Beyond Reward: Offline Preference-guided Policy Optimization

Yachen Kang [1 2] Diyuan Shi [2] Jinxin Liu [2] Li He [2] Donglin Wang [2]

Abstract

This study focuses on the topic of offline preference-based reinforcement learning (PbRL), a variant of conventional reinforcement learning that dispenses with the need for online interaction or specification of reward functions. Instead, the agent is provided with fixed offline trajectories and human preferences between pairs of trajectories to extract the dynamics and task information, respectively. Since the dynamics and task information are orthogonal, a naive approach would involve using preference-based reward learning followed by an off-the-shelf offline RL algorithm. However, this requires the separate learning of a scalar reward function, which is assumed to be an information bottleneck of the learning process. To address this issue, we propose the offline preference-guided policy optimization (OPPO) paradigm, which models offline trajectories and preferences in a one-step process, eliminating the need for separately learning a reward function. OPPO achieves this by introducing an offline hindsight information matching objective for optimizing a contextual policy and a preference modeling objective for finding the optimal context. OPPO further integrates a well-performing decision policy by optimizing the two objectives iteratively. Our empirical results demonstrate that OPPO effectively models offline preferences and outperforms prior competing baselines, including offline RL algorithms performed over either true or pseudo reward function specifications. Our code is available on the project website: https://sites.google.com/view/oppo-icml-2023.

1. Introduction

Deep reinforcement learning (RL) offers a versatile framework for acquiring task-oriented behaviors, as evidenced by a growing body of literature (Kohl & Stone, 2004; Kober & Peters, 2008; Kober et al., 2013; Silver et al., 2017; Kalashnikov et al., 2018; Vinyals et al., 2019). In this framework, the ”task” is frequently expressed as maximizing the cumulative reward of trajectories produced by deploying the learning policy in the corresponding environment. However, the above RL formulation presupposes two critical conditions for decision policy training: 1) an interactable environment, and 2) a pre-specified reward function. Regrettably, online interactions with the environment can be both expensive and hazardous (Mihatsch & Neuneier, 2002; Hans et al., 2008; Garcıa & Fernández, 2015), while developing a suitable reward function typically necessitates considerable human effort. Additionally, the heuristic rewards often employed may be insufficient to express the true intent (Hadfield-Menell et al., 2017).

To address these challenges, prior research has explored two approaches. First, some works have focused on the offline RL formulation (Fujimoto et al., 2019), where the learner has access to fixed offline trajectories along with a reward signal for each transition (or limited expert demonstrations). Second, others have considered the (online) preference-based RL formulation, where the task objective is conveyed to the learner through preferences of a human annotator between two trajectories rather than rewards for each transition. In pursuit of further advancements in this setting, we propose a novel approach that relaxes both of these requirements and advocates for offline preference-based RL (PbRL).

In the context of offline preference-based reinforcement learning (PbRL), where access to an offline dataset and labeled preferences between the offline trajectories is available, a common approach is to combine previous online PbRL methods with off-the-shelf offline RL algorithms (Shin & Brown, 2021). This two-step strategy, as illustrated in Fig.1 (left), typically involves training a reward function using the Bradley-Terry model (Bradley & Terry, 1952) in a supervised manner, followed by training the policy with any offline RL algorithm on the transitions relabeled via the learned reward function. However, the

[1] College of Computer Science and Technology, Zhejiang University, Hangzhou, Zhejiang, China [2] Machine Intelligence Lab (MiLAB) of the School of Engineering, Westlake University, Hangzhou, Zhejiang, China. Correspondence to: Donglin Wang <wangdonglin@westlake.edu.cn>, Yachen Kang <kangyachen@westlake.edu.cn>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

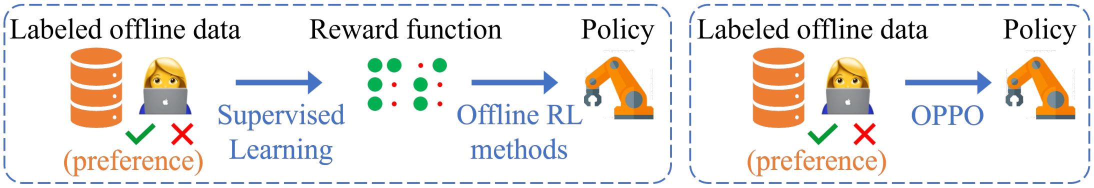


Figure 1. A flow diagram of previous offline PbRL algorithms (left) and our OPPO algorithm (right). Previous works require learning a separate reward function for modeling human preferences using the Bradley-Terry model. In contrast, our OPPO directly optimizes the policy network.

practice of separately learning a reward function that explains expert preferences may not directly instruct the policy on how to act optimally. This is because preference labels define the PbRL task, and the goal is to learn the most preferred trajectory by the annotator rather than to maximize the cumulative discounted proxy rewards of the policy rollouts. In cases of complex tasks, such as non-Markovian tasks, scalar rewards may create an information bottleneck in policy improvement, resulting in suboptimal behavior (Vamplew et al., 2022). Additionally, an isolated policy optimization may exploit loopholes in miscalibrated reward functions, leading to undesirable behaviors. Given these limitations, it is reasonable to question the necessity of learning a reward function, especially considering that it may not directly yield the optimal policy.

To achieve this objective, we present the offline preference-guided policy optimization (OPPO) approach, which is a one-step paradigm that simultaneously models offline preferences and learns the optimal decision policy without requiring the separate learning of a reward function (as illustrated in Figure 1 right). This is achieved through the use of two objectives: an offline hindsight information matching objective and a preference modeling objective. By iteratively optimizing these objectives, we derive a contextual policy $\pi(\mathbf{a}|\mathbf{s},\mathbf{z})$ to model the offline data and an optimal context $\mathbf{z}^*$ to model the preference. The main focus of OPPO is on both learning a high-dimensional $\mathbf{z}$-space and evaluating policies within such space. This high-dimensional $\mathbf{z}$-space captures more task-related information compared to scalar reward, making it ideal for policy optimization purposes. Furthermore, the optimal policy is obtained by conditioning the contextual policy $\pi(\mathbf{a}|\mathbf{s},\mathbf{z})$ on the learned optimal context $\mathbf{z}^*$.

Our main contribution can be summarized as follows. Firstly, we propose OPPO, a concise, stable, and one-step offline PbRL paradigm that avoids the need for separate reward function learning. Secondly, we present an instance of a preference-based hindsight information matching objective and a novel preference modeling objective over the context. Finally, extensive experiments are conducted to demonstrate the superiority of OPPO over previous competitive baselines and to analyze its performance.

2. Related Work

Since OPPO is at the intersection of Preference-Based Reinforcement Learning, Offline RL, and conditional RL we review the most relevant algorithms from these fields (see Table 1)

Online PbRL. Preference-based RL is also known as reinforcement learning from human feedback (RLHF). Several works have successfully utilized feedback from real humans to train RL agents (Arumugam et al., 2019; Christiano et al., 2017; Ibarz et al., 2018; Knox & Stone, 2009; Lee et al., 2021; Warnell et al., 2017). Christiano et al. (2017) scaled preference-based reinforcement learning to utilize modern deep learning techniques, and Ibarz et al. (2018) improved the efficiency of this method by introducing additional forms of feedback such as demonstrations. Recently, PEBBLE (Lee et al., 2021) proposed a feedback-efficient RL algorithm by utilizing off-policy learning and pre-training. SURF (Park et al., 2022) used pseudo-labeling to utilize unlabeled segments and proposed a novel data augmentation method called temporal cropping. All of the above methods require the agent to online interact with the environment, and they are all two-step strategies that require learning a scalar reward function separately.

Offline RL. To mitigate the impact of distribution shifts in offline RL, prior algorithms (a) constrain the action space (Fujimoto et al., 2019; Kumar et al., 2019a; Siegel et al., 2020; Zhuang et al., 2023), (b) incorporate value pessimism (Fujimoto et al., 2019; Kumar et al., 2020; Liu et al., 2022a), and (c) introduce pessimism into learned dynamics models (Kidambi et al., 2020; Yu et al., 2020). Another line of work explored learning a wide behavior distribution from the offline dataset by learning a task-agnostic set of skills, either with likelihood-based approaches (Ajay et al., 2020; Campos et al., 2020; Pertsch et al., 2020; Singh et al., 2020) or by maximizing mutual information (Eysenbach et al., 2018; Lu et al., 2020; Sharma et al., 2019). Some prior methods for RL is more similar to static supervised learning, such as Q-learning (Watkins, 1989; Mnih et al., 2013) and behavior cloning. In these mathods, the resulting agent's performance is positively correlated to the quality of data used for training. In addition to aforementioned RL methods, Srivastava et al. (2019) and Kumar et al. (2019b) studied

Table 1. A concise tabular representation of the differences between our method and related works.

	Method	**Supervised Signal**	**Training**	**Architectures**	**Learning a Separate Reward Function**
Imitation Learning	BC	Expert demonstration	Offline	MLP	×
Online PbRL	PEBBLE	Preference	Online	MLP	✓
	SURF	Preference	Online	MLP	✓
Offline RL	CQL	Ground Truth Reward	Offline	MLP	
	DT	Ground Truth Reward	Offline	Transformer	
Offline PbRL	OPAL	Preference	Offline	MLP	✓
	PT	Preference	Offline	Transformer	✓
	OPPO	Preference	Offline	Transformer	×

”upside-down” reinforcement learning (UDRL), seeking to model behaviors via a supervised loss conditioned on a target return. Ghosh et al. (2019); Liu et al. (2022b; 2021) extended prior UDRL methods to perform goal reaching by taking the goal state as the condition, and Paster et al. (2020) further used an LSTM for goal-conditioned online RL settings. DT (Chen et al., 2021) and TT (Janner et al., 2021) solved the problem via sequence modeling, since they believe sequence modeling enables to model behaviors without access to the reward, in a similar style to language (Radford et al., 2018) and images (Chen et al., 2020). Although the above methods can avoid online interaction between the agent and the environment, they all require ground truth reward or expert demonstrations to specify the task, which often requires a lot of human labor.

Offline PbRL. OPAL (Shin & Brown, 2021) first tried to solve offline PbRL by simply combining previous (online) PbRL method and off-the-shelf offline RL algorithm. PT (Kim et al., 2023) introduced a new preference model based on the weighted sum of non-Markovian rewards and utilized transformer-based architecture to design such model. Both of these works adopt naive two-step strategy with learning a reward function separately. To avoid information bottleneck that scalar reward may create (Vamplew et al., 2022), OPPO jointly models offline preferences and learns the optimal decision policy in a one-step paradigm. And in contrast to both supervised RL and UDRL, the purpose of our method is to search for the optimal solution supervised by a binary preference signal in the offline setting. Our method is not only working with sub-optimal demonstrations but also revealing optimal behaviors without injecting human priors about the optimal demonstration.

3. Preliminaries

We consider reinforcement learning (RL) in a Markov decision process (MDP) described by a tuple $(\mathcal{S}, \mathcal{A}, r, P, p_0, \gamma)$, where $\mathbf{s}_t \in \mathcal{S}$, $\mathbf{a}_t \in \mathcal{A}$, and $r_t = r(\mathbf{s}_t, \mathbf{a}_t)$ denote the state, action, and reward at timestep t, $P(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)$ denotes the transition dynamics, $p_0(\mathbf{s}_0)$ denotes the initial state distribution, and $\gamma \in [0, 1)$ denotes the discount factor. At each timestep t, the agent receives a state $\mathbf{s}_t$ from the environment and chooses an action $\mathbf{a}_t$ based on the policy $\pi(\mathbf{a}_t|\mathbf{s}_t)$. In the standard RL framework, the environment returns a reward r_t, and the agent transits to the next state $\mathbf{s}_{t+1}$. The expected return $\mathcal{J}_r(\pi) = \mathbb{E}_{\tau\sim\pi(\tau)} \sum_{k=0}^{\infty} \gamma^k r(s_{t+k}, a_{t+k})$ is defined as the expectation of discounted cumulative rewards, where $\tau = (\mathbf{s}_0, \mathbf{a}_0, \mathbf{s}_1, \mathbf{a}_1, \dots)$, $\mathbf{s}_0 \sim p_0(\mathbf{s}_0)$, $\mathbf{a}_t \sim \pi(\mathbf{a}_t|\mathbf{s}_t)$, and $\mathbf{s}_{t+1} \sim P(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)$. The agent's goal is to learn a policy π that maximizes the expected return.

3.1. Offline Preference-based reinforcement learning

In this work, we assume a fully offline setting in which the agent is not allowed to conduct online rollouts (over the MDP) during training but is provided with a static fixed dataset. The static dataset, $\mathcal{D} := \{\tau^0, \dots, \tau^N\}$, consists of pre-collected trajectories, where each trajectory τ^i contains a contiguous sequence of states and actions: $\tau^i := \{\mathbf{s}_0^i, \mathbf{a}_0^i, \mathbf{s}_1^i, \dots\}$. Such an offline setting is more challenging than the standard (online) setting as it removes the ability to explore the environment and collect additional feedback. Unlike imitation learning, we do not assume that the dataset comes from a single expert policy. Instead, the dataset $\mathcal{D}$ may contain trajectories collected by sub-optimal or even random behavior policies.

Generally, the standard offline RL assumes the existence of reward information for each state-action pair in $\mathcal{D}$. However, in the offline Preference-based RL (PbRL) framework, we assume that such reward is not accessible, while the agent can access offline preferences (between some pairs of trajectories (τ^i, τ^j)) that are labeled by an expert (human) annotator. Specifically, the annotator gives a feedback indicating which trajectory is preferred, i.e., $y \in \{0, 1, 0.5\}$, where 0 indicates $\tau^i \succ \tau^j$ (the event that trajectory τ^i is preferable to trajectory τ^j), 1 indicates $\tau^j \succ \tau^i$ (τ^j is preferable to τ^i), and 0.5 implies an equally preferable case. Each feedback is stored in a labeled offline dataset $\mathcal{D}_\succ$ as a triple (τ^i, τ^j, y). Given these preferences, the goal of PbRL is to find a policy $\pi(\mathbf{a}_t|\mathbf{s}_t)$ that maximizes the expected return $\mathcal{J}_{r_\psi}$, under the hypothetical reward function $r_\psi(\mathbf{s}_t, \mathbf{a}_t)$

consistent with human preferences. To enable this, previous works learn a reward function $r_\psi(\mathbf{s}_t, \mathbf{a}_t)$ and use the Bradley-Terry model (Bradley & Terry, 1952) to model the human preference, expressed here as a logistic function:

$$P[\tau^i \succ \tau^j] = \text{logistic}(\sum_t r_\psi(\mathbf{s}_t^i, \mathbf{a}_t^i) - \sum_t r_\psi(\mathbf{s}_t^j, \mathbf{a}_t^j)), \tag{1}$$

where $(\mathbf{s}_t^i, \mathbf{a}_t^i) \sim \tau^i$, $(\mathbf{s}_t^j, \mathbf{a}_t^j) \sim \tau^j$. Intuitively, this can be interpreted as the assumption that the probability of preferring a trajectory depends exponentially on the cumulative reward over the trajectory labeled by an underlying reward function. The reward function is then updated by minimizing the following cross-entropy loss:

$$- \mathop{\mathbb{E}}_{(\tau^i,\tau^j,y)\sim\mathcal{D}_\succ} \Big[(1-y) \log P[\tau^i \succ \tau^j] + y \log P[\tau^j \succ \tau^i]\Big]. \tag{2}$$

With the learned reward function r_ψ used to label each transition in the dataset, we can adopt an off-the-shelf offline RL algorithm to enable the policy learning.

3.2. Hindsight Information Matching

Beyond the typical iterative (offline) RL framework, information matching (IM) (Furuta et al., 2021) has been recently studied as an alternative problem specification in (offline) RL. The objective of IM in RL is to learn a contextual policy $\pi(\mathbf{a}|\mathbf{s}, \mathbf{z})$ whose trajectory rollouts satisfy the pre-defined desired information statistics value $\mathbf{z}$:

$$\min_\pi \mathop{\mathbb{E}}_{\substack{\mathbf{z}\sim p(\mathbf{z}) \\ \tau_\mathbf{z}\sim\pi(\mathbf{z})}} \left[\ell\left(\mathbf{z}, I(\tau_\mathbf{z})\right)\right], \tag{3}$$

where $p(\mathbf{z})$ is a prior, and $\pi(\mathbf{z})$ denotes the trajectory distribution generated by rolling out $\pi(\mathbf{a}|\mathbf{s}, \mathbf{z})$ in the environment. $I(\tau)$ is a function capturing the statistical information of a trajectory τ, such as the distribution statistics of state and reward, like mean, variance (Wainwright et al., 2008), and ℓ is a loss function.

On the one hand, if we set $p(\mathbf{z})$ as a prior distribution, optimizing Eq.3 corresponds to performing unsupervised (online) RL to learn a set of skills (Eysenbach et al., 2018; Sharma et al., 2019). On the other hand, if we set $p(\mathbf{z})$ as statistical information of a given off-policy trajectory (or state-action) distribution $\mathcal{D}(\tau)$ (or $\mathcal{D}(\mathbf{s}, \mathbf{a})$), Eq.3 corresponds to an objective for hindsight information matching in (offline) RL. For example, HER (Andrychowicz et al., 2017) and return-conditioned RL (upside-down RL (Srivastava et al., 2019; Kumar et al., 2019b; Chen et al., 2021; Janner et al., 2021)) use the above concept of hindsight: specifying any trajectory τ in the dataset as the hindsight target and setting the information $\mathbf{z}$ in Eq.3 as $I(\tau)$. Then, we provide the $I(\cdot)$-driven hindsight information matching (HIM) objective:

$$\min_\pi \mathop{\mathbb{E}}_{\substack{\tau\sim\mathcal{D}(\tau) \\ \tau_\mathbf{z}\sim\pi(\mathbf{z})}} \left[\ell\left(I(\tau), I(\tau_\mathbf{z})\right)\right], \tag{4}$$

where $\mathbf{z} := I(\tau)$. In HER, we set $I(\tau)$ as the final state in trajectory τ, and in reward-conditional RL, we set $I(\tau)$ as the return of trajectory τ. Thus, we can use the hindsight information $\mathbf{z} := I(\tau)$ to provide supervision for training the contextual policy $\pi(\mathbf{a}|\mathbf{s}, \mathbf{z})$. However, in the offline setting, sampling $\tau_\mathbf{z}$ from $\pi(\mathbf{z})$ is not accessible. Thus, we must model the environment transition dynamics besides $I(\cdot)$-driven hindsight information modeling. That is to say, we need to model the trajectory itself, *i.e.*, $\min_\pi \mathbb{E}_{\tau\sim\mathcal{D}(\tau),\tau_\mathbf{z}\sim\pi(\mathbf{z})} \left[\ell\left(\tau, \tau_\mathbf{z}\right)\right]$. Then, we provide the overall offline HIM objective:

$$\min_\pi \mathop{\mathbb{E}}_{\substack{\tau\sim\mathcal{D}(\tau) \\ \tau_\mathbf{z}\sim\pi(\mathbf{z})}} \left[\ell\left(I(\tau), I(\tau_\mathbf{z})\right) + \ell\left(\tau, \tau_\mathbf{z}\right)\right]. \tag{5}$$

To give an intuitive understanding of the above objective, we provide a simple example: considering hindsight $I(\cdot)$ being the return of a trajectory, optimizing $\ell\left(I(\tau), I(\tau_\mathbf{z})\right)$ ensures that the generated $\tau_\mathbf{z}$ will reach the same return as $\tau = I^{-1}(\mathbf{z})$. However, in the offline setting, we must ensure that the generated $\tau_\mathbf{z}$ stays in support of the offline data, eliminating the out-of-distribution (OOD) issue. Thus we minimize $\ell\left(\tau, \tau_\mathbf{z}\right)$ approximately. In implementation, directly optimizing $\ell\left(\tau, \tau_\mathbf{z}\right)$ is enough to ensure the hindsight information is matched, *e.g.*, $\ell\left(I(\tau), I(\tau_\mathbf{z})\right) < \epsilon$. Here, we explicitly formalize the $\ell\left(I(\tau), I(\tau_\mathbf{z})\right)$ term with particular emphasis on the requisite of hindsight information matching objective and meanwhile highlight the difference, see Section 4, between the above HIM objective (taking $I(\cdot)$ as a prior) and our OPPO formulation (requiring learning $I_\theta(\cdot)$).

By optimizing Eq.5, we can obtain a contextual policy $\pi(\mathbf{a}|\mathbf{s}, \mathbf{z})$. In the evaluation phase, the optimal policy $\pi(\mathbf{a}|\mathbf{s}, \mathbf{z}^*)$ can be specified by conditioning the policy on a selected target $\mathbf{z}^*$. For example, Decision Transformer (Chen et al., 2021) takes the desired performance as the target $\mathbf{z}^*$(e.g., specify maximum possible return to generate expert behavior), and RvS-G (Emmons et al., 2021) takes the goal state as the target $\mathbf{z}^*$.

4. OPPO: Offline Preference-guided Policy Optimization

In this section, we present our method, OPPO (offline preference-guided policy optimization), which adopts the hindsight information matching (HIM) objective in Section 4.1 to model an offline contextual policy $\pi(\mathbf{a}|\mathbf{s}, \mathbf{z})$, and introduces a triplet loss in Section 4.2 to model the human preference as well as the optimal context $\mathbf{z}^*$. At testing, we condition the policy on the optimal context $\mathbf{z}^*$ and thus

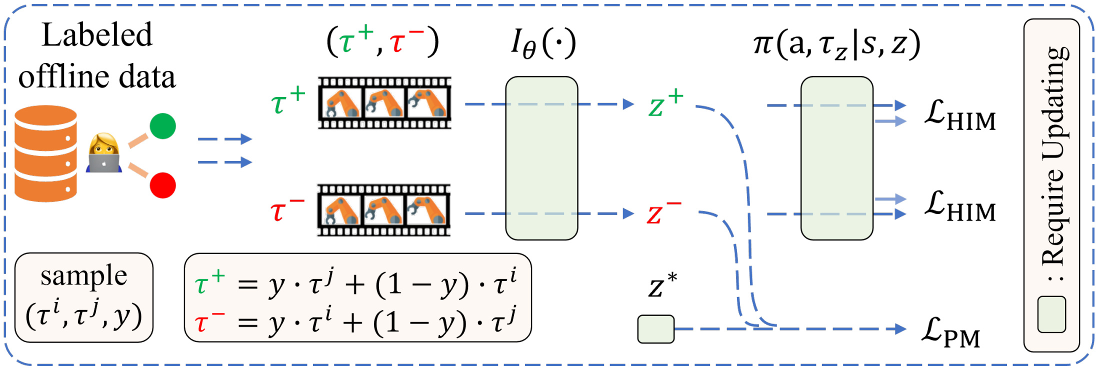

Labeled offline data
(τ^+, τ^-)
$I_\theta(\cdot)$
$\pi(a, \tau_z | s, z)$
τ^+
τ^-
z^+
z^-
z^*
$\mathcal{L}_{\text{HIM}}$
$\mathcal{L}_{\text{HIM}}$
$\mathcal{L}_{\text{PM}}$
: Require Updating
sample (τ^i, τ^j, y)
$\tau^+ = y \cdot \tau^j + (1-y) \cdot \tau^i$
$\tau^- = y \cdot \tau^i + (1-y) \cdot \tau^j$

Figure 2. OPPO first maps offline trajectories (both positive τ^+ and negative τ^-) to a latent space via the hindsight information extractor I_θ. It then optimizes the offline HIM objective $\mathcal{L}_{\text{HIM}}$. Finally, the belief of the optimal hindsight information $\mathbf{z}^*$ is updated to model the human preference with objective $\mathcal{L}_{\text{PM}}$. Meanwhile, the preference modeling loss also regularizes the learning of the hindsight information extractor I_θ.

conduct rollout with $\pi(\mathbf{a}|\mathbf{s}, \mathbf{z}^*)$. In principle, OPPO is compatible with any PbRL setting, including both online and offline. In the scope of our analysis and experiments, however, we focus on the offline setting to decouple exploration difficulties in online RL.

4.1. HIM-driven Policy Optimization

As described in Section 3.1, to directly implement the off-the-shelf offline RL algorithms, previous works in PbRL explicitly learn a reward function with Eq.2 (as shown in Fig.1 left). As an alternative to such a two-step approach, we seek to learn the policy directly from the preference-labeled offline dataset (as shown in Fig.1 right). Inspired by the offline HIM objective in Section 3.2, we propose to learn a contextual policy $\pi(\mathbf{a}|\mathbf{s}, \mathbf{z})$ in the offline PbRL setting. Assuming I_θ being a (learnable) network that encodes the hindsight information in PbRL, we formulate the following objective:

$$\min_{\pi, I_\theta} \mathcal{L}_{\text{HIM}} := \mathop{\mathbb{E}}_{\substack{\tau\sim\mathcal{D}(\tau)\\ \tau_{\mathbf{z}}\sim\pi(\mathbf{z})}} \Big[\ell\left(I_\theta(\tau), I_\theta(\tau_{\mathbf{z}})\right) + \ell\left(\tau, \tau_{\mathbf{z}}\right) \Big], \quad (6)$$

where $\mathbf{z} := I_\theta(\tau)$. Note that Eq.6 is a different instantiation of Eq.5 in which we learn the hindsight information extractor $I_\theta(\cdot)$ in the PRBL setting, while previous (offline) RL algorithms normally set $I(\cdot)$ to be a prior (Chen et al., 2021; Emmons et al., 2021). Such an encoder-decoder structure is now similar to Bi-directional Decision Transformer (BDT) proposed by (Furuta et al., 2021) for offline imitation learning. However, since expert demonstrations are missing in the PbRL setting, in Section 4.2, we propose to use the preference labels in $\mathcal{D}_\succ$ to extract hindsight information.

4.2. Preference Modeling

To make the hindsight information $I_\theta(\tau)$ in Eq.6 match the preference information in the (labeled) dataset $\mathcal{D}_\succ$, we construct the following preference modeling objective inspired by the contrastive loss in metric learning (Le-Khac et al., 2020):

$$\min_{\mathbf{z}^*, I_\theta} \mathop{\mathbb{E}}_{(\tau^i, \tau^j, y)\sim\mathcal{D}_\succ} \Big[\ell(\mathbf{z}^*, \mathbf{z}^+) - \ell(\mathbf{z}^*, \mathbf{z}^-) \Big], \quad (7)$$

where $\mathbf{z}^+$ and $\mathbf{z}^-$ represent the embedding of the preferable (positive) trajectory $I_\theta(y\tau^j + (1-y)\tau^i)$ and that of the less preferable (negative) trajectory $I_\theta(y\tau^i + (1-y)\tau^j)$, respectively. Closing to the idea of using regret for modeling preference (Knox et al., 2022; Chen et al., 2022), our basic assumption of designing the objective in Eq.7 is that humans normally conduct two-level comparisons before giving preferences between two trajectories (τ^i, τ^j): 1) separately judging the similarity between trajectory τ^i and the hypothetical optimal trajectory τ^*, i.e. $-\ell(\mathbf{z}^*, \mathbf{z}^i)$, and the similarity between trajectory τ^j and the hypothetical optimal one τ^*, $-\ell(\mathbf{z}^*, \mathbf{z}^j)$, and 2) judging the difference between the above two similarities ($-\ell(\mathbf{z}^*, \mathbf{z}^i)$ vs. $-\ell(\mathbf{z}^*, \mathbf{z}^j)$) and setting the trajectory with the higher similarity as the preferred one. Hence, optimizing Eq.7 guarantees finding the optimal embedding that is more similar to $\mathbf{z}^+$ and less similar to $\mathbf{z}^-$. To clarify, $\mathbf{z}^*$ is the corresponding contextual information for τ^*, whereas τ^* will always be preferred over any offline trajectories in the dataset.

In practice, to robustify the preference modeling, we optimize the following objective using the triplet loss in place of the objective in Eq.7:

$$\min_{\mathbf{z}^*, I_\theta} \mathcal{L}_{\text{PM}} := \mathbb{E}\Big[\max(\ell(\mathbf{z}^*, \mathbf{z}^+) - \ell(\mathbf{z}^*, \mathbf{z}^-) + m, 0) \Big], \quad (8)$$

where m is an arbitrarily set margin between positive and negative pairs. It is worth mentioning that the posterior of the optimal embedding $\mathbf{z}^*$ and the hindsight information extractor $I_\theta(\cdot)$ are updated alternatively to ensure learning stability. A better estimate of the optimal embedding helps

Algorithm 1 OPPO: Offline Preference-guided Policy Optimization

```
Require: Dataset $\mathcal{D} := \{\tau\}$ and labeled dataset $\mathcal{D}_{\succ} := \{(\tau^i, \tau^j, y)\}$, where $\tau^i \in \mathcal{D}$ and $\tau^j \in \mathcal{D}$. Return: $\pi(\mathbf{a}|\mathbf{s}, \mathbf{z})$ and $\mathbf{z}^*$.
1: Initialize policy network $\pi(\mathbf{a}|\mathbf{s}, \mathbf{z})$, hindsight information extractor $I_\theta : \tau \to \mathbf{z}$, and the optimal context embedding $\mathbf{z}^*$.
2: while not converged do
3:     Sample a batch of trajectories from $\mathcal{D}$: $\{\tau\}_\mathrm{B} \sim \mathcal{D}$.
4:     Update $\pi(\mathbf{a}|\mathbf{s}, \mathbf{z})$ and $I_\theta(\cdot)$ with sampled $\{\tau\}_\mathrm{B}$ using $\mathcal{L}_\mathrm{HIM}$.
5:     Sample a batch of preferences from $\mathcal{D}_{\succ}$: $\{(\tau^i, \tau^j, y)\}_\mathrm{B} \sim \mathcal{D}_{\succ}$.
6:     Update $I_\theta(\cdot)$ and the optimal $\mathbf{z}^*$ with sampled $\{(\tau^i, \tau^j, y)\}_\mathrm{B}$ using $\mathcal{L}_\mathrm{PM}$.
7: end while
```

the encoder to extract features to which the human labeler pays more attention. In contrast, a better hindsight information encoder, on the other hand, accelerates the search process for the optimal trajectory in the high-level embedding space. In this way, the loss function for the encoder consists of two parts: 1) a hindsight information matching loss in a supervised style as in Eq.6 and 2) a triplet loss as in Eq.8 to better incorporate the binary supervision provided by the preference-labeled dataset.

4.3. Training Objectives & Implementation Details

In our experiment, we consolidate ℓ in Eq.6 as MSE Loss and in Eq.8 as Euclidean Distance. In this case, we model $\mathbf{z}^*$ as a point in the $\mathbf{z}-$space, and the similarity measure ℓ is L_2 distance. An alternative option is to model $\mathbf{z}^*$ as a point sampled from a learned distribution in the $\mathbf{z}-$space, where ℓ is a measurement between two distributions, such as the KL divergence. Also, we add a normalization loss $\mathcal{L}_\text{norm}$ to constrain the L2 norm of all embeddings generated by hindsight information extractor I_θ.

$$\mathcal{L}_\text{total} := \mathcal{L}_\text{HIM} + \alpha\mathcal{L}_\text{PM} + \beta\mathcal{L}_\text{norm} \tag{9}$$

The architecture overview of OPPO is shown in Fig.2. OPPO models the hindsight information extractor I_θ as an encoder network $I_\theta : \tau \to \mathbf{z}$ and we use the BERT architecture. Furthermore, similar to DT (Chen et al., 2021), we use the GPT architecture to model $\pi(\mathbf{a}|\mathbf{s}, \mathbf{z})$, which predicts future actions via autoregressive modeling. For specific hyperparameter selection during the training process, please refer to the detailed description in Appendix A.1.3. Algorithm 1 details the training of OPPO, and the entire process is summarized as follows: 1) We sample a batch of trajectories from the dataset $\mathcal{D}$; 2) In Line 4, we use Eq.6 (the hindsight information matching loss) to update $\pi(\mathbf{a}|\mathbf{s}, \mathbf{z})$ and $I_\theta(\cdot)$ based on sampled trajectories; consequently, given the $\mathbf{z}$ extracted out of an offline trajectory by the extractor, the policy is able to reconstruct it; 3) Then, we sample a batch of preferences from the labeled dataset $\mathcal{D}_{\succ}$; 4) Finally, in Line 6, we update $I_\theta(\cdot)$ and $\mathbf{z}^*$ based on the sampled $\{(\tau^i, \tau^j, y)\}_\mathrm{B}$ using Eq.8, making the optimal embedding $\mathbf{z}^*$ near to the more preferred trajectory $\mathbf{z}^+$, and meanwhile further away from the less preferred trajectory $\mathbf{z}^-$.

In summary, OPPO learns a contextual policy $\pi(\mathbf{a}|\mathbf{s}, \mathbf{z})$, a context (hindsight information) encoder $I_\theta(\tau)$, and the optimal context, $\mathbf{z}^*$, for the optimal trajectory τ^*. Compared with previous PbRL works (first learning a reward function with Eq.2 and then learning offline policy with off-the-shelf offline RL algorithms), OPPO learns the optimal (offline) policy ($\pi(\mathbf{a}|\mathbf{s}, \mathbf{z}*)$) directly and thus avoids the potential information bottleneck caused by the limited information capacity of scalar reward assignment. Compared with the HIM-based offline RL algorithms (*e.g.*, DT (Chen et al., 2021) and RvS-G (Emmons et al., 2021)), OPPO does not need to manually specify the target context for the rollout policy $\pi(\mathbf{a}|\mathbf{s}, \cdot)$ at the testing phase.

5. Experiments

In this section, we evaluate and compare OPPO to other baselines in the offline PbRL setting. A central premise behind the design of OPPO is that the learned hindsight information encoder $I_\theta(\cdot)$ can capture preferences over different trajectories, as described by Eq.8. Our experiments are therefore designed to answer the following questions:

1. Does OPPO truly capture the preference? In other words, does the learned $\mathbf{z}$-space (encoded by the learned $I_\theta(\cdot)$) align with the given preference? Please refer to Section 5.1.

2. Can the learned optimal contextual policy $\pi(\mathbf{a}|\mathbf{s}, \mathbf{z}^*)$ outperform the policy $\pi(\mathbf{a}|\mathbf{s}, \mathbf{z})$ that is conditioned on any other context $\mathbf{z} \in \{I_\theta(\tau)|\tau \in \mathcal{D}\}$? Please refer to Section 5.2.

3. Can OPPO achieve the competitive performance compared with other offline PbRL baselines? Please refer to Section 5.3.

4. What benefits can we gain from designing the one-step offline PbRL, *i.e.*, iteratively conducting offline data modeling (Eq.6) and preference modeling (Eq.8)? Please refer to Section 5.4.

5. How does OPPO behave in terms of the amount of preference feedback? Please refer to Section 5.5.

6. Can OPPO attain satisfactory results by incorporating preference from real human instead of scripted teacher? Please refer to Section 5.6.

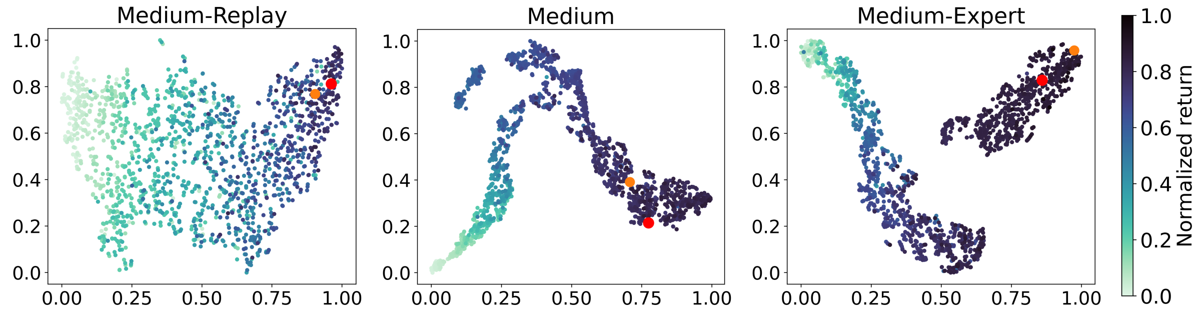


Figure 3. We utilize t-SNE to visualize the $\mathbf{z}$-space learned in Hopper environment, encoded with a well-trained $I_\theta(\cdot)$, including the embedding of random trajectories in $\mathcal{D}$, our learned $\mathbf{z}^*$ ("orange dot") and the actual optimal $\mathbf{z}^{**}$ ("red dot"), embedding of the best trajectory/policy learned with online reinforcement learning method. Color of the points represent the normalized return of the corresponding trajectory τ.

To answer the above questions, we evaluate OPPO on the continuous control tasks from the D4RL benchmark (Fu et al., 2020). Specifically, we choose Hopper, Walker, and Halfcheetah as three base tasks, with medium, medium-replay, medium-expert as the datasets for each task.

Table 2. Comparison of (normalized) performance when rollouting the contextual policy $\pi(\mathbf{a}|\mathbf{s},\cdot)$ conditioned on different context ($\mathbf{z}^*$, $\mathbf{z}_{\text{high}}$, and $\mathbf{z}_{\text{low}}$).

Environment	**Dataset**	$\mathbf{z}^*$	$\mathbf{z}_{\text{high}}$	$\mathbf{z}_{\text{low}}$
Hopper	Medium-Expert	**108.0** ± **5.1**	94.2 ± 24.3	79.1 ± 28.8
	Medium	**86.3** ± **3.2**	55.8 ± 7.9	51.6 ± 13.8
	Medium-Replay	**88.9** ± **2.3**	78.6 ± 26.3	26.6 ± 15.2
Walker	Medium-Expert	**105.0** ± **2.4**	**106.5** ± **9.1**	93.4 ± 7.4
	Medium	**85.0** ± **2.9**	64.9 ± 24.9	72.6 ± 10.6
	Medium-Replay	**71.7** ± **4.4**	55.7 ± 24.8	6.8 ± 1.7
Halfcheetah	Medium-Expert	**89.6** ± **0.8**	48.3 ± 14.4	42.6 ± 2.6
	Medium	**43.4** ± **0.2**	42.5 ± 3.9	42.4 ± 3.2
	Medium-Replay	**39.8** ± **0.2**	35.6 ± 8.5	33.9 ± 9.2
Sum		**717.7**	581.9	448.9

5.1. Can z-space align well with given preferences?

In this subsection, we probe that OPPO can enable well-aligned preferences over the $\mathbf{z}$-space encoded by the learned I_θ. We first sample random trajectories from the offline dataset $\mathcal{D}$, and encode them with the learned I_θ, and utilize t-SNE (van der Maaten & Hinton, 2008) as a tool to visualize the encoded $\mathbf{z}$, shown in Fig.3. The learned optimal $\mathbf{z}^*$ is marked with an orange dot. Besides, we also mark the (embedding of) optimal trajectory in the D4RL expert dataset, generated by the learned online optimal policy, with a red dot ($\mathbf{z}^{**}$).

According to Eq.8, embeddings near the actual optimal $\mathbf{z}^{**}$ in $\mathbf{z}$-space means they are more preferable implied by the preference label. Comparing the sampled trajectories (embeddings), we find OPPO successfully captures the preference. As illustrated in Fig.3, the trajectories (embeddings) that are near $\mathbf{z}^{**}$ often have high returns (points with a deeper color). Further, we observe that our learned optimal $\mathbf{z}^*$ constantly stays close to actual optimal $\mathbf{z}^{**}$, which suggests that our learned $\mathbf{z}^*$ preserves near-optimal behaviors. Thus, it gives justification that OPPO can make meaningful preference modeling.

5.2. Can $\pi(\mathbf{a}|\mathbf{s},\mathbf{z}^*)$ achieve better performance?

Fig.3 shows that our learned $I_\theta(\cdot)$ can produce a well-aligned context embedding $\mathbf{z}$-space exhibiting effective preference modeling across (embeddings of) trajectories. More importantly, context embeddings' preference property should be preserved when we condition the context on the learned contextual policy $\pi(\mathbf{a}|\mathbf{s},\cdot)$. In other words, $I_\theta(\cdot)$ should transfers the preference relationship from (τ^i,τ^j) to $(\ell(\mathbf{z}^i,\mathbf{z}^*),\ell(\mathbf{z}^j,\mathbf{z}^*))$; further, rolling out the contextual policy $\pi(\mathbf{a}|\mathbf{s},\cdot)$, $(\tau_{\mathbf{z}^i},\tau_{\mathbf{z}^j})$ should similarly preserve the above preference relationship.

To show that, we compare the performance of rollouts by the contextual policy $\pi(\mathbf{a}|\mathbf{s},\cdot)$ conditioned on different contexts in Table 2. We choose three context embeddings: $\mathbf{z}^*$, $\mathbf{z}_{\text{high}}$ (embedding of the trajectory with the highest return in $\mathcal{D}$), and $\mathbf{z}_{\text{low}}$ (embedding of the trajectory with the lowest return in $\mathcal{D}$) and provide respective rollout performances (averaged over 3 seeds). We discover that the contextual policy $\pi(\mathbf{a}|\mathbf{s},\mathbf{z})$ conditioned on $\mathbf{z}$ with a high (or low) return (of corresponding trajectory $\tau = I_\theta^{-1}(\mathbf{z})$) obtains an actual high (or low) return when rollouting this policy in the environment, *e.g.*, $\pi(\mathbf{a}|\mathbf{s},\mathbf{z}_{\text{high}})$ performs better than $\pi(\mathbf{a}|\mathbf{s},\mathbf{z}_{\text{low}})$ (thus preserving the hindsight preference relationship). Further, when conditioned on the learned optimal $\mathbf{z}^*$, $\pi(\mathbf{a}|\mathbf{s},\mathbf{z}^*)$ produces the best performance over that conditioned on all other offline embeddings. Notice that our learned optimal $\pi(\mathbf{a}|\mathbf{s},\mathbf{z}^*)$ performs better than the contextual policy $\pi(\mathbf{a}|\mathbf{s},\mathbf{z}_{\text{high}})$. This result implies that the trajectory of our optimal policy is generally better than other trajectories in the offline dataset.

Table 3. Performance comparison between OPPO and 4 offline (PbRL) baselines (DT+r, DT+r_ψ, CQL+r, and IQL+r) in D4RL Gym-Mujoco tasks, where results are reported over 3 seeds.

Environment	Dataset	Ours	DT+r	DT+r_ψ	CQL+r	IQL+r	BC
Hopper	Medium-Expert	**108.0 ± 5.1**	**111.0 ± 0.5**	95.6 ± 27.3	**111.0**	91.5	79.6
	Medium	**86.3 ± 3.2**	76.6 ± 3.9	73.3 ± 3.0	58.0	66.3	63.9
	Medium-Replay	**88.9 ± 2.3**	**87.8 ± 4.7**	72.5 ± 22.2	48.6	**94.7**	27.6
Walker	Medium-Expert	105.0 ± 2.4	**109.2 ± 0.3**	**109.7 ± 0.1**	98.7	**109.6**	36.6
	Medium	**85.0 ± 2.9**	80.9 ± 3.1	81.1 ± 2.1	79.2	78.3	77.3
	Medium-Replay	71.7 ± 4.4	**79.6 ± 3.1**	**80.4 ± 4.4**	26.7	73.9	36.9
HalfCheetah	Medium-Expert	**89.6 ± 0.8**	86.8 ± 1.3	**88.4 ± 0.7**	62.4	86.7	59.9
	Medium	43.4 ± 0.2	43.4 ± 0.1	43.2 ± 0.2	44.4	**47.4**	43.1
	Medium-Replay	39.8 ± 0.2	39.2 ± 0.3	38.8 ± 0.3	**46.2**	44.2	4.3
Sum		**717.7**	714.5	683.0	575.2	692.4	429.2

5.3. Performance of OPPO on Benchmark Tasks with Scripted Teacher

We have shown that OPPO produces a near-optimal context $\mathbf{z}^*$, and the learned contextual policy $\pi(\mathbf{a}|\mathbf{s}, \cdot)$ can preserve the hindsight preference. This subsection investigates whether the optimal policy $\pi(\mathbf{a}|\mathbf{s}, \mathbf{z}^*)$ can achieve competitive performance on the offline (PBRL) benchmark. For comparison, we consider three offline PbRL methods and BC as baselines: 1) DT+r: performing Decision Transformer (Chen et al., 2021) with ground-truth reward function, and the results are run by us; 2) DT+r_ψ: performing Decision Transformer with a learned reward function (using Eq.2); 3) CQL+r: performing CQL (Kumar et al., 2020) with ground-truth reward function; 3) IQL+r: performing IQL with ground-truth reward function, the results are reported from IQL (Kostrikov et al., 2022); 4) BC: performing bahavior cloning on the dataset, the results are reported from DT (Chen et al., 2021).

In Table 3, we show the performance of OPPO and baselines. We have the following observations. 1) OPPO has retained a comparable performance against the Decision Transformer trained using true rewards. OPPO is a PbRL approach requiring only (human) preferences, which have a more flexible and straightforward form of supervision in the real world. 2) Although DT+r_ψ also shows competitive results in these benchmarks, such a method needs a target of return-to-go determined by the human prior [1]. Our method, in contrast, avoids the need for such a prior target by searching across the $\mathbf{z}$-space. We argue that our searching method brings advantages because rewards are usually hard to obtain in real-world RL applications, where the preference is the only information accessible for training and deploying an RL method.

Table 4. Ablation study of one-step paradigm in Medium-Replay Dataset

Task	OPPO	OPPO-a
Hopper	**88.9 ± 2.3**	78.3 ± 7.1
Walker	**71.7 ± 4.4**	66.3 ± 1.6
HalfCheetah	**39.8 ± 0.2**	39.6 ± 0.1
Sum	**200.4**	184.2

5.4. Benefits of One-step Offline PbRL

We conduct an ablation study to analyze the benefit of iterating $\mathcal{L}_{\text{HIM}}$ and $\mathcal{L}_{\text{PM}}$ (for updating I_θ) in an one-step paradigm. Firstly, we remove I_θ from $\partial\mathcal{L}_{\text{PM}}/\partial\theta$ and only keep the optimal embedding $\mathbf{z}^*$ to be updated in Eq.8. Then, we continue to visualize the embedding $\mathbf{z}$-space for this ablation setting (OPPO-a), the t-SNE visualization shown in Fig.4. By comparing Fig.4 to Fig.3, we can see that the preference relationship in the embedding space (learned with OPPO-a) is all shuffled. In a less expressive $\mathbf{z}$-space, it is challenging to model the preference and find the optimal $\mathbf{z}^*$. Further, as shown in Table 4, the comparison results of medium-replay tasks demonstrate that such an ablation does cause the performance degradation.

Table 5. Ablation study of feedback quantity

Dataset	50k	1k	500
Medium-Expert	108.0 ± 5.1	102.9 ± 3.2	104.9 ± 4.1
Medium	86.3 ± 3.2	90.8 ± 2.0	77.5 ± 12.8
Medium-Replay	88.9 ± 2.3	60.4 ± 3.0	68.5 ± 22.8
Sum	283.1	254.2	250.9

5.5. Performance with Different Amount of Preference Feedback

For the Hopper task, we evaluate the impact of different amounts of preference labels on the performance of OPPO

[1] Preference-based relabelled rewards only participate in the training phase. During the evaluation phase of DT+r_ψ, we pass in the same target return-to-go value as in the original DT paper.

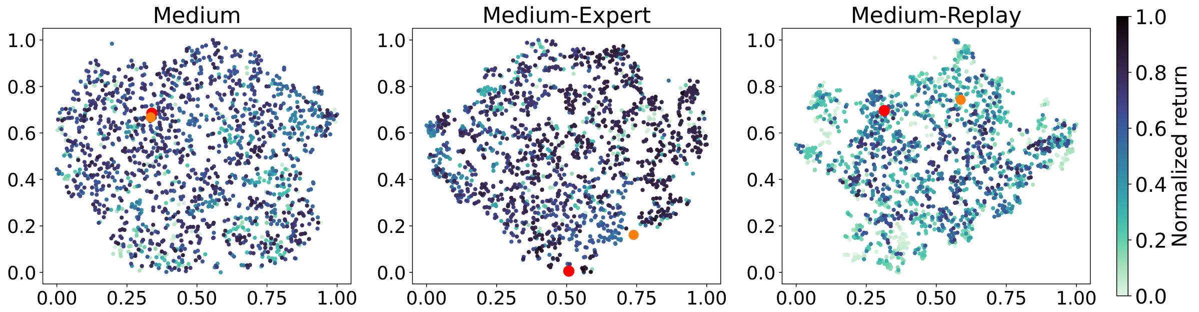


Figure 4. t-SNE visualization of the embedding space learned with OPPO-a in Hopper environment.

and show the results in Table 5. Specifically, OPPO is evaluated using the labels amount from 50k, 1k, 500, on the dataset from Medium-Expert, Medium, Medium Replay. As illustrated in the Table 5, OPPO performs the best when given 50k preference labels and achieves a total normalized score of 283.1 among the three datasets. However, the performance decreases at around 250 for feedback amount decreases to 1k and 500. Therefore, OPPO is robust to the variation in terms of the amount of preference feedback used for training.

Table 6. Performance OPPO with preference from real human, where results are reported over 3 seeds.

Environment	Dataset	IQL+r	IQL+PT	OPPO
Hopper	Medium-Expert	73.6 ± 41.5	69.0 ± 33.9	**107.8 ± 1.6**
	Medium-Replay	83.1 ± 15.8	84.5 ± 4.1	**93.1 ± 1.3**
Walker	Medium-Expert	107.8 ± 2.0	**110.1 ± 0.2**	106.4 ± 1.1
	Medium-Replay	73.1 ± 8.1	71.3 ± 10.3	**74.9 ± 0.7**
Locomotion Sum		337.47	334.9	**382.2**
Lift	Proficient-Human	**96.8 ± 1.8**	91.8 ± 5.9	94.7 ± 1.2
	Multi-Human	86.8 ± 2.8	86.8 ± 6.0	**98.7 ± 2.3**
Can	Proficient-Human	74.5 ± 6.8	69.7 ± 5.9	**75.3 ± 10.1**
	Multi-Human	56.3 ± 8.8	50.5 ± 6.5	**86.7 ± 12.7**
Robosuite Sum		314.3	298.7	**355.3**

5.6. Performance of OPPO on Benchmark Tasks with Real Human Teacher

We have conducted additional experiments using real human-labeled data on the Hopper and Walker tasks. The human preferences we used are obtained from the open-source dataset of PT (Kim et al., 2023), which is collected from actual human familiar with robotic tasks. Also, we have carried out experiments on the Robomimic dataset (Mandlekar et al., 2022), which offers a set of offline datasets on 7-DoF robot manipulation domains. In our experiments, we test our method on two tasks (Lift and Can), where the offline data are collected by either one proficient human teleoperators (Proficient-Human) or multiple human teleoperators with varying proficiency (Multi-Human), and the preference labels are also labeled by real human. In Table 6, we compare OPPO to IQL+r (Kostrikov et al., 2022) and IQL+PT (Kim et al., 2023). We find that our method outperforms IQL+PT in most tasks, and even achieves competitive or better preformance than IQL+r.

To sum up, through six experiments and the visualization of the results, we demonstrate that the **z**-space learned by the encoder is informative and visually interpretable. Besides, the ablation study proves that a preference-guided embedding space of context could improve task performance asymptotically by a non-neglectable margin. Moreover, OPPO can find an embedding to represent the context of the optimal trajectory, where the resulting trajectory is better than any offline trajectory in the dataset. Last but not least, in the offline setting with environment interaction disabled, our paradigm can acquire the optimal behaviors using binary preference labels between sub-optimal trajectories. As shown in the experiment results, OPPO achieves a competitive performance over DT trained using either true rewards or pseudo rewards.

6. Conclusion

This paper introduces offline preference-guided policy optimization (OPPO), a one-step offline PbRL paradigm. Unlike the previous PbRL approaches that learn policy from a pseudo-reward function (learning a separate reward function is a prerequisite), OPPO directly optimizes the policy in a high-level embedding space. To enable that, we propose an offline hindsight information matching (HIM) objective and a preference modeling objective. Empirically, we show that iterating the above two objectives can produce meaningful and preference-aligned embeddings of context. Moreover, conditioned on the learned optimal context, our HIM-based contextual policy can achieve competitive performance on standard offline (PbRL) tasks.

Acknowledgements

This work was supported by STI 2030—Major Projects (2022ZD0208800), and NSFC General Program (Grant No. 62176215).

References

Ajay, A., Kumar, A., Agrawal, P., Levine, S., and Nachum, O. Opal: Offline primitive discovery for accelerating offline reinforcement learning. *arXiv preprint arXiv:2010.13611*, 2020.

Andrychowicz, M., Wolski, F., Ray, A., Schneider, J., Fong, R., Welinder, P., McGrew, B., Tobin, J., Pieter Abbeel, O., and Zaremba, W. Hindsight experience replay. *Advances in neural information processing systems*, 30, 2017.

Arumugam, D., Lee, J. K., Saskin, S., and Littman, M. L. Deep reinforcement learning from policy-dependent human feedback. *arXiv: Learning*, 2019.

Bradley, R. A. and Terry, M. E. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324, December 1952. ISSN 0006-3444. doi: 10.2307/2334029. URL `https://doi.org/10.2307/2334029`.

Campos, V., Trott, A., Xiong, C., Socher, R., Giró-i Nieto, X., and Torres, J. Explore, discover and learn: Unsupervised discovery of state-covering skills. In *International Conference on Machine Learning*, pp. 1317–1327. PMLR, 2020.

Chen, L., Lu, K., Rajeswaran, A., Lee, K., Grover, A., Laskin, M., Abbeel, P., Srinivas, A., and Mordatch, I. Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems*, 34:15084–15097, 2021.

Chen, M., Radford, A., Child, R., Wu, J., Jun, H., Luan, D., and Sutskever, I. Generative pretraining from pixels. In *International conference on machine learning*, pp. 1691–1703. PMLR, 2020.

Chen, X., Zhong, H., Yang, Z., Wang, Z., and Wang, L. Human-in-the-loop: Provably efficient preference-based reinforcement learning with general function approximation. *arXiv preprint arXiv:2205.11140*, 2022.

Christiano, P. F., Leike, J., Brown, T. B., Martic, M., Legg, S., and Amodei, D. Deep reinforcement learning from human preferences. *arXiv: Machine Learning*, 2017.

Emmons, S., Eysenbach, B., Kostrikov, I., and Levine, S. Rvs: What is essential for offline rl via supervised learning? *arXiv preprint arXiv:2112.10751*, 2021.

Eysenbach, B., Gupta, A., Ibarz, J., and Levine, S. Diversity is all you need: Learning skills without a reward function. *arXiv preprint arXiv:1802.06070*, 2018.

Fu, J., Kumar, A., Nachum, O., Tucker, G., and Levine, S. D4rl: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*, 2020.

Fujimoto, S., Meger, D., and Precup, D. Off-policy deep reinforcement learning without exploration. In *International conference on machine learning*, pp. 2052–2062. PMLR, 2019.

Furuta, H., Matsuo, Y., and Gu, S. S. Generalized decision transformer for offline hindsight information matching. *arXiv preprint arXiv:2111.10364*, 2021.

Garcıa, J. and Fernández, F. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research*, 16(1):1437–1480, 2015.

Ghosh, D., Gupta, A., Fu, J., Reddy, A., Devin, C., Eysenbach, B., and Levine, S. Learning to reach goals without reinforcement learning. 2019.

Hadfield-Menell, D., Milli, S., Abbeel, P., Russell, S. J., and Dragan, A. Inverse reward design. *Advances in neural information processing systems*, 30, 2017.

Hans, A., Schneegaß, D., Schäfer, A. M., and Udluft, S. Safe exploration for reinforcement learning. In *ESANN*, pp. 143–148. Citeseer, 2008.

Ibarz, B., Leike, J., Pohlen, T., Irving, G., Legg, S., and Amodei, D. Reward learning from human preferences and demonstrations in Atari. In *Neural Information Processing Systems*, 2018.

Janner, M., Li, Q., and Levine, S. Offline reinforcement learning as one big sequence modeling problem. *Advances in neural information processing systems*, 34: 1273–1286, 2021.

Kalashnikov, D., Irpan, A., Pastor, P., Ibarz, J., Herzog, A., Jang, E., Quillen, D., Holly, E., Kalakrishnan, M., Vanhoucke, V., and Levine, S. QT-opt: Scalable deep reinforcement learning for vision-based robotic manipulation. In *Computer Vision and Pattern Recognition*, 2018.

Kidambi, R., Rajeswaran, A., Netrapalli, P., and Joachims, T. Morel: Model-based offline reinforcement learning. *Advances in neural information processing systems*, 33: 21810–21823, 2020.

Kim, C., Park, J., Shin, J., Lee, H., Abbeel, P., and Lee, K. Preference transformer: Modeling human preferences using transformers for RL. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=Peot1SFDX0`.

Knox, W. B. and Stone, P. Interactively shaping agents via human reinforcement. In *Proceedings of the fifth international conference on Knowledge capture - K-CAP '09*. ACM Press, 2009. doi: 10.1145/

1597735.1597738. URL https://doi.org/10.1145/1597735.1597738.

Knox, W. B., Hatgis-Kessell, S., Booth, S., Niekum, S., Stone, P., and Allievi, A. Models of human preference for learning reward functions. *arXiv preprint arXiv:2206.02231*, 2022.

Kober, J. and Peters, J. Policy search for motor primitives in robotics. In *Neural Information Processing Systems*, 2008.

Kober, J., Bagnell, J. A., and Peters, J. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 32(11):1238–1274, August 2013. ISSN 0278-3649, 1741-3176. doi: 10.1177/0278364913495721. URL https://doi.org/10.1177/0278364913495721.

Kohl, N. and Stone, P. Policy gradient reinforcement learning for fast quadrupedal locomotion. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004*. IEEE, 2004. doi: 10.1109/robot.2004.1307456. URL https://doi.org/10.1109/robot.2004.1307456.

Kostrikov, I., Nair, A., and Levine, S. Offline reinforcement learning with implicit q-learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=68n2s9ZJWF8.

Kumar, A., Fu, J., Soh, M., Tucker, G., and Levine, S. Stabilizing off-policy q-learning via bootstrapping error reduction. *Advances in Neural Information Processing Systems*, 32, 2019a.

Kumar, A., Peng, X. B., and Levine, S. Reward-conditioned policies. *arXiv preprint arXiv:1912.13465*, 2019b.

Kumar, A., Zhou, A., Tucker, G., and Levine, S. Conservative q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 33: 1179–1191, 2020.

Le-Khac, P. H., Healy, G., and Smeaton, A. F. Contrastive representation learning: A framework and review. *Ieee Access*, 8:193907–193934, 2020.

Lee, K., Smith, L. M., and Abbeel, P. PEBBLE: Feedback-efficient interactive reinforcement learning via relabeling experience and unsupervised pre-training. In *International Conference on Machine Learning*, 2021.

Liu, J., Shen, H., Wang, D., Kang, Y., and Tian, Q. Unsupervised domain adaptation with dynamics-aware rewards in reinforcement learning. In Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=u7Qb7pQk8tF.

Liu, J., Hongyin, Z., and Wang, D. DARA: Dynamics-aware reward augmentation in offline reinforcement learning. In *International Conference on Learning Representations*, 2022a. URL https://openreview.net/forum?id=9SDQB3b68K.

Liu, J., Wang, D., Tian, Q., and Chen, Z. Learn goal-conditioned policy with intrinsic motivation for deep reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 7558–7566, 2022b.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Lu, K., Grover, A., Abbeel, P., and Mordatch, I. Reset-free lifelong learning with skill-space planning. *arXiv preprint arXiv:2012.03548*, 2020.

Mandlekar, A., Xu, D., Wong, J., Nasiriany, S., Wang, C., Kulkarni, R., Fei-Fei, L., Savarese, S., Zhu, Y., and Martín-Martín, R. What matters in learning from offline human demonstrations for robot manipulation. In *Conference on Robot Learning*, pp. 1678–1690. PMLR, 2022.

Mihatsch, O. and Neuneier, R. Risk-sensitive reinforcement learning. *Machine learning*, 49(2):267–290, 2002.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Park, J., Seo, Y., Shin, J., Lee, H., Abbeel, P., and Lee, K. Surf: Semi-supervised reward learning with data augmentation for feedback-efficient preference-based reinforcement learning. *arXiv preprint arXiv:2203.10050*, 2022.

Paster, K., McIlraith, S. A., and Ba, J. Planning from pixels using inverse dynamics models. *arXiv preprint arXiv:2012.02419*, 2020.

Pertsch, K., Lee, Y., and Lim, J. J. Accelerating reinforcement learning with learned skill priors. *arXiv preprint arXiv:2010.11944*, 2020.

Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al. Improving language understanding by generative pre-training. 2018.

Sharma, A., Gu, S., Levine, S., Kumar, V., and Hausman, K. Dynamics-aware unsupervised discovery of skills. *arXiv preprint arXiv:1907.01657*, 2019.

Shin, D. and Brown, D. S. Offline preference-based apprenticeship learning. *arXiv preprint arXiv:2107.09251*, 2021.

Siegel, N. Y., Springenberg, J. T., Berkenkamp, F., Abdolmaleki, A., Neunert, M., Lampe, T., Hafner, R., Heess, N., and Riedmiller, M. Keep doing what worked: Behavioral modelling priors for offline reinforcement learning. *arXiv preprint arXiv:2002.08396*, 2020.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., Chen, Y., Lillicrap, T., Hui, F., Sifre, L., van den Driessche, G., Graepel, T., and Hassabis, D. Mastering the game of go without human knowledge. *Nature*, 550(7676):354–359, October 2017. ISSN 0028-0836, 1476-4687. doi: 10.1038/nature24270. URL `https://doi.org/10.1038/nature24270`.

Singh, A., Liu, H., Zhou, G., Yu, A., Rhinehart, N., and Levine, S. Parrot: Data-driven behavioral priors for reinforcement learning. *arXiv preprint arXiv:2011.10024*, 2020.

Srivastava, R. K., Shyam, P., Mutz, F., Jaśkowski, W., and Schmidhuber, J. Training agents using upside-down reinforcement learning. *arXiv preprint arXiv:1912.02877*, 2019.

Vamplew, P., Smith, B. J., Källström, J., Ramos, G., Rădulescu, R., Roijers, D. M., Hayes, C. F., Heintz, F., Mannion, P., Libin, P. J., et al. Scalar reward is not enough: A response to silver, singh, precup and sutton (2021). *Autonomous Agents and Multi-Agent Systems*, 36 (2):41, 2022.

van der Maaten, L. and Hinton, G. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008. URL `http://jmlr.org/papers/v9/vandermaaten08a.html`.

Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A., Chung, J., Choi, D. H., Powell, R., Ewalds, T., Georgiev, P., Oh, J., Horgan, D., Kroiss, M., Danihelka, I., Huang, A., Sifre, L., Cai, T., Agapiou, J. P., Jaderberg, M., Vezhnevets, A. S., Leblond, R., Pohlen, T., Dalibard, V., Budden, D., Sulsky, Y., Molloy, J., Paine, T. L., Gulcehre, C., Wang, Z., Pfaff, T., Wu, Y., Ring, R., Yogatama, D., Wünsch, D., McKinney, K., Smith, O., Schaul, T., Lillicrap, T., Kavukcuoglu, K., Hassabis, D., Apps, C., and Silver, D. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature*, 575 (7782):350–354, October 2019. ISSN 0028-0836, 1476-4687. doi: 10.1038/s41586-019-1724-z. URL `https://doi.org/10.1038/s41586-019-1724-z`.

Wainwright, M. J., Jordan, M. I., et al. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1–2):1–305, 2008.

Warnell, G., Waytowich, N. R., Lawhern, V. J., and Stone, P. Deep TAMER: Interactive agent shaping in high-dimensional state spaces. *arXiv: Artificial Intelligence*, 2017.

Watkins, C. J. C. H. Learning from delayed rewards. 1989.

Wattenberg, M., Viégas, F., and Johnson, I. How to use t-sne effectively. *Distill*, 2016. doi: 10.23915/distill.00002. URL `http://distill.pub/2016/misread-tsne`.

Yu, T., Thomas, G., Yu, L., Ermon, S., Zou, J. Y., Levine, S., Finn, C., and Ma, T. Mopo: Model-based offline policy optimization. *Advances in Neural Information Processing Systems*, 33:14129–14142, 2020.

Zhuang, Z., LEI, K., Liu, J., Wang, D., and Guo, Y. Behavior proximal policy optimization. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=3c13LptpIph`.

A. Appendix

A.1. Implementation details

A.1.1. CODEBASE.

Our code is based on Decision Transformer[2], and our implementation of OPPO is available at:

https://github.com/bkkgbkjb/OPPO

A.1.2. OPENAI GYM.

We choose the OpenAI Gym continuous control tasks from the D4RL benchmark (Fu et al., 2020). The different dataset settings are described below.

- Medium: 1 million timesteps generated by a ”medium” policy that achieves approximately one-third of the score of an expert policy.
- Medium-Replay: the replay buffer of an agent trained to the performance of a medium policy (approximately 25k-400k timesteps in our environments).
- Medium-Expert: 1 million timesteps generated by the medium policy concatenated with 1 million timesteps generated by an expert policy.

For details of these environments and datasets, please refer to D4RL for more information.

A.1.3. HYPERPARAMETERS

During the offline HIM phase, we weighted sum all three losses as in Eq.9 (with ratios listed in Table 7) and perform backpropagation, while in Preference Modeling phase, only $\mathcal{L}_{\text{PM}}$ is computed and backpropagated.

Table 7. Hyperparameters of coefficients of combined losses during Offline HIM.

Hyperparameter	Value
α	0.25 for halfcheetah-medium-expert 0.5 for others
β	0.05 for halfcheetah-medium-expert 0.1 for others

Our hyperparameters on all tasks are shown below in Table 8 and Table 9. Models were trained for 10^5 gradient steps using the AdamW optimizer Loshchilov & Hutter (2017) following PyTorch defaults.

Table 8. Hyperparameters of $\mathbf{z}^*$ searching for OpenAI Gym experiments.

Hyperparameter	Value
Number of dimensions	8 for halfcheetah 16 for others
Amount of feedback	50k
Type of optimizer	AdamW
Learning rate	10^{-2} for halfcheetah-medium-expert 10^{-3} for others
Weight decay	10^{-4}
Margin	1

[2] https://github.com/kzl/decision-transformer

Table 9. Hyperparameters of Transformer for OpenAI Gym experiments.

Hyperparameter	Value
Number of layers	3
Number of attention heads	2 for encoder transformer 1 for decision transformer
Embedding dimension	128
Nonlinearity function	ReLU
Batch size	64
context length K	20
Dropout	0.1
Learning rate	10^{-4}
Grad norm clip	0.25
Weight decay	10^{-4}
Learning rate decay	Linear warmup for first 10^5 training steps

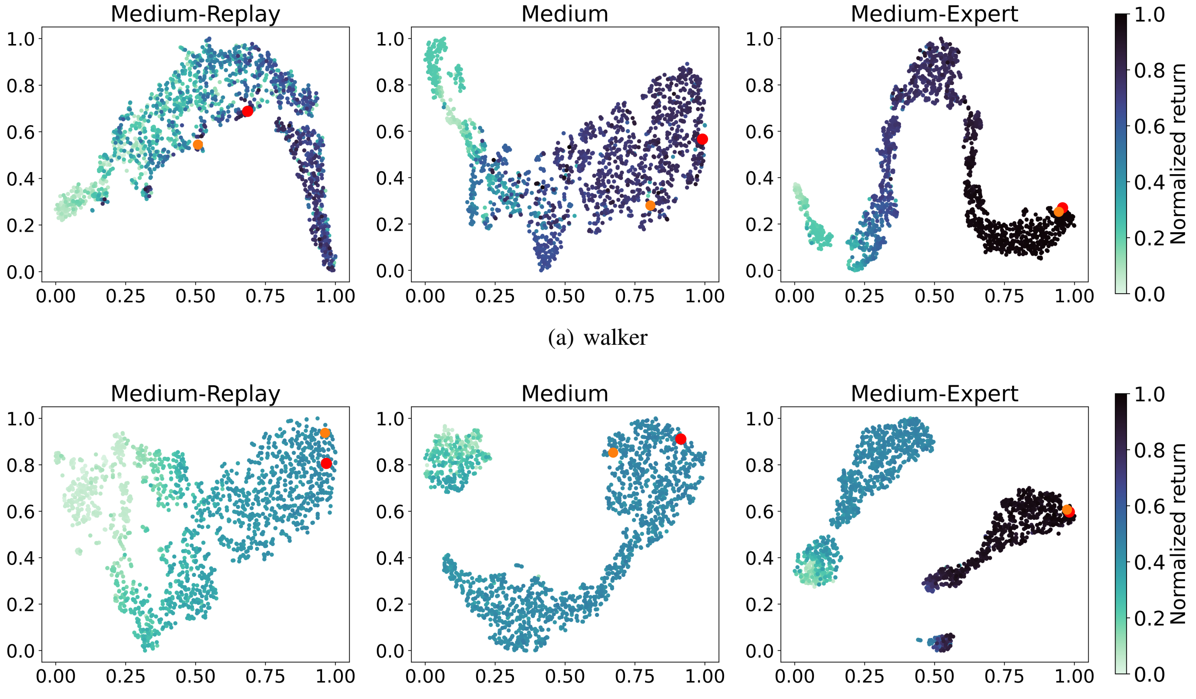
A.1.4. Computational resources.

The experiments were run on a computational cluster with 20x GeForce RTX 2080 Ti, and 4x NVIDIA Tesla V100 32GB for about 20 days.

A.2. Additional results

A.2.1. More visualization results on z-space.

We further show the t-sne results of OPPO in 5 with the setting described in Section 5.1 in Walker and HalfCheetah environments.



(a) walker

(b) halfcheetah

Figure 5. t-SNE of OPPO in Qalker and HalfCheetah including the embedding of random trajectories in $\mathcal{D}$, the colors of the points represent the normalized return of the corresponding trajectory τ.

Our primary purpose of using t-SNE is for visualization, to illustrate the structure of the learned $\mathbf{z}$-space in a more intuitive rather than quantitative manner. While t-SNE results are known to be hyper-parameter dependent (Wattenberg et al., 2016), we have included in Table 10 that listing the euclidean distances of $\mathbf{z}^*$ from $\mathbf{z}^{**}$ on different tasks.

Table 10. Euclidean distances of $\mathbf{z}^*$ from $\mathbf{z}^{**}$ on different tasks.

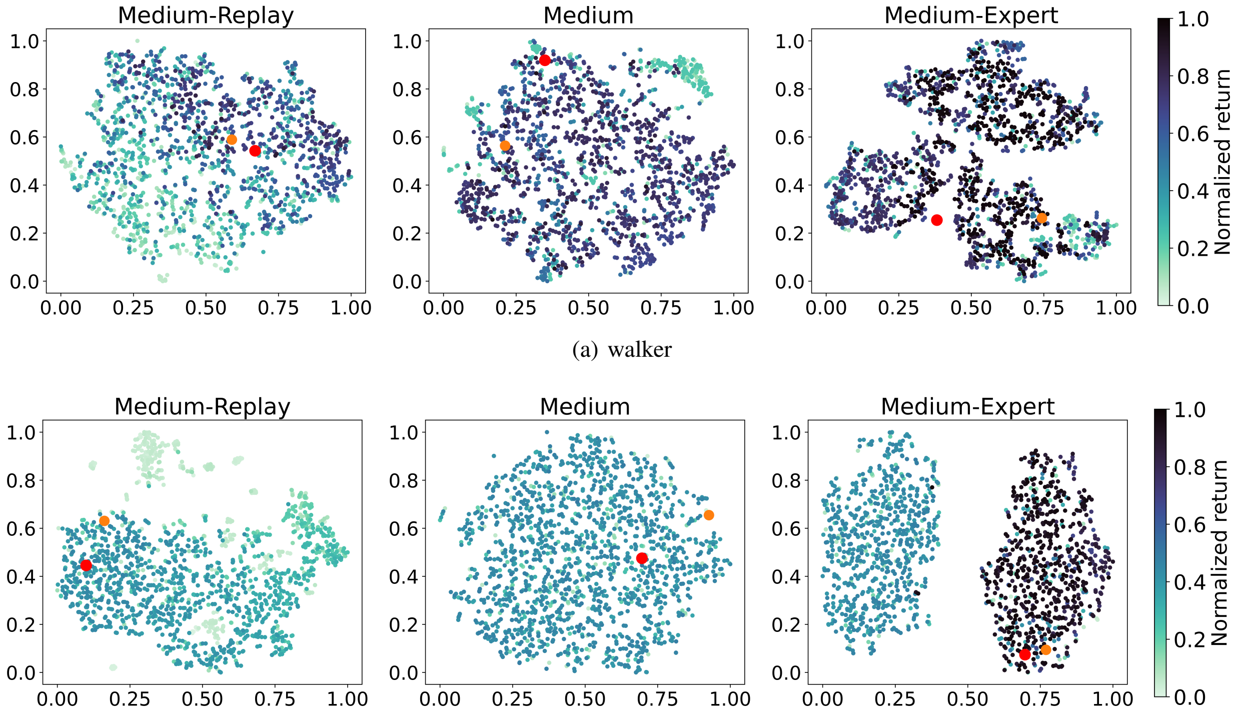
Environment	**Dataset**	$\|\|\mathbf{z}^* - \mathbf{z}^{**}\|\|_2$	**[Min, Lower quartile, Median, Upper quartile, Max]**	**Percentile Rank (PR)**
Hopper	Medium-Expert	12.68	[12.40, 13.29, 14.19, 15.01, 16.37]	97.9%
	Medium	15.50	[13.20, 13.92, 14.81, 15.70, 17.23]	30.1%
	Medium-Replay	13.33	[12.13, 13.24, 13.98, 14.83, 16.45]	72.0%
Walker	Medium-Expert	13.16	[12.62, 13.02, 14.13, 15.02, 16.40]	71.2%
	Medium	12.86	[12.01, 12.84, 13.21, 14.03, 15.79]	74.1%
	Medium-Replay	14.26	[10.90, 12.39, 13.14, 13.58, 15.18]	6.7%
HalfCheetah	Medium-Expert	10.42	[10.34, 10.82, 12.35, 12.71, 13.85]	99.7%
	Medium	3.22	[3.19, 3.40, 4.15, 4.54, 5.80]	99.9%
	Medium-Replay	1.89	[1.53, 1.98, 2.92, 3.69, 4.24]	79.9%

To provide some context, we also calculated the distances between the embeddings of trajectories in each dataset and $\mathbf{z}^*$, then gathered the minimum, lower quartile, median, upper quartile, and maximum values in Table 10. Additionally, we included percentile rankings for the distances between $\mathbf{z}^*$ and $\mathbf{z}^{**}$ within each dataset.

The results confirm the intuitions from the t-SNE plots, and the percentile rank maybe more informative.

A.2.2. More results of ablation study of one-step paradigm

We also show the t-sne results of the corresponding ablation study in 6 with the setting described in Section 5.4 in Walker and HalfCheetah environments.



(a) walker

(b) halfcheetah

Figure 6. t-SNE visualization of the embedding space learned with OPPO-a in walker and halfcheetah environments.

By comparing Fig.6 to Fig.5, we discover that the structure of $\mathbf{z}$-space significantly collapses in eight out of nine environments (except for halfcheetah medium-replay). More specifically, we can no longer recognize the distribution pattern and clusters that emerged in Fig.5, while such an observation is in line with our conclusion in the main text.

Table 11. Ablation study of one-step paradigm

Environment	**Dataset**	**OPPO**	**OPPO-a**
Hopper	Medium-Expert	**108.0** ± **5.1**	103.5 ± 4.4
	Medium	**86.3** ± **3.2**	69.2 ± 7.4
	Medium-Replay	**88.9** ± **2.3**	78.3 ± 7.1
Walker	Medium-Expert	**105.0** ± **2.4**	**108.8** ± **1.0**
	Medium	**85.0** ± **2.9**	80.7 ± 1.5
	Medium-Replay	**71.7** ± **4.4**	66.3 ± 1.6
HalfCheetah	Medium-Expert	**89.6** ± **0.8**	**90.1** ± **1.4**
	Medium	**43.4** ± **0.2**	**43.4** ± **0.2**
	Medium-Replay	**39.8** ± **0.2**	**39.6** ± **0.1**
Sum		**717.7**	679.8

However, it is also worth noting that the performance of OPPO-a in the D4RL benchmark is not hindered much by this uninformative $\mathbf{z}$-space, as shown in Table 11. We attribute this to the effectiveness of the preference modeling phase, where our method is still able to find a meaningful $\mathbf{z}^*$ in a less expressive $\mathbf{z}$-space.

This is also justified from t-SNE(Fig.6) as there our learned $\mathbf{z}^*$ (orange dot) locates just in the point of deep color.