HIP-RL: Hallucinated Inputs for Preference-based Reinforcement Learning in Continuous Domains

Chen Bo Calvin Zhang [1] **Giorgia Ramponi** [2]

Abstract

Preference-based Reinforcement Learning (PbRL) enables agents to learn policies based on preferences between trajectories rather than explicit reward functions. Previous approaches to PbRL are either experimental and successfully used in real-world applications but lack theoretical understanding, or they have strong theoretical guarantees but only for tabular settings. In this work, we propose a novel practical PbRL algorithm in the continuous domain called Hallucinated Inputs Preference-based RL (HIP-RL) which filled the gap between theory and practice. HIP-RL reparametrizes the set of transition models and uses hallucinated inputs to facilitate optimistic exploration in continuous state-action spaces by controlling the epistemic uncertainty. We construct regret bounds for HIP-RL and show that they are sublinear for Gaussian Process dynamic and reward models. Moreover, we experimentally demonstrate the effectiveness of HIP-RL.

1. Introduction

Reinforcement Learning (RL) (Sutton & Barto, 2018) showed promising results in recent years, in games (Tesauro et al., 1995; Mnih et al., 2015), robotics (Kober et al., 2013), and industrial (Wang & Usher, 2005) and medical problems (Zhao et al., 2011). However, one of the main challenges in using RL to solve real-world problems is in defining the reward function, since the learned policy can often be very sensitive to small variations. Consequently, selecting the correct reward function becomes crucial in training a good agent, often requiring extensive reward engineering efforts. In general, the design is demanding mainly for two reasons.

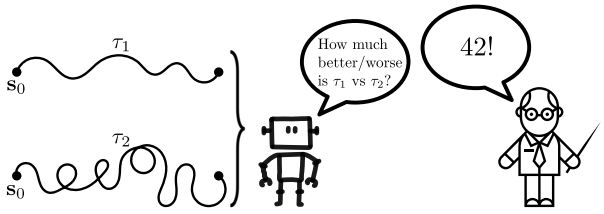


Figure 1. An illustration of PbRL. The agent only receives relative preference feedback, from which it has to learn an optimal control policy.

Firstly, determining the appropriate weighting for multiple desired goals presents a challenge. Balancing these goals effectively becomes crucial in achieving a satisfactory outcome. Secondly, the mathematical definition of abstract and vaguely defined objectives poses another obstacle (Christiano et al., 2017).

An easier way is to ask humans to provide feedback on a trajectory. This is what is called Preference-based RL (PbRL) (Wirth et al., 2017; Busa-Fekete et al., 2014). Rather than relying on explicit reward functions, PbRL enables agents to learn an optimal policy based on preferences between trajectories.

Despite the success of PbRL in many tasks, including robotics (Christiano et al., 2017) and games (Wirth & Fürnkranz, 2013), there is still little theoretical understanding. Novoseller et al. (2020); Xu et al. (2020); Pacchiano et al. (2021) study the tabular setting and provide convergence guarantees, however, their complexity bounds scale polynomially with the cardinality of the state-action space, making them not suitable in many practical applications. The first work on continuous state-action space was conducted by Chen et al. (2022), who prove a regret bound when employing general function approximators to learn both the transition and reward functions. However, their algorithm is intractable to implement in practice as it requires the construction of high-confidence sets over the transition, preference, and policy spaces, which is generally not computationally feasible.

Recent work on Model-Based RL (MBRL) using probabilis-

[1]Department of Computer Science, ETH Zurich, Zurich, Switzerland [2]ETH AI Center, ETH Zurich, Zurich, Switzerland. Correspondence to: Chen Bo Calvin Zhang <zhangca@ethz.ch>.

The Many Facets of Preference Learning Workshop at the International Conference on Machine Learning (ICML), Honolulu, Hawaii, USA, 2023.

tic dynamical models has shown great sample efficiency (Chua et al., 2018). These algorithms alternate between two phases: first, a policy is rolled out to collect data about the transition model, then this data is used to simulate transitions and optimize a policy over them. One of the reasons for the success of recent MBRL algorithms can be attributed to the ability to distinguish aleatoric and epistemic uncertainty when learning the model (Gal et al., 2016). This was however not done when optimizing the policy until recently. Curi et al. (2020); Kakade et al. (2020); Abeille & Lazaric (2020); Neu & Pike-Burke (2020) use the principle of Optimism-in-the-Face-of-Uncertainty (OFU) to perform provable optimistic exploration in MBRL. For example, Curi et al. (2020) augments the action space of the agent with an additional hallucinated control that allows the agent to control the epistemic uncertainty in the next state. In our work, we extended these concepts from RL to the PbRL setting.

Contributions In this paper, we investigate regret minimization for PbRL in continuous state-action spaces, with an unknown dynamical model and preference function.

This paper makes the following contributions:

1. We propose a novel algorithm for PbRL in the continuous state-action space setting called Hallucinated Inputs Preference-based RL (HIP-RL). By leveraging the concept of hallucinated inputs (Curi et al., 2020), our algorithm enables optimistic exploration in continuous domains.
2. We provide rigorous theoretical analysis and regret bounds for HIP-RL (Theorem 4.2). Specifically, we demonstrate sublinear regret bounds for Gaussian Process (GP) dynamics models and reward functions (Theorem 4.3).
3. We provide an experimental evaluation of our algorithm in Section 5 and show that even with a limited amount of preference feedbacks, it can perform as well as traditional RL algorithm.

To the best of our knowledge, HIP-RL is the first practical algorithm for PbRL in continuous domains with regret-bound guarantees.

1.1. Related Work

Preference-based RL Initially, the problem of PbRL has been tackled experimentally with success (Busa-Fekete et al., 2014; Wirth et al., 2016; 2017; Christiano et al., 2017). Only recently, there have been works analyzing the PbRL framework theoretically.

Novoseller et al. (2020) proposes an algorithm for PbRL based on Double Posterior Sampling (DPS) and proves asymptotic sublinear regret bounds for the finite horizon setting in tabular MDPs. Xu et al. (2020) shows near-optimal sample complexity in their finite-time analysis for PbRL. Pacchiano et al. (2021) proposes a formal framework to study the PbRL problem with a linearly parametrized reward function. Moreover, they present and analyze two algorithms, when the transition model is known and unknown. The authors show sublinear regret bounds for both algorithms.

PbOP (Chen et al., 2022) extends the discrete state-action space results to the infinite one, proposing an algorithm to learn the preference and transition function with a general function approximation. The regret of the proposed algorithm is shown to be sublinear in the number of episodes (in the general case). Moreover, the authors show that the bound is tight with respect to the feature dimension and the number of episodes in the linear setting. Unfortunately, the algorithm is intractable in practice.

PbRL is also closely related to the dueling bandits setting (Yue et al., 2012; Zoghi et al., 2015), which can be regarded as a particular instance of PbRL with only one state and 1-step horizon.

Model-Based RL Thanks to its sample efficiency, MBRL can be applied to many real-world settings for complex decision-making (Deisenroth et al., 2013). For example, Kaiser et al. (2019) propose SimPLe, a model-based deep RL algorithm to efficiently learn how to play Atari games, Chua et al. (2018) uses uncertainty-aware deep network dynamics models to solve high-dimensional continuous-control problems. However, all these works perform greedy exploitation with the current policy, which is in general sub-optimal.

Curi et al. (2020) propose H-UCRL, which reduces optimistic exploration to greedy exploitation by parametrizing the model space and augmenting the control space with hallucinated actions, which allow the agent to control the epistemic uncertainty in the next state. Kakade et al. (2020) prove tight confidence bounds for the setting of Curi et al. (2020). Abeille & Lazaric (2020) proved that the planning problem for linear quadratic regulators (LQR) (Mania et al., 2019) can be solved efficiently. Neu & Pike-Burke (2020), also reduce optimistic exploration to greedy exploitation using reward bonuses.

Liu et al. (2023) propose MoP-RL, a PbRL algorithm that combines MBRL and preference learning. The authors experimentally show the effectiveness of their algorithm.

2. Preliminaries

We consider an undiscounted, finite horizon Markov Decision Process M defined by a tuple $M = (\mathcal{S}, \mathcal{A}, \mathbb{P}, r, H)$,

where $\mathcal{S} \subseteq \mathbb{R}^p$ is the state space, $\mathcal{A} \subseteq \mathbb{R}^q$ is the action space, $\mathbb{P} : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$[1] are the transition probabilities, $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the latent reward function, unobserved by the agent during training, and H is the horizon. Given $\mathbf{s} \in \mathcal{S}, \mathbf{a} \in \mathcal{A}$, the dynamics can be written as

$$\mathbf{s}_{t+1} = f(\mathbf{s}_t, \mathbf{a}_t) + \boldsymbol{\omega}_t \tag{1}$$

where $f : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ is the true transition function, and $\boldsymbol{\omega}_t \in \mathbb{R}^p$ is *i.i.d.*, additive noise.

Assumption 2.1. The true transition function f is L_f-Lipschitz continuous and for all time steps $t \geq 0$, the noise $\boldsymbol{\omega}_t$ is element-wise σ-sub-Gaussian.

This is a standard assumption when using Gaussian Process models (Srinivas et al., 2009; Chowdhury & Gopalan, 2019; Curi et al., 2020).

We study the episodic setting with K episodes, where each episode has a horizon H. We aim to learn a deterministic, time-dependent control policy $\pi_t : \mathcal{S} \to \mathcal{A}$ from a set Π that selects action $\mathbf{a}_t = \pi_t(\mathbf{s}_t)$, from preference feedback over trajectories.

Let us first define some notations for our analysis. Executing policy π on the environment starting from state $\mathbf{s}_0$ yields a trajectory $\tau = (\mathbf{s}_0, \mathbf{a}_0, \mathbf{s}_1, \mathbf{a}_1, \ldots, \mathbf{s}_{H-1}, \mathbf{a}_{H-1})$. We denote the set of all possible trajectories of length H by

$$\Gamma_H = \{\tau = (\mathbf{s}_0, \mathbf{a}_0, \ldots, \mathbf{s}_{H-1}, \mathbf{a}_{H-1}) \mid \mathbf{s}_t \in \mathcal{S}, \mathbf{a}_t \in \mathcal{A}\}.$$

Then, given a trajectory τ at iteration k, we will denote each state and action at time step t as $\mathbf{s}_{t,k}$ and $\mathbf{a}_{t,k}$. Instead, given two trajectories τ_1, τ_2, we use a superscript to denote the trajectory to which each state and action belongs, i.e., $\mathbf{s}^1_{t,k}$ and $\mathbf{a}^1_{t,k}$ belong to trajectory τ_1 (similarly for τ_2). When the episode k is not important for our analysis, we simplify notation by simply using $\mathbf{s}_t = \mathbf{s}_{t,k}$ and $\mathbf{s}^i_t = \mathbf{s}^i_{t,k}$ (similarly for actions).

Trajectory Preference Given two trajectories $\tau_1, \tau_2 \in \Gamma_H$, we define the preference function $g : \Gamma_H \times \Gamma_H \to \mathbb{R}$ as the utility difference of τ_1 and τ_2:

$$g(\tau_1, \tau_2; r) = \sum_{t=0}^{H-1} r(\mathbf{s}^1_t, \mathbf{a}^1_t) - r(\mathbf{s}^2_t, \mathbf{a}^2_t),$$

which is the difference in the cumulative rewards obtained over the trajectories. This preference feedback is effectively what the learner receives from the environment; the learner has no access to the true reward feedback, but only to a measure of how much trajectory performances differ.

[1] $\Delta(\mathcal{S})$ denotes the probability distribution over $\mathcal{S}$.

Policy Preference Given two policies $\pi_1, \pi_2 \in \Pi$, we define the preference function over policies, overloading the notation of g, as

$$g(\pi_1, \pi_2; f, r) = \mathop{\mathbb{E}}_{\tau_1 \sim (f,\pi_1), \tau_2 \sim (f,\pi_2)} \left[g(\tau_1, \tau_2; r) \right], \tag{2}$$

where $\tau_i \sim (f, \pi_i)$ denotes that τ_i is sampled from the environment with dynamics f using policy π_i.

Assumption 2.2 (Optimality)**.** There exists a policy $\pi^* \in \Pi$ such that

$$g(\pi^*, \pi; f, r) \geq 0, \quad \forall \pi \in \Pi,$$

so that the objective of regret minimization can be well-defined.

Objective We aim to find a control policy that minimizes cumulative regret, defined as

$$\mathrm{Reg}(K) = \sum_{k=1}^{K} \mathrm{reg}(k) = \sum_{k=1}^{K} g(\pi^*, \pi_k; f, r), \tag{3}$$

where $\mathrm{reg}(k) = g(\pi^*, \pi_k; f, r)$ is the regret at iteration k.

2.1. Hallucinated Upper Confidence Reinforcement Learning

We briefly review the Hallucinated Upper Confidence Reinforcement Learning (H-UCRL) algorithm by Curi et al. (2020) since we will use ideas from it.

Assumption 2.3 (Calibrated Dynamics Model)**.** The learned dynamics model is calibrated with respect to the true dynamics f, i.e., there exists a sequence of positive β_k such that, with probability $(1 - \delta)$, for all $k \geq 0$ and for all $\mathbf{s}, \mathbf{a} \in \mathcal{S} \times \mathcal{A}$, it holds that:

$$|f(\mathbf{s}, \mathbf{a}) - \boldsymbol{\mu}_k(\mathbf{s}, \mathbf{a})| \leq \beta_k \boldsymbol{\sigma}_k(\mathbf{s}, \mathbf{a}),$$

element-wise, where $\boldsymbol{\sigma}_k(\cdot) = \mathrm{diag}(\boldsymbol{\Sigma}_k(\cdot))$.

Model Learning In MBRL, the agent selects a policy π_k in each episode and executes it for H steps. During each episode, the agent collects data $\mathcal{D}_k = \{(\mathbf{s}_t, \mathbf{a}_t, \mathbf{s}_{t+1})\}_{t=0}^{H-1}$. The collected data, $\mathcal{D}_{1:k} = \cup_{i=1}^{k} \mathcal{D}_i$, can then be used to estimate a model $\tilde{f}$ using a frequentist model with mean and covariance estimates $\boldsymbol{\mu}_k(\mathbf{s}, \mathbf{a})$ and $\boldsymbol{\Sigma}_k(\mathbf{s}, \mathbf{a})$, or a Bayesian model with posterior $p(\tilde{f} \mid \mathcal{D}_{1:k})$, defining $\boldsymbol{\mu}_k(\cdot) = \mathbb{E}_{\tilde{f} \sim p(\tilde{f} \mid \mathcal{D}_{1:k})}[\tilde{f}(\cdot)]$ and $\boldsymbol{\Sigma}^2_k(\cdot) = \mathrm{Var}[\tilde{f}(\cdot)]$.

Exploration Strategy The authors propose a novel exploration strategy to optimistically explore the environment and maximize the cumulative reward

$$J(\tilde{f}, \pi) = \mathop{\mathbb{E}}_{\tilde{\boldsymbol{\omega}}_{0:H-1}} \left[\sum_{t=0}^{H-1} r(\tilde{\mathbf{s}}_t, \tilde{\mathbf{a}}_t) \mid \mathbf{s}_0, \pi \right], \tag{4}$$

where $\tilde{\mathbf{s}}_{t+1} = \tilde{f}(\tilde{\mathbf{s}}_t, \pi(\tilde{\mathbf{s}}_t)) + \tilde{\boldsymbol{\omega}}_t$.

H-UCRL is a variant of the Upper Confidence Reinforcement Learning (UCRL) algorithm (Auer et al., 2008) that reparametrizes the statistical model of the dynamics at each iteration k as

$$\tilde{f}(\mathbf{s}, \mathbf{a}) = \boldsymbol{\mu}_{k-1}(\mathbf{s}, \mathbf{a}) + \beta_{k-1}\boldsymbol{\sigma}_{k-1}(\mathbf{s}, \mathbf{a})\eta(\mathbf{s}, \mathbf{a}), \quad (5)$$

where $\eta : \mathcal{S} \times \mathcal{A} \to [-1, 1]^p$. Then, the policy at each iteration k is

$$\pi_k = \arg\max_{\pi \in \Pi} \max_{\eta \in [-1,1]^p} J(\tilde{f}, \pi). \quad (6)$$

The authors show that the optimization problem in H-UCRL can be solved efficiently by a greedy oracle that acts on both the action and the hallucinated control η. In practice, this is implemented with a combination of offline policy search and online planning.

Algorithm 1 HIP-RL

Initialize $\pi_0, \mathbf{s}_0, \tilde{f}, \hat{r}$
Initialize a set of preferences $P = \emptyset$
Initialize a set of trajectories $T = \{\tau_0 \sim (f, \pi_0)\}$
for $k = 1, \ldots, K$ **do**
  $\pi_k, \eta_k = \texttt{PolicySearch}(\tilde{f}, \hat{r}, \pi_{k-1})$
  $\tau_k = [\,]$
  for $t = 1, \ldots, H$ **do**
    $\tau_k = \tau_k.\texttt{append}(\mathbf{s}_{t-1,k})$
    $(\mathbf{a}_{t-1,k}, \mathbf{a}'_{t-1,k}) = \texttt{Plan}(\mathbf{s}_{t-1,k}; \tilde{f}, \hat{r})$
    $\mathbf{s}_{t,k} = f(\mathbf{s}_{t-1,k}, \mathbf{a}_{t-1,k}) + \boldsymbol{\omega}_{t-1,k}$
    $\tau_k = \tau_k.\texttt{append}(\mathbf{a}_{t-1,k})$
  end for
  Sample $\tau \sim T$
  Add new trajectory $T = T \cup \{\tau_k\}$
  Add new preference $P = P \cup \{g(\tau_k, \tau; r)\}$
  Update reward model $\hat{r} = \texttt{EstReward}(P, T)$
  Update transition model $\tilde{f} = \texttt{EstDynamics}(T)$
end for

3. HIP-RL: Hallucinated Inputs Preference-based Reinforcement Learning

We propose a novel practical algorithm called Hallucinated Inputs Preference-based RL (HIP-RL) that combines the ideas of H-UCRL and preference-based reinforcement learning. The main idea is to learn a dynamics model $\tilde{f}$ and a reward function $\hat{r}$ from the set of trajectories T and preferences P and then use these for the policy search and optimistic planning in H-UCRL.

We will show that the uncertainty induced by the learned reward function can be bounded. Therefore we can prove regret bounds the proposed algorithm.

Transition Model Learning We adopt the approach used in the H-UCRL algorithm (Curi et al., 2020) to learn the transition model. In each episode, the agent collects transitions while following the current policy π_k and adds this information to the set of trajectories T. Using this data, we estimate a model $\tilde{f}$. We assume a calibrated dynamics model with respect to the true dynamics f.

Reward Function Learning One possible interpretation of the preference function g is that it is a distance function between two trajectories. Therefore, we can use the preference function to learn a reward function $\hat{r}$ that is consistent with the preferences. We can also interpret the reward estimate $\hat{r}$ as a predictor of the preference function, i.e., human preferences can be viewed as arising from some latent reward function.

Assumption 3.1 (Calibrated Preference Function)**.** The preference function induced by the reward function $\hat{r}$ is calibrated with respect to the true preference function g, i.e., there exists a sequence of positive α_k such that, with probability $(1-\delta)$, for all $k \geq 0$ and all $\tau, \tau' \in \Gamma_H$, it holds that:

$$|g(\tau, \tau'; r) - \mu_k(\tau, \tau'; \hat{r})| \leq \alpha_k \sigma_k(\tau, \tau'; \hat{r}),$$

where $\mu_k(\tau, \tau'; \hat{r})$ and $\sigma_k(\tau, \tau'; \hat{r})$ are the mean and standard deviation of the predicted preference $g(\tau, \tau'; \hat{r})$.

One possible approach to learning the reward function is to minimize the mean squared error between the predicted preferences and the observed preferences. In particular, we can learn a reward function $\hat{r}$ by solving the following optimization problem

$$\min_{\hat{r}} \quad \frac{1}{|P|} \sum_{(g,\tau,\tau') \in P} \left(g(\tau, \tau'; r) - g(\tau, \tau'; \hat{r})\right)^2, \quad (7)$$

where $(g, \tau, \tau') \in P$ is a shorthand for the preference over trajectories $g(\tau, \tau'; r)$ and the trajectories τ and τ' in the set of preferences P. Note that the mean squared error is only one possible choice of the loss function. We can also use other loss functions for regression problems.

Offline Policy Search and Online Planning Having estimated the transition model $\tilde{f}$ and the reward function $\hat{r}$, we can solve the optimization in (6) to obtain the next policy π_k and hallucinated control η_k. Curi et al. (2020) prove that it is sufficient to optimize over Lipschitz-continuous bounded functions, therefore one can optimize over Lipschitz-continuous $\eta(\cdot)$. This allows us to use a greedy oracle that acts on both the action and the hallucinated control η. In practice, this is implemented with a combination of offline policy search and online planning (Lowrey et al., 2018). We note in Algorithm 1 that in the

planning step, e.g., using Model Predictive Control (MPC) (Morari & Lee, 1999), we obtain both the true action $\mathbf{a}$ and the hallucinated action $\mathbf{a}'$, but during execution, we only execute the true action $\mathbf{a}$.

4. Theoretical Guarantees

Our objective is to learn a policy that performs as well as the optimal policy π^*. One way to measure the performance of a policy is to measure the cumulative regret $\mathrm{Reg}(K)$, which quantifies how much the optimal policy π^* is preferred over the learned policy. If we show that the cumulative regret $\mathrm{Reg}(K)$ is sublinear in K, then we can know that the preference of the optimal policy over the learned policy vanishes as K goes to infinity. In other words, the learned policy will eventually perform as well as the optimal policy.

Assumption 4.1 (Lipschitz Continuity)**.** We assume that the preference function g is L_g Lipschitz continuous, the policy π is L_π Lipschitz continuous, the functions $\boldsymbol{\mu}_k$ and $\boldsymbol{\sigma}_k$ are L_{μ_f} and L_{σ_f} Lipschitz continuous, and the functions μ_k and σ_k are L_{μ_g} and L_{σ_g} Lipschitz continuous.

Transition Function and Reward Function Complexity We define the complexity of the transition function and reward function as

$$I_K^f(\mathcal{S},\mathcal{A}) = \max_{\tau_1,\dots,\tau_K\in\Gamma_H} \sum_{k=1}^{K} \sum_{\mathbf{s},\mathbf{a}\in\tau_k} \|\boldsymbol{\sigma}_{k-1}(\mathbf{s},\mathbf{a})\|_2^2 \quad (8)$$

and

$$I_K^g(\Gamma_H,\Gamma_H) = \max_{\substack{(\tau_1,\tau_1'),\dots,(\tau_K,\tau_K')\\ \in\Gamma_H\times\Gamma_H}} \sum_{k=1}^{K} \left|\sigma_{k-1}(\tau_k,\tau_k';\hat{r})\right|^2. \quad (9)$$

Theorem 4.2. *Under Assumptions 2.1, 2.2, 2.3, 3.1, and 4.1, for any $K \geq 1$, with probability $(1-\delta)$, the regret of Algorithm 1 is at most*

$$\mathrm{Reg}(K) \leq \mathcal{O}\Bigg(\sqrt{2K}\Bigg[HL_1\sqrt{2HI_K^f(\mathcal{S},\mathcal{A})} + L_2\sqrt{I_K^g(\Gamma_H,\Gamma_H)}\Bigg]\Bigg). \quad (10)$$

We will provide a proof sketch of the Theorem 4.2 in this section. The full proof can be found in Appendix A.

Proof sketch. Let us first note that we can write the episodic regret in Equation (3) as

$$\begin{aligned}
\mathrm{reg}(k) &= g(\pi^*,\pi_k;f,r) && (11)\\
&= g(\pi^*,\pi_k;f,r) - g(\pi^*,\pi_k;\tilde{f},r) && (12)\\
&\quad + g(\pi^*,\pi_k;\tilde{f},r) - g(\pi^*,\pi_k;\tilde{f},\hat{r})\\
&\quad + g(\pi^*,\pi_k;\tilde{f},\hat{r})\\
&\leq \left|g(\pi^*,\pi_k;f,r) - g(\pi^*,\pi_k;\tilde{f},r)\right| && (13)\\
&\quad + \left|g(\pi^*,\pi_k;\tilde{f},r) - g(\pi^*,\pi_k;\tilde{f},\hat{r})\right|\\
&\quad + g(\pi^*,\pi_k;\tilde{f},\hat{r}).
\end{aligned}$$

We will then bound each of the terms in Equation (13) separately. By Lemma A.3, we have

$$\begin{aligned}
&\left|g(\pi^*,\pi_k,f,r) - g(\pi^*,\pi_k,\tilde{f},r)\right|\\
&\quad\leq \beta_{k-1}\bar{L}_f^{H-1}L_g\sqrt{(1+L_\pi)}\,\mathop{\mathbb{E}}_{\boldsymbol{\omega}}[A+B],
\end{aligned} \quad (14)$$

where

$$A = \sum_{t=0}^{H-1}\sum_{j=0}^{t-1}\left\|\boldsymbol{\sigma}_{k-1}(\mathbf{s}_{j,k}^*)\right\|_2,$$
$$B = \sum_{t=0}^{H-1}\sum_{j=0}^{t-1}\left\|\boldsymbol{\sigma}_{k-1}(\mathbf{s}_{j,k}^k)\right\|_2,$$

and $\mathbf{s}_{j,k}^*$ and $\mathbf{s}_{j,k}^k$ are the states visited by the optimal policy and the learned policy at time k respectively.

By Lemma A.4, we have

$$\begin{aligned}
&\left|g(\pi^*,\pi_k,\tilde{f},r) - g(\pi^*,\pi_k,\tilde{f},\hat{r})\right|\\
&\quad\leq 2\alpha_{k-1}\mathop{\mathbb{E}}_{\boldsymbol{\omega}}\left[\left|\sigma_{k-1}(\tau^*,\tau_k;\hat{r})\right|\right],
\end{aligned} \quad (15)$$

where τ^* indicates that the trajectory was obtained by executing the optimal policy and τ_k is obtained from π_k.

Lastly, by Lemma A.5, we have

$$g(\pi^*,\pi_k,\tilde{f},\hat{r}) \leq 0. \quad (16)$$

Using Lemma A.7, we can then bound the squared regret at time k, then using this bound, we can obtain the final bound on the cumulative regret by Lemma A.8 and Corollary A.9. □

Gaussian Process Model In order to prove that the learned policy indeed performs as well as the optimal policy, we need to show that the regret in Theorem 4.2 is sublinear in K. Equivalently, we need to show that the complexity of the transition function and the reward function is sublinear in K. We will show that this is the case when we use Gaussian Process dynamics models and reward functions.

Table 1. Performance of HIP-RL and standard PPO agent. Each algorithm was evaluated by averaging 10 rollouts of the model.

ENVIRONMENT	HIP-RL	PPO
INVPENDULUM	1000.0 ± 0.0	1000.0 ± 0.0
HALFCHEETAH	2638.2 ± 133.5	1877.5 ± 58.7

Theorem 4.3. *Under Assumptions 2.1, 2.2, 2.3, 3.1, and 4.1, for any $K \geq 1$, with probability $(1-\delta)$, the regret of Algorithm 1 is at most*

$$\mathrm{Reg}(K) \leq \mathcal{O}\left(\sqrt{4K(p+q)}\left[L_1 H^2 p\sqrt{\log(pHK)} + L_2 H\sqrt{\log(K)}\right]\right) \tag{17}$$

if the transition and the reward models are learned using Gaussian Processes with squared exponential kernels.

Proof. The proof is a direct consequence of Lemma 17 in (Curi et al., 2020) and the bounds on the information capacity of Gaussian Processes with squared exponential kernels by (Srinivas et al., 2009; Krause & Ong, 2011). □

5. Experimental Evaluation

We present the findings of a series of initial experiments[2], where we evaluate the performance of HIP-RL on two Mujoco environments (Todorov et al., 2012). In our implementation, we model the reward function as a probabilistic neural network and the transition model as a 5-head probabilistic ensemble. The policy search step is performed by using a modified version of the Proximal Policy Optimization (PPO) algorithm (Schulman et al., 2017) to support hallucinated inputs, our learned transition dynamics and reward function. Further details on the choice of hyperparameters can be found in Appendix B.

During our experiments, we include a buffer consisting of 5 episodes, where a random agent is deployed to gather trajectory and preference data. As depicted in Figure 2, we observe that with less than 30 preferences, our method successfully learns effective behavior in both the Inverted-Pendulum and Half-Cheetah environments.

Furthermore, we compare the performance of our agents to that of two agents trained using PPO with the actual dynamics and reward functions. The results presented in Table 1 demonstrate that we surpass the performance of PPO trained for 1×10^6 steps.

[2]The open-source implementation can be found at https://github.com/calvincbzhang/hip-rl.

6. Conclusion and Future Work

In this paper, we introduced a novel Preference-based RL (PbRL) algorithm in continuous state-action spaces called Hallucinated Inputs Preference-based RL (HIP-RL). HIP-RL leverages the concept of hallucinated inputs (H-UCRL) to enable optimistic exploration and achieve convergence guarantees. We provided sublinear regret bounds for Gaussian Process (GP) dynamics and reward functions, bridging the gap between preference-based RL and continuous control tasks. Finally, we demonstrate the feasibility and efficiency of implementing our algorithm by showcasing its ability to learn a high-quality policy with a minimal number of preference feedbacks required.

Some interesting future research directions include investigating the setting in which comparisons are not only pairwise, which would allow using previously seen trajectories more efficiently. Another direction for future research concerns non-linear reward function, by using tools from concave utility RL (Hazan et al., 2019). On the practical side, similar algorithms could be developed and tested in real-world settings, where alignment with human objectives is of particular importance. Our contributions pave the way for practical implementations of PbRL in real-world scenarios such as autonomous driving and robotic manipulation.

Disclosure of Funding

Giorgia Ramponi is partially funded by Google Brain and by the ETH AI Center.

References

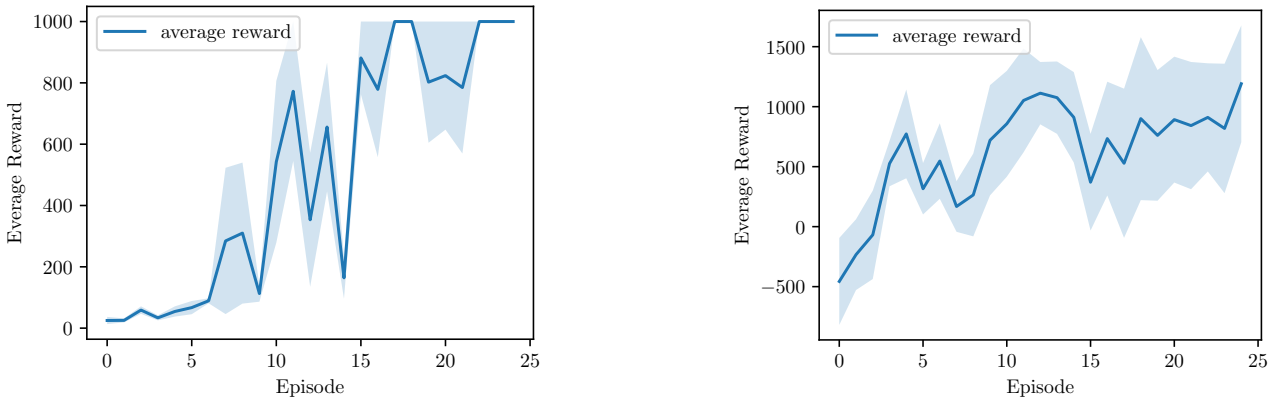
Abeille, M. and Lazaric, A. Efficient optimistic exploration in linear-quadratic regulators via lagrangian relaxation. In *International Conference on Machine Learning*, pp. 23–31. PMLR, 2020.

Auer, P., Jaksch, T., and Ortner, R. Near-optimal regret bounds for reinforcement learning. *Advances in neural information processing systems*, 21, 2008.

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

Busa-Fekete, R., Szörényi, B., Weng, P., Cheng, W., and Hüllermeier, E. Preference-based reinforcement learning: evolutionary direct policy search using a preference-based racing algorithm. *Machine learning*, 97:327–351, 2014.

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.



(a) InvertedPendulum-v4 (b) HalfCheetah-v4

Figure 2. Learning curves for Inverted-Pendulum and Half-Cheetah (averaged of 4 runs).

Chowdhury, S. R. and Gopalan, A. Online learning in kernelized markov decision processes. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 3197–3205. PMLR, 2019.

Christiano, P. F., Leike, J., Brown, T., Martic, M., Legg, S., and Amodei, D. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

Chua, K., Calandra, R., McAllister, R., and Levine, S. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. *Advances in neural information processing systems*, 31, 2018.

Curi, S., Berkenkamp, F., and Krause, A. Efficient model-based reinforcement learning through optimistic policy search and planning. *Advances in Neural Information Processing Systems*, 33:14156–14170, 2020.

Deisenroth, M. P., Neumann, G., Peters, J., et al. A survey on policy search for robotics. *Foundations and Trends® in Robotics*, 2(1–2):1–142, 2013.

Gal, Y. et al. Uncertainty in deep learning. 2016.

Hazan, E., Kakade, S., Singh, K., and Van Soest, A. Provably efficient maximum entropy exploration. In *International Conference on Machine Learning*, pp. 2681–2691. PMLR, 2019.

Kaiser, L., Babaeizadeh, M., Milos, P., Osinski, B., Campbell, R. H., Czechowski, K., Erhan, D., Finn, C., Kozakowski, P., Levine, S., et al. Model-based reinforcement learning for atari. *arXiv preprint arXiv:1903.00374*, 2019.

Kakade, S., Krishnamurthy, A., Lowrey, K., Ohnishi, M., and Sun, W. Information theoretic regret bounds for online nonlinear control. *Advances in Neural Information Processing Systems*, 33:15312–15325, 2020.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

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

Krause, A. and Ong, C. Contextual gaussian process bandit optimization. *Advances in neural information processing systems*, 24, 2011.

Liu, Y., Datta, G., Novoseller, E., and Brown, D. S. Efficient preference-based reinforcement learning using learned dynamics models. *arXiv preprint arXiv:2301.04741*, 2023.

Lowrey, K., Rajeswaran, A., Kakade, S., Todorov, E., and Mordatch, I. Plan online, learn offline: Efficient learning and exploration via model-based control. *arXiv preprint arXiv:1811.01848*, 2018.

Mania, H., Tu, S., and Recht, B. Certainty equivalence is efficient for linear quadratic control. *Advances in Neural Information Processing Systems*, 32, 2019.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. Human-level control through deep reinforcement learning. *nature*, 518(7540): 529–533, 2015.

Morari, M. and Lee, J. H. Model predictive control: past, present and future. *Computers & Chemical Engineering*, 23(4-5):667–682, 1999.

Neu, G. and Pike-Burke, C. A unifying view of optimism in episodic reinforcement learning. *Advances in Neural Information Processing Systems*, 33:1392–1403, 2020.

Novoseller, E., Wei, Y., Sui, Y., Yue, Y., and Burdick, J. Dueling posterior sampling for preference-based reinforcement learning. In *Conference on Uncertainty in Artificial Intelligence*, pp. 1029–1038. PMLR, 2020.

Pacchiano, A., Saha, A., and Lee, J. Dueling rl: Reinforcement learning with trajectory preferences. *arXiv preprint arXiv:2111.04850*, 2021.

Raffin, A., Hill, A., Gleave, A., Kanervisto, A., Ernestus, M., and Dormann, N. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021. URL http://jmlr.org/papers/v22/20-1364.html.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Srinivas, N., Krause, A., Kakade, S. M., and Seeger, M. Gaussian process optimization in the bandit setting: No regret and experimental design. *arXiv preprint arXiv:0912.3995*, 2009.

Sutton, R. S. and Barto, A. G. *Reinforcement learning: An introduction*. MIT press, 2018.

Tesauro, G. et al. Temporal difference learning and td-gammon. *Communications of the ACM*, 38(3):58–68, 1995.

Todorov, E., Erez, T., and Tassa, Y. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033. IEEE, 2012. doi: 10.1109/IROS.2012.6386109.

Wang, Y.-C. and Usher, J. M. Application of reinforcement learning for agent-based production scheduling. *Engineering applications of artificial intelligence*, 18(1):73–82, 2005.

Wirth, C. and Fürnkranz, J. A policy iteration algorithm for learning from preference-based feedback. In *Advances in Intelligent Data Analysis XII: 12th International Symposium, IDA 2013, London, UK, October 17-19, 2013. Proceedings 12*, pp. 427–437. Springer, 2013.

Wirth, C., Fürnkranz, J., and Neumann, G. Model-free preference-based reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016.

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

Xu, Y., Wang, R., Yang, L., Singh, A., and Dubrawski, A. Preference-based reinforcement learning with finite-time guarantees. *Advances in Neural Information Processing Systems*, 33:18784–18794, 2020.

Yue, Y., Broder, J., Kleinberg, R., and Joachims, T. The k-armed dueling bandits problem. *Journal of Computer and System Sciences*, 78(5):1538–1556, 2012.

Zhao, Y., Zeng, D., Socinski, M. A., and Kosorok, M. R. Reinforcement learning strategies for clinical trials in nonsmall cell lung cancer. *Biometrics*, 67(4):1422–1433, 2011.

Zoghi, M., Karnin, Z. S., Whiteson, S., and De Rijke, M. Copeland dueling bandits. *Advances in neural information processing systems*, 28, 2015.

A. Proof of Theorem 4.2

We will first prove a few ancillary lemmas that will be used in the proof of the main theorem.

Lemma A.1 (Lemma 1 in Curi et al. (2020)). *Under Assumption 2.3, for any sequence generated by the true system in Equation (1), there exists a function $\eta : \mathbb{R}^p \to [-1,1]^p$ such that $\mathbf{s}_{t,k} = \tilde{\mathbf{s}}_{t,k}$ if $\boldsymbol{\omega} = \tilde{\boldsymbol{\omega}}$.*

Lemma A.2 (Lemma 4 in Curi et al. (2020)). *Under Assumptions 2.1, 2.3 and 4.1, let $\bar{L}_f = 1 + L_{fc} + 2\beta_{k-1} L_{\sigma_f} \sqrt{(1+L_\pi)}$, with $L_{fc} = L_f \sqrt{(1+L_\pi)}$. Then, for all iterations $k > 0$, any function $\eta : \mathbb{R}^p \times \mathbb{R}^q \to [-1,1]^p$ and any sequence $\boldsymbol{\omega}_t$ with $\tilde{\boldsymbol{\omega}}_t = \boldsymbol{\omega}_t$, $\pi \in \Pi$, with $1 \le t \le H$, we have that*

$$\|\mathbf{s}_{t,k} - \tilde{\mathbf{s}}_{t,k}\|_2 \le 2\beta_{k-1} \bar{L}_f^{H-1} \sum_{j=0}^{t-1} \|\boldsymbol{\sigma}_{k-1}(\mathbf{s}_{j,k})\|_2 . \tag{18}$$

Lemma A.3. *Let $\pi_1 = \pi^*, \pi_2 = \pi_k \in \Pi$, then under the assumptions of Theorem 4.2, we have that with probability at least $(1-\delta)$,*

$$\left| g(\pi_1, \pi_2; f, r) - g(\pi_1, \pi_2; \hat{f}, r) \right| \le 2\beta_{k-1} \bar{L}_f^{H-1} L_g \sqrt{(1+L_\pi)} \mathop{\mathbb{E}}_{\boldsymbol{\omega}} \left[\sum_{i=1,2} \sum_{t=0}^{H-1} \sum_{j=0}^{t-1} \left\| \boldsymbol{\sigma}_{k-1}(\mathbf{s}^i_{j,k}) \right\|_2 \right] . \tag{19}$$

Proof. At episode k, consider $\pi_1 = \pi^*, \pi_2 = \pi_k \in \Pi$, then we have that (dropping the dependence on k in the following derivation to lighten the notation)

$$\begin{aligned}
\left| g(\pi_1, \pi_2; f, r) - g(\pi_1, \pi_2; \tilde{f}, r) \right| &= \left| \mathop{\mathbb{E}}_{\tilde{\tau}_1 \sim (\tilde{f}, \pi_1), \tilde{\tau}_2 \sim (\tilde{f}, \pi_2)} [g(\tilde{\tau}_1, \tilde{\tau}_2; r)] - \mathop{\mathbb{E}}_{\tau_1 \sim (f, \pi_1), \tau_2 \sim (f, \pi_2)} [g(\tau_1, \tau_2; r)] \right| && (20) \\
&= \left| \mathop{\mathbb{E}}_{\tilde{\boldsymbol{\omega}}} [g(\tilde{\tau}_1, \tilde{\tau}_2; r)] - \mathop{\mathbb{E}}_{\boldsymbol{\omega}} [g(\tau_1, \tau_2; r)] \right| && (21) \\
&= \left| \mathop{\mathbb{E}}_{\tilde{\boldsymbol{\omega}} = \boldsymbol{\omega}} [g(\tilde{\tau}_1, \tilde{\tau}_2; r) - g(\tau_1, \tau_2; r)] \right| && (22) \\
&\le \mathop{\mathbb{E}}_{\tilde{\boldsymbol{\omega}} = \boldsymbol{\omega}} \left| [g(\tilde{\tau}_1, \tilde{\tau}_2; r) - g(\tau_1, \tau_2; r)] \right| , && (23)
\end{aligned}$$

where the second and third equality hold by Lemma A.1 and the inequality follows from Jensen's inequality.

By L_g-Lipschitz continuity of the preference function, we have

$$\begin{aligned}
|[g(\tilde{\tau}_1, \tilde{\tau}_2; r) - g(\tau_1, \tau_2; r)]| &\le L_g \left\| (\tilde{\tau}_1 - \tau_1, \tilde{\tau}_2 - \tau_2) \right\|_2 && (24) \\
&= L_g \sqrt{\|\tilde{\tau}_1 - \tau_2\|_2^2 + \|\tilde{\tau}_2 - \tau_2\|_2^2} && (25) \\
&= L_g \sqrt{\sum_{i=1,2} \left\| (\tilde{\mathbf{s}}^i_0, \tilde{\mathbf{a}}^i_0, \ldots, \tilde{\mathbf{s}}^i_{H-1}, \tilde{\mathbf{a}}^i_{H-1}) - (\mathbf{s}^i_0, \mathbf{a}^i_0, \ldots, \mathbf{s}^i_{H-1}, \mathbf{a}^i_{H-1}) \right\|_2^2} && (26) \\
&= L_g \sqrt{\sum_{i=1,2} \left\| \tilde{\mathbf{s}}^i_0 - \mathbf{s}^i_0, \pi_i(\tilde{\mathbf{s}}^i_0) - \pi_i(\mathbf{s}^i_0), \ldots, \tilde{\mathbf{s}}^i_{H-1} - \mathbf{s}^i_{H-1}, \pi_i(\tilde{\mathbf{s}}^i_{H-1}) - \pi_i(\mathbf{s}^i_{H-1}) \right\|_2^2} && (27) \\
&= L_g \sqrt{\sum_{i=1,2} \left\| \tilde{\mathbf{s}}^i_0 - \mathbf{s}^i_0 \right\|_2^2 + \left\| \pi_i(\tilde{\mathbf{s}}^i_0) - \pi_i(\mathbf{s}^i_0) \right\|_2^2 + \cdots + \left\| \pi_i(\tilde{\mathbf{s}}^i_{H-1}) - \pi_i(\mathbf{s}^i_{H-1}) \right\|_2^2} && (28) \\
&\le L_g \sqrt{\sum_{i=1,2} \left\| \tilde{\mathbf{s}}^i_0 - \mathbf{s}^i_0 \right\|_2^2 + L_\pi \left\| \tilde{\mathbf{s}}^i_0 - \mathbf{s}^i_0 \right\|_2^2 + \cdots + L_\pi \left\| \tilde{\mathbf{s}}^i_{H-1} - \mathbf{s}^i_{H-1} \right\|_2^2} && (29) \\
&= L_g \sqrt{(1+L_\pi) \sum_{i=1,2} \sum_{t=0}^{H-1} \left\| \tilde{\mathbf{s}}^i_t - \mathbf{s}^i_t \right\|_2^2} && (30) \\
&\le L_g \sum_{i=1,2} \sum_{t=0}^{H-1} \sqrt{(1+L_\pi)} \left\| \tilde{\mathbf{s}}^i_t - \mathbf{s}^i_t \right\|_2 , && (31)
\end{aligned}$$

where Equation (29) follows from L_π-Lipschitz continuity of the policy.

Hence,

$$\left|g(\pi_1,\pi_2,f,r)-g(\pi_1,\pi_2,\tilde{f},r)\right| \leq \mathop{\mathbb{E}}_{\boldsymbol{\omega}}\left[L_g \sum_{i=1,2}\sum_{t=0}^{H-1}\sqrt{(1+L_\pi)}\left\|\tilde{\mathbf{s}}_t^i-\mathbf{s}_t^i\right\|_2\right]. \tag{32}$$

Now, using Lemma A.2, we have

$$\left|g(\pi_1,\pi_2,f,r)-g(\pi_1,\pi_2,\tilde{f},r)\right| \leq 2\beta_{k-1}\bar{L}_f^{H-1}L_g\sqrt{(1+L_\pi)}\mathop{\mathbb{E}}_{\boldsymbol{\omega}}\left[\sum_{i=1,2}\sum_{t=0}^{H-1}\sum_{j=0}^{t-1}\left\|\boldsymbol{\sigma}_{k-1}(\mathbf{s}_{j,k}^i)\right\|_2\right], \tag{33}$$

concluding the proof. □

Lemma A.4. *Let $\pi_1=\pi^*,\pi_2=\pi_k\in\Pi$, then under the assumptions of Theorem 4.2, we have that with probability at least $(1-\delta)$,*

$$\left|g(\pi_1,\pi_2;\hat{f},r)-g(\pi_1,\pi_2;\hat{f},\hat{r})\right| \leq 2\alpha_{k-1}\mathop{\mathbb{E}}_{\boldsymbol{\omega}}\left[\left|\sigma_{g_{k-1}}(\tau_1,\tau_2;\hat{r})\right|\right]. \tag{34}$$

Proof. At episode k, consider $\pi_1=\pi^*,\pi_2=\pi_k\in\Pi$, then we have that (dropping the dependence on k in the following derivation to lighten the notation)

$$\begin{aligned}\left|g(\pi_1,\pi_2;\tilde{f},r)-g(\pi_1,\pi_2;\tilde{f},\hat{r})\right| &= \left|\mathop{\mathbb{E}}_{\boldsymbol{\omega}}[g(\tau_1,\tau_2;r)-g(\tau_1,\tau_2;\hat{r})]\right| &(35)\\ &\leq \mathop{\mathbb{E}}_{\boldsymbol{\omega}}\left|g(\tau_1,\tau_2;r)-g(\tau_1,\tau_2;\hat{r})\right|. &(36)\end{aligned}$$

Using Assumption 3.1, we have

$$\begin{aligned}|g(\tau_1,\tau_2;r)-g(\tau_1,\tau_2;\hat{r})| &= |g(\tau_1,\tau_2;r)-\mu_{k-1}(\tau_1,\tau_2;\hat{r})-\alpha_{k-1}\sigma_{k-1}(\tau_1,\tau_2;\hat{r})| &(37)\\ &\leq |g(\tau_1,\tau_2;r)-\mu_{k-1}(\tau_1,\tau_2;\hat{r})|+\alpha_{k-1}|\sigma_{k-1}(\tau_1,\tau_2;\hat{r})| &(38)\\ &\leq 2\alpha_{k-1}|\sigma_{k-1}(\tau_1,\tau_2;\hat{r})|. &(39)\end{aligned}$$

Therefore, we have

$$\left|g(\pi_1,\pi_2;\tilde{f},r)-g(\pi_1,\pi_2;\tilde{f},\hat{r})\right| \leq 2\alpha_{k-1}\mathop{\mathbb{E}}_{\boldsymbol{\omega}}\left[|\sigma_{k-1}(\tau_1,\tau_2;\hat{r})|\right], \tag{40}$$

which concludes the proof. □

Lemma A.5. *Let $\pi^*\in\Pi$ be the optimal policy and π_k be the policy at iteration k, then*

$$g(\pi^*,\pi_k;\tilde{f},\hat{r})\leq 0. \tag{41}$$

Proof. We simply observe that π_k is the optimal policy for the estimated model and reward function at iteration k. Hence, its preference $g(\pi_k,\pi,\tilde{f},\hat{r})\geq 0,\forall\pi\in\Pi$. This holds in particular for $\pi=\pi^*$, so we have that $g(\pi^*,\pi_k,\tilde{f},\hat{r})\leq 0$. □

Corollary A.6. *Let $\pi_1=\pi^*,\pi_2=\pi_k\in\Pi$, then under the assumptions of Theorem 4.2, we have that with probability at least $(1-\delta)$,*

$$\begin{aligned}\mathrm{reg}(k) &= g(\pi_1,\pi_2;f,r) &(42)\\ &\leq 2\beta_{k-1}\bar{L}_f^{H-1}L_g\sqrt{(1+L_\pi)}\mathop{\mathbb{E}}_{\boldsymbol{\omega}}\left[\sum_{i=1,2}\sum_{t=0}^{H-1}\sum_{j=0}^{t-1}\left\|\boldsymbol{\sigma}_{k-1}(\mathbf{s}_{j,k}^i)\right\|_2\right]+2\alpha_{k-1}\mathop{\mathbb{E}}_{\boldsymbol{\omega}}\left[|\sigma_{k-1}(\tau_1,\tau_2;\hat{r})|\right]. &(43)\end{aligned}$$

Lemma A.7. *Let $\pi_1 = \pi^*, \pi_2 = \pi_k \in \Pi$, then under the assumptions of Theorem 4.2, we have that with probability at least $(1-\delta)$,*

$$\mathrm{reg}(k)^2 \leq \sum_{i=1,2} 4H^3 L_1^2 \mathop{\mathbb{E}}_{\boldsymbol{\omega}} \left[\sum_{t=1}^{H-1} \left\| \boldsymbol{\sigma}_{k-1}(\mathbf{s}_{t,k}^i) \right\|_2^2 \right] + 2L_2^2 \mathop{\mathbb{E}}_{\boldsymbol{\omega}} \left[|\sigma_{k-1}(\tau_1, \tau_2; \hat{r})|^2 \right], \tag{44}$$

where $L_1 = 2\beta_{k-1} \bar{L}_f^{H-1} L_g \sqrt{(1+L_\pi)}$ and $L_2 = 2\alpha_{k-1}$.

Proof. By Corollary A.6, we have

$$\begin{aligned}
\mathrm{reg}(k) &= g(\pi^*, \pi_k; f, r) && (45)\\
&\leq 2\beta_{k-1} \bar{L}_f^{H-1} L_g \sqrt{(1+L_\pi)} \mathop{\mathbb{E}}_{\boldsymbol{\omega}} \left[\sum_{i=1,2} \sum_{t=0}^{H-1} \sum_{j=0}^{t-1} \left\| \boldsymbol{\sigma}_{k-1}(\mathbf{s}_{j,k}^i) \right\|_2 \right] + 2\alpha_{k-1} \mathop{\mathbb{E}}_{\boldsymbol{\omega}} \left[|\sigma_{k-1}(\tau_1, \tau_2; \hat{r})| \right] && (46)\\
&\leq T \underbrace{2\beta_{k-1} \bar{L}_f^{H-1} L_g \sqrt{(1+L_\pi)}}_{L_1} \sum_{i=1,2} \mathop{\mathbb{E}}_{\boldsymbol{\omega}} \left[\sum_{t=0}^{H-1} \left\| \boldsymbol{\sigma}_{k-1}(\mathbf{s}_{t,k}^i) \right\|_2 \right] + \underbrace{2\alpha_{k-1}}_{L_2} \mathop{\mathbb{E}}_{\boldsymbol{\omega}} \left[|\sigma_{k-1}(\tau_1, \tau_2; \hat{r})| \right] && (47)\\
&= \sum_{i=1,2} T L_1 \mathop{\mathbb{E}}_{\boldsymbol{\omega}} \left[\sum_{t=0}^{H-1} \left\| \boldsymbol{\sigma}_{k-1}(\mathbf{s}_{t,k}^i) \right\|_2 \right] + L_2 \mathop{\mathbb{E}}_{\boldsymbol{\omega}} \left[|\sigma_{k-1}(\tau_1, \tau_2; \hat{r})| \right]. && (48)
\end{aligned}$$

Therefore, we have

$$\begin{aligned}
\mathrm{reg}(k)^2 &\leq \left(\sum_{i=1,2} H L_1 \mathop{\mathbb{E}}_{\boldsymbol{\omega}} \left[\sum_{t=0}^{H-1} \left\| \boldsymbol{\sigma}_{k-1}(\mathbf{s}_{t,k}^i) \right\|_2 \right] + L_2 \mathop{\mathbb{E}}_{\boldsymbol{\omega}} \left[|\sigma_{k-1}(\tau_1, \tau_2; \hat{r})| \right] \right)^2 && (49)\\
&\leq 2 \left[\left(\sum_{i=1,2} H L_1 \mathop{\mathbb{E}}_{\boldsymbol{\omega}} \left[\sum_{t=0}^{H-1} \left\| \boldsymbol{\sigma}_{k-1}(\mathbf{s}_{t,k}^i) \right\|_2 \right] \right)^2 + \left(L_2 \mathop{\mathbb{E}}_{\boldsymbol{\omega}} \left[|\sigma_{k-1}(\tau_1, \tau_2; \hat{r})| \right] \right)^2 \right] && (50)\\
&\leq \sum_{i=1,2} 4H^2 L_1^2 \left(\mathop{\mathbb{E}}_{\boldsymbol{\omega}} \left[\sum_{t=0}^{H-1} \left\| \boldsymbol{\sigma}_{k-1}(\mathbf{s}_{t,k}^i) \right\|_2 \right] \right)^2 + 2L_2^2 \left(\mathop{\mathbb{E}}_{\boldsymbol{\omega}} \left[|\sigma_{k-1}(\tau_1, \tau_2; \hat{r})| \right] \right)^2 && (51)\\
&\leq \sum_{i=1,2} 4H^3 L_1^2 \mathop{\mathbb{E}}_{\boldsymbol{\omega}} \left[\sum_{t=0}^{H-1} \left\| \boldsymbol{\sigma}_{k-1}(\mathbf{s}_{t,k}^i) \right\|_2^2 \right] + 2L_2^2 \mathop{\mathbb{E}}_{\boldsymbol{\omega}} \left[|\sigma_{k-1}(\tau_1, \tau_2; \hat{r})|^2 \right], && (52)
\end{aligned}$$

where we used Jensen's inequality and that for any n real numbers $x_1, \ldots, x_n \in \mathbb{R}$, it the AM-QM (arithmetic mean-quadratic mean) inequality holds, i.e.,

$$\left(\sum_{i=1}^{n} x_i \right)^2 \leq \sum_{i=1}^{n} n x_i^2. \tag{53}$$

□

Lemma A.8. *Let $\pi_1 = \pi^*, \pi_2 = \pi_k \in \Pi$, then under the assumptions of Theorem 4.2, we have that with probability at least $(1-\delta)$,*

$$\mathrm{Reg}(K)^2 \leq K \sum_{k=1}^{K} \left[\sum_{i=1,2} 4H^3 L_1^2 \mathop{\mathbb{E}}_{\boldsymbol{\omega}} \left[\sum_{t=0}^{H-1} \left\| \boldsymbol{\sigma}_{k-1}(\mathbf{s}_{t,k}^i) \right\|_2^2 \right] + 2L_2^2 \mathop{\mathbb{E}}_{\boldsymbol{\omega}} \left[|\sigma_{k-1}(\tau_1, \tau_2; \hat{r})|^2 \right] \right]. \tag{54}$$

Proof. By the definition of cumulative regret, the AM-QM inequality and Lemma A.7, we have

$$\mathrm{Reg}(K)^2 = \left(\sum_{k=1}^{K} \mathrm{reg}(k)\right)^2 \tag{55}$$

$$\leq K \sum_{k=1}^{K} \mathrm{reg}(k)^2 \tag{56}$$

$$\leq K \sum_{k=1}^{K} \left[\sum_{i=1,2} 4H^3 L_1^2 \mathop{\mathbb{E}}_{\boldsymbol{\omega}} \left[\sum_{t=0}^{H-1} \left\| \boldsymbol{\sigma}_{k-1}(\mathbf{s}_{t,k}^i) \right\|_2^2 \right] + 2L_2^2 \mathop{\mathbb{E}}_{\boldsymbol{\omega}} \left[|\sigma_{k-1}(\tau_1, \tau_2; \hat{r})|^2 \right] \right]. \tag{57}$$

□

Corollary A.9. *Let $\pi_1 = \pi^*, \pi_2 = \pi_k \in \Pi$, then under the assumptions of Theorem 4.2, we have that with probability at least $(1-\delta)$,*

$$\mathrm{Reg}(K)^2 \leq 2K \left(2H^3 L_1^2 I_K^f(\mathcal{S}, \mathcal{A}) + L_2^2 I_K^g(\Gamma_H, \Gamma_H) \right). \tag{58}$$

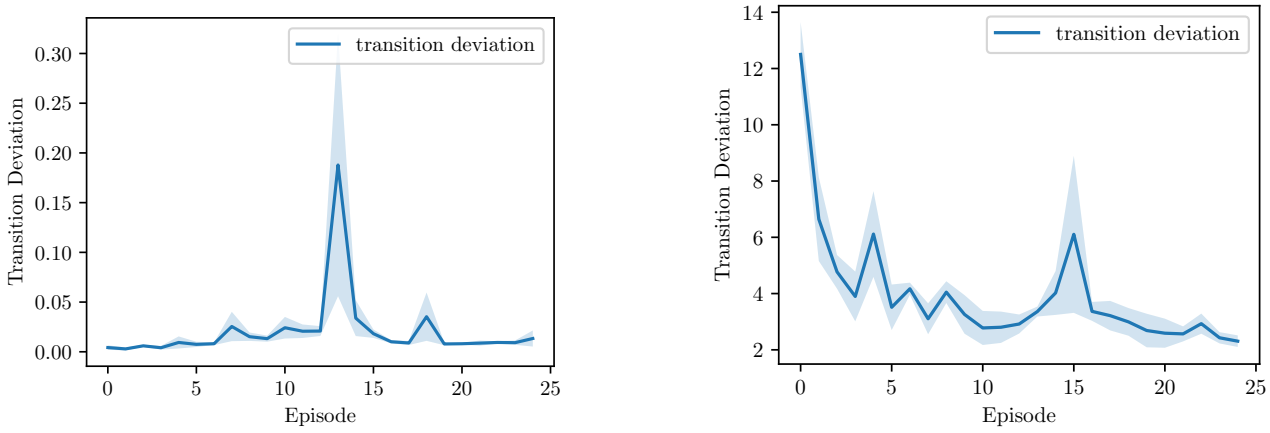
Theorem 4.2. *Under Assumptions 2.1, 2.2, 2.3, 3.1, and 4.1, for any $K \geq 1$, with probability $(1-\delta)$, the regret of Algorithm 1 is at most*

$$\mathrm{Reg}(K) \leq \mathcal{O}\left(\sqrt{2K} \left[HL_1 \sqrt{2HI_K^f(\mathcal{S}, \mathcal{A})} + L_2 \sqrt{I_K^g(\Gamma_H, \Gamma_H)} \right] \right). \tag{10}$$

Proof. This is a direct consequence of Corollary A.9. □

B. Experimental Details

Transition Model We employ an ensemble of probabilistic neural networks as in Chua et al. (2018). We train each of the 5 networks in the ensemble using Adam (Kingma & Ba, 2014) with a learning rate of 1×10^{-3}. The network architecture consists of a two-layer network with 32 units and Rectified Linear Units (ReLU) activations. Figure 3 shows the deviation of the predicted state from the true observed state.



(a) InvertedPendulum-v4 (b) HalfCheetah-v4

Figure 3. Transition function deviation for Inverted-Pendulum and Half-Cheetah (averaged over 4 runs).

Table 2. Hyperparameters for InvertedPendulum-v4

Parameter	Value
learning_rate	0.00025
n_steps	500
batch_size	256
n_epochs	5
gamma	0.999
gae_lambda	0.9
clip_range	0.4
ent_coef	1.5e-7
vf_coef	0.2
max_grad_norm	0.3
total_timesteps	100000

Table 3. Hyperparameters for HalfCheetah-v4

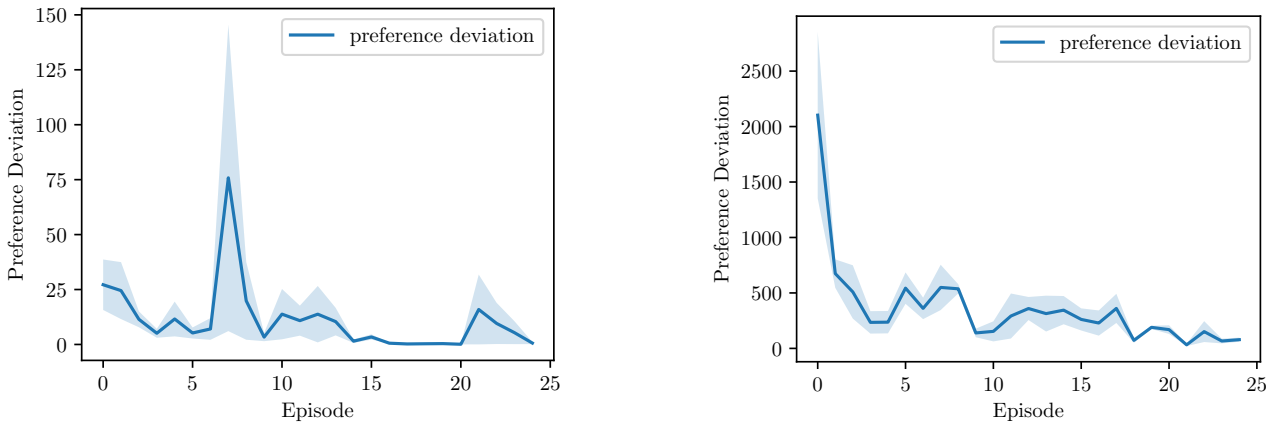
Parameter	Value
learning_rate	1.0633e-05
n_steps	512
batch_size	64
n_epochs	20
gamma	0.98
gae_lambda	0.92
clip_range	0.1
ent_coef	0.000401762
vf_coef	0.58096
max_grad_norm	0.8
total_timesteps	100000

Reward Model Similarly to the transition model, we use a probabilistic neural network with the same architecture for our reward function. We use again the Adam optimizer and a learning rate of 1×10^{-2}. Since we do not have access to the true rewards, we optimize the mean squared error on the preferences induced by the true and learned reward functions,

$$\mathcal{L}(\hat{r}) = \frac{1}{|P|} \sum_{(g,\tau,\tau') \in P} \left(g(\tau, \tau'; r) - g(\tau, \tau'; \hat{r}) \right)^2 \tag{59}$$

$$= \frac{1}{|P|} \sum_{(g,\tau,\tau') \in P} \left(\sum_{(\mathbf{s},\mathbf{a}) \in \tau} (r(\mathbf{s},\mathbf{a}) - \hat{r}(\mathbf{s},\mathbf{a})) - \sum_{(\mathbf{s}',\mathbf{a}') \in \tau'} (r(\mathbf{s}',\mathbf{a}') - \hat{r}(\mathbf{s}',\mathbf{a}')) \right)^2 . \tag{60}$$

Figure 4 shows the deviation of the preference induced by the predicted reward from the true preference.



(a) InvertedPendulum-v4 (b) HalfCheetah-v4

Figure 4. Preference function deviation for Inverted-Pendulum and Half-Cheetah (averaged over 4 runs).

Policy Search and Planning We use a modified version of PPO (Schulman et al., 2017) from Stable Baselines 3 (Raffin et al., 2021) to learn an optimal policy at every step on the learned transition model and reward function. This required a modification of the two Mujoco environments taken into consideration. We do this by modifying and extending the class definition of each environment provided in the Gymnasium library, an extension of OpenAI Gym (Brockman et al., 2016). Tables 2 and 3 provide the hyperparameters used for our experiments. For the planning step, we simply sample from the action distribution given by the learned policy.