BEYOND REWARD: OFFLINE PREFERENCE-GUIDED POLICY OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

In this work, we study offline preference-based reinforcement learning (PbRL), which relaxes the two fundamental supervisory signals in standard reinforcement learning (online accessible transition dynamics and rewards). In other words, the agent is provided with fixed offline trajectory transitions and human preferences between pairs of trajectories. Due to the orthogonality property of rewards and dynamics, one common practice is combining prior PbRL-based reward learning objectives with off-the-shelf offline RL algorithms to bridge preference modeling and offline learning. However, such two isolated optimizations require learning a separate reward function and thus place an information bottleneck on reward learning (the bridge). As an alternative, we propose offline preference-guided policy optimization (OPPO), an end-to-end offline PbRL formulation, which jointly learns to model the preference (for finding the optimal task policy) and the offline data (for eliminating OOD). In particular, OPPO introduces an offline hindsight information matching objective and a preference modeling objective. Then, iterating the two objectives over, we can directly extract a well-performing decision policy, avoiding a separate reward learning. We empirically show that OPPO can effectively model the offline preference and outperform prior competing baselines (including the offline RL algorithms performed over the true reward function).

1 INTRODUCTION

Deep reinforcement learning (RL) presents a flexible framework for learning task-oriented behaviors (Kohl & Stone, 2004; Kober & Peters, 2008; Kober et al., 2013; Silver et al., 2017; Kalashnikov et al., 2018; Vinyals et al., 2019), where the "task" is often expressed as maximizing the cumulative reward sum of trajectories generated by rollouting the learning policy in the corresponding environment. However, the above RL formulation implies two indispensable prerequisites for the training of decision policy: 1) an interactable environment and 2) a pre-specified reward function. Unfortunately, 1) online interaction with the environment can be costly and unsafe (Mihatsch & Neuneier, 2002; Hans et al., 2008; Garcıa & Fernández, 2015), and 2) designing a suitable reward function often requires expensive human effort, while the heuristic rewards often used are sometimes incapable of conveying the true intention (Hadfield-Menell et al., 2017).

To relax these requirements, previous works have either 1) focused on the offline RL formulation (the online rollout is inaccessible) (Fujimoto et al., 2019), where the learner has access to fixed offline trajectories along with a reward signal for each transition (or along with limited expert demonstrations), or 2) considered the (online) preference-based RL (PbRL) formulation (Christiano et al., 2017; Bıyık & Sadigh, 2018; Sadigh et al., 2017; Biyik et al., 2020; Lee et al., 2021a), where messages about the task objective are passed to the learner through preferences of a (human) annotator between two trajectories rather than rewards for each transition. To further progress in this setting, we propose relaxing both requirements and advocating for offline PbRL.

In the offline PbRL setting with access to an offline dataset and labeled preferences between offline trajectories, it is straightforward to combine previous (online) PbRL methods and off-the-shelf offline RL algorithms (Shin & Brown, 2021). As shown in Fig. 1 (left), we can first use the Bradley-Terry model (Bradley & Terry, 1952) to model the preference label and supervisedly learn a reward function (normally adopted in prior PbRL methods), and then train the policy with any offline RL algorithm on the transitions relabeled via the learned reward function. Intuitively, such a procedure

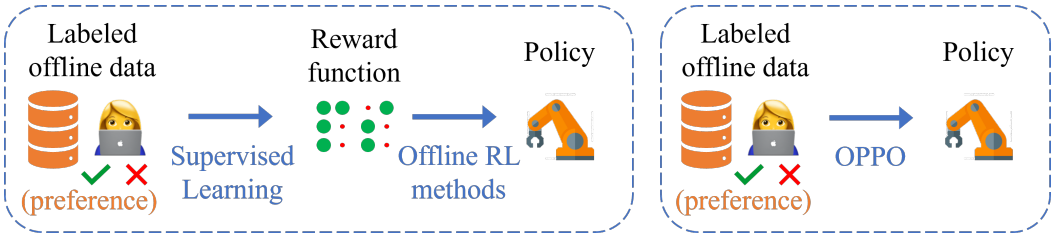


Figure 1: A flow diagram of previous offline PbRL algorithms (left) and our OPPO algorithm (right). Previous works require learning a separate reward function for modeling human preference using the Bradley-Terry model. In contrast, our OPPO directly learns the policy network.

allows preference modeling via a separate reward function. Fundamentally, however, learning a separate reward function that explains expert preferences does not directly instruct the policy how to act. Given that the objective of RL is to take actions optimally, prior PbRL algorithms are evaluated on the quality of the optimal actions implied by the rewards they learn. If the learned reward function is miscalibrated, an isolated policy optimization may learn to exploit loopholes in the relabeled rewards, resulting in unwanted behaviors. Then why must we learn a reward function considering it possibly being not able to directly yield optimal actions?

To this end, we propose offline preference-guided policy optimization (OPPO), an end-to-end formulation that jointly models the offline preference and learns the optimal decision policy without learning a separate reward function (as shown in Fig. 1 right). Specifically, we explicitly introduce a hindsight information encoder with which we further design an offline hindsight information matching objective and a preference modeling objective. Via optimizing the two objectives iteratively we derive a contextual policy $\pi(\mathbf{a}|\mathbf{s},\mathbf{z})$ to model the offline data and concurrently optimize an optimal contextual variable $\mathbf{z}^*$ to model the preference. Then, we arrive at the optimal policy by having the contextual policy $\pi(\mathbf{a}|\mathbf{s},\mathbf{z}^*)$ conditioned on the learned optimal $\mathbf{z}^*$.

In summary, our contributions include 1) OPPO: a simple, stable and end-to-end offline PbRL method that avoids learning a separate reward function, 2) an instance of preference-based hindsight information matching objective and a novel preference modeling objective over the contextual, and 3) extensive experiments to show and analyze the outstanding performance of OPPO against previous competitive baselines.

2 Related Work

Preference-based RL. Preference-based RL is also known as learning from human feedback. 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., 2021b; 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, Lee et al. (2021b) proposed a feedback-efficient RL algorithm by utilizing off-policy learning and pre-training. Park et al. (2022) used pseudo-labeling to utilize unlabeled segments and proposed a novel augmentation called temporal cropping to augment labeled data.

Offline RL. To mitigate the impact of distribution shift in offline RL, prior algorithms (a) constrain the action space (Fujimoto et al., 2019; Kumar et al., 2019a; Siegel et al., 2020), (b) incorporate value pessimism (Fujimoto et al., 2019; Kumar et al., 2020), 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). Shin & Brown (2021) tried to solve offline PbRL setting by simply combining previous (online) PbRL methods and off-the-shelf offline RL algorithms.

Supervised learning in RL. Some prior methods for reinforcement learning bear more resemblance to static supervised learning, such as Q-learning (Watkins, 1989; Mnih et al., 2013) and behavior cloning. In these cases, the resulting agent's performance is positively correlated to the quality of data used for training. On the other hand, Srivastava et al. (2019) and Kumar et al. (2019b) studied "upside-down" reinforcement learning (UDRL), seeking to model behaviors via a supervised loss conditioned on a target return. Ghosh et al. (2019) 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. Chen et al. (2021) and Janner et al. (2021) solved the problem via sequence modeling. 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). 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 can not only work with sub-optimal demonstrations but also reveal optimal behaviors without injecting human priors about the optimal demonstrations.

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 its policy $\pi(\mathbf{a}_t|\mathbf{s}_t)$. In the standard RL framework, the environment returns a reward r_t and the agent transitions 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 cannot 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 some pre-collected trajectories, where each trajectory τ^i consists of 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 some sub-optimal or even random behavior policies.

Further, the standard offline RL assumes the existence of reward signals for each state-action pair in $\mathcal{D}$. However, in the offline Preference-based RL (PbRL) framework, we do not assume such rewards are accessible. Instead, 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 accumulative reward over the trajectory labeled by an underlying reward function. The reward function is then updated by minimizing the following cross-entropy loss:

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

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 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 some (pre-defined) desired information statistics value $\mathbf{z}$:

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

where $p(\mathbf{z})$ is a prior, $\pi(\tau_{\mathbf{z}})$ 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 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; Liu et al., 2022; 2021). 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:

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

where $\mathbf{z} := I(\tau)$. In HER, we can set $I(\tau)$ as the final state in trajectory τ; 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 offline settings, sampling $\tau_{\mathbf{z}}$ from $\pi(\tau_{\mathbf{z}})$ is not accessible. Thus, we are required to 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.*, $\min_\pi \mathbb{E}_{\tau\sim\mathcal{D}(\tau),\tau_{\mathbf{z}}\sim\pi(\tau_{\mathbf{z}})}\left[\ell\left(\tau,\tau_{\mathbf{z}}\right)\right]$. Then, we provide the overall offline HIM objective:

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

To give an intuitive understanding of above objective, here 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 the generated $\tau_{\mathbf{z}}$ will reach the same return as $\tau = I^{-1}(\mathbf{z})$. However, in the offline setting, we must ensure the generated $\tau_{\mathbf{z}}$ stay in support of the offline data, eliminating the out-of-distribution (OOD) issues. 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 formalizes 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 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); 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), that adopts the hindsight information matching (HIM) objective in Section 3.2 to model an offline contextual policy $\pi(\mathbf{a}|\mathbf{s},\mathbf{z})$, and introduces a triplet loss to model the human preference as well as the optimal contextual variable $\mathbf{z}^*$. At testing, we condition the policy on the optimal $\mathbf{z}^*$ and thus conduct rollout with $\pi(\mathbf{a}|\mathbf{s},\mathbf{z}^*)$. In principle, OPPO can be paired with any PbRL settings, including

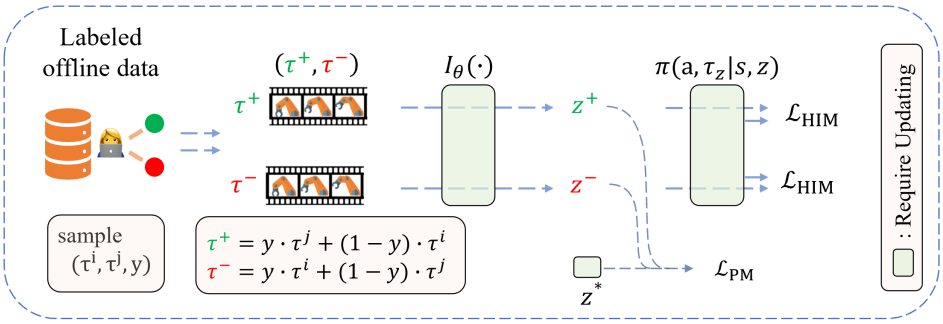


Figure 2: OPPO first maps offline trajectories (both positive τ^+ and negative τ^-) to a latent space via the hindsight information extractor I_θ. It then optimizes the offline HIM objective $\mathcal{L}_{\text{HIM}}$. Finally, the belief of the optimal hindsight information $\mathbf{z}^*$ is updated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

both online and offline algorithms. 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 offline 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 is a (learnable) network that encodes the hindsight information in PbRL, we formulate the following objective:

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

where $\mathbf{z} := I_\theta(\tau)$. Note that Eq. 6 is a different instantiation of Eq. 5 where 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 with Bi-directional Decision Transformer (BDT) proposed by (Furuta et al., 2021) for offline imitation learning. However, since expert demonstrations is missing in the PbRL setting, in Section 4.2, we propose to use the preference label in $\mathcal{D}_\succ$ to extract hindsight information.

4.2 Preference Modeling

To make the hindsight information $I_\theta(\tau)$ in Eq. 6 to 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:

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

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 of the above two similarities ($-\ell(\mathbf{z}^*, \mathbf{z}^i)$ vs. $-\ell(\mathbf{z}^*, \mathbf{z}^j)$) and setting the trajectory with high 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 $\mathbf{z}^-$.

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 contextual embedding $\mathbf{z}^*$.
2: **while** not converged **do**
3: Sample a batch of trajectories from $\mathcal{D}$: $\{\tau\}_{\text{B}} \sim \mathcal{D}$.
4: Update $\pi(\mathbf{a}|\mathbf{s}, \mathbf{z})$ and $I_\theta(\cdot)$ with sampled $\{\tau\}_{\text{B}}$ using $\mathcal{L}_{\text{HIM}}$.
5: Sample a batch of preferences from $\mathcal{D}_{\succ}$: $\{(\tau^i, \tau^j, y)\}_{\text{B}} \sim \mathcal{D}_{\succ}$.
6: Update $I_\theta(\cdot)$ and the optimal $\mathbf{z}^*$ with sampled $\{(\tau^i, \tau^j, y)\}_{\text{B}}$ using $\mathcal{L}_{\text{PM}}$.
7: **end while**

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

$$\min_{\mathbf{z}^*, I_\theta} \mathcal{L}_{\text{PM}} := \mathbb{E}_{(\tau^i, \tau^j, y) \sim \mathcal{D}_{\succ}} \Big[\max(\ell(\mathbf{z}^*, \mathbf{z}^+) - \ell(\mathbf{z}^*, \mathbf{z}^-) + \text{margin}, 0) \Big]. \tag{8}$$

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 provides hints for the encoder to extract more valuable features, while 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.

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 variable, $\mathbf{z}^*$, 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 separately learning a reward function. Compared with the HIM-based offline RL algorithms (*e.g.*, Decision Transformer (Chen et al., 2021) and RvS-G (Emmons et al., 2021)), at the testing phase, OPPO does not need to manually specify the target contextual variable for the rollout policy $\pi(\mathbf{a}|\mathbf{s}, \cdot)$.

4.3 Architecture & Implementation Details

As shown in Fig. 2, OPPO models the hindsight information extractor I_θ as an encoder network $I_\theta : \tau \to \mathbf{z}$, we use the BERT architecture. And similar to 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.

Algorithm 1 details the training of OPPO. The entire process is summarized as follows: 1) we sample a batch of trajectories from the dataset $\mathcal{D}$, 2) use Eq. 6 to update $\pi(\mathbf{a}|\mathbf{s}, \mathbf{z})$ and $I_\theta(\cdot)$ based on sampled trajectories, 3) then sample a batch of preferences from the labeled dataset $\mathcal{D}_{\succ}$, and finally, 4) update $I_\theta(\cdot)$ and the posterior of $\mathbf{z}^*$ based on sampled $\{(\tau^i, \tau^j, y)\}_{\text{B}}$ using Eq. 8.

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 these types of preferences? In other words, does the learned $\mathbf{z}$-space (encoded by the learned $I_\theta(\cdot)$) align with given preferences?

2) Can the learned optimal contextual policy $\pi(\mathbf{a}|\mathbf{s}, \mathbf{z}^*)$ obtain better performance than policy $\pi(\mathbf{a}|\mathbf{s}, \mathbf{z})$ that is conditioned on all the other contextual variables $\mathbf{z} \in \{I_\theta(\tau)|\tau \in \mathcal{D}\}$?

3) Can OPPO also achieve competitive performance against other offline (PbRL) algorithms?
4) What benefits, if any, do we gain from designing the end-to-end offline preference learning, *i.e.*, iteratively conducting offline data modeling (Equation 6) and preference modeling (Equation 8)?

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 environments, with medium, medium-replay, medium-expert as dataset for each environment. To build our labeled offline preference dataset $\mathcal{D}_{\succ} := \{(\tau^i, \tau^j, y)\}$, following prior PbRL benchmark (Lee et al., 2021a), we use the following stochastic preference model (to simulate a human annotator and) to label preference between trajectories (τ^i, τ^j):

$$y \sim P[\tau^i \succ \tau^j; \beta, \gamma] = \text{logistic}(\beta \sum_{t=1}^{H} \gamma^{H-t} r(\mathbf{s}_t^i, \mathbf{a}_t^i) - \beta \sum_{t=1}^{H} \gamma^{H-t} r(\mathbf{s}_t^j, \mathbf{a}_t^j)), \tag{9}$$

where factors γ and β are used to model myopic and rational behaviors (of human annotator) respectively and are both set to 1 in our experiment.

5.1 Does the learned $\mathbf{z}$-space align with given preferences?

In this subsection, we probe that OPPO can enable well-aligned preference over the $\mathbf{z}$-space encoded by the learned I_θ. We first sample random trajectories from offline dataset $\mathcal{D}$, encode them with 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 with an orange dot. Besides, we also mark the (embedding of) optimal trajectory in 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, 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.

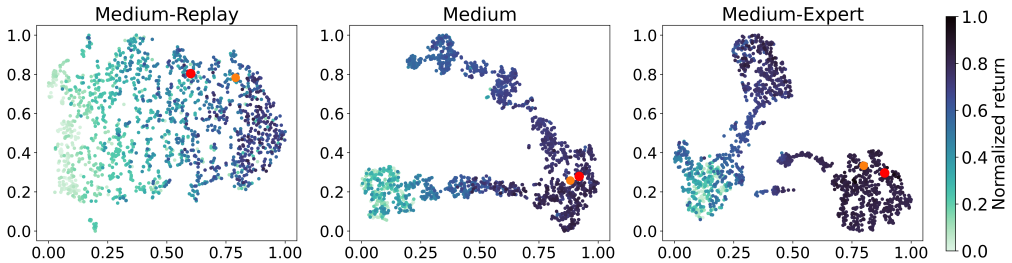


Figure 3: We utilize t-SNE to visualize the $\mathbf{z}$-space learned in Hopper environment, encoded with a well-trained $I_\theta(\cdot)$, including the embedding of random trajectories in $\mathcal{D}$, our learned $\mathbf{z}^*$ ("orange dot") and the actual optimal $\mathbf{z}^{**}$ ("red dot"), 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.

5.2 Can $\pi(\mathbf{a}|\mathbf{s}, \mathbf{z})$ conditioned the learned $\mathbf{z}^*$ enable better performance?

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

Table 1: Comparison of (normalized) performance when rollouting the contextual policy $\pi(\mathbf{a}|\mathbf{s}, \cdot)$ when conditioning on different embeddings ($\mathbf{z}^*$, $\mathbf{z}_{\text{high}}$, and $\mathbf{z}_{\text{low}}$).

Environment	Dataset	$\mathbf{z}^*$	$\mathbf{z}_{\text{high}}$	$\mathbf{z}_{\text{low}}$
Hopper	Medium-Expert	**107.8** $\pm$ 1.0	94.2 $\pm$ 24.3	79.1 $\pm$ 28.8
	Medium	**67.4** $\pm$ 3.5	55.8 $\pm$ 7.9	51.6 $\pm$ 13.8
	Medium-Replay	**88.2** $\pm$ 1.7	78.6 $\pm$ 26.3	26.6 $\pm$ 15.2
Walker	Medium-Expert	**106.8** $\pm$ 2.8	106.5 $\pm$ 9.1	93.4 $\pm$ 7.4
	Medium	**81.6** $\pm$ 2.8	64.9 $\pm$ 24.9	72.6 $\pm$ 10.6
	Medium-Replay	**70.6** $\pm$ 5.1	55.7 $\pm$ 24.8	6.8 $\pm$ 1.7
Halfcheetah	Medium-Expert	**89.6** $\pm$ 0.8	48.3 $\pm$ 14.4	42.6 $\pm$ 2.6
	Medium	**43.4** $\pm$ 0.2	42.5 $\pm$ 3.9	42.4 $\pm$ 3.2
	Medium-Replay	**39.8** $\pm$ 0.2	35.6 $\pm$ 8.5	33.9 $\pm$ 9.2
Sum		**695.3**	581.9	448.9

To show that, we compare the performance of rollouts by the contextual policy $\pi(\mathbf{a}|\mathbf{s}, \cdot)$ when conditioning on different embeddings. In Table 1, we choose 3 contextual embeddings: $\mathbf{z}^*$, $\mathbf{z}_{\text{high}}$ (embedding of a trajectory with the highest return in $\mathcal{D}$), and $\mathbf{z}_{\text{low}}$ (embedding of a trajectory with the lowest return in $\mathcal{D}$) and provide rollout performance (averaged over 3 seeds). We discover that the contextual $\pi(\mathbf{a}|\mathbf{s}, \mathbf{z})$ conditioned on $\mathbf{z}$ with high (or low) return (of corresponding trajectory $\tau = I_\theta^{-1}(\mathbf{z})$) obtain 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 conditioning on the learned optimal $\mathbf{z}^*$, $\pi(\mathbf{a}|\mathbf{s}, \mathbf{z}^*)$ produces the best performance over that conditioned other 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 better than any trajectory in offline dataset.

5.3 Can OPPO achieve competitive performance on offline (PbRL) benchmark?

We have shown that OPPO produces a near-optimal embedding $\mathbf{z}^*$, 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 offline (PBRL) benchmark. For comparison, we introduce 3 offline (PbRL) baselines: 1) DT+r: performing Decision Transformer with ground-truth reward function; 2) DT+r_ψ: performing Decision Transformer with a learned reward function (using Eq. 2); 3) CQL+r: performing CQL with ground-truth reward function;

Table 2: Performance comparison between OPPO and 3 offline (PbRL) baselines (DT+r, DT+r_ψ, and CQL+r) in D4RL Gym-Mujoco tasks, where results are reported over 3 seeds.

Environment	Dataset	Ours	DT+r	DT+r_ψ	CQL+r	BC
Hopper	Medium-Expert	107.8 $\pm$ 1.0	107.6 $\pm$ 1.8	95.6 $\pm$ 27.3	**111.0**	79.6
	Medium	67.4 $\pm$ 3.5	67.6 $\pm$ 1.0	**73.3** $\pm$ 3.0	58.0	63.9
	Medium-Replay	**88.2** $\pm$ 1.7	82.7 $\pm$ 7.0	72.5 $\pm$ 22.2	48.6	27.6
Walker	Medium-Expert	106.8 $\pm$ 2.8	108.1 $\pm$ 0.2	**109.7** $\pm$ 0.1	98.7	36.6
	Medium	**81.6** $\pm$ 2.8	74.0 $\pm$ 1.4	81.1 $\pm$ 2.1	79.2	77.3
	Medium-Replay	70.6 $\pm$ 5.1	66.6 $\pm$ 3.0	**80.4** $\pm$ 4.4	26.7	36.9
HalfCheetah	Medium-Expert	**89.6** $\pm$ 0.8	86.8 $\pm$ 1.3	88.4 $\pm$ 0.7	62.4	59.9
	Medium	43.4 $\pm$ 0.2	42.6 $\pm$ 0.1	43.2 $\pm$ 0.2	**44.4**	43.1
	Medium-Replay	39.8 $\pm$ 0.2	36.6 $\pm$ 0.8	38.8 $\pm$ 0.3	**46.2**	4.3
Sum		**695.3**	672.6	683.0	575.2	429.2

In Table 2, we provide our comparison results. 1) OPPO has retained a comparable performance against the Decision Transformer trained using true rewards. Consider that OPPO is a PbRL approach requiring only (human) preference, wich is a more flexible and straightforward form of supervision in the real world. 2) Although DT+r_ψ also shows competitive result in these benchmarks, such method need target return-to-go value determined by human prior as required by DT. [1]. However, our method, by searching across z-space, avoids the need of such prior target. We argue that this searching method brings advantages as rewards is usually hard to obtain in real-world RL applications. And preference is the only information easily accessible for training and **deploying** an RL method.

5.4 Do we get any benefits by iterating offline HIM and preference modeling?

Here, we conduct the ablation study to understand better the benefit of iterating $\mathcal{L}_{\text{HIM}}$ and $\mathcal{L}_{\text{PM}}$ (for updating I_θ). For ablation study, we removed I_θ from $\partial\mathcal{L}_{\text{PM}}/\partial\theta$ and are left with gradient of the optimal embedding $\mathbf{z}^*$ only in Eq. 8. Then, we continue to visualize the embedding $\mathbf{z}$-space for this ablation setting (OPPO-a), and we show the t-SNE visualization 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 3, the comparison results of medium-replay tasks confirm that such an ablation does cause performance degradation.

Table 3: Ablation study

Task	OPPO	OPPO-a
Hopper	**88.2** $\pm$ 1.7	78.3 $\pm$ 7.1
Walker	**70.6** $\pm$ 5.1	66.3 $\pm$ 1.6
HalfCheetah	**39.8** $\pm$ 0.2	39.6 $\pm$ 0.1
Sum	**198.6**	184.2

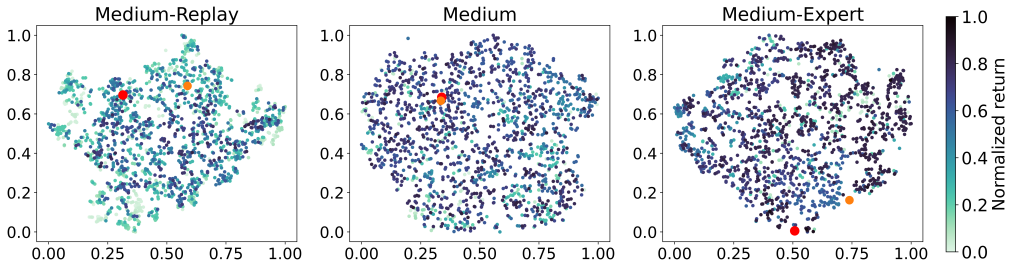


Figure 4: t-SNE visualization of the embedding space learned with OPPO-a in Hopper environment.

6 Conclusion

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

Through the visualization, we demonstrate that the $\mathbf{z}$-space learned by the encoder is informative and visually interpretable. Besides, the ablation study proves that a preference-guided embedding space could improve task performance asymptotically by a non-neglectable margin. Moreover, OPPO can find a contextual variable to represent the embedding 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 method 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.

[1] Preference-based relabelled rewards only participate in 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.

REFERENCES

Anurag Ajay, Aviral Kumar, Pulkit Agrawal, Sergey Levine, and Ofir Nachum. Opal: Offline primitive discovery for accelerating offline reinforcement learning. *arXiv preprint arXiv:2010.13611*, 2020.

Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, OpenAI Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. *Advances in neural information processing systems*, 30, 2017.

Dilip Arumugam, Jun Ki Lee, Sophie Saskin, and Michael L. Littman. Deep reinforcement learning from policy-dependent human feedback. *arXiv: Learning*, 2019.

Erdem Biyik, Nicolas Huynh, Mykel Kochenderfer, and Dorsa Sadigh. Active preference-based Gaussian process regression for reward learning. In *Robotics: Science and Systems XVI*. Robotics: Science and Systems Foundation, July 2020. doi: 10.15607/rss.2020.xvi.041. URL `https://doi.org/10.15607/rss.2020.xvi.041`.

Ralph Allan Bradley and Milton E. Terry. 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`.

Erdem Bıyık and Dorsa Sadigh. Batch active preference-based learning of reward functions. *arXiv: Learning*, 2018.

Víctor Campos, Alexander Trott, Caiming Xiong, Richard Socher, Xavier Giró-i Nieto, and Jordi Torres. Explore, discover and learn: Unsupervised discovery of state-covering skills. In *International Conference on Machine Learning*, pp. 1317–1327. PMLR, 2020.

Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Misha Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems*, 34:15084–15097, 2021.

Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *International conference on machine learning*, pp. 1691–1703. PMLR, 2020.

Xiaoyu Chen, Han Zhong, Zhuoran Yang, Zhaoran Wang, and Liwei Wang. Human-in-the-loop: Provably efficient preference-based reinforcement learning with general function approximation. *arXiv preprint arXiv:2205.11140*, 2022.

Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *arXiv: Machine Learning*, 2017.

Scott Emmons, Benjamin Eysenbach, Ilya Kostrikov, and Sergey Levine. Rvs: What is essential for offline rl via supervised learning? *arXiv preprint arXiv:2112.10751*, 2021.

Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is all you need: Learning skills without a reward function. *arXiv preprint arXiv:1802.06070*, 2018.

Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4rl: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*, 2020.

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

Hiroki Furuta, Yutaka Matsuo, and Shixiang Shane Gu. Generalized decision transformer for offline hindsight information matching. *arXiv preprint arXiv:2111.10364*, 2021.

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

Dibya Ghosh, Abhishek Gupta, Justin Fu, Ashwin Reddy, Coline Devin, Benjamin Eysenbach, and Sergey Levine. Learning to reach goals without reinforcement learning. 2019.

Dylan Hadfield-Menell, Smitha Milli, Pieter Abbeel, Stuart J Russell, and Anca Dragan. Inverse reward design. *Advances in neural information processing systems*, 30, 2017.

Alexander Hans, Daniel Schneegaß, Anton Maximilian Schäfer, and Steffen Udluft. Safe exploration for reinforcement learning. In *ESANN*, pp. 143–148. Citeseer, 2008.

Borja Ibarz, Jan Leike, Tobias Pohlen, Geoffrey Irving, Shane Legg, and Dario Amodei. Reward learning from human preferences and demonstrations in Atari. In *Neural Information Processing Systems*, 2018.

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

Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, and Sergey Levine. QT-opt: Scalable deep reinforcement learning for vision-based robotic manipulation. In *Computer Vision and Pattern Recognition*, 2018.

Rahul Kidambi, Aravind Rajeswaran, Praneeth Netrapalli, and Thorsten Joachims. Morel: Model-based offline reinforcement learning. *Advances in neural information processing systems*, 33: 21810–21823, 2020.

W. Bradley Knox and Peter Stone. 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.

W Bradley Knox, Stephane Hatgis-Kessell, Serena Booth, Scott Niekum, Peter Stone, and Alessandro Allievi. Models of human preference for learning reward functions. *arXiv preprint arXiv:2206.02231*, 2022.

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

Jens Kober, J. Andrew Bagnell, and Jan Peters. 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.

N. Kohl and P. Stone. 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.

Aviral Kumar, Justin Fu, Matthew Soh, George Tucker, and Sergey Levine. Stabilizing off-policy q-learning via bootstrapping error reduction. *Advances in Neural Information Processing Systems*, 32, 2019a.

Aviral Kumar, Xue Bin Peng, and Sergey Levine. Reward-conditioned policies. *arXiv preprint arXiv:1912.13465*, 2019b.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 33:1179–1191, 2020.

Kimin Lee, Laura Smith, Anca Dragan, and Pieter Abbeel. B-pref: Benchmarking preference-based reinforcement learning. 2021a.

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

Jinxin Liu, Hao Shen, Donglin Wang, Yachen Kang, and Qiangxing Tian. Unsupervised domain adaptation with dynamics-aware rewards in reinforcement learning. *Advances in Neural Information Processing Systems*, 34:28784–28797, 2021.

Jinxin Liu, Donglin Wang, Qiangxing Tian, and Zhengyu Chen. 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, 2022.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Kevin Lu, Aditya Grover, Pieter Abbeel, and Igor Mordatch. Reset-free lifelong learning with skill-space planning. *arXiv preprint arXiv:2012.03548*, 2020.

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

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Jongjin Park, Younggyo Seo, Jinwoo Shin, Honglak Lee, Pieter Abbeel, and Kimin Lee. Surf: Semi-supervised reward learning with data augmentation for feedback-efficient preference-based reinforcement learning. *arXiv preprint arXiv:2203.10050*, 2022.

Keiran Paster, Sheila A McIlraith, and Jimmy Ba. Planning from pixels using inverse dynamics models. *arXiv preprint arXiv:2012.02419*, 2020.

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

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

Dorsa Sadigh, Anca Dragan, Shankar Sastry, and Sanjit Seshia. Active preference-based learning of reward functions. In *Robotics: Science and Systems XIII*. Robotics: Science and Systems Foundation, July 2017. doi: 10.15607/rss.2017.xiii.053. URL https://doi.org/10.15607/rss.2017.xiii.053.

Archit Sharma, Shixiang Gu, Sergey Levine, Vikash Kumar, and Karol Hausman. Dynamics-aware unsupervised discovery of skills. *arXiv preprint arXiv:1907.01657*, 2019.

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

Noah Y Siegel, Jost Tobias Springenberg, Felix Berkenkamp, Abbas Abdolmaleki, Michael Neunert, Thomas Lampe, Roland Hafner, Nicolas Heess, and Martin Riedmiller. Keep doing what worked: Behavioral modelling priors for offline reinforcement learning. *arXiv preprint arXiv:2002.08396*, 2020.

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel, and Demis Hassabis. 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.

Avi Singh, Huihan Liu, Gaoyue Zhou, Albert Yu, Nicholas Rhinehart, and Sergey Levine. Parrot: Data-driven behavioral priors for reinforcement learning. *arXiv preprint arXiv:2011.10024*, 2020.

Rupesh Kumar Srivastava, Pranav Shyam, Filipe Mutz, Wojciech Jaśkowski, and Jürgen Schmidhuber. Training agents using upside-down reinforcement learning. *arXiv preprint arXiv:1912.02877*, 2019.

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

Oriol Vinyals, Igor Babuschkin, Wojciech M. Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H. Choi, Richard Powell, Timo Ewalds, Petko Georgiev, Junhyuk Oh, Dan Horgan, Manuel Kroiss, Ivo Danihelka, Aja Huang, Laurent Sifre, Trevor Cai, John P. Agapiou, Max Jaderberg, Alexander S. Vezhnevets, Rémi Leblond, Tobias Pohlen, Valentin Dalibard, David Budden, Yury Sulsky, James Molloy, Tom L. Paine, Caglar Gulcehre, Ziyu Wang, Tobias Pfaff, Yuhuai Wu, Roman Ring, Dani Yogatama, Dario Wünsch, Katrina McKinney, Oliver Smith, Tom Schaul, Timothy Lillicrap, Koray Kavukcuoglu, Demis Hassabis, Chris Apps, and David Silver. 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.

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

Garrett Warnell, Nicholas R. Waytowich, Vernon J. Lawhern, and Peter Stone. Deep TAMER: Interactive agent shaping in high-dimensional state spaces. *arXiv: Artificial Intelligence*, 2017.

Christopher John Cornish Hellaby Watkins. Learning from delayed rewards. 1989.

Tianhe Yu, Garrett Thomas, Lantao Yu, Stefano Ermon, James Y Zou, Sergey Levine, Chelsea Finn, and Tengyu Ma. Mopo: Model-based offline policy optimization. *Advances in Neural Information Processing Systems*, 33:14129–14142, 2020.

A APPENDIX

A.1 IMPLEMENTATION DETAILS

Codebase. Our code is based on Decision Transformer: `https://github.com/kzl/decision-transformer`. We provide our source code in the supplementary material.

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 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.

Training Objectives In our experiment, we consolidate ℓ in Equation 6 as MSELoss and in Equation 8 as Euclidean Distance. Also, we add a normalization loss $\mathcal{L}_{\text{norm}}$ to constraint the L2 norm of all embeddings generated by hindsight information extractor I_θ.

$$\mathcal{L}_{\text{total}} := \mathcal{L}_{\text{HIM}} + \alpha\mathcal{L}_{\text{PM}} + \beta\mathcal{L}_{\text{norm}} \tag{10}$$

During Offline HIM phase, we weighted sum all these 3 losses as in Equation 10 (with ratios listed in Table 4) and perform backpropagation, while in Preference Modeling phase, only $\mathcal{L}_{\text{PM}}$ is computed and backpropagated.

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

Table 4: Hyperparameters of coefficients of combined losses during Offline HIM.

Hyperparameters Our hyperparameters on all tasks are shown below in Tab. 5 and Tab. 6. Models were trained for 10^5 gradient steps using the AdamW optimizer Loshchilov & Hutter (2017) following PyTorch defaults.

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

Table 5: Hyperparameters of $\mathbf{z}^*$ searching for OpenAI Gym experiments.

Computational resources. The experiments were run on a computational cluster with 22x GeForce RTX 2080 Ti, and 4x NVIDIA Tesla V100 32GB for 20 days.

A.2 ADDITIONAL RESULTS

More visualization results on z-space. We further show the t-sne results of OPPO and the corresponding ablation study in the setting described in Section 5.1 and Section 5.4 in walker and

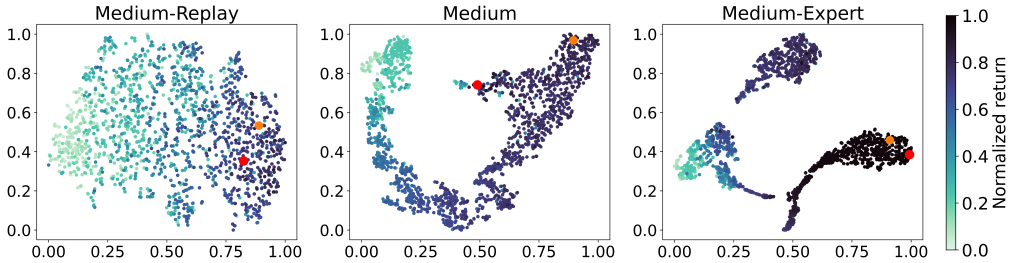
Hyperparameter	Value
Number of layers	3
Number of attention heads	2 for encoder transformer 1 for decision transformer
Embedding dimension	128
Nonlinearity function	ReLU
Batch size	64
context length K	20
Dropout	0.1
Learning rate	10^{-4}
Grad norm clip	0.25
Weight decay	10^{-4}
Learning rate decay	Linear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

Table 6: Hyperparameters of Transformer for OpenAI Gym experiments.

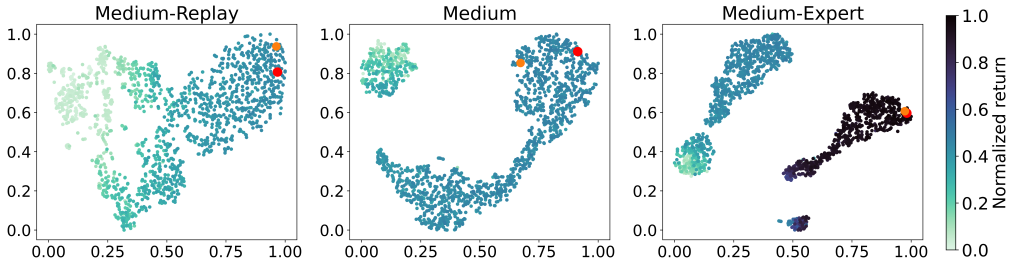
halfcheetah environments. By comparing Fig.6 to Fig.5, we discover that the structure of $\mathbf{z}$-space significantly collapses in eight out of nine environments (except for halfcheetah medium-replay). More specifically, we can no longer recognize the distribution pattern and clusters emerged in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

However, it is also worth noting that the performance of OPPO-a in D4RL benchmark, is not hindered much by this uninformative z-space, as shown in 7. We attribute this to the effectiveness of preference modelling phase, where our method is still able to find a meaningful $\mathbf{z}^*$ in a less expressive z-space.

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

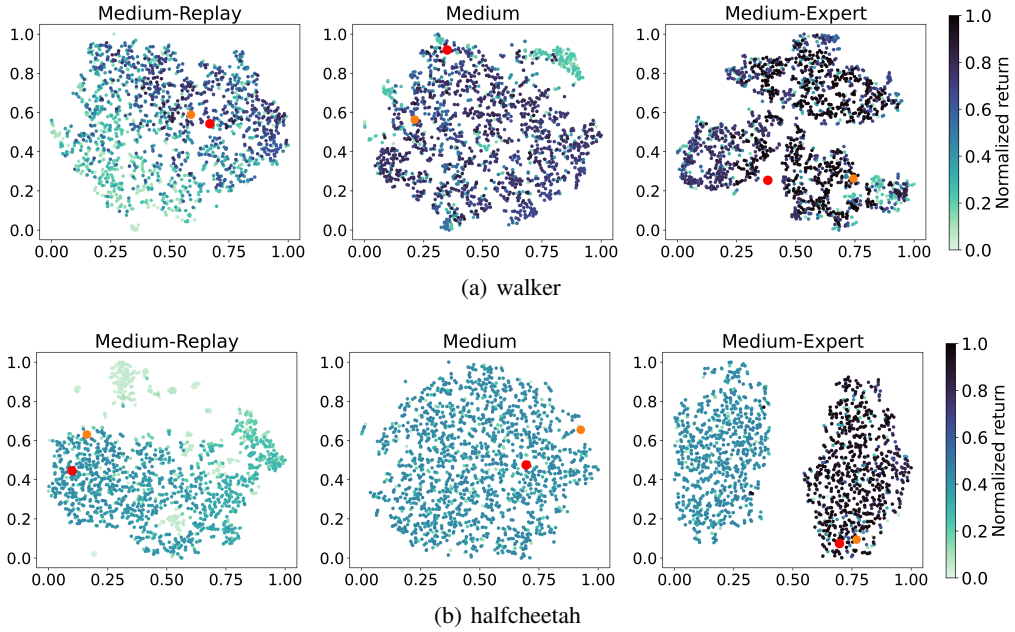


(a) walker



(b) halfcheetah

Figure 5: t-SNE of OPPO in walker and halfcheetah environments, 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 RL method. Color of the points represents the normalized return of the corresponding trajectory τ.



(a) walker

(b) halfcheetah

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

Environment	Dataset	OPPO(Ours)	OPPO(as)
Hopper	Medium-Expert	**107.8** $\pm$ 1.0	103.5 $\pm$ 4.4
	Medium	67.4 $\pm$ 3.5	**69.2** $\pm$ 7.4
	Medium-Replay	**88.2** $\pm$ 1.7	78.3 $\pm$ 7.1
Walker	Medium-Expert	106.8 $\pm$ 2.8	**108.8** $\pm$ 1.0
	Medium	**81.6** $\pm$ 2.8	80.7 $\pm$ 1.5
	Medium-Replay	**70.6** $\pm$ 5.1	66.3 $\pm$ 1.6
HalfCheetah	Medium-Expert	89.6 $\pm$ 0.8	**90.1** $\pm$ 1.4
	Medium	**43.4** $\pm$ 0.2	**43.4** $\pm$ 0.2
	Medium-Replay	**39.8** $\pm$ 0.2	39.6 $\pm$ 0.1
Sum		**695.3**	679.8

Table 7: Ablation study