to interact with different objects. These tasks simulate real-world environments, such as pressing buttons, opening doors, and moving objects, making Meta-World a valuable platform for testing the generalization and performance of learning algorithms in high-dimensional, real-world-like settings.

In our experiments, following prior works [6, 15, 25], human feedback is provided by a scripted teacher, who provides preferences between pairs of agent trajectory segments according to the ground truth reward function. For all experiments, we present the mean and standard deviation across five runs.

5.2 Hyperparameter Settings

In our experiments, we compare PRIDE against PEBBLE [25], using the same underlying RL algorithm (SAC) and hyperparameter settings as in the original paper. [1] One key hyperparameter in PEBBLE is the total amount of *human feedback* collected. Human feedback is provided in batches at regular timesteps, and the reward model is updated after each batch until the maximum number of feedback queries is reached.

The main distinction in PRIDE lies in the introduction of two additional hyperparameters: *synthetic-to-real sampling ratio* and *diffusion model retraining interval*. The *synthetic-to-real sampling ratio* determines the proportion of synthetic experiences to real experiences used when training the agent, while the *diffusion model*

[1] We adopt the same batch sizes in SAC as used in PEBBLE: a batch size of 1024 for DMControl tasks and 512 for Meta-World tasks.

retraining interval defines the number of timesteps after which the diffusion model is retrained to generate new synthetic experiences.

5.3 Main Results

Figure 2 shows the mean episode return for PRIDE and PEBBLE on the Quadruped-walk, Cheetah-run, and Walker-walk tasks, with 2000 human feedback queries for the Quadruped-walk task and 1000 human feedback queries for the Walker-walk and Cheetah-run tasks. In each task, PEBBLE and PRIDE use the same number of feedback queries for fair comparison. We can see that PRIDE outperforms PEBBLE on all three locomotion tasks. In both Quadruped-walk and Walker-walk, PRIDE performs better than PEBBLE both in terms of absolute performance and learning speed, demonstrating the advantages of using the diverse synthetic data generated by the diffusion model. In the Cheetah-run task, PRIDE achieves similar asymptotic performance to PEBBLE but is more sample efficient.

Figure 3 shows the success rate for PRIDE and PEBBLE on the Button Press, Door Open, Sweep Into, and Drawer-open tasks from Meta-world. The results show that PRIDE demonstrates faster learning in some tasks compared to PEBBLE. In Button Press, PRIDE converges significantly faster than PEBBLE, reaching a higher success rate much earlier in training. Similarly, in Door Open, compared to PEBBLE, PRIDE achieves a higher success rate (close to 100%) with lower variance across seeds. In other two tasks, both PRIDE and PEBBLE achieve similar performance with high variance.

5.4 Feedback Efficiency Analysis

To further evaluate the feedback efficiency of our approach, we compare the performance of PRIDE to PEBBLE under conditions where PEBBLE receives significantly more human feedback. As shown in Figure 4, PRIDE achieves comparable or superior performance to PEBBLE with significantly fewer human feedback queries on all four tasks tested. This demonstrates the ability of PRIDE to reduce reliance on human feedback by leveraging the novel and diverse synthetic experiences generated by the diffusion model. In Quadruped-walk, PRIDE converges significantly faster than PEBBLE, achieving similar asymptotic performance. In Cheetah-run, PRIDE requires only 25% of the human feedback compared to PEBBLE while still achieving similar final performance. Our experimental results show that PRIDE not only matches but can surpass PEBBLE in terms of final performance, even when using at least 50% less human feedback. This validates both the sample efficiency and feedback efficiency of our method, and its potential in significantly reducing the burden of human supervision in preference-based RL.

5.5 The Effect of the Synthetic-to-Real Sampling Ratio

To further demonstrate the benefits of leveraging synthetic experiences in training the agent with PRIDE, we evaluate the performance of PRIDE on the Quadruped-walk task with varying synthetic-to-real sampling ratio r. For instance, a sampling ratio of $r = 0.3$ indicates that 30% of the training experiences are sampled from the synthetic replay buffer, while 70% are sampled from the real replay buffer. As shown in Figure 5, PRIDE is sensitive to the hyperparameter r. Sampling a small percentage of synthetic experiences ($r = 0.3$) or relying entirely on the synthetic experiences ($r = 1$) both result in poor absolute performance and slow learning speed. There exists some intermediate value ($r = 0.5$) that provides the best trade-off, demonstrating the importance of utilising both real and synthetic data with the right balance to maximize learning efficiency in PRIDE. If the sampling ratio is too low, the additional synthetic data might not significantly improve the diversity of the training data, thereby limiting its impact on learning performance. On the other hand, if the sampling ratio is too high, the algorithm may overfit to the synthetic data, some of which may deviate from the real data.

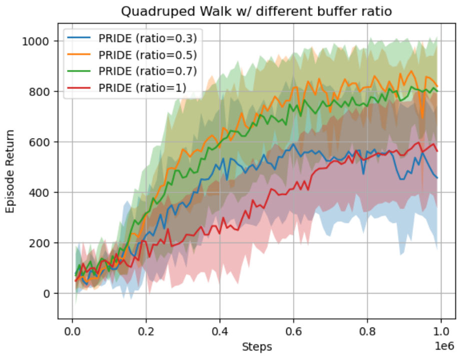


Figure 5: Mean episode return for PRIDE on Quadruped-walk with different synthetic-to-real sampling ratio values. The mean across 5 random seeds is plotted and the standard deviation is shown shaded.

6 CONCLUSION

In this work, we proposed PRIDE, a novel preference-based RL approach that integrates diffusion models into the training process to improve both sample and feedback efficiency. PRIDE continually trains a diffusion model to generate a large number of synthetic experiences that mimic real agent-environment interactions. These synthetic experiences are novel and diverse, which can be used to augment the limited real agent experiences to broaden the training data available to the agent, leading to improved sample efficiency and final performance.

Through extensive experiments on locomotion tasks from the DeepMind Control Suite and robotic manipulation tasks from Meta-world, we demonstrated that PRIDE outperforms PEBBLE in most of the scenarios tested. Furthermore, PRIDE achieves comparable or superior performance with at least 50% less human feedback compared to PEBBLE in some tasks, demonstrating the benefits of using diffusion models to reduce the amount of human effort required for efficient learning. Our integration of diffusion models into preference-based RL opens up new possibilities for human-in-the-loop RL research, offering a promising direction for future work on learning from limited human feedback.

For future work, we aim to explore strategies to better trade-off between learning performance and the cost of continually training the diffusion model. Additionally, we will develop an approach to dynamically adjust the synthetic-to-real sampling ratio based on the agent's learning progress to further improve performance.

ACKNOWLEDGMENTS

We would like to thank the UK-Ukraine Research & Innovation Twinning Grant Scheme for supporting Alona Tenytska's research visit to the University of Liverpool, during which this work was conducted.

REFERENCES

[1] OpenAI: Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Józefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, Jonas Schneider, Szymon Sidor, Josh Tobin, Peter Welinder, Lilian Weng, and Wojciech Zaremba. 2020. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research* 39, 1 (2020), 3–20.

[2] Matthew Baucum, Anahita Khojandi, and Rama Vasudevan. 2020. Improving deep reinforcement learning with transitional variational autoencoders: A healthcare application. *IEEE Journal of Biomedical and Health Informatics* 25, 6 (2020), 2273–2280.

[3] Jan Beirlant, Edward J Dudewicz, László Györfi, Edward C Van der Meulen, et al. 1997. Nonparametric entropy estimation: An overview. *International Journal of Mathematical and Statistical Sciences* 6, 1 (1997), 17–39.

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

[5] Jie Cheng, Gang Xiong, Xingyuan Dai, Qinghai Miao, Yisheng Lv, and Fei-Yue Wang. 2024. RIME: Robust Preference-based Reinforcement Learning with Noisy Preferences. In *International Conference on Machine Learning*.

[6] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, Vol. 30.

[7] Florinel-Alin Croitoru, Vlad Hondru, Radu Tudor Ionescu, and Mubarak Shah. 2023. Diffusion models in vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45, 9 (2023), 10850–10869.

[8] Rong Gao, Haifeng Xia, Jing Li, Donghua Liu, Shuai Chen, and Gang Chun. 2019. DRCGR: Deep reinforcement learning framework incorporating CNN and GAN-based for interactive recommendation. In *2019 IEEE international conference on data mining (ICDM)*. IEEE, 1048–1053.

[9] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2020. Generative adversarial networks. *Commun. ACM* 63, 11 (2020), 139–144.

[10] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. *Advances in Neural Information Processing Systems* 27.

[11] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. 2018. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*. Pmlr, 1861–1870.

[12] Elad Hazan, Sham Kakade, Karan Singh, and Abby Van Soest. 2019. Provably efficient maximum entropy exploration. In *International Conference on Machine Learning*. 2681–2691.

[13] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, Vol. 33. 6840–6851.

[14] Simon Holk, Daniel Marta, and Iolanda Leite. 2024. POLITE: Preferences Combined with Highlights in Reinforcement Learning. In *2024 IEEE International Conference on Robotics and Automation*. IEEE, 2288–2295.

[15] Borja Ibarz, Jan Leike, Tobias Pohlen, Geoffrey Irving, Shane Legg, and Dario Amodei. 2018. Reward learning from human preferences and demonstrations in atari. In *Advances in Neural Information Processing Systems*, Vol. 31.

[16] Charles Isbell, Christian R Shelton, Michael Kearns, Satinder Singh, and Peter Stone. 2001. A social reinforcement learning agent. In *International Conference on Autonomous Agents*.

[17] Michael Janner, Yilun Du, Joshua B Tenenbaum, and Sergey Levine. 2022. Planning with diffusion for flexible behavior synthesis. *International Conference on Machine Learning*.

[18] Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, et al. 2018. Scalable deep reinforcement learning for vision-based robotic manipulation. In *Conference on Robot Learning*. 651–673.

[19] Diederik P Kingma and Max Welling. 2014. Auto-encoding variational bayes. In *International Conference on Learning Representations*.

[20] Diederik P Kingma, Max Welling, et al. 2019. An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning* 12, 4 (2019), 307–392.

[21] W Bradley Knox and Peter Stone. 2009. Interactively shaping agents via human reinforcement: The TAMER framework. In *Proceedings of the International Conference on Knowledge Capture*. 9–16.

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

[23] Jens Kober and Jan Peters. 2011. Policy search for motor primitives in robotics: Special Issue on Empirical Evaluations in Reinforcement Learning. *Machine Learning* 84, 1–2 (2011), 171–203.

[24] Shir Kozlovsky, Elad Newman, and Miriam Zacksenhouse. 2022. Reinforcement Learning of Impedance Policies for Peg-in-Hole Tasks: Role of Asymmetric Matrices. *IEEE Robotics and Automation Letters* 7, 4 (2022), 10898–10905.

[25] Kimin Lee, Laura Smith, and Pieter Abbeel. 2021. Pebble: Feedback-efficient interactive reinforcement learning via relabeling experience and unsupervised pre-training. In *International Conference on Machine Learning*.

[26] Lisa Lee, Benjamin Eysenbach, Emilio Parisotto, Eric Xing, Sergey Levine, and Ruslan Salakhutdinov. 2019. Efficient exploration via state marginal matching. *arXiv preprint arXiv:1906.05274* (2019).

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

[28] Hao Liu and Pieter Abbeel. 2021. Behavior from the void: Unsupervised active pre-training. In *Advances in Neural Information Processing Systems*, Vol. 34. 18459–18473.

[29] Robert Loftin, James MacGlashan, Bei Peng, Matthew Taylor, Michael Littman, Jeff Huang, and David Roberts. 2014. A strategy-aware technique for learning behaviors from discrete human feedback. In *AAAI Conference on Artificial Intelligence*.

[30] Robert Loftin, Bei Peng, James MacGlashan, Michael L Littman, Matthew E Taylor, Jeff Huang, and David L Roberts. 2015. Learning behaviors via human-delivered discrete feedback: modeling implicit feedback strategies to speed up learning. *Autonomous Agents and Multi-agent Systems* 30 (2015), 30–59.

[31] Robert Loftin, Bei Peng, James MacGlashan, Michael L Littman, Matthew E Taylor, Jeff Huang, and David L Roberts. 2016. Learning behaviors via human-delivered discrete feedback: modeling implicit feedback strategies to speed up learning. *Autonomous Agents and Multi-Agent Systems* (2016).

[32] Cong Lu, Philip Ball, Yee Whye Teh, and Jack Parker-Holder. 2023. Synthetic experience replay. In *Advances in Neural Information Processing Systems*, Vol. 36.

[33] James MacGlashan, Mark K Ho, Robert Loftin, Bei Peng, Guan Wang, David L Roberts, Matthew E Taylor, and Michael L Littman. 2017. Interactive learning from policy-dependent human feedback. In *International Conference on Machine Learning*.

[34] Azalia Mirhoseini, Anna Goldie, Mustafa Yazgan, Joe Wenjie Jiang, Ebrahim Songhori, Shen Wang, Young-Joon Lee, Eric Johnson, Omkar Pathak, Azade Nazi, et al. 2021. A graph placement methodology for fast chip design. *Nature* 594, 7862 (2021), 207–212.

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

[36] Andrew Y Ng, Stuart Russell, et al. 2000. Algorithms for inverse reinforcement learning.. In *International Conference on Machine Learning*, Vol. 1. 2.

[37] Fei Ni, Jianye Hao, Yao Mu, Yifu Yuan, Yan Zheng, Bin Wang, and Zhixuan Liang. 2023. Metadiffuser: Diffusion model as conditional planner for offline meta-rl. In *International Conference on Machine Learning*. PMLR, 26087–26105.

[38] Jongjin Park, Younggyo Seo, Jinwoo Shin, Honglak Lee, Pieter Abbeel, and Kimin Lee. 2022. SURF: Semi-supervised reward learning with data augmentation for feedback-efficient preference-based reinforcement learning. In *International Conference on Learning Representations*.

[39] Younggyo Seo, Lili Chen, Jinwoo Shin, Honglak Lee, Pieter Abbeel, and Kimin Lee. 2021. State entropy maximization with random encoders for efficient exploration. In *International Conference on Machine Learning*. 9443–9454.

[40] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. 2016. Mastering the game of Go with deep neural networks and tree search. *Nature* 529, 7587 (2016), 484–489.

[41] Harshinder Singh, Neeraj Misra, Vladimir Hnizdo, Adam Fedorowicz, and Eugene Demchuk. 2003. Nearest neighbor estimates of entropy. *American Journal of Mathematical and Management Sciences* 23, 3-4 (2003), 301–321.

[42] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. 2015. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*. 2256–2265.

[43] Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yuval Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, et al. 2018. DeepMind Control Suite. *arXiv preprint arXiv:1801.00690* (2018).

[44] Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yuval Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, et al. 2020. dm_control: Software and Tasks for Continuous Control. *arXiv preprint arXiv:2006.12983* (2020).

[45] Ana C Tenorio-Gonzalez, Eduardo F Morales, and Luis Villasenor-Pineda. 2010. Dynamic reward shaping: training a robot by voice. In *Advances in Artificial Intelligence–IBERAMIA 2010: 12th Ibero-American Conference on AI*.

[46] Andrea Lockerd Thomaz, Cynthia Breazeal, et al. 2006. Reinforcement learning with human teachers: Evidence of feedback and guidance with implications

for learning performance. In *AAAI Conference on Artificial Intelligence*, Vol. 6. 1000–1005.

[47] Zhendong Wang, Jonathan J Hunt, and Mingyuan Zhou. 2023. Diffusion policies as an expressive policy class for offline reinforcement learning. *International Conference on Learning Representations*.

[48] Garrett Warnell, Nicholas Waytowich, Vernon Lawhern, and Peter Stone. 2018. Deep tamer: Interactive agent shaping in high-dimensional state spaces. In *AAAI conference on artificial intelligence*, Vol. 32.

[49] Quantao Yang, Alexander Dürr, Elin Anna Topp, Johannes A. Stork, and Todor Stoyanov. 2022. Variable Impedance Skill Learning for Contact-Rich Manipulation. *IEEE Robotics and Automation Letters* 7, 3 (2022), 8391–8398.

[50] Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. 2021. Meta-World: A Benchmark and Evaluation for Multi-Task and Meta Reinforcement Learning. *arXiv preprint arXiv:1910.10897* (2021).

[51] Huixin Zhan, Feng Tao, and Yongcan Cao. 2021. Human-guided robot behavior learning: A gan-assisted preference-based reinforcement learning approach. *IEEE Robotics and Automation Letters* 6, 2 (2021), 3545–3552.

[52] Tengteng Zhang and Hongwei Mo. 2021. Reinforcement learning for robot research: A comprehensive review and open issues. *International Journal of Advanced Robotic Systems* 18, 3 (2021).

[53] Zhengbang Zhu, Hanye Zhao, Haoran He, Yichao Zhong, Shenyu Zhang, Haoquan Guo, Tingting Chen, and Weinan Zhang. 2023. Diffusion models for reinforcement learning: A survey. *arXiv preprint arXiv:2311.01223* (2023).