# Hallucinated Adversarial Control for Conservative Offline Policy Evaluation

**Jonas Rothfuss**[*1] **Bhavya Sukhija**[*1] **Tobias Birchler**[*1] **Parnian Kassraie**[1] **Andreas Krause**[1]

[1]ETH Zurich, Switzerland

## Abstract

We study the problem of *conservative off-policy evaluation (COPE)* where given an offline dataset of environment interactions, collected by other agents, we seek to obtain a (tight) lower bound on a policy's performance. This is crucial when deciding whether a given policy satisfies certain minimal performance/safety criteria before it can be deployed in the real world. To this end, we introduce HAMBO, which builds on an uncertainty-aware learned model of the transition dynamics. To form a conservative estimate of the policy's performance, HAMBO hallucinates worst-case trajectories that the policy may take, within the margin of the models' epistemic confidence regions. We prove that the resulting COPE estimates are valid lower bounds, and, under regularity conditions, show their convergence to the true expected return. Finally, we discuss scalable variants of our approach based on Bayesian Neural Networks and empirically demonstrate that they yield reliable and tight lower bounds in various continuous control environments.

## 1 INTRODUCTION

Reinforcement learning methods require many interactions with their environment to successfully learn and evaluate policies. Therefore, they are rarely applied in challenging real-world applications such as medicine [Murphy et al., 2001], education [Mandel et al., 2014] or autonomous driving [Kiran et al., 2021], where a policy can only be deployed in the environment if it exceeds a pre-specified performance threshold or fulfills certain safety criteria. This leaves us with a challenging problem: How do we know whether a policy fulfills the necessary criteria so that it can safely interact with the environment, without testing it on the environment, and in the process, compromising safety?

Off-policy evaluation (OPE) aims to solve this problem by estimating the performance of an evaluation policy, using only offline data that was previously collected by other agents [e.g. Precup et al., 2001, Dudík et al., 2011]. In practice, offline datasets are often recorded interactions of a human expert with the environment. Since the evaluation policy typically induces a different action-state distribution than offline data, OPE methods often have to make predictions under strong distribution shifts. As a result, most existing OPE estimators suffer from high variance and are prone to overestimating the performance of the policy [Thomas et al., 2015]. In safety-critical applications, we can not risk and deploy a policy that is potentially much worse than what the OPE estimate suggests. Therefore, we aim for *conservative off-policy evaluation (COPE)* which seeks a (tight) lower bound on the evaluation policy's expected return that holds with high probability. Once deployed, the policy may end up exploring areas that were not included in the offline data. Thus, reliably bounding the worst-case performance can be quite challenging.

We develop a novel *model-based* COPE approach that hinges upon two key ideas: *epistemic uncertainty* and *pessimism*. In particular, our approach –HAMBO – builds on a learned statistical model of the transition dynamics that is able to quantify epistemic uncertainty. To obtain a valid lower bound on the policy performance, HAMBO hallucinates adversarial/worst-case trajectories the agent may take within the epistemic confidence sets of the model.

We prove that HAMBO reliably yields a high-probability bound on the true expected return of the policy, even when the offline data does not cover the areas explored by the evaluation policy (Proposition 3.2). Under regularity conditions, we further show that our conservative estimate *converges* from below to the true expected return (Theorem 3.8). To the best of our knowledge, HAMBO is the first provably consistent and conservative approach for OPE in continuous action-state spaces. We then propose scalable

[*]Equal contribution.

Bayesian neural network (BNN) variants of HAMBO and empirically evaluate them on various continuous control tasks. Importantly, we demonstrate that, *even when the regularity conditions are not met*, HAMBO reliably provides tight lower bounds on the true expected return.

## 2 PROBLEM SETTING

We consider a finite horizon Markov decision process (MDP) $\mathcal{M} = (\mathcal{S}, \mathcal{A}, p_0, p, r, T)$ with continuous state and action spaces $\mathcal{S} \subseteq \mathbb{R}^{d_s}$ and $\mathcal{A} \subseteq \mathbb{R}^{d_a}$, initial state distribution $p_0(\boldsymbol{s}_0)$, reward function $r(\boldsymbol{a}_t, \boldsymbol{s}_t)$ and horizon $T \in \mathbb{N}$. In particular, we consider stochastic transition dynamics that are governed by $\boldsymbol{s}_{t+1} = \boldsymbol{f}(\boldsymbol{s}_t, \boldsymbol{a}_t) + \boldsymbol{\epsilon}_t$ where $\boldsymbol{f} : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ is unknown and $\boldsymbol{\epsilon}_t \in \mathbb{R}^{d_s}$ is independent, additive transition noise with distribution $p_{\boldsymbol{\epsilon}}(\boldsymbol{\epsilon}_t | \boldsymbol{s}_t, \boldsymbol{a}_t)$. Hence, the transition distribution $p$ follows as $p(\boldsymbol{s}_{t+1} | \boldsymbol{s}_t, \boldsymbol{a}_t) = p_{\boldsymbol{\epsilon}}(\boldsymbol{s}_{t+1} - f(\boldsymbol{s}_t, \boldsymbol{a}_t) | \boldsymbol{s}_t, \boldsymbol{a}_t)$. For simplicity, we assume that the reward function is known. However, all results can straightforwardly be extended to unknown rewards.

The agent interacts with the environment according to a policy $\pi(\boldsymbol{a}_t | \boldsymbol{s}_t)$, which is a distribution over actions, conditioned on the current state $\boldsymbol{s}_t$. The performance of a policy is typically measured by its expected return $J(\pi) := J_p(\pi) := \mathbb{E}_{\boldsymbol{s}_0 \sim p_0}[V^{\pi}_{p,0}(\boldsymbol{s}_0)]$ where $V^{\pi}_t(\boldsymbol{s}) := V^{\pi}_{p,t}(\boldsymbol{s}) := \mathbb{E}_{p,\pi}[G_t | S_t = \boldsymbol{s}]$ is the value function and $G_t := \sum_{t'=t+1}^{T} r(\boldsymbol{s}_{t'}, \boldsymbol{a}_{t'})$ is the return. For simplicity, we omit a discount factor in the return computation. However, all results presented can be straightforwardly adapted to discounted rewards. Furthermore, we denote the *occupancy measure* of policy $\pi$ as

$$\rho^{\pi}(\boldsymbol{s}, \boldsymbol{a}) := \frac{1}{T} \sum_{t=0}^{T-1} p(\boldsymbol{s}_t = s, \boldsymbol{a}_t = a | \pi, \mathcal{M}) \ ,$$

that is, the probability density function of being in state $\boldsymbol{s}$ and performing action $\boldsymbol{a}$ at any point of time $t = 0, ..., T-1$.

We study the problem of offline policy evaluation where the task is to evaluate the performance, i.e. estimate the expected return $J(\pi_e)$, of a given evaluation policy $\pi_e$ while only using an offline dataset $\mathcal{D}_b = \{(\boldsymbol{s}_i, \boldsymbol{a}_i, r_i, \boldsymbol{s}'_i)\}_{i=1}^{n}$ of observed transitions. The key challenge in OPE is the distribution shift between the (unknown) behavior policy $\pi_b$ which generated the dataset $\mathcal{D}_b$ and the policy $\pi_e$ which we would like to evaluate. If $\pi_b$ differs from $\pi_e$, their state occupancy measures $\rho^{\pi_b}$ and $\rho^{\pi_e}$ can look significantly different. As a result, the dataset $\mathcal{D}_b$ which is generated based on $\rho^{\pi_b}$ may contain many samples in regions of the state-action space which $\pi_e$ is unlikely to visit and limited data in regions that are relevant for accurately evaluating $\pi_e$. In some cases, the support of $\rho^{\pi_b}$ might not even contain the support of $\rho^{\pi_e}$, i.e., $\exists (\boldsymbol{s}, \boldsymbol{a}) \in \mathcal{S} \times \mathcal{A} : \rho^{\pi_e}(\boldsymbol{s}, \boldsymbol{a}) > 0 \wedge \rho^{\pi_b}(\boldsymbol{s}, \boldsymbol{a}) = 0$. Since OPE methods have to make predictions under such strong distribution shifts their estimates suffer from high variance and are prone to overestimate the performance of the policy. OPE is particularly relevant in applications where we need to ensure a certain level of performance before a policy can be deployed online. Hence, it is often important to reliably determine whether or not the policy $\pi_e$ meets its minimum performance requirements. We formalize this problem as *conservative offline policy evaluation* (see Definition 2.1) where we want to ideally find a tight lower bound on the expected return that holds with high-probability:

**Definition 2.1** (Conservative Offline Policy Evaluation). *Let $\mathcal{M}$ be an MDP and $\mathcal{D}_b \in (\mathcal{S} \times \mathcal{A} \times \mathbb{R} \times \mathcal{S})^n$ a dataset of transitions, collected with a behavior policy $\pi_b$ on $\mathcal{M}$. Then the task of conservative OPE is: Given the offline dataset $\mathcal{D}_b$, a policy $\pi_e$ to evaluate and a confidence level $\delta \in (0, 1)$, find the largest possible lower-bound $b \in \mathbb{R}$, which satisfies $b \leq J(\pi_e)$ with probability at least $1 - \delta$.*

In some applications [e.g., Brunke et al., 2022], safety criteria are not directly encoded in the reward and instead, are expressed as additional constraints in the form of $\mathbb{E}_{(\boldsymbol{s},\boldsymbol{a}) \sim \rho^{\pi_e}}[c_i(\boldsymbol{s}, \boldsymbol{a})] \geq 0$. To determine with high confidence whether $\pi_e$ meets these constraints, we can apply COPE to each $c_i$ individually.

## 3 COPE VIA ADVERSARIAL TRANSITION MODELS

We take a model-based approach to COPE, and use a statistical model to estimate which transition functions $\boldsymbol{h} : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ from a hypothesis space $\mathcal{H}$ are plausible given the offline data $\mathcal{D}_b$ of size $n$. Then, we employ this statistical model of the transition dynamics to estimate the policy value $J(\pi_e)$. For this estimate, we want to guarantee with high probability that it does not exceed the true policy value. To ensure this, we need to be able to reliably quantify the *epistemic uncertainty* of our model estimates.

Uncertainty quantification can be done with either a frequentist approach that produces mean and confidence estimate $\boldsymbol{\mu}_n(\boldsymbol{s}, \boldsymbol{a})$ and $\boldsymbol{\sigma}_n(\boldsymbol{s}, \boldsymbol{a})$ or with a Bayesian model that maintains a posterior distribution $p(\boldsymbol{h} | \mathcal{D}_b)$ over dynamics models in $\mathcal{H}$. In the Bayesian case, we denote $\boldsymbol{\mu}_n(\boldsymbol{s}, \boldsymbol{a}) := \mathbb{E}_{\boldsymbol{h} \sim p(\boldsymbol{h} | \mathcal{D}_b)}[\boldsymbol{h}(\boldsymbol{s}, \boldsymbol{a})]$ as the posterior mean and $\boldsymbol{\sigma}^2_n(\boldsymbol{s}, \boldsymbol{a}) := \operatorname{diag}(\mathbb{E}_{\boldsymbol{h}, \boldsymbol{h}' \sim p(h | \mathcal{D}_b)}[\boldsymbol{h}(\boldsymbol{s}, \boldsymbol{a}) \boldsymbol{h}'(\boldsymbol{s}, \boldsymbol{a})^{\top}])$ as the posterior variance. In either case, we require that our statistical model of $\boldsymbol{h}$ is calibrated:

**Assumption 3.1** (Calibrated model). *A statistical model $(\boldsymbol{\mu}_n, \boldsymbol{\sigma}_n, \beta_n)$, with $\beta_n(\delta) \in \mathbb{R}^+$ as a scalar function that depends on the confidence level $\delta \in (0, 1]$, is calibrated with respect to $\boldsymbol{f}$ if, with probability at least $1 - \delta$, for all $(\boldsymbol{s}, \boldsymbol{a}) \in \mathcal{S} \times \mathcal{A}$ and $j = 1, \ldots, d_s$*

$$|\mu_{n,j}(\boldsymbol{s}, \boldsymbol{a}) - f_j(\boldsymbol{s}, \boldsymbol{a})| \leq \beta_n(\delta) \sigma_{n,j}(\boldsymbol{s}, \boldsymbol{a}),$$

*where $\mu_{n,j}$ and $\sigma_{n,j}$ denote the $j$-th element in the vector-valued functions $\boldsymbol{\mu}_n$ and $\boldsymbol{\sigma}_n$, respectively.*

Popular statistical models for transition dynamics that capture epistemic uncertainty are *Gaussian Processes (GPs)*

[Rasmussen and Williams, 2005], *Probabilistic Neural Network Ensembles* [Lakshminarayanan et al., 2017] and *Bayesian Neural Networks* [Blundell et al., 2015]. In later sections, we will attend to these specific choices of model in more detail and discuss when they are calibrated.

### 3.1 THE HAMBO FRAMEWORK

If our model is calibrated, we can, with high probability, use the confidence region

$$[\boldsymbol{\mu}_n(\boldsymbol{s},\boldsymbol{a}) - \beta_n(\delta)\boldsymbol{\sigma}_n(\boldsymbol{s},\boldsymbol{a}), \boldsymbol{\mu}_n(\boldsymbol{s},\boldsymbol{a}) + \beta_n(\delta)\boldsymbol{\sigma}_n(\boldsymbol{s},\boldsymbol{a})]$$

which is a $d_s$-dimensional hypercube, as a proxy for the true dynamics $\boldsymbol{f}(\boldsymbol{s},\boldsymbol{a})$. We then pessimistically select transitions within this region, to guarantee a high probability lower bound on the policy value $J(\pi_e)$. We do so, by introducing an adversary $\boldsymbol{\eta} : \mathcal{S} \times \mathcal{A} \to [-1,1]^{d_s}$ that, for every $(\boldsymbol{s},\boldsymbol{a}) \in \mathcal{S} \times \mathcal{A}$ picks a transition from the confidence region, thereby inducing the following hallucinated transition distribution:

$$\tilde{p}_{\boldsymbol{\eta}}(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t,\boldsymbol{a}_t) := p_{\boldsymbol{\epsilon}}\big(\boldsymbol{s}_{t+1} - \boldsymbol{\mu}_n(\boldsymbol{s}_t,\boldsymbol{a}_t) - \beta_n\boldsymbol{\eta}(\boldsymbol{s}_t,\boldsymbol{a}_t)\boldsymbol{\sigma}_n(\boldsymbol{s}_t,\boldsymbol{a}_t)\big). \tag{1}$$

This allows us to obtain a conservative value estimate for $\pi_e$

$$\tilde{J}(\pi_e) := \min_{\boldsymbol{\eta}} J_{\tilde{p}_{\boldsymbol{\eta}}}(\pi_e)\ . \tag{2}$$

This equation summarizes our approach *hallucinated adversarial model-based off-policy evaluation (*HAMBO*)* and Algorithm 1 presents the pseudo-code. Here, the expected reward $J_{\tilde{p}_\eta}(\pi_e)$ of $\pi_e$ under the hallucinated transition model $\tilde{p}_\eta$ can, e.g., be estimated via Monte Carlo estimation (i.e., generating trajectory rollouts and averaging the respecting returns). To find the adversary $\boldsymbol{\eta}(\boldsymbol{s},\boldsymbol{a})$ which minimizes (2), we can view $\boldsymbol{\eta}(\boldsymbol{s},\boldsymbol{a})$ as policy that aims to maximize $-J_{\tilde{p}_\eta}(\pi_e)$ and solve the corresponding optimal control problem. Importantly, with high probability, $\tilde{J}(\pi_e)$ is a lower bound on the true policy value $J(\pi_e)$:

**Proposition 3.2** (Valid lower bound)**.** *Given a calibrated model $(\boldsymbol{\mu}_n, \boldsymbol{\sigma}_n, \beta_n(\delta))$, the* HAMBO *estimates satisfy $\tilde{J}(\pi_e) \leq J(\pi_e)$, with probability greater than $1-\delta$.*

While Proposition 3.2 shows that our estimate $\tilde{J}(\pi_e)$ fulfills the requirements of COPE, $\tilde{J}(\pi_e)$ could potentially be very loose. However, we can further establish a worst-case lower bound on $\tilde{J}(\pi_e)$, if $\boldsymbol{f}, r, \boldsymbol{\sigma}_n$ and $\pi_e$ are continuous. Formally, we make the following Lipschitz continuity assumption:

**Assumption 3.3.** *(Lipschitz continuity) $\boldsymbol{f}$ is $L_f$-Lipschitz, $r$ is $L_r$-Lipschitz, $\boldsymbol{\sigma}$ is $L_\sigma$-Lipschitz and $\pi_e$ is $L_\pi$-Lipschitz w.r.t. the Wasserstein-1 distance, i.e., for all $\boldsymbol{s}, \boldsymbol{s}' \in \mathcal{S}$*

$$\mathcal{W}_1(\pi(\boldsymbol{a}|\boldsymbol{s}), \pi(\boldsymbol{a}|\boldsymbol{s}')) \leq L_\pi \|\boldsymbol{s} - \boldsymbol{s}\|_2. \tag{3}$$

Here, the continuity assumption on $\pi$ is expressed in terms the Wasserstein-1 distance and implies that a small change in the state space only induces a proportionally small change in the conditional action distribution of the policy. For instance, this is the case for policies that can be reparametrized with a Lipschitz function which is very common in practice:

**Example 3.4.** *Any policy $\pi(\boldsymbol{a}|\boldsymbol{s})$ that can be reparametrized as $\boldsymbol{g}(\boldsymbol{s},\boldsymbol{\zeta})$, where $\boldsymbol{\zeta} \sim p(\boldsymbol{\zeta})$ and $\boldsymbol{g}$ is $L_g$-Lipschitz, is also $L_g$-Lipschitz w.r.t. the $\mathcal{W}_1$- distance.*

Such Lipschitz assumptions are common in model-based OPE [e.g. Fonteneau et al., 2009, Paduraru, 2013] and RL more broadly [e.g. Berkenkamp et al., 2017, Curi et al., 2020], and, e.g, hold in many real-world control problems. With these regularity assumptions, we bound how far away the HAMBO estimate $\tilde{J}(\pi_e)$ is from the true policy value:

**Theorem 3.5.** *Under Assumption 3.1 and 3.3 we have, with probability at least $1-\delta$, that*

$$J(\pi_e) - \tilde{J}(\pi_e) \leq C_n \mathop{\mathbb{E}}_{(\boldsymbol{s},\boldsymbol{a})\sim\rho^{\pi_e}} [\|\boldsymbol{\sigma}(\boldsymbol{s},\boldsymbol{a})\|_2]$$

*where*

$$C_n := \bar{L}_r \left(1+\sqrt{d_s}\right) \beta_n T^2 \left(1+\bar{L}_f+(1+\sqrt{d_s})\beta_n\bar{L}_\sigma\right)^{T-1}$$

*with $\bar{L}_r := L_r(1+L_\pi)$ and $\bar{L}_f, \bar{L}_\sigma$ defined analogously.*

Tightness of the HAMBO lower bound $\tilde{J}(\pi_e)$ depends on the following key factors: Lipschitz-regularity, episode horizon $T$, and epistemic uncertainty. Mainly, smaller Lipschitz constants, shorter episode lengths improve the bound. Moreover, the smaller the expected epistemic standard deviation $\boldsymbol{\sigma}_n(\boldsymbol{s}_t,\boldsymbol{a}_t)$ under the state occupancy measure of $\pi_e$, the tighter the bound. While the first two factors are generally dictated by the problem instance, the epistemic uncertainty can be reduced by using more offline data (in the relevant areas of the state-action space). If we can show that the epistemic uncertainty shrinks sufficiently fast with the number of offline data points $n$ (i.e., faster than $\mathcal{O}(\beta_n^T)$), then we can prove that $\tilde{J}(\pi_e)$ converges to the true policy value as $n \to \infty$.

### 3.2 HAMBO WITH SMOOTH GP FUNCTIONS

In this section, we discuss the application of GPs for constructing calibrated confidence regions to be used for HAMBO. For the transition dynamics, we consider vector-valued functions $\boldsymbol{f}(\boldsymbol{s},\boldsymbol{a}) \mapsto (f_1(\boldsymbol{s},\boldsymbol{a}), ..., f_{d_s}(\boldsymbol{s},\boldsymbol{a}))$ such that the scalar-valued functions $f_j \in \mathcal{H}_k$ reside in a Reproducing Kernel Hilbert Space (RKHS) $\mathcal{H}_k$ with kernel function $k(\cdot,\cdot)$ and have bounded RKHS norm, i.e. $\|f_j\|_k \leq B$. We denote this space by $\boldsymbol{f} \in \mathcal{H}_{k,B}^{d_s} = \{[f_1, ..., f_{d_s}] : \|f_j\|_k \leq B, j = 1, ..., d_s\}$. We assume that transition noise $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma_\epsilon^2 \boldsymbol{I})$ is normally distributed with variance $\sigma_\epsilon^2$.

By fitting a zero-mean Gaussian Process $\mathcal{GP}(0,k)$ on each dimension $j = 1, ..., d_s$ of the next state $\boldsymbol{s}_{t+1}$, we can use the posterior means and variances to construct calibrated confidence sets. For brevity, we denote $\boldsymbol{x} := (\boldsymbol{s},\boldsymbol{a})$, so that

$$\begin{aligned} \mu_{n,j}(\boldsymbol{x}) &= \boldsymbol{k}_n^\top(\boldsymbol{x})(\boldsymbol{K}_n + \sigma_\epsilon^2 \boldsymbol{I})^{-1}\boldsymbol{y}_{n,j} \\ \sigma_{n,j}^2(\boldsymbol{x}) &= k(\boldsymbol{x},\boldsymbol{x}) - \boldsymbol{k}_n^T(\boldsymbol{x})(\boldsymbol{K}_n + \bar{\sigma}^2 \boldsymbol{I})^{-1}\boldsymbol{k}_n(\boldsymbol{x}) \end{aligned} \tag{4}$$

where $\boldsymbol{y}_{n,j} = [s'_{i,j}]_{i \leq n}^\top$ is the vector the $j$-th element of the observed next states $\boldsymbol{s}'_i$, $\boldsymbol{k}_n(\boldsymbol{x}) = [k(\boldsymbol{x}, \boldsymbol{x}_i)]_{i \leq n}^\top$, and $\boldsymbol{K}_n = [k(\boldsymbol{x}_i, \boldsymbol{x}_l)]_{i,l \leq n}$ is the kernel matrix. By concatenating the element-wise posterior mean and standard deviation, we obtain $\boldsymbol{\mu}_n(\boldsymbol{x}) = [\mu_{n,j}(\boldsymbol{x})]_{j \leq d_s}^\top$ and $\boldsymbol{\sigma}_n(\boldsymbol{x}) = [\sigma_{n,j}(\boldsymbol{x})]_{j \leq d_s}^\top$. Using this, we can construct calibrated confidence intervals that fulfill Assumption 3.1:

**Lemma 3.6** (Calibrated GP confidence sets)**.** *Let $\boldsymbol{f} \in \mathcal{H}_{k,B}^{d_s}$. Suppose $\boldsymbol{\mu}_n$ and $\boldsymbol{\sigma}_n$ are the posterior mean and variance of a GP with kernel $k$, fitted to $n$ noisy evaluations of $\boldsymbol{f}$. There exists $\beta_n(\delta)$, for which the tuple $(\boldsymbol{\mu}_n, \boldsymbol{\sigma}_n, \beta_n(\delta))$ satisfies Assumption 3.1 w.r.t. function $\boldsymbol{f}$.*

In Appendix B.2 we prove this lemma using results of Chowdhury and Gopalan [2017] and give the exact expression for a $\beta_n(\delta)$ that satisfies it. Generally, depends on the maximum information capacity $\gamma_n$ of the kernel (see Appx. B.2 or [Srinivas et al., 2012]).

In the described GP setting, we can also show Lipschitz continuity of $\boldsymbol{f}$ and $\boldsymbol{\sigma}$, if the kernel function $k$ is sufficiently regular:

**Lemma 3.7.** *If the kernel metric $d_k(\boldsymbol{x}, \boldsymbol{x}') := (k(\boldsymbol{x}, \boldsymbol{x}) + k(\boldsymbol{x}', \boldsymbol{x}') - 2k(\boldsymbol{x}, \boldsymbol{x}'))^{\frac{1}{2}}$ is $L_k$-Lipschitz, then every $\boldsymbol{f} \in \mathcal{H}_{k,B}^{d_s}$ is Lipschitz with $L_f = \sqrt{d_s} B L_k$ and the posterior standard deviation $\boldsymbol{\sigma}$ is Lipschitz with $L_\sigma = \sqrt{d_s} L_k$.*

For common kernels, the kernel metric is Lipschitz continuous, and thus Lemma 3.7 applies. For instance, for the linear kernel we have $L_k = 1$, for the RBF kernel we have $L_k = 1/\ell$ and for the Matern-$\nu$ kernel we have $L_k = \sqrt{\nu/(\nu-1)}/\ell$, where $\ell$ is the lengthscale and $\nu$ the smoothness parameter of the Matern kernel.

We can conclude that conditions of Proposition 3.2 and Theorem 3.5 are met when a GP is used for learning the transition dynamics from offline data. Hence, when the reward and the policy are Lipschitz, the HAMBO estimate satisfies

$$J(\pi_e) - C_n \mathop{\mathbb{E}}_{\rho^{\pi_e}} [\|\boldsymbol{\sigma}_n(\boldsymbol{s}, \boldsymbol{a})\|_2] \leq \tilde{J}(\pi_e) \leq J(\pi_e)$$

with high probability. We can show that given a dataset of i.i.d. trajectories, the difference term shrinks with $n$ sufficiently fast:

**Theorem 3.8** (Consistency of HAMBO)**.** *Let $r$ be $L_r$-Lipschitz, $\pi$ be $L_\pi$-Lipschitz w.r.t. the $\mathcal{W}_1$-distance and $\boldsymbol{f} \in \mathcal{H}_{k,B}^{d_s}$ where $k$ is a kernel for which $\beta_n(\delta)$ depends polylogarithmically on $n$. Suppose both $\rho^{\pi_e}$ and $\rho^{\pi_b}$ have a compact support and $supp(\rho^{\pi_e}) \subseteq supp(\rho^{\pi_b})$ and $\mathcal{D}_b$ consists of $n$ data points from i.i.d. trajectories according to the behavior policy $\pi_b$. Then as $n \to \infty$,*

$$\tilde{J}_n(\pi_e) \xrightarrow{\text{a.s.}} J(\pi_e).$$

The theorem implies that $\tilde{J}(\pi_e)$ is not only a conservative estimator for $J(\pi_e)$, but under certain regularity conditions, it is also a consistent estimator of the policy's true value. In Appendix B.3 we prove this theorem and give the exact rate at which the HAMBO estimate converges to the true value of $\pi_e$. This rate depends on the choice of kernel, time horizon $T$, and dimensions of the environment $(d_a, d_s)$. As an example, if $k$ is a Linear or RBF kernel, with high probability $|\tilde{J}_n(\pi_e) - J(\pi_e)| = \tilde{O}\left(n^{-1/2}\right)$ where $\tilde{O}$ omits polylogarithmic factors. To the best of our knowledge, Theorem 3.8 is the first result that shows the consistency of a model-based finite-horizon OPE method on a continuous environment.

## 4 HAMBO WITH NEURAL NETWORKS

In practice, we often want to evaluate policies in settings where the state and action spaces are higher-dimensional, and have access to larger amounts of offline data. In such environments, GPs become unpractical as they tend to generalize poorly in high-dimensional domains and their inference becomes prohibitively expensive for larger datasets.

**The NN-based Statistical Model.** In this section, we discuss practical variants of HAMBO which employ neural networks that scale more favorably to large datasets and high-dimensional domains. Crucially, we need to be able to quantify epistemic uncertainty. For this purpose, we employ Bayesian Neural Networks (BNNs) which model $\boldsymbol{h}_{\boldsymbol{\theta}}(\boldsymbol{s}, \boldsymbol{a})$ as a neural network function where $\boldsymbol{\theta}$ are the parameters of the neural network. BNNs presume a prior distribution $p(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}; 0, \lambda \boldsymbol{I})$ and maintain an approximation of the posterior $p(\boldsymbol{\theta}|\mathcal{D}_b) \propto p(\mathcal{D}_b|\boldsymbol{\theta})p(\boldsymbol{\theta})$ over neural network parameters. We use an independent Gaussian likelihood $p(\mathcal{D}_b|\boldsymbol{\theta}) = \prod_{i=1}^n \mathcal{N}(\boldsymbol{s}'_i; \boldsymbol{h}_{\boldsymbol{\theta}}(\boldsymbol{s}_i, \boldsymbol{a}_i), \boldsymbol{\nu}_{\boldsymbol{\theta}}^2(\boldsymbol{s}_i, \boldsymbol{a}_i))$ where $\boldsymbol{\nu}_{\boldsymbol{\theta}}^2(\boldsymbol{s}, \boldsymbol{a})$ is the vector of transition noise variances which is also predicted by the BNN.

We use Stein Variational Gradient Descent (SVGD) [Liu and Wang, 2016] to approximate the posterior as a set of $K$ particles $\Theta = \{\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_K\}$. We form the mean prediction of our model as the average prediction of the $K$ NNs:

$$\boldsymbol{\mu}_\Theta(\boldsymbol{s}, \boldsymbol{a}) = \frac{1}{K} \sum_{k=1}^K \boldsymbol{h}_{\boldsymbol{\theta}_k}(\boldsymbol{s}, \boldsymbol{a}).$$

Similarly, we estimate the epistemic variance as

$$\boldsymbol{\sigma}_{\Theta,e}^2(\boldsymbol{s}, \boldsymbol{a}) = \frac{1}{K} \sum_{k=1}^K (\boldsymbol{h}_{\boldsymbol{\theta}_k}(\boldsymbol{s}, \boldsymbol{a}) - \boldsymbol{\mu}_\Theta(\boldsymbol{s}, \boldsymbol{a}))^2.$$

The overall predictive distribution is the equally weighted mixture of all $K$ NN-based conditional Gaussians, i.e.,

$$p(\boldsymbol{s}'|\boldsymbol{s}, \boldsymbol{a}, \mathcal{D}_b) = \frac{1}{K} \sum_{k=1}^K \mathcal{N}(\boldsymbol{s}'; \boldsymbol{h}_{\boldsymbol{\theta}_k}(\boldsymbol{s}, \boldsymbol{a}), \boldsymbol{\nu}_{\boldsymbol{\theta}_k}^2(\boldsymbol{s}, \boldsymbol{a})) \tag{5}$$

whose variance is $\boldsymbol{\sigma}_\Theta^2(\boldsymbol{s}, \boldsymbol{a}) = \boldsymbol{\sigma}_{\Theta,e}^2(\boldsymbol{s}, \boldsymbol{a}) + \boldsymbol{\sigma}_{\Theta,a}^2(\boldsymbol{s}, \boldsymbol{a})$, where $\boldsymbol{\sigma}_{\Theta,a}^2(\boldsymbol{s}, \boldsymbol{a}) := \frac{1}{K} \sum_{k=1}^K \boldsymbol{\nu}_{\boldsymbol{\theta}_k}^2(\boldsymbol{s}, \boldsymbol{a})$ represents aleatoric and $\boldsymbol{\sigma}_{\Theta,e}^2(\boldsymbol{s}, \boldsymbol{a})$ the epistemic uncertainty.

**Calibrating the Model.** Since our BNN model uses approximate inference and a potentially misspecified prior, it may not satisfy the calibration condition of Assumption 3.1. Thus, we re-calibrate the model's uncertainty estimates with a calibration set $\mathcal{D}_c \subset \mathcal{D}_b$ that is withheld from the training. In particular, we use temperature scaling which chooses $\tau > 0$ such that the scaled predictive distribution (5) with variance $\tau^2 \boldsymbol{\sigma}^2_\Theta(\boldsymbol{s}, \boldsymbol{a})$ has a minimal empirical calibration error on $\mathcal{D}_c$ [Kuleshov et al., 2018]. Algorithm 3 formalizes this technique. Note that re-calibrating the BNN model does not guarantee formal calibration in the sense of Assumption 3.1. However, in our experiments, we found it to reliably yield a conservative value estimate $\tilde{J}(\pi_e)$.

### 4.1 PRACTICAL NN-BASED HAMBO VARIANTS

In the following, we discuss three ways of constructing adversarially hallucinated transition models based on our BNN model described in Equation (5). The formal pseudo-code of all algorithms is presented in Appendix A.

**Continuous Adversary (HAMBO-CA).** This approach directly reflects the hallucinated adversarial transition model, introduced in (1) and (2). The adversary $\boldsymbol{\eta}(\boldsymbol{s}, \boldsymbol{a}) \in [-1, 1]^{d_s}$ chooses mean of the Gaussian transition probability from the epistemic confidence set, i.e.,

$$\tilde{p}_{\boldsymbol{\eta}}(\boldsymbol{s}'|\boldsymbol{s}, \boldsymbol{a}) := \mathcal{N}\big(\boldsymbol{s}'; \boldsymbol{\mu}_\Theta(\boldsymbol{s}, \boldsymbol{a}) + \tau^2 \boldsymbol{\eta}(\boldsymbol{s}, \boldsymbol{a}) \boldsymbol{\sigma}^2_{\Theta,e}, \boldsymbol{\sigma}^2_{\Theta,a}(\boldsymbol{s}, \boldsymbol{a})\big).$$

To get the corresponding conservative value estimate $\tilde{J}(\pi_e)$, we need to solve the minimization problem $\min_{\boldsymbol{\eta}} J_{\tilde{p}_{\boldsymbol{\eta}}}(\pi_e)$. For this, we parameterize the adversary $\boldsymbol{\eta}(\boldsymbol{s}, \boldsymbol{a})$ as a neural network policy and use Soft Actor-Critic (SAC) [Haarnoja et al., 2018b] to maximize the negative return.

**Discrete Adversary (HAMBO-DA).** Our BNN posterior is approximated by a set of $K$ NNs whose mean squared error difference corresponds to epistemic uncertainty. Thus, we can also construct a pessimistic transition model by letting the adversary choose which of the $K$ NNs to pick. In this case, the adversary $\vartheta(k|\boldsymbol{s}, \boldsymbol{a})$ is a categorical distribution over the NN indices $\{1, ..., K\}$. The hallucinated transition model follows as:

$$\tilde{p}_\vartheta(\boldsymbol{s}'|\boldsymbol{s}, \boldsymbol{a}) := \sum_{k=1}^{K} \vartheta(k|\boldsymbol{s}, \boldsymbol{a}) \mathcal{N}\big(\boldsymbol{s}'; \boldsymbol{h}_{\boldsymbol{\theta}_k}(\boldsymbol{s}, \boldsymbol{a}), \boldsymbol{\nu}^2_{\boldsymbol{\theta}_k}(\boldsymbol{s}, \boldsymbol{a})\big).$$

Here, the adversary stochastically picks one of NN models at every step $t = 0, ..., T-1$. For this reason, we refer to this variant as DA1 (Algorithm 5). The corresponding value estimate follows as $\tilde{J}_{\text{DA1}}(\pi_e) = \min_\vartheta J_{\tilde{p}_\vartheta}(\pi_e)$. We solve the optimization problem by parameterizing the adversary $\vartheta$ as a NN policy and use the clipped double DQN algorithm Fujimoto et al. [2018] to maximize the negative return.

Alternatively, we can constrain the adversary so that it has to commit to one of the $K$ NN models for the entire trajectory. We refer to this variant as DAINF (Algorithm 6). In this case, the transition model corresponds to the predictive distribution of one of the NNs $p_{\boldsymbol{\theta}_k}(\boldsymbol{s}'|\boldsymbol{s}, \boldsymbol{a}) = \mathcal{N}\big(\boldsymbol{s}'; \boldsymbol{h}_{\boldsymbol{\theta}_k}(\boldsymbol{s}, \boldsymbol{a}), \boldsymbol{\nu}^2_{\boldsymbol{\theta}_k}(\boldsymbol{s}, \boldsymbol{a})\big)$, and the value estimate follows as the minimum the policy values under each of the models, i.e., $\tilde{J}_{\text{DAinf}}(\pi_e) = \min_{k \in \{1,...,K\}} J_{p_{\boldsymbol{\theta}_k}}(\pi_e)$. If $K$ is larger (e.g., $K > 20$), we recommend taking the empirical $\delta$ quantile of the policy values $\{J_{\tilde{p}_k}(\pi_e)\}_{k=1}^K$ instead of the minimum. In this case, DAINF has similarities to the model-based bootstrap approach of Kostrikov and Nachum [2020].

Naturally, the value estimates of DAINF are less pessimistic than those of DA1, i.e. $\tilde{J}_{\text{DAinf}}(\pi_e) \geq \tilde{J}_{\text{DA1}}(\pi_e)$, because the adversary cannot change which model it picks throughout the trajectory. In the experiment section, we investigate whether DAINF is still conservative enough to reliably yield lower bounds on the true policy values $J(\pi_e)$.

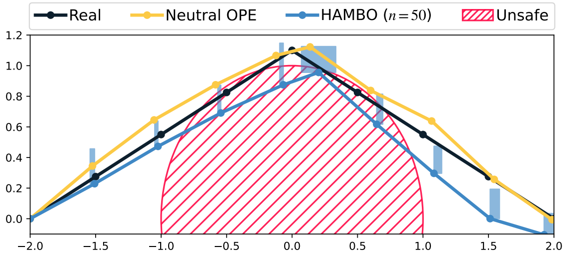

Figure 1: Hallucinated trajectories for model-based OPE and pessimistic HAMBO. While OPE overestimates the performance of the unsafe policy, HAMBO correctly gives a conservative estimates through its adversarial transition model. The adversary chooses the worst-case trajectory with the confidence sets (shaded blue areas).

## 5 EXPERIMENTS

We start this section by illustrating the inner workings of HAMBO with a toy example to show why pessimism is crucial for COPE. We demonstrate that the convergence guarantees from Section 3.2 materialize in practice for GP models. Finally, we empirically evaluate and compare the practical variants of HAMBO with BNNs on various continuous control tasks. For comparability between our environments, we shift and scale all our results so that the true policy return value $J(\pi_e)$ is 1.

### 5.1 ILLUSTRATIVE EXAMPLE

To illuminate the core idea of HAMBO and why pessimism is crucial for COPE, we conduct experiments on a toy environment which we call PointSafety (see Figure 1). In this environment, the agent navigates in the two-dimensional plane by applying actions $\boldsymbol{a} \in [-0.5, 0.5]^2$ such that its position (i.e, state $\boldsymbol{s} \in \mathcal{S} = \mathbb{R}^2$) changes to $\boldsymbol{s}_{t+1} = \boldsymbol{s}_t + \boldsymbol{a}_t$. The agent always starts on the left $\boldsymbol{s}_0 = (-2, 0)$ and aims to go to its goal on the right $\boldsymbol{s}_{\text{fin}} = (2, 0)$. However, the unit circle is a danger zone, in which the agent is subject to highly negative rewards (red shaded area).

We consider evaluation policies $\pi_y$ with an intermediate

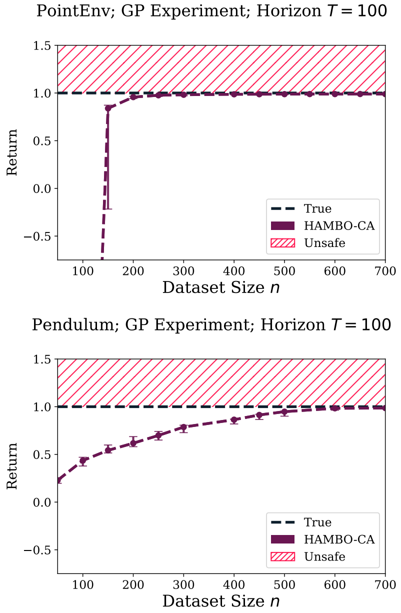

Figure 2: GP-based HAMBO for increasing offline dataset sizes $n$ evaluated on the PointEnv and Pendulum-v1. The lower bound approaches the true return.

goal $\boldsymbol{s}_{\text{im}} = (0, y)$ on the y-axis that goes in a straight line from $\boldsymbol{s}_0$ to $\boldsymbol{s}_{\text{im}}$ and then in a straight line from $\boldsymbol{s}_{\text{im}}$ to the goal $\boldsymbol{s}_{\text{fin}}$. Note that policies $\pi_y$ with $|y| \leq 1.155$ are unsafe.

We generate an offline dataset by rolling out the behavior policy $\pi_{1.6}$ with Gaussian action noise with a standard deviation of $0.1$. Then, we evaluate $\pi_{1.1}$, which is unsafe (see black trajectory), by rolling it out using HAMBO-CA.

We compare this to a neutral variant that predicts the next state with the predictive mean $\boldsymbol{\mu}_\Theta(\boldsymbol{s}, \boldsymbol{a})$, i.e., without pessimism. As we can observe from the yellow trajectory, it falsely estimates $\pi_{1.1}$ as safe, that is, it predicts that the trajectory lies outside of the danger zone. The trajectories with the adversarial transition model and the corresponding epistemic confidence sets for every step are depicted in Fig 1. The adversary successfully moves the prediction towards the danger zone within the confidence set, and, thus, correctly estimates the policy to be unsafe. Overall, this demonstrates a failure case of (neutral) off-policy evaluation and shows how HAMBO reliably gives a conservative estimate of the policy value through its pessimistic transition model.

## 5.2 EMPIRICAL CONVERGENCE OF HAMBO

For GP models, we show that HAMBO estimates converge to the true policy values (Theorem 3.8). Now, we empirically evaluate the behavior of GP-based HAMBO with an RBF kernel, as the number of offline data points grows. To this end, we consider two environments; a simple 2D PointEnv ($\mathcal{S} = \mathbb{R}^2$, $\mathcal{A} = [-1, 1]^2$), similar to the PointSaftey environment, and the Pendulum-v1 environment from the OpenAI Gym [Brockman et al., 2016]. In the PointEnv, the agent has to navigate the origin and accordingly receives the negative distance to the origin as a reward.

To generate the offline dataset, we collect transition data by uniformly sampling states and actions from the state and action space respectively. For the PointEnv, we restrict the sampled states to $[-40, 40]^2$ which covers the relevant part of the state space. As the evaluation policy, we use a proportional controller for the PointEnv, and a controller learned with SAC for the Pendulum.

Figure 2 plots the HAMBO estimates $\tilde{J}(\pi_e)$ for a varying number of offline datapoints $n = |\mathcal{D}_b|$. We notice that in the PointEnv, when we have insufficient data (here, ca. $n \leq 150$), the epistemic confidence regions of our GP model are large enough so that the transition model adversary sometimes manages to steer the policy outside the data support where the epistemic uncertainty is even higher. As a result, we see that $\tilde{J}(\pi_e)$ are initially far below the true expected return $J(\pi_e)$. However, as $n$ increases, the GP uncertainty regions become smaller, and, as we can observe in Figure 2, $\tilde{J}(\pi_e)$ becomes an increasingly tighter lower bound, approaching $J(\pi_e)$ for both the environments.

## 5.3 HAMBO FOR CONTINUOUS CONTROL

We evaluate the NN-based HAMBO methods from Section 4 on the continuous control tasks Pendulum-v1, Hopper-v3 and HalfCheetah-v3 from the OpenAI Gym and compare them to respective neutral (i.e., non-pessimistic) OPE algorithms.

Our general methodology is as follows: For a given environment, we first train a policy using the SAC algorithm Haarnoja et al. [2018a,b] and save several checkpoints of the agent. Then, some of the mediocre-performing checkpoints are rolled out to generate an offline dataset. After that, a given policy (usually one of the best checkpoints) is evaluated with the NN-based HAMBO variants.

We compare our approach to neutral OPE variants that do not use a pessimistic transition model [Fonteneau et al., 2013]. In particular, we consider various trajectory uncertainty propagation methods from Chua et al. [2018], employed in the context of OPE: First, we consider OPE-DS, where the transition model is approximated by a Gaussian $p(\boldsymbol{s}'|\boldsymbol{s}, \boldsymbol{a}) = \mathcal{N}(\boldsymbol{s}'; \boldsymbol{\mu}_\Theta(\boldsymbol{s}, \boldsymbol{a}); \boldsymbol{\sigma}^2_\Theta(\boldsymbol{s}, \boldsymbol{a}))$, here the variance is the sum of the epistemic and aleatoric variance. Second, we consider OPE-TS1 where the transition model is the mixture of predictive Gaussians in (5). This means that, in every step, one of the NN models is chosen uniformly at random to compute the next state distribution. Third, we consider OPE-TSINF, where, for every episode, we randomly commit to one of the $K$ NNs.

We investigate the following three aspects: 1) whether a method yields reliable lower bounds, 2) the effect of the offline dataset size, and 3) the curse of long horizons. Figure 3 and 4 report the estimated expected policy returns, averaged over 5 seeds, alongside the corresponding

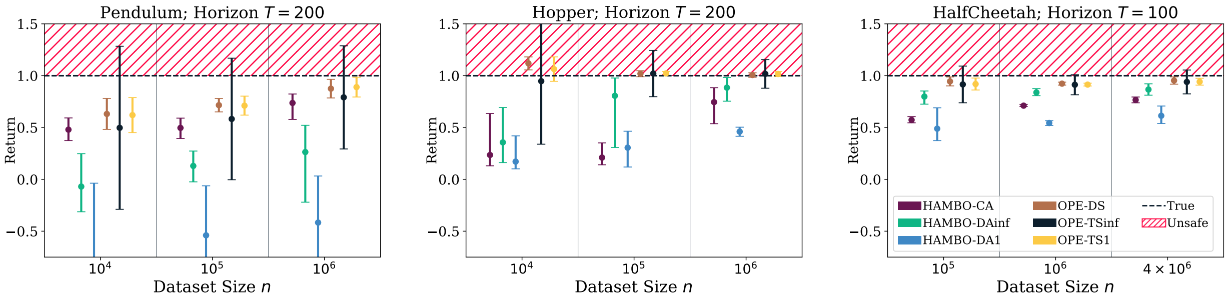

Figure 3: HAMBO variants and neutral OPE baselines for continuous control. Unlike neutral OPE, which frequently overestimates the true expected return, HAMBO always yields a valid lower bound, which becomes more accurate with $n$.

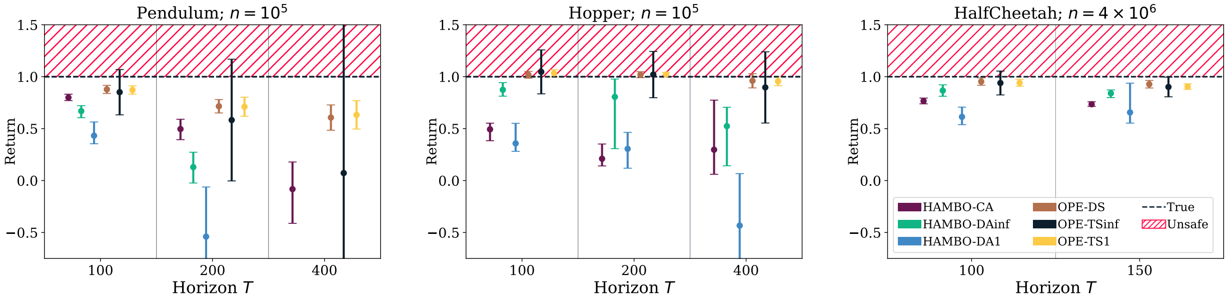

Figure 4: HAMBO variants and neutral OPE baselines for different horizons. With longer horizons, the variance of the neutral OPE estimates increases and HAMBO lower bounds become looser.

confidence intervals.

**Reliable Lower Bounds.** The HAMBO variants are designed to give reliable lower bounds on the true expected return. The results in Figure 3 and 4 empirically confirm that, across all seeds, all NN-based HAMBO variants reliably provide lower bounds on $J(\pi_e)$, and, thus, fulfill the COPE requirements from Definition 2.1. In contrast, the neutral OPE variants which do not introduce pessimism w.r.t. the epistemic uncertainty of the transition model fail to do so. In many cases, they overestimate the true policy value, particularly in the Hopper environment. This demonstrates the importance of pessimism in model-based COPE and affirms the validity of HAMBO, even with BNN models, where calibration (Definition 3.1) cannot be formally proven.

**Offline Dataset Size and Tightness.** The difference between HAMBO estimates $\tilde{J}(\pi_e)$ and the true expected reward $J(\pi_e)$ depends on the strength of the transition adversary, which is limited by the size of the epistemic confidence sets. As the size of the offline datasets $\mathcal{D}_b$ increases, we can generally expect the epistemic uncertainty to shrink. Thus, the adversary $\boldsymbol{\eta}$ becomes less powerful and the HAMBO estimates become an increasingly tight lower bound.

In Figure 3, we empirically investigate this effect by varying the offline dataset size $n$. As we hypothesized, we can observe the general trend that the HAMBO estimates come close to the true policy value, as $n$ increases. Moveover, we observe that the HAMBO-DA1 estimates are always strictly smaller than those of the HAMBO-DAINF variant. This is expected, since in the DA1 variant, the adversary can pick the worst-case NN transition model at every step while in the case of DAINF the adversary can only do so per trajectory, and, thus has less power. Since our experiment results indicate that the pessimism in HAMBO-DAINF is sufficient to obtain reliable lower bounds in practice, we conclude that HAMBO-DAINF is the preferred choice among the two. While HAMBO-DAINF performs better in Hopper and HalfCheetah, HAMBO-CA yields the tightest lower bounds in the Pendulum environment.

**The Curse of the Long Horizons.** Finally, we investigate the effect of the horizon length $T$ on our COPE estimates. Over the course of a trajectory, the transition model estimation errors can compound and lead to large discrepancies. This is a well-studied phenomenon in model-based RL [e.g. see Janner et al., 2019]. In our case, this is reflected by the worst-case lower bound in Theorem 3.5 which depends exponentially on $T$.

To evaluate the empirical effect of horizon length, we report the (C)OPE estimates for an offline dataset of size $n = 10^5$ across varying horizon lengths: $T$ = 100, 200 and 400 for the Pendulum and Hopper. For HalfCheetah, we only report horizon lengths of $T$ = 100 and 150. Figure 4 displays the corresponding results. For an increasing horizon length, the variance of the neutral variants increases and the lower bounds of the conservative HAMBO estimates become looser. However, the observed decline in tightness in Figure 4 is much less pronounced than the exponential decline of the worst-case bound in Theorem 3.5.

For large horizon lengths, it can happen that the hallucinated trajectory under the pessimistic transition model strives far outside the support of the offline data. In such cases, unlike neutral OPE methods, HAMBO will still provide lower bounds on the true expected return. However, these bounds can be very pessimistic. For instance, this can be observed in the case of Pendulum, where for $T = 400$ the estimates of HAMBO-DA1 and HAMBO-DAINF go out of the chart. Making accurate long-horizon predictions is generally very hard. For instance, this is discussed extensively in the context of model-based RL in Janner et al. [2019]. Often, a discount factor is used when computing returns to alleviate these issues. We highlight that we work with undiscounted returns and continuous state-action spaces, and, thus, operate in the most challenging setting for OPE.

## 6 RELATED WORK

This work mainly contributes to the literature on off-policy evaluation for MDPs, which we divide to three categories.

**Model-Free OPE.** The key challenge in OPE is to the distribution shift between behavior and evaluated policy. A popular natural approach to correct the distribution mismatch is to use importance sampling (IS) ratios to re-weight the rewards collected by the behavior policy [Precup et al., 2000, Dudík et al., 2011] or to adjust the recursive updates when estimating the values directly via the Bellman equation [Precup et al., 2001, Sutton et al., 2015, Hallak and Mannor, 2017]. Some work also combine both approaches to obtain a more favorable bias-variance trade-off [Jiang and Li, 2016, Thomas and Brunskill, 2016]. Unlike HAMBO, these approaches are model-free, i.e., they do not learn a model of the state transitions. However, they suffer from three key disadvantages: First, they have notoriously high variance, especially if the evaluated policy differs a lot from the behavior policy Levine et al. [2020]. Second, they require the support of the behavior occupancy measure $\rho^{\pi_b}$ to contain the support of $\rho^{\pi_e}$ which is often not the case. In contrast, HAMBO still provides valid estimates in this scenario. Third, to compute the importance ratios, they assume access to the distribution of behavior policy which is almost never the case in practical applications where data is often collected by human experts. HAMBO does not require access to the behavior policy and, thus, is much more broadly applicable.

A recent line of work [Nachum et al., 2019a,b, Zhang et al., 2020, Yang et al., 2020] estimates the state occupancy correction ratios via a form of fixed point iteration, and does not require access to the behavior policy. However, the Bellmann-like fixed point iteration is not applicable to the finite horizon case that we study in this paper. In addition, due to the fixed point iteration, it is very hard to quantify the uncertainty or bound error that is associated with such OPE estimates, making them poorly suited to COPE.

**Model-Based OPE.** This approach first learns the transition dynamics, to then simulate rollouts with the evaluation policy $\pi_e$ and thereby estimate the expected reward of $\pi_e$ [e.g., Fonteneau et al., 2013, Hanna et al., 2017, Kostrikov and Nachum, 2020]. Due to error in predicting the transitions, the resulting OPE estimate may overestimate the policy's performance which is prohibited in safety-critical applications. Our approach additionally simulates pessimistic trajectories using the model's epistemic uncertainty, to avoid overestimation. Further, to the best of our knowledge, Theorem 3.8 is the first consistency result for model-based OPE.

**COPE and High-Confidence OPE.** We study the problem of COPE which seeks a high-probability lower bound on the expected return. This is is closely related to estimating confidence bounds for OPE. Thomas et al. [2015] provide such confidence bounds for IS-based OPE estimates. However, due to the high variance of IS estimates, such bounds are often very loose [Levine et al., 2020]. Assuming that the $Q$-function resides in an RKHS, Feng et al. [2020, 2021], Shi et al. [2021] propose a model-free variational constrained optimization problem to directly give confidence intervals for $J(\pi)$. However these approaches only work for discounted, infinite-horizon MDPs or linear Q-functions, thus, are not generally applicable to our finite-horizon setting. Hanna et al. [2017], Kostrikov and Nachum [2020] use model-based bootstrapping to construct confidence intervals for the OPE estimates. Kostrikov and Nachum [2020] prove the asymptotic correctness of the bootstrap confidence intervals only for finite state-action spaces. In contrast, we show the validity of our COPE estimates non-asymptotically for any $|\mathcal{D}_b|$, and in continuous state-action spaces. Alternatively, Fonteneau et al. [2009] and Paduraru [2013] employ a Lipschitz argument to obtain valid COPE estimates. Our derivation of the worst-case lower bound in Theorem 3.5 also uses Lipschitz continuity. However, the HAMBO estimate provide a tighter lower bound on the true policy value, as we use the local confidence intervals rather than the global Lipschitz constants to introduce pessimism. Furthermore, unlike the mentioned work, HAMBO does not require knowledge of the Lipschitz constant and works with sub-Gaussian noise.

## 7 CONCLUSION

HAMBO, a novel approach for COPE that forms a pessimistic estimate of the expected return by hallucinating adversarial trajectories within the epistemic confidence regions of the estimated transition model. We formally prove the validity and consistency of the resulting COPE estimates. We propose various scalable NN-based variants of HAMBO and empirically demonstrate that they give reliable and tight lower bounds on the true expected return.

Importantly, our approach does not require access to the probability distribution of the behavior policy and gives reliable estimates, even when the support evaluation policy's occupancy measure is not contained in the offline data distribution. This makes HAMBO particularly relevant for safety-critical real-world applications, where the offline data is mostly collected by human experts and we need to make reliable decisions about whether a given policy is good enough to be deployed.

HAMBO can be naturally combined with other offline reinforcement learning (ORL) algorithms to solve safety-critical ORL tasks. We leave this for future work to investigate.

### Acknowledgements

This research was supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program grant agreement no. 815943 and the Swiss National Science Foundation under NCCR Automation, grant agreement 51NF40 180545. Jonas Rothfuss was supported by an Apple Scholars in AI/ML fellowship. We thank Sebastian Curi for contributing to the initial idea for this project.

### References

Felix Berkenkamp, Matteo Turchetta, Angela Schoellig, and Andreas Krause. Safe model-based reinforcement learning with stability guarantees. *Advances in Neural Information Processing Systems (NeurIPS)*, 30, 2017.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *International Conference on Machine Learning (ICML)*, 2015.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv*, 2016.

Lukas Brunke, Melissa Greeff, Adam W Hall, Zhaocong Yuan, Siqi Zhou, Jacopo Panerati, and Angela P Schoellig. Safe learning in robotics: From learning-based control to safe reinforcement learning. *Annual Review of Control, Robotics, and Autonomous Systems*, 2022.

Sayak Ray Chowdhury and Aditya Gopalan. On kernelized multi-armed bandits. In *International Conference on Machine Learning*, 2017.

Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, 2006.

Sebastian Curi, Felix Berkenkamp, and Andreas Krause. Efficient model-based reinforcement learning through optimistic policy search and planning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Miroslav Dudík, John Langford, and Lihong Li. Doubly robust policy evaluation and learning. In *International Conference on Machine Learning (ICML)*, 2011.

Yihao Feng, Tongzheng Ren, Ziyang Tang, and Qiang Liu. Accountable off-policy evaluation with kernel bellman statistics. In *International Conference on Machine Learning (ICML)*, 2020.

Yihao Feng, Ziyang Tang, na zhang, and qiang liu. Non-asymptotic confidence intervals of off-policy evaluation: Primal and dual bounds. In *International Conference on Learning Representations (ICLR)*, 2021.

Raphael Fonteneau, Susan Murphy, Louis Wehenkel, and Damien Ernst. Inferring bounds on the performance of a control policy from a sample of trajectories. In *IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning*, 2009.

Raphael Fonteneau, Susan A Murphy, Louis Wehenkel, and Damien Ernst. Batch mode reinforcement learning based on the synthesis of artificial trajectories. *Annals of Operations Research*, 208:383–416, 2013.

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

Scott Fujimoto, Herke van Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International Conference on Machine Learning (ICML)*, 2018.

Tuomas Haarnoja, Aurick Zhou, P. Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning (ICML)*, 2018a.

Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, and Sergey Levine. Soft actor-critic algorithms and applications. *arXiv*, 2018b.

Assaf Hallak and Shie Mannor. Consistent on-line off-policy evaluation. In *International Conference on Machine Learning (ICML)*, 2017.

Josiah P. Hanna, Peter Stone, and Scott Niekum. Bootstrapping with models: Confidence intervals for off-policy evaluation. In *AAAI*, 2017.

Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to trust your model: Model-based policy optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Nan Jiang and Lihong Li. Doubly robust off-policy value evaluation for reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2016.

Rahul Kidambi, Aravind Rajeswaran, Praneeth Netrapalli, and Thorsten Joachims. Morel: Model-based offline reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

B Ravi Kiran, Ibrahim Sobh, Victor Talpaert, Patrick Mannion, Ahmad A Al Sallab, Senthil Yogamani, and Patrick Pérez. Deep reinforcement learning for autonomous driving: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 2021.

Ilya Kostrikov and Ofir Nachum. Statistical bootstrapping for uncertainty estimation in off-policy evaluation. *arXiv*, 2020.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. In *International Conference on Learning Representations (ICLR)*, 2022.

Volodymyr Kuleshov, Nathan Fenner, and Stefano Ermon. Accurate uncertainties for deep learning using calibrated regression. In *International Conference on Machine Learning (ICML)*, 2018.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv*, 2020.

Qiang Liu and Dilin Wang. Stein variational gradient descent: A general purpose bayesian inference algorithm. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

Travis Mandel, Yun-En Liu, Sergey Levine, Emma Brunskill, and Zoran Popovic. Offline policy evaluation across representations with applications to educational games. In *AAMAS*, 2014.

Susan A Murphy, Mark J van der Laan, James M Robins, and Conduct Problems Prevention Research Group. Marginal mean models for dynamic regimes. *Journal of the American Statistical Association*, 2001.

Ofir Nachum, Yinlam Chow, Bo Dai, and Lihong Li. Dualdice: Behavior-agnostic estimation of discounted stationary distribution corrections. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019a.

Ofir Nachum, Bo Dai, Ilya Kostrikov, Yinlam Chow, Lihong Li, and Dale Schuurmans. Algaedice: Policy gradient from arbitrary experience. *arXiv*, 2019b.

Cosmin Paduraru. *Off-policy evaluation in Markov decision processes*. PhD thesis, McGill University, 2013.

Doina Precup, Richard S. Sutton, and Satinder P. Singh. Eligibility traces for off-policy policy evaluation. In *International Conference on Machine Learning (ICML)*, 2000.

Doina Precup, Richard S Sutton, and Sanjoy Dasgupta. Off-policy temporal-difference learning with function approximation. In *International Conference on Machine Learning (ICML)*, 2001.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.

Marc Rigter, Bruno Lacerda, and Nick Hawes. Rambo-rl: Robust adversarial model-based offline reinforcement learning. *arXiv*, 2022.

Chengchun Shi, Runzhe Wan, Victor Chernozhukov, and Rui Song. Deeply-debiased off-policy interval estimation. In *International Conference on Machine Learning (ICML)*, 2021.

Niranjan Srinivas, Andreas Krause, Sham M. Kakade, and Matthias W. Seeger. Information-theoretic regret bounds for gaussian process optimization in the bandit setting. *IEEE Transactions on Information Theory*, 2012.

Richard S. Sutton, A. Rupam Mahmood, and Martha White. An emphatic approach to the problem of off-policy temporal-difference learning. *arXiv*, 2015.

Philip Thomas and Emma Brunskill. Data-efficient off-policy policy evaluation for reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2016.

Philip Thomas, Georgios Theocharous, and Mohammad Ghavamzadeh. High-confidence off-policy evaluation. *AAAI*, 2015.

Sattar Vakili, Kia Khezeli, and Victor Picheny. On information gain and regret bounds in gaussian process bandits. In *International Conference on Artificial Intelligence and Statistics*, 2021.

Mengjiao Yang, Ofir Nachum, Bo Dai, Lihong Li, and Dale Schuurmans. Off-policy evaluation via the regularized lagrangian. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

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

Tianhe Yu, Aviral Kumar, Rafael Rafailov, Aravind Rajeswaran, Sergey Levine, and Chelsea Finn. Combo: Conservative offline model-based policy optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Ruiyi Zhang, Bo Dai, Lihong Li, and Dale Schuurmans. Gendice: Generalized offline estimation of stationary values. In *International Conference on Learning Representations (ICLR)*, 2020.

# A ALGORITHM AND EXPERIMENT DETAILS

In the following, we provide algorithmic formalizations and implementation details of the HAMBO framework and its practical variants which were discussed in the main paper.

## A.1 GENERIC HAMBO ALGORITHM FROM SECTION 3.1

First, we formalize the general HAMBO framework from Section 3.1:

**Algorithm 1** HAMBO Framework

```
Require: Offline dataset $\mathcal{D}_b$, evaluation policy $\pi_e$, reward function $r(\cdot,\cdot)$, Horizon $T$, initial state distribution $p_0(\boldsymbol{s}_0)$
  $(\boldsymbol{\mu}_n, \boldsymbol{\sigma}_n, \beta_n) \leftarrow \mathrm{TrainModel}(\mathcal{D}_b)$    ▷ Train statistical model with offline data
  $\tilde{p}_{\boldsymbol{\eta}}(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t, \boldsymbol{a}_t) \leftarrow p_{\boldsymbol{\epsilon}}\big(\boldsymbol{s}_{t+1} - \boldsymbol{\mu}_n(\boldsymbol{s}_t, \boldsymbol{a}_t) - \beta_n \boldsymbol{\eta}(\boldsymbol{s}_t, \boldsymbol{a}_t)\boldsymbol{\sigma}_n(\boldsymbol{s}_t, \boldsymbol{a}_t)\big)$    ▷ Set up adversarial transition model.
  $\tilde{J}(\pi) \leftarrow \min_{\boldsymbol{\eta}} \mathbb{E}_{\boldsymbol{s}_0 \sim p_0}[\mathbb{E}_{p_{\boldsymbol{\eta}},\pi}[\sum_{t=0}^{T} r(\boldsymbol{s}_t, \boldsymbol{a}_t)]]$    ▷ Optimize adversary to get pessimistic value estimate.
  return $\tilde{J}(\pi)$
```

We estimate $J_{\tilde{p}_{\boldsymbol{\eta}}}(\pi_e) = \mathbb{E}_{\boldsymbol{s}_0 \sim p_0}[\mathbb{E}_{p_{\boldsymbol{\eta}},\pi}[\sum_{t=0}^{T} r(\boldsymbol{s}_t, \boldsymbol{a}_t)]]$ via Monte Carlo estimation, i.e., we roll out $L$ trajectories and estimate the expectation as the average of the trajectory return:

$$\hat{J}_{\tilde{p}_{\boldsymbol{\eta}}}(\pi_e) = \frac{1}{L} \sum_{l=1}^{L} \sum_{t=0}^{T} r(\boldsymbol{s}_{l,t}, \boldsymbol{a}_{l,t}) \;\; \text{where} \;\; \boldsymbol{s}_{l,0} \sim p_0, \; \boldsymbol{a}_{l,t} \sim \pi(\boldsymbol{a}|\boldsymbol{s}_{l,t}), \; \boldsymbol{s}_{l,t+1} \sim \tilde{p}_{\boldsymbol{\eta}}(\boldsymbol{s}'|\boldsymbol{s}_{l,t}, \boldsymbol{a}_{l,t}) \tag{6}$$

The optimization of the advesary corresponds to a standard optimal control problem for which we use traditional methods such as trajectory optimization or model-free RL algorithms such as SAC.

## A.2 BNN BASED HAMBO VARIANTS

### A.2.1 The BNN model

We use fully connected neural networks with 4 hidden layers each of size 256 with ReLU activation functions. Before training, the offline data inputs and targets are standardized. The NN takes the concatenated state and action as input (i.e., $d_s + d_a$ dimensional) and outputs a vector of size $2d_s$ which is split into two vectors of size $d_s$. The first one corresponds to the mean prediction $\boldsymbol{h}_{\boldsymbol{\theta}}(\boldsymbol{s}, \boldsymbol{a})$ and the second one is the raw the aleatoric standard deviation which is fed through a softplus function to ensure positivity of $\boldsymbol{\nu}^2_{\boldsymbol{\theta}}(\boldsymbol{s}, \boldsymbol{a})$

As BNN prior we use a standard Normal distribution over the NN parameters $\boldsymbol{\theta}$, i.e., $p(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}; \boldsymbol{0}, \boldsymbol{I})$. However, as commonly done for BNNs to alleviate the problems of prior misspecification, we add a temperature parameter $\tau$ to the prior, so that we have $p(\boldsymbol{\theta}|\mathcal{D}_b) \propto p(\mathcal{D}_b|\boldsymbol{\theta})p(\boldsymbol{\theta})^{\tau}$. This hyper-parameter is chosen to as $\tau = 0.0001$ for Pendulum and Hopper and $\tau = 0.01$ for the HalfCheetah control environment.

We use Stein Variational Gradient Descent (SVGD) [Liu and Wang, 2016] for approximate posterior inference. In particular, we approximate the posterior $p(\boldsymbol{\theta}|\mathcal{D}_b)$ with $K$ NN particles $\{\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_K\}$. After randomly initializing the parameters of the $K$ NNs, the parameters are iteratively updated with the SVGD update rule:

$$\boldsymbol{\theta}_k \leftarrow \boldsymbol{\theta}_k + \frac{1}{K} \sum_{k'=1}^{K} \left[k(\boldsymbol{\theta}_{k'}, \boldsymbol{\theta}_k)\nabla_{\boldsymbol{\theta}_{k'}} \log p(\boldsymbol{\theta}_{k'}|\mathcal{D}_b) + \nabla_{\boldsymbol{\theta}_{k'}} k(\boldsymbol{\theta}'_k, \boldsymbol{\theta}_k)\right] \;\; \forall k = 1, ..., K \; . \tag{7}$$

Here $k(\cdot,\cdot)$ is a kernel function on the space of NN parameters vectors. In our experiments, we use an RBF kernel $k(\boldsymbol{\theta}, \boldsymbol{\theta}') = \exp\left(-\|\boldsymbol{\theta} - \boldsymbol{\theta}'\|_2^2/(2\ell)\right)$ with a length scale of $\ell = 10$ and $K = 5$ NN particles. Note that the kernel here is different from the one in Section 3.2. Algorithm 2 summarizes how to obtain the SVGD BNN posterior approximation.

**Algorithm 2** SVGD

```
Require: Training data $\mathcal{D}$, number of particles $K$
  Initialize NN parameter vectors $\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_K$        ▷ Initialize SVGD particles.
  while not converged do
    $\log p(\boldsymbol{\theta}_k|\mathcal{D}) \leftarrow \log p(\mathcal{D}|\boldsymbol{\theta}_k) + \tau \log p(\boldsymbol{\theta}_k) \ \ \forall k = 1, ..., K$
    $\boldsymbol{\theta}_k \leftarrow \boldsymbol{\theta}_k + \frac{1}{K}\sum_{k'=1}^{K} \left[k(\boldsymbol{\theta}_{k'}, \boldsymbol{\theta}_k)\nabla_{\boldsymbol{\theta}_{k'}} \log p(\boldsymbol{\theta}_{k'}|\mathcal{D}) + \nabla_{\boldsymbol{\theta}_{k'}} k(\boldsymbol{\theta}'_k, \boldsymbol{\theta}_k)\right] \ \ \forall k = 1, ..., K$
  end while
  return $\{\boldsymbol{h}_{\boldsymbol{\theta}_1}, \boldsymbol{\nu}^2_{\boldsymbol{\theta}_1}, \cdots, \boldsymbol{h}_{\boldsymbol{\theta}_K}, \boldsymbol{\nu}^2_{\boldsymbol{\theta}_K}\}$        ▷ Return the $K$ NN predictive mean and aleatoric variance functions
```

### A.3 RECALIBRATION OF THE BNN UNCERTAINTY ESTIMATES

To obtain well-calibrated confidence sets for HAMBO, we recalibrate the BNNs predictive distribution. In particular, we use temperature scaling based on the regression calibration error Kuleshov et al. [2018]. We perform re-calibration based on the predictive distribution $\mathcal{N}(\boldsymbol{\mu}_\Theta(\boldsymbol{s}, \boldsymbol{a}), \boldsymbol{\sigma}^2_\Theta(\boldsymbol{s}, \boldsymbol{a}))$. The calibration error compares the predictive quantiles of this Normal distribution with the corresponding empirical frequencies of data points, that fall below the predicted quantiles. Formally, we define $\Phi^{-1}_{\boldsymbol{\tau}}(\alpha; \boldsymbol{s}, \boldsymbol{a}) : [0,1]^{d_s} \to \mathcal{S}$ as the quantile function (inverse cumulative density function) of $\mathcal{N}(\boldsymbol{\mu}_\Theta(\boldsymbol{s}, \boldsymbol{a}), \boldsymbol{\tau}^2\boldsymbol{\sigma}^2_\Theta(\boldsymbol{s}, \boldsymbol{a}))$ where $\boldsymbol{\tau} \in \mathbb{R}^{d_s}$ is the temperature scaling vector. Given a calibration dataset $\mathcal{D}_c = \{(\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{s}')\}$, the calibration error [Kuleshov et al., 2018] for multivariate distributions follows as

$$\mathrm{CalErr}(\boldsymbol{\tau}) := \frac{1}{d_s}\sum_{j=1}^{d_s} \frac{1}{|A|} \sum_{\alpha \in A} \left(\mathrm{EmpFreq}(\alpha, \boldsymbol{\tau})_j - \alpha\right)^2 \ , \tag{8}$$

where $A = \{0.1, \cdots, 0.9, 0.99\}$ is a set of confidence levels and

$$\mathrm{EmpFreq}(\alpha; \boldsymbol{\tau}) := \frac{1}{|\mathcal{D}_c|} \sum_{(\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{s}') \in \mathcal{D}_c} \mathbf{1}\{\boldsymbol{s}' \leq \Phi^{-1}_{\boldsymbol{\tau}}(\alpha; \boldsymbol{s}, \boldsymbol{a})\} \tag{9}$$

is a vector-valued function of the (per dimension) empirical frequencies of the prediction targets that fall below the $\alpha$ quantile. Finally, we recalibrate the BNN predictions, by choosing the variance scaling vector $\boldsymbol{\tau}$ such that the calibration error is minimized, i.e., we choose

$$\boldsymbol{\tau}^* = \arg\min_{\boldsymbol{\tau}} \mathrm{CalErr}(\boldsymbol{\tau}) \ . \tag{10}$$

Algorithm 3 summarizes this BNN re-calibration procedure:

**Algorithm 3** CALIBRATEBNN

```
Require: calibration dataset $\mathcal{D}_c$, predictive mean $\boldsymbol{\mu}(\cdot, \cdot)$, predictive variance $\boldsymbol{\sigma}^2(\cdot, \cdot)$
  $A \leftarrow \{0.1, \cdots, 0.9, 0.99\}$.        ▷ Fix a set of confidence levels.
  $\Phi^{-1}(\cdot; \boldsymbol{s}, \boldsymbol{a}, \boldsymbol{\tau}) : [0,1] \mapsto \mathbb{R}^{d_s}$ as the inverse CDF of the Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}(\boldsymbol{s}, \boldsymbol{a}), \boldsymbol{\tau}^2\boldsymbol{\sigma}^2(\boldsymbol{x}_i))$.
  Define $\mathrm{EmpFreq}(\alpha; \boldsymbol{\tau}) \leftarrow \frac{1}{|\mathcal{D}_c|}\sum_{(\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{s}') \in \mathcal{D}_c} \mathbf{1}\{\boldsymbol{s}' \leq \Phi^{-1}_{\boldsymbol{\tau}}(\alpha; \boldsymbol{s}, \boldsymbol{a}, \boldsymbol{\tau})\}$
  Define $\mathrm{CalErr}(\boldsymbol{\tau}) \leftarrow \frac{1}{d_s}\sum_{j=1}^{d_s} \frac{1}{|A|}\sum_{\alpha \in A} \left(\mathrm{EmpFreq}(\alpha, \boldsymbol{\tau})_j - \alpha\right)^2$
  return $\arg\min_{\boldsymbol{\tau}} \mathrm{CalErr}(\boldsymbol{\tau})$        ▷ Choose $\boldsymbol{\tau}$ that minimizes the calibration error
```

### A.4 THE NN-BASED HAMBO VARIANTS

Here, we provide algorithmic descriptions of the NN-Based HAMBO variants from Section 4 as well as details about their implementation and how the corresponding experiments were conducted.

#### A.4.1 HAMBO with a Continuous Adversary (HAMBO-CA)

HAMBO-CA directly reflects the hallucinated adversarial transition model, introduced in Section 3. The adversary $\boldsymbol{\eta}(\boldsymbol{s}, \boldsymbol{a}) \in [-1, 1]^{d_s}$ chooses the mean of the Gaussian transition probability from the epistemic confidence set, i.e.,

$$\tilde{p}_{\boldsymbol{\eta}}(\boldsymbol{s}'|\boldsymbol{s}, \boldsymbol{a}) := \mathcal{N}\big(\boldsymbol{s}'; \boldsymbol{\mu}_\Theta(\boldsymbol{s}, \boldsymbol{a}) + \tau^2\boldsymbol{\eta}(\boldsymbol{s}, \boldsymbol{a})\boldsymbol{\sigma}^2_{\Theta,e}, \boldsymbol{\sigma}^2_{\Theta,a}(\boldsymbol{s}, \boldsymbol{a})\big) \tag{11}$$

For obtaining the corresponding conservative value estimate $\tilde{J}(\pi_e) = \min_{\boldsymbol{\eta}} J_{\tilde{p}_{\boldsymbol{\eta}}}(\pi_e)$, we need to find the adversary $\boldsymbol{\eta}^\star$ that minimizes the expected return. For this, we parameterize the adversary $\boldsymbol{\eta}$ as a neural network policy with two hidden layers of size 256 with ReLU activations and a tanh squashed Gaussian conditional distribution over the adversary actions in $[-1, 1]^{d_s}$. We use SAC [Haarnoja et al., 2018b] to maximize the negative expected return of the adversary policy. As usual, to stabilize the SAC training and avoid Q-value overestimation, we use double critics and trailing target critics. The SAC training is conducted in rounds consisting of rollouts of 1000 episodes under the hallucinated transition model where actions are chosen by $\pi_e$, followed by 1000 gradient steps on the SAC objectives. For the gradient steps, we use a batch size of 1024 and the Adam optimizer with a learning rate of $10^{-3}$ for critic and policy and $5 * 10^{-5}$ for the SAC entropy parameter. After SAC has converged, we take the adversary policy $\boldsymbol{\eta}^\star$ and estimate the expected return $\hat{J}_{\tilde{p}_{\boldsymbol{\eta}^\star}}(\pi_e)$ of $\pi_e$ under the adversary transition model, induced by $\boldsymbol{\eta}^\star$ with $L = 10^4$ trajectories (see Eq. 6). The HAMBO-CA method is summarized in Algorithm 4.

**Algorithm 4** HAMBO-CA

```
Require: Offline dataset $\mathcal{D}_b$, evaluation policy $\pi_e$, Number of BNN particles $K$
    Select a subset of $\mathcal{D}_b$ as calibration set $\mathcal{D}_c$
    $\{\boldsymbol{h}_{\boldsymbol{\theta}_1}, \boldsymbol{\nu}^2_{\boldsymbol{\theta}_1}, \cdots, \boldsymbol{h}_{\boldsymbol{\theta}_K}, \boldsymbol{\nu}^2_{\boldsymbol{\theta}_K}\} \leftarrow \mathrm{SVGD}(\mathcal{D}_b \setminus \mathcal{D}_c, K)$    ▷ Train BNN via SVGD and get predictive NN functions
    $\boldsymbol{\mu}_\Theta(\boldsymbol{s}, \boldsymbol{a}) \leftarrow \frac{1}{K}\sum_{k=1}^K \boldsymbol{h}_{\boldsymbol{\theta}_k}(\boldsymbol{s}, \boldsymbol{a})$    ▷ Calculate posterior mean.
    $\boldsymbol{\sigma}^2_{\Theta,e}(\boldsymbol{s}, \boldsymbol{a}) \leftarrow \frac{1}{K}\sum_{k=1}^K (\boldsymbol{h}_{\boldsymbol{\theta}_k}(\boldsymbol{s}, \boldsymbol{a}) - \boldsymbol{\mu}_\Theta(\boldsymbol{s}, \boldsymbol{a}))^2$    ▷ Calculate epistemic uncertainty.
    $\boldsymbol{\sigma}^2_{\Theta,a}(\boldsymbol{s}, \boldsymbol{a}) \leftarrow \frac{1}{K}\sum_{k=1}^K \boldsymbol{\nu}^2_{\boldsymbol{\theta}_k}(\boldsymbol{s}, \boldsymbol{a})$    ▷ Calculate aleatoric uncertainty.
    $\tau \leftarrow \mathrm{CalibrateBNN}(\mathcal{D}_c, \boldsymbol{\mu}_\Theta, \boldsymbol{\sigma}^2_{\Theta,e} + \boldsymbol{\sigma}^2_{\Theta,a})$    ▷ Calibrate the model
    Initialize adversary policy $\boldsymbol{\eta}$
    $\tilde{p}_{\boldsymbol{\eta}}(\boldsymbol{s}'|\boldsymbol{s}, \boldsymbol{a}) \leftarrow \mathcal{N}\big(\boldsymbol{s}'; \boldsymbol{\mu}_\Theta(\boldsymbol{s}, \boldsymbol{a}) + \tau^2\boldsymbol{\eta}(\boldsymbol{s}, \boldsymbol{a})\boldsymbol{\sigma}^2_{\Theta,e}, \boldsymbol{\sigma}^2_{\Theta,a}(\boldsymbol{s}, \boldsymbol{a})\big)$    ▷ Setup hallucinated adversarial transition model
    $\boldsymbol{\eta}^\star \leftarrow \mathrm{SoftActorCritic}(-J_{\tilde{p}_{\boldsymbol{\eta}}}(\pi_e), \boldsymbol{\eta})$    ▷ Train adversary $\boldsymbol{\eta}$ via SAC to maximize the negative return
    $\tilde{J}(\pi_e) \leftarrow \hat{J}_{\tilde{p}_{\boldsymbol{\eta}^\star}}(\pi_e)$    ▷ Estimate expected return of $\pi_e$ via sampling (see Eq. 6)
    return $\tilde{J}(\pi_e)$
```

### A.4.2 HAMBO with a Discrete Adversary (HAMBO-DA1 and HAMBO-DAINF)

In the case of HAMBO-DA1 the adversary $\vartheta$ has discrete action $\{1, ..., K\}$, i.e. picking one of the $K$ particles. We parameterize the adversary policy as a neural network with two hidden layers of size 256 with ReLU activations and softmax-categorical distribution over the $K$ discrete actions. To train this adversary policy, we use clipped double DQN [Fujimoto et al., 2018]. The double DQN training is conducted in rounds consisting of rollouts of 1000 episodes under the hallucinated transition model where actions are chosen by $\pi_e$, followed by 1000 gradient steps on the DQN objectives. For the gradient steps, we use a batch size of 1024 and the Adam optimizer with a learning rate of $10^{-3}$. Once double DQN has converged, we take the adversary policy $\vartheta^\star$ and estimate the expected return $\hat{J}_{\tilde{p}_{\vartheta^\star}}(\pi_e)$ of $\pi_e$ under the adversary transition model, induced by $\vartheta^\star$ with $L = 10^4$ trajectories (see Eq. 6). The overall HAMBO-DA1 method is summarized in Algorithm 5.

**Algorithm 5** HAMBO-DA1

```
Require: Offline dataset $\mathcal{D}_b$, evaluation policy $\pi_e$, Number of BNN particles $K$
    $\{\boldsymbol{h}_{\boldsymbol{\theta}_1}, \boldsymbol{\nu}^2_{\boldsymbol{\theta}_1}, \cdots, \boldsymbol{h}_{\boldsymbol{\theta}_K}, \boldsymbol{\nu}^2_{\boldsymbol{\theta}_K}\} \leftarrow \mathrm{SVGD}(\mathcal{D}_b, K)$    ▷ Train BNN via SVGD and get predictive NN functions
    Initialize adversary policy $\vartheta$
    $\tilde{p}_\vartheta(\boldsymbol{s}'|\boldsymbol{s}, \boldsymbol{a}) := \sum_{k=1}^K \vartheta(k|\boldsymbol{s}, \boldsymbol{a})\mathcal{N}\big(\boldsymbol{s}'; \boldsymbol{h}_{\boldsymbol{\theta}_k}(\boldsymbol{s}, \boldsymbol{a}), \boldsymbol{\nu}^2_{\boldsymbol{\theta}_k}(\boldsymbol{s}, \boldsymbol{a})\big)$    ▷ Setup hallucinated adversarial transition model
    $\vartheta^\star \leftarrow \mathrm{DoubleDQN}(-J_{\tilde{p}_{\boldsymbol{\eta}}}(\pi_e), \vartheta)$    ▷ Train adversary $\vartheta$ via to maximize the negative return
    $\tilde{J}(\pi_e) \leftarrow \hat{J}_{\tilde{p}_{\vartheta^\star}}(\pi_e)$    ▷ Estimate expected return of $\pi_e$ via sampling (see Eq. 6)
    return $\tilde{J}(\pi_e)$
```

In contrast to HAMBO-DA1, HAMBO-DAINF uses a weaker adversary that has to commit to one of the BNN particles for the entire trajectory. As a result, the corresponding pessimistic HAMBO estimate can simply be chosen as the minimum of the expected evaluation policy return under each of the NN models in the particle approximation, i.e. $J(\pi_e) = \min_{k\in\{1,...,K\}} J_{p_{\boldsymbol{\theta}_k}}(\pi_e)$. The HAMBO-DAINF method is summarized in Algorithm 5.

**Algorithm 6** HAMBO-DAINF

```
Require: Offline dataset $\mathcal{D}_b$, evaluation policy $\pi_e$, Number of BNN particles $K$
    $\{\boldsymbol{h}_{\boldsymbol{\theta}_1}, \boldsymbol{\nu}^2_{\boldsymbol{\theta}_1}, \cdots, \boldsymbol{h}_{\boldsymbol{\theta}_K}, \boldsymbol{\nu}^2_{\boldsymbol{\theta}_K}\} \leftarrow \mathrm{SVGD}(\mathcal{D}_b, K)$    ▷ Train BNN via SVGD and get predictive NN functions
    $p_{\boldsymbol{\theta}_k}(\boldsymbol{s}'|\boldsymbol{s}, \boldsymbol{a}) \leftarrow \mathcal{N}\big(\boldsymbol{s}'; \boldsymbol{h}_{\boldsymbol{\theta}_k}(\boldsymbol{s}, \boldsymbol{a}), \boldsymbol{\nu}^2_{\boldsymbol{\theta}_k}(\boldsymbol{s}, \boldsymbol{a})\big)$
    $\tilde{J}(\pi_e) \leftarrow \min_{k \in \{1,\dots,K\}} \hat{J}_{p_{\boldsymbol{\theta}_k}}(\pi_e)$    ▷ Estimate return of $\pi_e$ for each model (see Eq. 6) and take minimum
    return $\tilde{J}(\pi_e)$
```

# B PROOFS AND DERIVATIONS

***Proof of Proposition 3.2.*** By Assumption 3.1 we have, with probability $1 - \delta$, uniformly over $\mathcal{S} \times \mathcal{A}$, that

$$|\boldsymbol{\mu}_n(\boldsymbol{s}, \boldsymbol{a}) - f(\boldsymbol{s}, \boldsymbol{a})| \leq \beta_n(\delta)\boldsymbol{\sigma}_n(\boldsymbol{s}, \boldsymbol{a}) \,. \tag{12}$$

Hence, there exists an (adversary) mapping $\boldsymbol{\eta}^\dagger : \mathcal{S} \times \mathcal{A} \mapsto [-1, 1]^{d_s}$ such that every $\forall\, \boldsymbol{s}, \boldsymbol{a} \in \mathcal{S} \times \mathcal{A}$ we have

$$f(\boldsymbol{s}, \boldsymbol{a}) = \boldsymbol{\mu}_n(\boldsymbol{s}, \boldsymbol{a}) + \beta_n(\delta)\boldsymbol{\eta}^\dagger(\boldsymbol{s}, \boldsymbol{a})\boldsymbol{\sigma}_n(\boldsymbol{s}, \boldsymbol{a}) \,, \tag{13}$$

and, thus the hallucinated transition model is equal to the true transition dynamics, i.e.,

$$\tilde{p}_{\boldsymbol{\eta}^\dagger}(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t, \boldsymbol{a}_t) = p_{\boldsymbol{\epsilon}}\big(\boldsymbol{s}_{t+1} - \boldsymbol{\mu}_n(\boldsymbol{s}, \boldsymbol{a}) + \beta\boldsymbol{\eta}(\boldsymbol{s}, \boldsymbol{a})^\dagger\boldsymbol{\sigma}_n(\boldsymbol{s}, \boldsymbol{a})\big) = p_{\boldsymbol{\epsilon}}\big(\boldsymbol{s}_{t+1} - f(\boldsymbol{s}, \boldsymbol{a})\big) = p(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t, \boldsymbol{a}_t) \,. \tag{14}$$

Finally, we can use this to show

$$\tilde{J}(\pi_e) := \min_{\boldsymbol{\eta}} J_{\tilde{p}_{\boldsymbol{\eta}}}(\pi_e) \leq J_{\tilde{p}_{\boldsymbol{\eta}^\dagger}}(\pi_e) = J_p(\pi_e) = J(\pi_e) \,, \tag{15}$$

which concludes the proof. □

***Proof of Example 3.4.*** If $\pi(\boldsymbol{a}|\boldsymbol{s})$ can be reparametrized as $g(\boldsymbol{s}, \boldsymbol{\zeta})$, where $\boldsymbol{\zeta} \sim p(\boldsymbol{\zeta})$ and $g$ is $L_g$-Lipschitz, we have that the two random variables are equal in distribution, i.e. $\boldsymbol{a} \overset{d}{=} g(\boldsymbol{s}, \boldsymbol{\zeta})$ . Therefore,

$$\begin{aligned}
\mathcal{W}_1(\pi(\boldsymbol{a}|\boldsymbol{s}), \pi(\boldsymbol{a}|\boldsymbol{s}')) &= \inf_{\gamma \in \Gamma(\pi(\boldsymbol{a}|\boldsymbol{s}), \pi(\boldsymbol{a}|\boldsymbol{s}'))} \mathbb{E}_{\boldsymbol{a}, \boldsymbol{a}' \sim \gamma}\left[\|\boldsymbol{a} - \boldsymbol{a}'\|_2\right] && (16)\\
&\leq \mathbb{E}_{\boldsymbol{a}, \boldsymbol{a}' \sim \tilde{\gamma}}\left[\|\boldsymbol{a} - \boldsymbol{a}'\|_2\right] = \mathbb{E}_{\boldsymbol{\zeta}}\left[\|g(\boldsymbol{s}, \boldsymbol{\zeta}) - g(\boldsymbol{s}', \boldsymbol{\zeta})\|_2\right] && (17)\\
&\leq L_g\|\boldsymbol{s} - \boldsymbol{s}'\|_2 && (18)
\end{aligned}$$

where $\tilde{\gamma}(\boldsymbol{a}, \boldsymbol{a}')$ is the joint probability distribution of $(g(\boldsymbol{s}, \boldsymbol{\zeta}), g(\boldsymbol{s}', \boldsymbol{\zeta}))$, and, thus a coupling. Hence, we have shown that $\pi(\boldsymbol{a}, \boldsymbol{s})$ is $L_g$-Lipschitz w.r.t. the Wasserstein-1 distance. □

## B.1 PROOF OF THEOREM 3.5

The following lemmata will be used to prove the theorem.

**Lemma B.1** (Reparameterizability of two random variables with covariates)**.** *Let $X$ and $Y$ be random variables with finite expectation and corresponding probability distributions $p$ and $q$. Then, we can reparameterize $Y$ as $Y \overset{d}{=} X + \boldsymbol{\omega}_{|X}$, where $\boldsymbol{\omega}_{|X}$ is a covariate that is generally dependent on $X$ and satisfies*

$$\mathbb{E}_X\mathbb{E}_{\boldsymbol{\omega}|X}\left[\|\boldsymbol{\omega}\|_2\right] = \mathcal{W}_1(p, q) \,, \tag{19}$$

*where $\mathcal{W}_1(p, q)$ is the Wasserstein-1 distance between $p$ and $q$.*

*Proof.* Recall that the Wasserstein-1 distance is defined as infimum over couplings between $p$ and $q$, i.e.,

$$\mathcal{W}_1 = \inf_{\gamma \in \Gamma(p,q)} \mathbb{E}_{X', Y' \sim \gamma}\left[\|X' - Y'\|_2\right] \tag{20}$$

If the expectation of $p$ and $q$ is finite, then the infimum over couplings in (20) is attained for some $\gamma^*(x,y)$. Now we construct the covariate $\boldsymbol{\omega}_{|X}$ which is defined by applying the change of variable $g_x(x,y) \mapsto (x, y-x) = (x, \boldsymbol{\omega})$ to $\gamma^*$, so that we get $\tilde{\gamma}^*(x,\boldsymbol{\omega}) = \gamma^*(x, x+\boldsymbol{\omega})$. The conditional distribution of the covariate $\boldsymbol{\omega}_{|X}$ is

$$\tilde{\gamma}^*(\boldsymbol{\omega}|x) = \frac{\gamma^*(x, x+\boldsymbol{\omega})}{\gamma^*(x)} = \frac{\gamma^*(x, x+\boldsymbol{\omega})}{p(x)}$$

.

Now, given our construction of $\boldsymbol{\omega}_{|X}$, we aim to show that $Y \overset{d}{=} X + \boldsymbol{\omega}_{|X}$. Define the random variable $Z := X + \boldsymbol{\omega}_{|X}$. Then we have

$$\begin{aligned} p(z) &= \int_{\mathcal{X}} p(x, z-x) dx = \int_{\mathcal{X}} p(x) \tilde{\gamma}^*(\underbrace{z-x}_{\boldsymbol{\omega}}|x) dx \qquad (21) \\ &= \int_{\mathcal{X}} \gamma^*(x, x + (z-x)) dx = \int_{\mathcal{X}} \gamma^*(x, z) dx = q(z) \qquad (22) \end{aligned}$$

which shows that the pdf of $z$ is $q$, the probability density of $Y$. Since $\gamma^*(x,y)$ is the coupling that minimizes the transport cost, we can write

$$\mathcal{W}_1(p,q) = \inf_{\gamma \in \Gamma(p,q)} \mathbb{E}_{x',y' \sim \gamma} \left[ \|x' - y'\|_2 \right] = \mathbb{E}_{x',y' \sim \gamma^*} \left[ \|x' - y'\|_2 \right] = \mathbb{E}_{x' \sim p(x')} \mathbb{E}_{\boldsymbol{\omega} \sim \tilde{\gamma}^*(\boldsymbol{\omega}|x')} \left[ \|\boldsymbol{\omega}\|_2 \right] \qquad (23)$$

which shows that $\mathbb{E}_X \mathbb{E}_{\boldsymbol{\omega}|X} \left[ \|\boldsymbol{\omega}\|_2 \right] = \mathcal{W}_1(p,q)$, and, thus concludes the proof. □

**Corollary B.2.** *Let $\pi(\boldsymbol{a}|\boldsymbol{s})$ be $L_\pi$-Lipschitz w.r.t. the Wasserstein-1 distance. For any arbitrary but fixed $\boldsymbol{s}, \boldsymbol{s}' \in \mathcal{S}$ we denote $A$ and $A'$ as the random variables that follow the conditional distributions $\pi(\boldsymbol{a}|\boldsymbol{s})$ and $\pi(\boldsymbol{a}'|\boldsymbol{s}')$ respectively. Then, we can construct a covariate $\boldsymbol{\omega}_{|A}$ such that $A' \overset{d}{=} A + \boldsymbol{\omega}_{|A}$ and $\mathbb{E}_A \mathbb{E}_{\boldsymbol{\omega}|A} \left[ \|\boldsymbol{\omega}\|_2 \right] \leq L_\pi \|\boldsymbol{s} - \boldsymbol{s}'\|_2$.*

*Proof.* The corollary directly follows from Lemma B.1 and the definition of the $L_\pi$-Lipschitz continuity w.r.t. the Wasserstein-1, i.e., that $\forall$ $\boldsymbol{s}, \boldsymbol{s}' \in \mathcal{S}$ we have that $\mathcal{W}_1(\pi(\boldsymbol{a}|\boldsymbol{s}), \pi(\boldsymbol{a}|\boldsymbol{s}')) \leq L_\pi \|\boldsymbol{s} - \boldsymbol{s}'\|_2$. □

**Lemma B.3** (Lipschitz continuity of Wasserstein-one distance implies Lipschitz continuity in expectation)**.** *Let $f : \mathcal{X}_1 \times \mathcal{X}_2 \to \mathcal{Y}$ be $L_f$ Lipschitz continuous and $x_2$ a random variable with distribution $p(\cdot|x_1)$ that is $L_p$ Lipschitz w.r.t. the Wasserstein-1 distance. Then we have*

$$\mathop{\mathbb{E}}_{x_2 \sim p(\cdot|x_1)} \left[ f(x_1, x_2) \right] - \mathop{\mathbb{E}}_{x_2' \sim p(\cdot|x_1')} \left[ f(x_1', x_2') \right] \leq \bar{L}_f ||x_1 - x_1'||.$$

*with $\bar{L}_f = L_f(1 + L_p)$.*

*Proof.*

$$\begin{aligned} \mathop{\mathbb{E}}_{x_2 \sim p(\cdot|x_1)} \left[ f(x_1, x_2) \right] - \mathop{\mathbb{E}}_{x_2' \sim p(\cdot|x_1')} \left[ f(x_1', x_2') \right] &= \mathop{\mathbb{E}}_{x_2 \sim p(\cdot|x_1)} \left[ f(x_1, x_2) \right] - \mathop{\mathbb{E}}_{x_2 \sim p(\cdot|x_1)} \left[ \mathop{\mathbb{E}}_{\boldsymbol{\omega} \sim \tilde{\gamma}^*(\boldsymbol{\omega}|x_2)} \left[ f(x_1', x_2 + \boldsymbol{\omega}) \right] \right] \\ &\qquad \text{(Lemma B.1.)} \\ &= \mathop{\mathbb{E}}_{x_2 \sim p(\cdot|x_1)} \left[ \mathop{\mathbb{E}}_{\boldsymbol{\omega} \sim \tilde{\gamma}^*(\boldsymbol{\omega}|x_2)} \left[ f(x_1, x_2) - f(x_1', x_2 + \boldsymbol{\omega}) \right] \right] \\ &\leq L_f \mathop{\mathbb{E}}_{x_2 \sim p(\cdot|x_1)} \left[ \mathop{\mathbb{E}}_{\boldsymbol{\omega} \sim \tilde{\gamma}^*(\boldsymbol{\omega}|x_2)} \left[ ||x_1 - x_1'||_2 + ||\boldsymbol{\omega}||_2 \right] \right] \quad \text{(Lipschitzness of } f\text{)} \\ &\leq L_f ||x_1 - x_1'||_2 + L_f L_p ||x_1 - x_1'||_2 \qquad \text{(Corollary B.2)} \\ &= L_f (1 + L_p) ||x_1 - x_1'||_2. \end{aligned}$$

□

In the following, we bound the difference between the pessimistic and true return with the distance between the true and pessimistic trajectory using the Lipschitz continuity of the reward function and the policy's Wasserstein-one distance.

**Lemma B.4** (Bound on difference between pessimistic and true return estimate)**.** *Under Assumption 3.3 we have*

$$\left|J(\pi_e) - \tilde{J}(\pi_e)\right| \leq \bar{L}_r \mathop{\mathbb{E}}_{\boldsymbol{\epsilon}_{0:T-1}, a_{0:T}} \left[ \mathop{\mathbb{E}}_{\boldsymbol{\omega}_{0:T}} \left[ \sum_{t=0}^{T-1} ||\boldsymbol{s}_t - \tilde{\boldsymbol{s}}_t||_2 \right]\right].$$

*where* $\bar{L}_r = L_r(1 + L_\pi)$.

*Proof.* We have

$$\begin{aligned}
\left|J(\pi_e) - \tilde{J}(\pi_e)\right| &= \left| \mathop{\mathbb{E}}_{\boldsymbol{s}_{0:T}, \boldsymbol{a}_{0:T}} \left[ \sum_{t=0}^{T-1} r(\boldsymbol{s}_t, \boldsymbol{a}_t) \right] - \mathop{\mathbb{E}}_{\tilde{\boldsymbol{s}}_{0:T}, \tilde{\boldsymbol{a}}_{0:T}} \left[ \sum_{t=0}^{T-1} r(\tilde{\boldsymbol{s}}_t, \tilde{\boldsymbol{a}}_t) \right] \right| \\
&= \left| \mathop{\mathbb{E}}_{\boldsymbol{\epsilon}_{0:T}, a_{0:T}} \left[ \sum_{t=0}^{T-1} r(\boldsymbol{s}_t, \boldsymbol{a}_t) \right] - \mathop{\mathbb{E}}_{\boldsymbol{\epsilon}_{0:T}, \boldsymbol{a}_{0:T}} \left[ \mathop{\mathbb{E}}_{\boldsymbol{\omega}_{0:T}} \left[ \sum_{t=0}^{T-1} r(\tilde{\boldsymbol{s}}_t, \boldsymbol{a}_t + \boldsymbol{\omega}_t) \right]\right] \right| && \text{(Lemma B.1)} \\
&= \left| \mathop{\mathbb{E}}_{\boldsymbol{\epsilon}_{0:T}, \boldsymbol{a}_{0:T}} \left[ \mathop{\mathbb{E}}_{\boldsymbol{\omega}_{0:T}} \left[ \sum_{t=0}^{T-1} r(\boldsymbol{s}_t, \boldsymbol{a}_t) - r(\tilde{\boldsymbol{s}}_t, \boldsymbol{a}_t + \boldsymbol{\omega}_t) \right]\right] \right|.
\end{aligned}$$

From Lemma B.1, know that $\mathbb{E}\, \boldsymbol{\omega} = \mathcal{W}_1(\pi(\cdot|\boldsymbol{s}_t), \pi(\cdot|\tilde{\boldsymbol{s}}_t))$, where $\pi(\cdot|\cdot)$ is continuous w.r.t. the WD-1 distance. Therefore,

$$\begin{aligned}
\left|J(\pi_e) - \tilde{J}(\pi_e)\right| &=\leq \mathop{\mathbb{E}}_{\boldsymbol{\epsilon}_{0:T}, \boldsymbol{a}_{0:T}} \left[ \mathop{\mathbb{E}}_{\boldsymbol{\omega}_{0:T}} \left[ \sum_{t=0}^{T-1} L_r(1 + L_\pi)||\boldsymbol{s}_t - \tilde{\boldsymbol{s}}_t||_2 \right]\right] && \text{(Lemma B.3)} \\
&= \bar{L}_r \mathop{\mathbb{E}}_{\boldsymbol{\epsilon}_{0:T}, \boldsymbol{a}_{0:T}} \left[ \mathop{\mathbb{E}}_{\boldsymbol{\omega}_{0:T}} \left[ \sum_{t=0}^{T-1} ||\boldsymbol{s}_t - \tilde{\boldsymbol{s}}_t||_2 \right]\right]. && (\bar{L}_r = L_r(1 + L_\pi))
\end{aligned}$$

□

Next, we bound the distance between the true and pessimistic trajectory with the epistemic uncertainty around the true trajectory.

**Lemma B.5** (Bound on pessimistic and true trajectory)**.** *Under Assumption 3.1 and 3.3 with probability at least* $1 - \delta$ *for all* $\boldsymbol{\eta} : \mathcal{S} \to [-1, 1]^{d_s}$ *we have for all* $t \in \{0, \dots, T\}$ *that*

$$\mathop{\mathbb{E}}_{\boldsymbol{\epsilon}_{0:T}, \boldsymbol{a}_{0:T}} \left[ \mathop{\mathbb{E}}_{\boldsymbol{\omega}_{0:T}} [\|\boldsymbol{s}_{t+1} - \tilde{\boldsymbol{s}}_{t+1}\|_2] \right] \leq \left(1 + \sqrt{d_s}\right) \beta \textstyle\sum_{i=0}^{(t+1)-1} \left(\bar{L}_f + \left(1 + \sqrt{d_s}\right) \beta \bar{L}_\sigma\right)^{(t+1)-1-i} \displaystyle\mathop{\mathbb{E}}_{\boldsymbol{\epsilon}_{0:T}, \boldsymbol{a}_{0:T}} \left[ \mathop{\mathbb{E}}_{\boldsymbol{\omega}_{0:T}} [\|\boldsymbol{\sigma}_n(\boldsymbol{s}_i, \boldsymbol{a}_i)\|_2] \right].$$

*Proof.* We prove by induction. For $t = 1$ we have

$$\begin{aligned}
\mathop{\mathbb{E}}_{\boldsymbol{\epsilon}_{0:T}, \boldsymbol{a}_{0:T}} \left[ \mathop{\mathbb{E}}_{\boldsymbol{\omega}_{0:T}} [\|\boldsymbol{s}_1 - \tilde{\boldsymbol{s}}_1\|_2] \right] &= \mathop{\mathbb{E}}_{\boldsymbol{\epsilon}_{0:T}, \boldsymbol{a}_{0:T}} \left[ \mathop{\mathbb{E}}_{\boldsymbol{\omega}_{0:T}} [\|\boldsymbol{f}(\boldsymbol{s}_0, \boldsymbol{a}_0) + \boldsymbol{\epsilon}_0 - \boldsymbol{\mu}_n(\boldsymbol{s}_0, \boldsymbol{a}_0) - \beta \boldsymbol{\sigma}_n(\boldsymbol{s}_0, \boldsymbol{a}_0)\boldsymbol{\eta}(\boldsymbol{s}_0, \boldsymbol{a}_0) - \boldsymbol{\epsilon}_0\|_2] \right] \\
& && \text{(Lemma B.1)} \\
&\leq \mathop{\mathbb{E}}_{\boldsymbol{\epsilon}_{0:T}, \boldsymbol{a}_{0:T}} \left[ \mathop{\mathbb{E}}_{\boldsymbol{\omega}_{0:T}} [\|\boldsymbol{f}(\boldsymbol{s}_0, \boldsymbol{a}_0) - \boldsymbol{\mu}_n(\boldsymbol{s}_0, \boldsymbol{a}_0)\|_2 + \|\beta_n \boldsymbol{\sigma}_n(\boldsymbol{s}_0, a_0)\boldsymbol{\eta}(\boldsymbol{s}_0)\|_2] \right] \\
&\leq \left(1 + \sqrt{d_s}\right) \beta_n \mathop{\mathbb{E}}_{\boldsymbol{\epsilon}_{0:T}, \boldsymbol{a}_{0:T}} \left[ \mathop{\mathbb{E}}_{\boldsymbol{\omega}_{0:T}} [\|\boldsymbol{\sigma}_n(\boldsymbol{s}_0, \boldsymbol{a}_0)\|_2] \right] && (\boldsymbol{\eta} \in [-1, 1]^{d_s})
\end{aligned}$$

We get the induction hypothesis that for an arbitrary but fixed $t \geq 0$ we have

$$\mathop{\mathbb{E}}_{\boldsymbol{\epsilon}_{0:T}, \boldsymbol{a}_{0:T}} \left[ \mathop{\mathbb{E}}_{\boldsymbol{\omega}_{0:T}} [\|\boldsymbol{s}_t - \tilde{\boldsymbol{s}}_t\|_2] \right] \leq \left(1 + \sqrt{d_s}\right) \beta_n \textstyle\sum_{i=0}^{t-1} \left(\bar{L}_f + \left(1 + \sqrt{d_s}\right) \beta \bar{L}_\sigma\right)^{t-1-i} \displaystyle\mathop{\mathbb{E}}_{\boldsymbol{\epsilon}_{0:T}, \boldsymbol{a}_{0:T}} \left[ \mathop{\mathbb{E}}_{\boldsymbol{\omega}_{0:T}} [\|\boldsymbol{\sigma}_n(\boldsymbol{s}_i, \boldsymbol{a}_i)\|_2] \right]$$

Now for the induction step we can first derive

$$
\begin{aligned}
\mathop{\mathbb{E}}_{\boldsymbol{\epsilon}_{0:T},\boldsymbol{a}_{0:T}}\left[\mathop{\mathbb{E}}_{\boldsymbol{\omega}_{0:T}}\left[\|\boldsymbol{s}_{t+1}-\tilde{\boldsymbol{s}}_{t+1}\|_2\right]\right] &= \mathop{\mathbb{E}}_{\boldsymbol{\epsilon}_{0:T},\boldsymbol{a}_{0:T}}\left[\mathop{\mathbb{E}}_{\boldsymbol{\omega}_{0:T}}\left[\|f(\boldsymbol{s}_t,\boldsymbol{a}_t)+\boldsymbol{\epsilon}_t-\boldsymbol{\mu}_n(\tilde{\boldsymbol{s}}_t,\tilde{\boldsymbol{a}}_t)-\beta_n\boldsymbol{\sigma}_n(\tilde{\boldsymbol{s}}_t,\tilde{\boldsymbol{a}}_t)\boldsymbol{\eta}(\tilde{\boldsymbol{s}}_t,\tilde{\boldsymbol{a}}_t)-\boldsymbol{\epsilon}_t\|_2\right]\right] \\
&= \mathop{\mathbb{E}}_{\boldsymbol{\epsilon}_{0:T},\boldsymbol{a}_{0:T}}\left[\mathop{\mathbb{E}}_{\boldsymbol{\omega}_{0:T}}\left[\|f(\boldsymbol{s}_t,\boldsymbol{a}_t)-\boldsymbol{\mu}_n(\tilde{\boldsymbol{s}}_t,\boldsymbol{a}_t+\boldsymbol{\omega}_t)-\beta_n\boldsymbol{\sigma}_n(\tilde{\boldsymbol{s}}_t,\boldsymbol{a}_t+\boldsymbol{\omega}_t)\boldsymbol{\eta}(\tilde{\boldsymbol{s}}_t,\boldsymbol{a}_t+\boldsymbol{\omega}_t)\|_2\right]\right] \tag*{(Lemma B.1)} \\
&\le \mathop{\mathbb{E}}_{\boldsymbol{\epsilon}_{0:T},\boldsymbol{a}_{0:T}}\left[\mathop{\mathbb{E}}_{\boldsymbol{\omega}_{0:T}}\left[\|f(\boldsymbol{s}_t,\boldsymbol{a}_t)-f(\tilde{\boldsymbol{s}}_t,\boldsymbol{a}_t+\boldsymbol{\omega}_t)+f(\tilde{\boldsymbol{s}}_t,\boldsymbol{a}_t+\boldsymbol{\omega}_t)-\boldsymbol{\mu}_n(\tilde{\boldsymbol{s}}_t,\boldsymbol{a}_t+\boldsymbol{\omega}_t)\|_2\right]\right] \\
&\quad + \mathop{\mathbb{E}}_{\boldsymbol{\epsilon}_{0:T},\boldsymbol{a}_{0:T}}\left[\mathop{\mathbb{E}}_{\boldsymbol{\omega}_{0:T}}\left[\|\beta_n\boldsymbol{\sigma}_n(\tilde{\boldsymbol{s}}_t,\boldsymbol{a}_t+\boldsymbol{\omega}_t)\boldsymbol{\eta}(\tilde{\boldsymbol{s}}_t,\boldsymbol{a}_t+\boldsymbol{\omega}_t))\|_2\right]\right] \\
&\le \mathop{\mathbb{E}}_{\boldsymbol{\epsilon}_{0:T},\boldsymbol{a}_{0:T}}\left[\mathop{\mathbb{E}}_{\boldsymbol{\omega}_{0:T}}\left[\|f(\boldsymbol{s}_t,\boldsymbol{a}_t)-f(\tilde{\boldsymbol{s}}_t,\boldsymbol{a}_t+\boldsymbol{\omega}_t)\|_2+\|f(\tilde{\boldsymbol{s}}_t,\boldsymbol{a}_t+\boldsymbol{\omega}_t)-\boldsymbol{\mu}_n(\tilde{\boldsymbol{s}}_t,\boldsymbol{a}_t+\boldsymbol{\omega}_t)\|_2\right]\right] \\
&\quad + \mathop{\mathbb{E}}_{\boldsymbol{\epsilon}_{0:T},\boldsymbol{a}_{0:T}}\left[\mathop{\mathbb{E}}_{\boldsymbol{\omega}_{0:T}}\left[\|\beta_n\boldsymbol{\sigma}_n(\tilde{\boldsymbol{s}}_t,\boldsymbol{a}_t+\boldsymbol{\omega}_t)\boldsymbol{\eta}(\tilde{\boldsymbol{s}}_t,\boldsymbol{a}_t+\boldsymbol{\omega}_t)\|_2\right]\right] \\
&\le \mathop{\mathbb{E}}_{\boldsymbol{\epsilon}_{0:T},\boldsymbol{a}_{0:T}}\left[\mathop{\mathbb{E}}_{\boldsymbol{\omega}_{0:T}}\left[\bar{L}_f\|\boldsymbol{s}_t-\tilde{\boldsymbol{s}}_t\|_2+\left(1+\sqrt{d_s}\right)\beta_n\|\boldsymbol{\sigma}_n(\tilde{\boldsymbol{s}}_t,\boldsymbol{a}_t+\boldsymbol{\omega}_t)\|_2\right]\right] \tag*{(Lemma B.3 and $\boldsymbol{\eta}\in[-1,1]^{d_s}$)}
\end{aligned}
$$

By applying the triangle inequality and adding and subtracting $\boldsymbol{\sigma}_n$ to the second term,

$$
\begin{aligned}
\mathop{\mathbb{E}}_{\boldsymbol{\epsilon}_{0:T},\boldsymbol{a}_{0:T}}\left[\mathop{\mathbb{E}}_{\boldsymbol{\omega}_{0:T}}\left[\|\boldsymbol{s}_{t+1}-\tilde{\boldsymbol{s}}_{t+1}\|_2\right]\right] &\le \mathop{\mathbb{E}}_{\boldsymbol{\epsilon}_{0:T},\boldsymbol{a}_{0:T}}\left[\mathop{\mathbb{E}}_{\boldsymbol{\omega}_{0:T}}\left[\bar{L}_f\|\boldsymbol{s}_t-\tilde{\boldsymbol{s}}_t\|_2+\left(1+\sqrt{d_s}\right)\beta_n\|\boldsymbol{\sigma}_n(\tilde{\boldsymbol{s}}_t,\boldsymbol{a}_t+\boldsymbol{\omega}_t)\|_2\right]\right] \\
&= \mathop{\mathbb{E}}_{\boldsymbol{\epsilon}_{0:T},\boldsymbol{a}_{0:T}}\left[\mathop{\mathbb{E}}_{\boldsymbol{\omega}_{0:T}}\left[\bar{L}_f\|\boldsymbol{s}_t-\tilde{\boldsymbol{s}}_t\|_2\right]\right] \\
&\quad + \mathop{\mathbb{E}}_{\boldsymbol{\epsilon}_{0:T},\boldsymbol{a}_{0:T}}\left[\mathop{\mathbb{E}}_{\boldsymbol{\omega}_{0:T}}\left[\left(1+\sqrt{d_s}\right)\beta_n\|\boldsymbol{\sigma}_n(\tilde{\boldsymbol{s}}_t,\boldsymbol{a}_t+\boldsymbol{\omega}_t)-\boldsymbol{\sigma}_n(\boldsymbol{s}_t,\boldsymbol{a}_t)+\boldsymbol{\sigma}_n(\boldsymbol{s}_t,\boldsymbol{a}_t)\|_2\right]\right] \\
&\le \mathop{\mathbb{E}}_{\boldsymbol{\epsilon}_{0:T},\boldsymbol{a}_{0:T}}\left[\mathop{\mathbb{E}}_{\boldsymbol{\omega}_{0:T}}\left[\bar{L}_f\|\boldsymbol{s}_t-\tilde{\boldsymbol{s}}_t\|_2+\left(1+\sqrt{d_s}\right)\beta_n\left(\|\boldsymbol{\sigma}_n(\tilde{\boldsymbol{s}}_t,\boldsymbol{a}_t+\boldsymbol{\omega}_t)-\boldsymbol{\sigma}_n(\boldsymbol{s}_t,\boldsymbol{a}_t)\|_2\right)\right]\right] \\
&\quad + \left(1+\sqrt{d_s}\right)\beta_n\mathop{\mathbb{E}}_{\boldsymbol{\epsilon}_{0:T},\boldsymbol{a}_{0:T}}\left[\mathop{\mathbb{E}}_{\boldsymbol{\omega}_{0:T}}\left[\left(\|\boldsymbol{\sigma}_n(\boldsymbol{s}_t,\boldsymbol{a}_t)\|_2\right)\right]\right] \\
&\le \mathop{\mathbb{E}}_{\boldsymbol{\epsilon}_{0:T},\boldsymbol{a}_{0:T}}\left[\mathop{\mathbb{E}}_{\boldsymbol{\omega}_{0:T}}\left[\bar{L}_f\|\boldsymbol{s}_t-\tilde{\boldsymbol{s}}_t\|_2+\left(1+\sqrt{d_s}\right)\beta_n\left(\bar{L}_\sigma\|\boldsymbol{s}_t-\tilde{\boldsymbol{s}}_t\|_2+\|\boldsymbol{\sigma}_n(\boldsymbol{s}_t,\boldsymbol{a}_t)\|_2\right)\right]\right] \tag*{(Lemma B.1)} \\
&= \mathop{\mathbb{E}}_{\boldsymbol{\epsilon}_{0:T},\boldsymbol{a}_{0:T}}\left[\mathop{\mathbb{E}}_{\boldsymbol{\omega}_{0:T}}\left[\left(\bar{L}_f+\left(1+\sqrt{d_s}\right)\beta_n\bar{L}_\sigma\right)\|\boldsymbol{s}_t-\tilde{\boldsymbol{s}}_t\|_2+\left(1+\sqrt{d_s}\right)\beta_n\|\boldsymbol{\sigma}_n(\boldsymbol{s}_t,\boldsymbol{a}_t)\|_2\right]\right]
\end{aligned}
$$

Next, we apply the induction hypothesis

$$
\begin{aligned}
\mathop{\mathbb{E}}_{\boldsymbol{\epsilon}_{0:T},\boldsymbol{a}_{0:T}}\left[\mathop{\mathbb{E}}_{\boldsymbol{\omega}_{0:T}}\left[\|\boldsymbol{s}_{t+1}-\tilde{\boldsymbol{s}}_{t+1}\|_2\right]\right] &\le \mathop{\mathbb{E}}_{\boldsymbol{\epsilon}_{0:T},\boldsymbol{a}_{0:T}}\left[\mathop{\mathbb{E}}_{\boldsymbol{\omega}_{0:T}}\left[\left(\bar{L}_f+\left(1+\sqrt{d_s}\right)\beta_n\bar{L}_\sigma\right)\|\boldsymbol{s}_t-\tilde{\boldsymbol{s}}_t\|_2+\left(1+\sqrt{d_s}\right)\beta_n\|\boldsymbol{\sigma}_n(\boldsymbol{s}_t,\boldsymbol{a}_t)\|_2\right]\right] \\
&\le \left[\left(\bar{L}_f+\left(1+\sqrt{d_s}\right)\beta_n\bar{L}_\sigma\right)\left(1+\sqrt{d_s}\right)\beta_n\right] \\
&\quad\times\left(\sum_{i=0}^{t-1}\left(\bar{L}_f+\left(1+\sqrt{d_s}\right)\beta_n\bar{L}_\sigma\right)^{t-1-i}\mathop{\mathbb{E}}_{\boldsymbol{\epsilon}_{0:T},\boldsymbol{a}_{0:T}}\left[\mathop{\mathbb{E}}_{\boldsymbol{\omega}_{0:T}}\left[\|\boldsymbol{\sigma}_n(\boldsymbol{s}_i,\boldsymbol{a}_i)\|_2\right]\right]\right. \\
&\qquad\left.+\mathop{\mathbb{E}}_{\boldsymbol{\epsilon}_{0:T},\boldsymbol{a}_{0:T}}\left[\mathop{\mathbb{E}}_{\boldsymbol{\omega}_{0:T}}\left[\|\boldsymbol{\sigma}_n(\boldsymbol{s}_t,\boldsymbol{a}_t)\|_2\right]\right]\right) \\
&= \left(1+\sqrt{d_s}\right)\beta_n\sum_{i=0}^{(t+1)-1}\left(\bar{L}_f+\left(1+\sqrt{d_s}\right)\beta_n\bar{L}_\sigma\right)^{(t+1)-1-i}\mathop{\mathbb{E}}_{\boldsymbol{\epsilon}_{0:T},\boldsymbol{a}_{0:T}}\left[\mathop{\mathbb{E}}_{\boldsymbol{\omega}_{0:T}}\left[\|\boldsymbol{\sigma}_n(\boldsymbol{s}_i,\boldsymbol{a}_i)\|_2\right]\right]
\end{aligned}
$$

□

Using the above lemmas, we present the proof to the main theorem.

***Proof of Theorem 3.5.***

$$\begin{aligned}\left|J(\pi_e) - \tilde{J}(\pi_e)\right| &\le \bar{L}_r \mathop{\mathbb{E}}_{\boldsymbol{\epsilon}_{0:T}, \boldsymbol{a}_{0:T}} \left[ \mathop{\mathbb{E}}_{\boldsymbol{\omega}_{0:T}} \left[ \sum_{t=0}^{T-1} ||\boldsymbol{s}_t - \tilde{\boldsymbol{s}}_t||_2 \right]\right] && \text{(Lemma B.4)} \\ &\le \bar{L}_r \sum_{t=0}^{T-1} \left(1+\sqrt{d_s}\right) \beta_n \sum_{i=0}^{(t+1)-1} \left(\bar{L}_f + \left(1+\sqrt{d_s}\right) \beta_n \bar{L}_\sigma\right)^{(t+1)-1-i} \mathop{\mathbb{E}}_{\boldsymbol{\epsilon}_{0:T}, \boldsymbol{a}_{0:T}} \left[ \mathop{\mathbb{E}}_{\boldsymbol{\omega}_{0:T}} \left[ \|\boldsymbol{\sigma}_n(\boldsymbol{s}_i, \boldsymbol{a}_i)\|_2 \right]\right] \\ & && \text{(Lemma B.5)}\end{aligned}$$

Since $\bar{L}_f \ge 1$, then for all $0 \le i \le t$ and $0 \le t \le T-1$,

$$\left(\bar{L}_f + \left(1+\sqrt{d_s}\right) \beta_n \bar{L}_\sigma\right)^{(t+1)-1-i} \le \left(\bar{L}_f + \left(1+\sqrt{d_s}\right) \beta_n \bar{L}_\sigma\right)^{T-1}$$

which allows us to write,

$$\begin{aligned}\left|J(\pi_e) - \tilde{J}(\pi_e)\right| &\le \bar{L}_r \left(1+\sqrt{d_s}\right) \beta_n \left(1 + \bar{L}_f + \left(1+\sqrt{d_s}\right) \beta_n \bar{L}_\sigma\right)^{T-1} T \mathop{\mathbb{E}}_{\boldsymbol{\epsilon}_{0:T}, \boldsymbol{a}_{0:T}} \left[ \mathop{\mathbb{E}}_{\boldsymbol{\omega}_{0:T}} \left[ \sum_{t=0}^{T-1} \|\boldsymbol{\sigma}_n(\boldsymbol{s}_t, \boldsymbol{a}_t)\|_2 \right]\right] \\ &= \bar{L}_r \left(1+\sqrt{d_s}\right) \beta_n \left(1 + \bar{L}_f + \left(1+\sqrt{d_s}\right) \beta_n \bar{L}_\sigma\right)^{T-1} T \mathop{\mathbb{E}}_{\boldsymbol{s}_{0:T}, \boldsymbol{a}_{0:T}} \left[ \sum_{t=0}^{T-1} \|\boldsymbol{\sigma}_n(\boldsymbol{s}_t, \boldsymbol{a}_t)\|_2 \right] \\ &= \left[\bar{L}_r \left(1+\sqrt{d_s}\right) \beta_n \left(1 + \bar{L}_f + \left(1+\sqrt{d_s}\right) \beta_n \bar{L}_\sigma\right)^{T-1} T\right] \\ &= \left[\bar{L}_r \left(1+\sqrt{d_s}\right) \beta_n \left(1 + \bar{L}_f + \left(1+\sqrt{d_s}\right) \beta_n \bar{L}_\sigma\right)^{T-1} T\right] \\ &\quad \times \left(\int_{\mathcal{S}\times\mathcal{A}} \sum_{t=0}^{T-1} p(s_t = \boldsymbol{s}, a_t = \boldsymbol{a} | \pi_e, \mathcal{M}) \|\boldsymbol{\sigma}_n(\boldsymbol{s}, \boldsymbol{a})\|_2 d\boldsymbol{s} d\boldsymbol{a}\right) \\ &= \bar{L}_r \left(1+\sqrt{d_s}\right) \beta_n \left(1 + \bar{L}_f + \left(1+\sqrt{d_s}\right) \beta_n \bar{L}_\sigma\right)^{T-1} T \int_{\mathcal{S}\times\mathcal{A}} T \rho^{\pi_e}(\boldsymbol{s}, \boldsymbol{a}) \|\boldsymbol{\sigma}_n(\boldsymbol{s}, \boldsymbol{a})\|_2 d\boldsymbol{s} d\boldsymbol{a} && \text{(See Eq. 2)} \\ &= \bar{L}_r \left(1+\sqrt{d_s}\right) \beta_n \left(1 + \bar{L}_f + \left(1+\sqrt{d_s}\right) \beta_n \bar{L}_\sigma\right)^{T-1} T^2 \mathop{\mathbb{E}}_{\boldsymbol{s}, \boldsymbol{a} \sim \rho^{\pi_e}} \left[ \|\boldsymbol{\sigma}_n(\boldsymbol{s}, \boldsymbol{a})\|_2 \right]\end{aligned}$$

□

In summary, the deviation between the true and pessimistic return is proportional to the expected epistemic uncertainty of the evaluation policy state-occupancy measure $\rho^{\pi_e}$, and the constant $C_n$ defined as

$$C_n := \bar{L}_r \left(1+\sqrt{d_s}\right) \beta_n T^2 \left(1 + \bar{L}_f + \left(1+\sqrt{d_s}\right) \beta_n \bar{L}_\sigma\right)^{T-1}.$$

In Appendix B.3, we provide consistency guarantees for our method. In particular, we prove under further assumptions on the true dynamics function $f$, that $\left|J(\pi_e) - \tilde{J}(\pi_e)\right| \to 0$, for $n \to \infty$.

### B.2 PROOF OF KNOWN RESULTS FOR KERNEL METHODS

We first recall the notion of maximum mutual information [Srinivas et al., 2012, Cover and Thomas, 2006]. The mutual information $I(\boldsymbol{x}_{1:n}; k)$ quantifies the reduction in uncertainty due to the observations $\boldsymbol{x}_{1:n}$. Given a GP model $\mathcal{GP}(0, k(\cdot, \cdot))$ and gaussian noise assumption, mutual information is equal to

$$I(\boldsymbol{x}_{1:n}) = \frac{1}{2} \log\det(\boldsymbol{I} + \sigma_\epsilon^{-2} \boldsymbol{K})$$

with the kernel matrix $\boldsymbol{K} = [k(\boldsymbol{x}_i, \boldsymbol{x}_j)]_{i,j \le n}$. The maximum information capacity or maximum mutual information of a kernel $k$ is an upper bound on the mutual information, and is defined as

$$\gamma_n = \max_{\boldsymbol{x}_{1:n}} I(\boldsymbol{x}_{1:n}).$$

Table 1 shows the growth rate of $\gamma_n$ with $n$ for multiple different kernels.

*Proof of Lemma 3.6.* Let $\gamma_n$ be the maximum mutual information of $\mathcal{GP}(0, k(\cdot,\cdot))$. Set $\beta_n(\delta) := \left(B + \sigma_\epsilon\sqrt{2(\gamma_n + 1 + \ln(d_s/\delta)}\right)$. Element-wise application of Theorem 2 in Chowdhury and Gopalan [2017] over the dimensions of $\mathcal{S}$ and taking a union bound proves the lemma. □

*Proof of Lemma 3.7.* First, we prove the Lipschitz continuity of $\boldsymbol{f}$. By the Cauchy-Schwartz inequality, we have $\forall\, \boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}$

$$|f_j(\boldsymbol{x}) - f_j(\boldsymbol{x}')| = |\langle f_j, k(\boldsymbol{x},\cdot) - k(\boldsymbol{x}',\cdot)\rangle_k| \leq \|f_j\|_k\, d_k(\boldsymbol{x},\boldsymbol{x}') \tag{24}$$

Since $\|f_j\|_k \leq B, \forall j = 1, ..., d_s$ and $d_k(\boldsymbol{x},\boldsymbol{x}')$ is $L_k$-Lipschitz, we have that

$$\|\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{f}(\boldsymbol{x}')\|_2 = \sqrt{\sum_{j=1}^{d_s}(f_j(\boldsymbol{x}) - f_j(\boldsymbol{x}'))^2} \leq \sqrt{d_s B^2 d_k(\boldsymbol{x},\boldsymbol{x}')^2} = \sqrt{d_s} B d_k(\boldsymbol{x},\boldsymbol{x}') \leq \sqrt{d_s} B L_k \|\boldsymbol{x} - \boldsymbol{x}'\|_2;. \tag{25}$$

Next, we show the Lipschitz continuity of the GP standard deviation. By Lemma 12 in Curi et al. [2020], we have, independent of $n$, $|\boldsymbol{\sigma}_n(\boldsymbol{x}) - \boldsymbol{\sigma}_n(\boldsymbol{x})| \leq d_k(\boldsymbol{x},\boldsymbol{x}')$ for the GP standard deviation. Now, we make a similar argument as above:

$$\|\boldsymbol{\sigma}_n(\boldsymbol{x}) - \boldsymbol{\sigma}_n(\boldsymbol{x})\|_2 \leq \sqrt{d_s d_k^2(\boldsymbol{x},\boldsymbol{x}')} \leq \sqrt{d_s} L_k \|\boldsymbol{x} - \boldsymbol{x}'\|_2 \tag{26}$$

which shows that $\boldsymbol{\sigma}_n(\cdot)$ is $\sqrt{d_s}L_k$-Lipschitz. □

### B.3 PROOF OF THEOREM 3.8

For showing consistency of our lower bound in Theorem 3.5 for the GP case, we first prove that the uncertainty with respect to an i.i.d., data sampling distribution $p(x)$ shrinks in expectation.

**Lemma B.6** (Shrinking uncertainty in expectation)**.** *Let $p(x)$ denote a data sampling distribution with a compact support. Then the following holds for sequences $\{x_i\}_{i=0}^{n-1}$ sampled i.i.d. from $p(x)$,*

$$C_n^2 \mathop{\mathbb{E}}_{\boldsymbol{x}_{1:n}\sim p}\left[\sigma^2(x_n|\{x_i\}_{i=0}^{n-1})\right] \leq C_n^2 \mathop{\mathbb{E}}_{x_{1:n-1}\sim p}\left[\sigma^2(x_{n-1}|\{x_i\}_{i=0}^{n-2})\right]. \tag{27}$$

*Proof.*

$$\begin{aligned}
C_n^2 \mathop{\mathbb{E}}_{\boldsymbol{x}_{1:n}\sim p}\left[\sigma^2(x_n|\{x_i\}_{i=0}^{n-1})\right] &= C_n^2 \mathop{\mathbb{E}}_{x_{1:n-1}\sim p}\left[\mathop{\mathbb{E}}_{x_n\sim p}\left[\sigma^2(x_n|\{x_i\}_{i=0}^{n-1})\right]\right] \\
&= C_n^2 \mathop{\mathbb{E}}_{x_{1:n-1}\sim p}\left[\mathop{\mathbb{E}}_{x\sim p}\left[\sigma^2(x|\{x_i\}_{i=0}^{n-1})\right]\right] \\
&\leq C_n^2 \mathop{\mathbb{E}}_{x_{1:n-1}\sim p}\left[\mathop{\mathbb{E}}_{x\sim p}\left[\sigma^2(x|\{x_i\}_{i=0}^{n-2})\right]\right] && \text{(Monotonicity of variance)} \\
&= C_n^2 \mathop{\mathbb{E}}_{x_{1:n-2}\sim p}\left[\mathop{\mathbb{E}}_{x_{n-1}\sim p}\left[\mathop{\mathbb{E}}_{x\sim p}\left[\sigma^2(x|\{x_i\}_{i=0}^{n-2})|x_{n-1}\right]\right]\right] \\
&= C_n^2 \mathop{\mathbb{E}}_{x_{1:n-2}\sim p}\left[\mathop{\mathbb{E}}_{x\sim p}\left[\sigma^2(x|\{x_i\}_{i=0}^{n-2})\right]\right] && \text{(All points are sampled i.i.d from } p\text{)} \\
&= C_n^2 \mathop{\mathbb{E}}_{x_{1:n-2}\sim p}\left[\mathop{\mathbb{E}}_{x_{n-1}\sim p}\left[\sigma^2(x_{n-1}|\{x_i\}_{i=0}^{n-2})\right]\right] \\
&= C_n^2 \mathop{\mathbb{E}}_{x_{1:n-1}\sim p}\left[\sigma^2(x_{n-1}|\{x_i\}_{i=0}^{n-2})\right].
\end{aligned}$$

□

**Lemma B.7** (Bound on expectation of uncertainty at $n$)**.** *Let $p(\boldsymbol{x})$ denote the data sampling distribution with a compact support. Then the following holds for sequences $\{\boldsymbol{x}_i\}_{i=0}^{n-1}$ sampled i.i.d. from $p(\boldsymbol{x})$,*

$$nC_n^2 \mathop{\mathbb{E}}_{\boldsymbol{x}_{1:n}\sim p}\left[\sigma^2(\boldsymbol{x}_n|\{\boldsymbol{x}_i\}_{i=0}^{n-1})\right] \leq C_n^2 \mathop{\mathbb{E}}_{\boldsymbol{x}_{1:n}\sim p}\left[\sum_{j=1}^{n}\sigma^2(\boldsymbol{x}_j|\{\boldsymbol{x}_i\}_{i=0}^{j-1})\right]. \tag{28}$$

*Moreover, we have*

$$C_n^2 \mathop{\mathbb{E}}_{\boldsymbol{x}_{1:n}\sim p}\left[\sigma^2(\boldsymbol{x}_n|\{\boldsymbol{x}_i\}_{i=0}^{n-1})\right] \leq \frac{C_n^2\gamma_n}{n},$$

*where $\gamma_n$ represents the maximum information gain (Srinivas et al. [2012], Cover and Thomas [2006]).*

*Proof.* We prove by induction. For $n = 1$, Eq. 28 holds trivially. Now assume $n > 1$,

$$\begin{aligned}
nC_n^2 \mathop{\mathbb{E}}_{\boldsymbol{x}_{1:n}\sim p}\left[\sigma^2(\boldsymbol{x}_n|\{\boldsymbol{x}_i\}_{i=0}^{n-1})\right] &= C_n^2 \mathop{\mathbb{E}}_{\boldsymbol{x}_{1:n}\sim p}\left[\sigma^2(\boldsymbol{x}_n|\{\boldsymbol{x}_i\}_{i=0}^{n-1})\right] + (n-1)C_n^2 \mathop{\mathbb{E}}_{\boldsymbol{x}_{1:n}\sim p}\left[\sigma^2(\boldsymbol{x}_n|\{\boldsymbol{x}_i\}_{i=0}^{n-1})\right] \\
&\leq C_n^2 \mathop{\mathbb{E}}_{\boldsymbol{x}_{1:n}\sim p}\left[\sigma^2(\boldsymbol{x}_n|\{\boldsymbol{x}_i\}_{i=0}^{n-1})\right] + (n-1)C_n^2 \mathop{\mathbb{E}}_{\boldsymbol{x}_{1:n-1}\sim p}\left[\sigma^2(\boldsymbol{x}_{n-1}|\{\boldsymbol{x}_i\}_{i=0}^{n-2})\right] \\
&\qquad \text{(Lemma B.6)} \\
&\leq C_n^2 \mathop{\mathbb{E}}_{\boldsymbol{x}_{1:n}\sim p}\left[\sigma^2(\boldsymbol{x}_n|\{\boldsymbol{x}_i\}_{i=0}^{n-1})\right] + C_n^2 \mathop{\mathbb{E}}_{\boldsymbol{x}_{1:n-1}\sim p}\left[\sum_{j=1}^{n-1}\sigma^2(\boldsymbol{x}_j|\{\boldsymbol{x}_i\}_{i=0}^{j-1})\right] \\
&\qquad \text{(By induction hypothesis)} \\
&= C_n^2 \mathop{\mathbb{E}}_{\boldsymbol{x}_{1:n}\sim p}\left[\sum_{j=1}^{n}\sigma^2(\boldsymbol{x}_j|\{\boldsymbol{x}_i\}_{i=0}^{j-1})\right].
\end{aligned}$$

Note, $\sum_{j=1}^{n}\sigma^2(\boldsymbol{x}_{j+1}|\{\boldsymbol{x}_i\}_{i=0}^{j})$ is a measure of the mutual information associated to the sampling scheme, and lower bounds the mutual information. The mutual information $I(\boldsymbol{x}_{1:n})$ quantifies the reduction in uncertainty due to the observations $\boldsymbol{x}_{1:n}$ Cover and Thomas [2006]. When $f \in \mathcal{H}_k$, mutual information is equal to

$$I(\boldsymbol{x}_{1:n}) = \frac{1}{2}\log\det(\boldsymbol{I} + \lambda^{-1}\boldsymbol{K})$$

with the kernel matrix $\boldsymbol{K} = [k(\boldsymbol{x}_i, \boldsymbol{x}_j)]_{i,j\leq n}$. Moreover,

$$\mathop{\mathbb{E}}_{\boldsymbol{x}_{1:n}\sim p}\left[\sum_{j=1}^{n}\sigma^2(x_{j+1}|\{x_i\}_{i=0}^{j})\right] \leq I(\boldsymbol{x}_{1:n}).$$

The maximum information gain, is an upper bound on the mutual information, and is defined as

$$\gamma_n = \max_{\boldsymbol{x}_{1:n}} I(\boldsymbol{x}_{1:n}).$$

Therefore, by definition of $\gamma_n$, it is greater than the mutual information of all sampling schemes within the the support of $p(x)$.

$$C_n^2 \mathop{\mathbb{E}}_{\boldsymbol{x}_{1:n}\sim p}\left[\sigma^2(x_n|\{x_i\}_{i=0}^{n-1})\right] \leq \frac{C_n^2\gamma_n}{n}$$

Srinivas et al. [2012] derive the bounds on $\gamma_n$ (see Table 1) for linear, RBF, and Matèrn kernels on compact and convex sets. Hence, we obtain that for the linear and RBF kernel, $C_n^2\gamma_n$ grows sublinearly in $n$, i.e., $C_n^2\gamma_n/n \to 0$ for $n \to \infty$. □

**Lemma B.8.** *Let $p(\boldsymbol{x})$ denote a distribution with a compact support, and assume that $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{n-1}$ are i.i.d. samples from $p$. Then, the following holds with probability at least $1-\delta$,*

$$\mathbb{P}\left(\mathop{\mathbb{E}}_{\boldsymbol{x}\sim p}[C_n\sigma_n(\boldsymbol{x})] = \mathcal{O}\left((1 + 1/\sqrt{\delta})\sqrt{\frac{\gamma_n^{T+1}}{n}}\right), \forall \boldsymbol{x}_{1:n-1}\right) \geq 1-\delta, \qquad \forall n \in \mathbb{N}.$$

| Kernel | Bounds on $\gamma_n$ for $x \in \mathbb{R}^d$ |
|---|---|
| Linear | $\mathcal{O}(d \log n)$ |
| RBF | $\mathcal{O}((\log n)^{d+1})$ |
| Matèrn $\nu > 1/2$ | $\mathcal{O}(n^{\frac{d}{2\nu+d}} \log^{\frac{2\nu}{2\nu+d}}(n))$ |

Table 1: Bounds on $\gamma_n$ from [Vakili et al., 2021, Theorem 5.]

*Moreover, for the linear and RBF kernel we have*

$$\mathbb{P}\left(\mathop{\mathbb{E}}_{x\sim p}\left[C_n\sigma_n(x)\right] \to 0 \textit{ for } n \to \infty\right) = 1.$$

*Proof.* From Lemma B.7, we have

$$\mathop{\mathbb{E}}_{x_{1:n-1}\sim p}\left[C_n^2 \mathop{\mathbb{E}}_{x\sim p}\left[\sigma_n^2(x)\right]\right] \leq \frac{C_n^2\gamma_n}{n}. \tag{29}$$

Using the Markov inequality, we get

$$\mathbb{P}(X \geq a) \leq \frac{\mathbb{E}\left[X\right]}{a}.$$

Let $X$ denote $C_n^2 \mathop{\mathbb{E}}_{x\sim p}\left[\sigma_n^2(x)\right]$. Then we have,

$$\mathbb{P}(C_n^2 \mathop{\mathbb{E}}_{x\sim p}\left[\sigma_n^2(x)\right] \geq a) \leq \frac{\mathop{\mathbb{E}}_{x_{1:n-1}\sim p}\left[C_n^2 \mathop{\mathbb{E}}_{x\sim p}\left[\sigma_n^2(x)\right]\right]}{a} \leq \frac{C_n^2\gamma_n}{na}.$$

Therefore, for $n \to \infty$, $C_n^2 \mathop{\mathbb{E}}_{x\sim p}\left[\sigma_n^2(x)\right] \to 0$ almost surely if $\frac{C_n^2\gamma_n}{n} \to 0$ for $n \to \infty$. Now by definition of $C_n$ (Theorem 3.5) and plugging in the choice of $\beta_n$ (Lemma 3.6), we have $\frac{C_n^2\gamma_n}{n} \propto \frac{\gamma_n^{T+1}}{n}$. For the linear and RBF kernel, we have (see Table 1)

$$\frac{\gamma_n^{T+1}}{n} = \mathcal{O}\left(d^{T+1}\frac{(\log n)^{T+1}}{n}\right) \tag{Linear kernel}$$

$$\frac{\gamma_n^{T+1}}{n} = \mathcal{O}\left(\frac{(\log n)^{(d+1)(T+1)}}{n}\right). \tag{RBF kernel}$$

In both cases, $\frac{C_n^2\gamma_n}{n} \to 0$ for $n \to \infty$.

Now to recover the rate of convergence, let $v = \mathop{\mathbb{E}}_{x\sim p}\left[C_n\sigma_n(x)\right]$. We study its variance and expectation with respect to $x_{1:n-1} \sim p$ for a fixed $n$. We have

$$\text{Var}[v] \leq \mathbb{E}\left[v^2\right] \leq \mathop{\mathbb{E}}_{x_{1:n-1}\sim p}\left[C_n^2 \mathop{\mathbb{E}}_{x\sim p}\left[\sigma_n^2(x)\right]\right] \leq \frac{C_n^2\gamma_n}{n}.$$

Additionally, $\mathbb{E}\left[v\right] \leq \sqrt{\mathbb{E}\left[v^2\right]} \leq C_n\sqrt{\frac{\gamma_n}{n}}$. Now, we apply the Chebyshev inequality, i.e.,

$$\mathbb{P}(|v - \mathop{\mathbb{E}}_{v}\left[v\right]| \geq a) \leq \frac{\text{Var}[v]}{a^2} \leq \frac{\mathop{\mathbb{E}}_{v}\left[v^2\right]}{a^2}.$$

Therefore, for $a^2 = \frac{\mathop{\mathbb{E}}_{v}\left[v^2\right]}{\delta}$, we have with probability at least $1-\delta$,

$$\begin{aligned} v &\leq \mathop{\mathbb{E}}_{v}\left[v\right] + a \\ &= \mathop{\mathbb{E}}_{v}\left[v\right] + \sqrt{\frac{\mathop{\mathbb{E}}_{v}\left[v^2\right]}{\delta}} \\ &\leq \left(1 + \frac{1}{\sqrt{\delta}}\right)\sqrt{\mathop{\mathbb{E}}_{v}\left[v^2\right]}. \end{aligned}$$

Next, we plug in the definition of $v$, to get

$$\begin{aligned}\underset{x\sim p}{\mathbb{E}}\left[C_n\sigma_n(x)\right] &\leq \left(1+\frac{1}{\sqrt{\delta}}\right)\sqrt{\underset{x_{1:n-1}\sim p}{\mathbb{E}}\left[\underset{x\sim p}{\mathbb{E}}\left[C_n^2\sigma_n^2(x)\right]\right]} \\ &\leq \left(1+\frac{1}{\sqrt{\delta}}\right)C_n\sqrt{\frac{\gamma_n}{n}} = \mathcal{O}\left(\left(1+1/\sqrt{\delta}\right)\sqrt{\frac{\gamma_n^{T+1}}{n}}\right).\end{aligned}$$

with probability at least $1-\delta$. □

*Proof of Theorem 3.8 (Consistency of Hambo).* For the GP case we prove that the well calibration assumption, and the Lipschitz continuity of $f$ and $\sigma$ are satisfied (see Lemmas 3.6 and 3.7). This allows us to apply Theorem 3.5 and Proposition 3.2, which gives with probability at least $1-\delta$ that,

$$J(\pi_e) \geq \tilde{J}(\pi_e) \geq J(\pi_e) - C_n \underset{\boldsymbol{s},\boldsymbol{a}\sim\rho^{\pi_e}}{\mathbb{E}}\left[\|\boldsymbol{\sigma}_n(\boldsymbol{s},\boldsymbol{a})\|_2\right]. \tag{30}$$

To prove consistency, we then only need to show that $C_n \underset{\boldsymbol{s},\boldsymbol{a}\sim\rho^{\pi_e}}{\mathbb{E}}\left[\|\boldsymbol{\sigma}_n(\boldsymbol{s},\boldsymbol{a})\|_2\right]$ goes to 0 for $n\to\infty$. Since the support of the behavioural policy's state-occupancy measure $\rho^{\pi_b}$ is compact, and supp$(\rho^{\pi_e})\subseteq$ supp$(\rho^{\pi_b})$, we have $\rho^{\pi_b}(s,a)\geq\hat{C}\rho^{\pi_e}(s,a)$ for all $(s,a)\in\mathcal{S}\times\mathcal{A}$, and some $\hat{C}>0$, i.e., the importance sampling ratio is bounded. We can then write,

$$\begin{aligned}C_n \underset{\boldsymbol{s},\boldsymbol{a}\sim\rho^{\pi_e}}{\mathbb{E}}\left[\|\boldsymbol{\sigma}_n(\boldsymbol{s},\boldsymbol{a})\|_2\right] &\leq \sum_{i=1}^{d_s}\underset{\boldsymbol{s},\boldsymbol{a}\sim\rho^{\pi_e}}{\mathbb{E}}\left[C_n\sigma_n^i(\boldsymbol{s},\boldsymbol{a})\right] \\ &= \sum_{i=1}^{d_s}\underset{\boldsymbol{s},\boldsymbol{a}\sim\rho^{\pi_b}}{\mathbb{E}}\left[C_n\sigma_n^i(\boldsymbol{s},\boldsymbol{a})\frac{\rho^{\pi_e}(s,a)}{\rho^{\pi_b}(s,a)}\right] \\ &\leq \frac{1}{\hat{C}}\sum_{i=1}^{d_s}\underset{\boldsymbol{s},\boldsymbol{a}\sim\rho^{\pi_b}}{\mathbb{E}}\left[C_n\sigma_n^i(\boldsymbol{s},\boldsymbol{a})\right].\end{aligned}$$

Following Lemma B.8, with $\boldsymbol{x}=(\boldsymbol{s},\boldsymbol{a})$ and $p=\rho^e$, we have

$$\mathbb{P}\left(\sum_{i=1}^{d_s}\underset{\boldsymbol{s},\boldsymbol{a}\sim\rho^{\pi_e}}{\mathbb{E}}\left[C_n\sigma_n^i(\boldsymbol{s},\boldsymbol{a})\right]\to 0 \text{ for } n\to\infty\right)=1.$$

Moreover, by taking a union bound over the dimensions $1,\cdots,d_s$, Lemma B.8 implies that with probability greater than $1-\delta$, for any set of i.i.d. trajectories,

$$\sum_{i=1}^{d_s}\underset{\boldsymbol{s},\boldsymbol{a}\sim\rho^{\pi_e}}{\mathbb{E}}\left[C_n\sigma_n^i(\boldsymbol{s},\boldsymbol{a})\right] = \mathcal{O}\left(d_s\left(1+\sqrt{d_s/\delta}\right)\sqrt{\frac{\gamma_n^{T+1}}{n}}\right).$$

□

## C HAMBO FOR OFFLINE REINFORCEMENT LEARNING

OPE methods are commonly used in offline reinforcement learning (ORL) Levine et al. [2020] to recommend/learn an optimal policy. Moreover, ORL methods also suffer from distribution shifts and are susceptible to overestimation, i.e., overestimating the performance of the recommended policy. Therefore, in principle, a good COPE method can be applied for ORL applications. To this end, we propose a natural modification of HAMBO-CA for ORL.

$$\tilde{J}(\pi^*) := \max_{\pi}\min_{\boldsymbol{\eta}} J_{\tilde{p}_{\boldsymbol{\eta}}}(\pi). \tag{31}$$

Our proposed method induces pessimism with respect to the epistemic uncertainty of the learned transition model to tackle distribution shifts. Similar, to HAMBO-CA, we can also use the HAMBO-DS1 variant to induce pessimsm.

|  | Ours |  | Model-based |  |  |  | Model-free |  |
|---|---|---|---|---|---|---|---|---|
|  | HAMBO-CA | HAMBO-DA1 | Rigter et al. [2022] | Yu et al. [2021] | Yu et al. [2020] | Kidambi et al. [2020] | Kumar et al. [2020] | Kostrikov et al. [2022] |
| HalfCheetah-random | 37.1 | 35.1 | 39.5 | 38.8 | 35.4 | 25.6 | 19.6 | - |
| HalfCheetah-medium | 66.9 | 67.9 | 77.9 | 54.2 | 69.5 | 42.1 | 49.0 | 47.4 |

Table 2: Comparisons on the HalfCheetah from the D4RL benchmark suite. Results of the other algorithms are taken from Rigter et al. [2022].

We compare our HAMBO-based ORL variants to other ORL algorithms on the OpenAI Gym tasks from the D4RL benchmark Fu et al. [2020]. Specifically, we consider the HalfCheetah environment with data sets generated with a random and a medicore-performing policy. Our results are presented in table 2.

The max-min optimization in eq (31) is typically very challenging. For our experiments we use the soft actor critic algorithm to train the policy and adversary together (DQN algorithm is used for the HAMBO-DA1 variant).

Note, our proposed ORL algorithms recommend the policy with the best lower bound and not the best expected return (see eq 31). Therefore, in general, they may fail to recommend the optimal policy. This is the price we pay for inducing robustness in our ORL methods. However, in practice (see table 2) we observer that the HAMBO based ORL methods perform competitively to the start of the art in the field.