Hallucinated Adversarial Control for Conservative Offline Policy Evaluation - Supplementary Material

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

[1]ETH Zurich, Switzerland

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 π_e, reward function $r(\cdot, \cdot)$, Horizon T, initial state distribution $p_0(\boldsymbol{s}_0)$
$(\boldsymbol{\mu}_n, \boldsymbol{\sigma}_n, \beta_n) \leftarrow \text{TrainModel}(\mathcal{D}_b)$ ▷ 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:

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

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

[*]Equal contribution.

Accepted for the 39th Conference on Uncertainty in Artificial Intelligence (UAI 2023).

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., $p(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}; \mathbf{0}, \boldsymbol{I})$. However, as commonly done for BNNs to alleviate the problems of prior misspecification, we add a temperature parameter τ 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:

$$\boldsymbol{\theta}_k \leftarrow \boldsymbol{\theta}_k + \frac{1}{K} \sum_{k'=1}^{K} \left[k(\boldsymbol{\theta}_{k'}, \boldsymbol{\theta}_k) \nabla_{\boldsymbol{\theta}_{k'}} \log p(\boldsymbol{\theta}_{k'}|\mathcal{D}_b) + \nabla_{\boldsymbol{\theta}_{k'}} k(\boldsymbol{\theta}'_k, \boldsymbol{\theta}_k) \right] \quad \forall k = 1, ..., K \,. \tag{2}$$

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

$$\text{CalErr}(\boldsymbol{\tau}) \coloneqq \frac{1}{d_s} \sum_{j=1}^{d_s} \frac{1}{|A|} \sum_{\alpha \in A} \left(\text{EmpFreq}(\alpha, \boldsymbol{\tau})_j - \alpha\right)^2 \,, \tag{3}$$

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

$$\text{EmpFreq}(\alpha; \boldsymbol{\tau}) \coloneqq \frac{1}{|\mathcal{D}_c|} \sum_{(\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{s}') \in \mathcal{D}_c} \mathbf{1}\{\boldsymbol{s}' \leq \Phi^{-1}_{\boldsymbol{\tau}}(\alpha; \boldsymbol{s}, \boldsymbol{a})\} \tag{4}$$

is a vector-valued function of the (per dimension) empirical frequencies of the prediction targets that fall below the α 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

$$\boldsymbol{\tau}^* = \arg\min_{\boldsymbol{\tau}} \text{CalErr}(\boldsymbol{\tau}) \,. \tag{5}$$

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 $\text{EmpFreq}(\alpha; \boldsymbol{\tau}) \leftarrow \frac{1}{|\mathcal{D}_c|} \sum_{(\boldsymbol{s},\boldsymbol{a},\boldsymbol{s}')\in\mathcal{D}_c} \mathbf{1}\{\boldsymbol{s}' \leq \Phi^{-1}_{\boldsymbol{\tau}}(\alpha; \boldsymbol{s}, \boldsymbol{a}, \boldsymbol{\tau})\}$
Define $\text{CalErr}(\boldsymbol{\tau}) \leftarrow \frac{1}{d_s}\sum_{j=1}^{d_s} \frac{1}{|A|} \sum_{\alpha\in A} \left(\text{EmpFreq}(\alpha, \boldsymbol{\tau})_j - \alpha\right)^2$
return $\arg\min_{\boldsymbol{\tau}} \text{CalErr}(\boldsymbol{\tau})$ ▷ 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.,

$$\tilde{p}_{\boldsymbol{\eta}}(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{a}) \coloneqq \mathcal{N}\big(\boldsymbol{s}'; \boldsymbol{\mu}_\Theta(\boldsymbol{s},\boldsymbol{a}) + \tau^2\boldsymbol{\eta}(\boldsymbol{s},\boldsymbol{a})\boldsymbol{\sigma}^2_{\Theta,e}, \boldsymbol{\sigma}^2_{\Theta,a}(\boldsymbol{s},\boldsymbol{a})\big) \tag{6}$$

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., 2018] 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 π_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 π_e under the adversary transition model, induced by $\boldsymbol{\eta}^\star$ with $L = 10^4$ trajectories (see Eq. 1). The HAMBO-CA method is summarized in Algorithm 4.

Algorithm 4 HAMBO-CA

Require: Offline dataset $\mathcal{D}_b$, evaluation policy π_e, Number of BNN particles K
Select a subset of $\mathcal{D}_b$ as calibration set $\mathcal{D}_c$
$\{\boldsymbol{h}_{\boldsymbol{\theta}_1}, \boldsymbol{\nu}^2_{\boldsymbol{\theta}_1}, \cdots, \boldsymbol{h}_{\boldsymbol{\theta}_K}, \boldsymbol{\nu}^2_{\boldsymbol{\theta}_K}\} \leftarrow \text{SVGD}(\mathcal{D}_b \setminus \mathcal{D}_c, K)$ ▷ 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.
$\tau \leftarrow \text{CalibrateBNN}(\mathcal{D}_c, \boldsymbol{\mu}_\Theta, \boldsymbol{\sigma}^2_{\Theta,e} + \boldsymbol{\sigma}^2_{\Theta,a})$ ▷ 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
$\boldsymbol{\eta}^\star \leftarrow \text{SoftActorCritic}(-J_{\tilde{p}_{\boldsymbol{\eta}}}(\pi_e), \boldsymbol{\eta})$ ▷ 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 π_e via sampling (see Eq. 1)
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 ϑ 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 π_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 π_e under the adversary transition model, induced by $\vartheta^\star$ with $L = 10^4$ trajectories (see Eq. 1). The overall HAMBO-DA1 method is summarized in Algorithm 5.

Algorithm 5 HAMBO-DA1

Require: Offline dataset $\mathcal{D}_b$, evaluation policy π_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 ϑ
$\tilde{p}_\vartheta(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{a}) := \sum_{k=1}^K \vartheta(k|\boldsymbol{s},\boldsymbol{a})\mathcal{N}\big(\boldsymbol{s}'; \boldsymbol{h}_{\boldsymbol{\theta}_k}(\boldsymbol{s},\boldsymbol{a}), \boldsymbol{\nu}^2_{\boldsymbol{\theta}_k}(\boldsymbol{s},\boldsymbol{a})\big)$ ▷ Setup hallucinated adversarial transition model
$\vartheta^\star \leftarrow \mathrm{DoubleDQN}(-J_{\tilde{p}_{\boldsymbol{\eta}}}(\pi_e), \vartheta)$ ▷ Train adversary ϑ via to maximize the negative return
$\tilde{J}(\pi_e) \leftarrow \hat{J}_{\tilde{p}_{\vartheta^\star}}(\pi_e)$ ▷ Estimate expected return of π_e via sampling (see Eq. 1)
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. $J(\pi_e) = \min_{k\in\{1,\dots,K\}} J_{p_{\boldsymbol{\theta}_k}}(\pi_e)$. The HAMBO-DAINF method is summarized in Algorithm 5.

Algorithm 6 HAMBO-DAINF

Require: Offline dataset $\mathcal{D}_b$, evaluation policy π_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 π_e for each model (see Eq. 1) 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

$$|\boldsymbol{\mu}_n(\boldsymbol{s},\boldsymbol{a}) - f(\boldsymbol{s},\boldsymbol{a})| \leq \beta_n(\delta)\boldsymbol{\sigma}_n(\boldsymbol{s},\boldsymbol{a})\,. \tag{7}$$

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

$$f(\boldsymbol{s},\boldsymbol{a}) = \boldsymbol{\mu}_n(\boldsymbol{s},\boldsymbol{a}) + \beta_n(\delta)\boldsymbol{\eta}^\dagger(\boldsymbol{s},\boldsymbol{a})\boldsymbol{\sigma}_n(\boldsymbol{s},\boldsymbol{a})\,, \tag{8}$$

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

$$\tilde{p}_{\boldsymbol{\eta}^\dagger}(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t,\boldsymbol{a}_t) = p_{\boldsymbol{\epsilon}}\big(\boldsymbol{s}_{t+1} - \boldsymbol{\mu}_n(\boldsymbol{s},\boldsymbol{a}) + \beta\boldsymbol{\eta}(\boldsymbol{s},\boldsymbol{a})^\dagger\boldsymbol{\sigma}_n(\boldsymbol{s},\boldsymbol{a})\big) = p_{\boldsymbol{\epsilon}}\big(\boldsymbol{s}_{t+1} - f(\boldsymbol{s},\boldsymbol{a})\big) = p(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t,\boldsymbol{a}_t)\,. \tag{9}$$

Finally, we can use this to show

$$\tilde{J}(\pi_e) := \min_{\boldsymbol{\eta}} J_{\tilde{p}_{\boldsymbol{\eta}}}(\pi_e) \leq J_{\tilde{p}_{\boldsymbol{\eta}^\dagger}}(\pi_e) = J_p(\pi_e) = J(\pi_e)\,, \tag{10}$$

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,

$$\begin{aligned}
\mathcal{W}_1(\pi(\boldsymbol{a}|\boldsymbol{s}), \pi(\boldsymbol{a}|\boldsymbol{s}')) &= \inf_{\gamma\in\Gamma(\pi(\boldsymbol{a}|\boldsymbol{s}),\pi(\boldsymbol{a}|\boldsymbol{s}'))} \mathbb{E}_{\boldsymbol{a},\boldsymbol{a}'\sim\gamma}\left[\|\boldsymbol{a}-\boldsymbol{a}'\|_2\right] && (11)\\
&\leq \mathbb{E}_{\boldsymbol{a},\boldsymbol{a}'\sim\tilde{\gamma}}\left[\|\boldsymbol{a}-\boldsymbol{a}'\|_2\right] = \mathbb{E}_{\boldsymbol{\zeta}}\left[\|g(\boldsymbol{s},\boldsymbol{\zeta}) - g(\boldsymbol{s}',\boldsymbol{\zeta})\|_2\right] && (12)\\
&\leq L_g\|\boldsymbol{s}-\boldsymbol{s}'\|_2 && (13)
\end{aligned}$$

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*

$$\mathbb{E}_X \mathbb{E}_{\boldsymbol{\omega}|X} \left[\|\boldsymbol{\omega}\|_2 \right] = \mathcal{W}_1(p, q) \ , \tag{14}$$

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.,

$$\mathcal{W}_1 = \inf_{\gamma \in \Gamma(p,q)} \mathbb{E}_{X',Y' \sim \gamma} \left[\|X' - Y'\|_2 \right] \tag{15}$$

If the expectation of p and q is finite, then the infimum over couplings in (15) 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 γ^*, 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

$$p(z) = \int_{\mathcal{X}} p(x, z - x) dx = \int_{\mathcal{X}} p(x) \tilde{\gamma}^*(\underbrace{z - x}_{\boldsymbol{\omega}} | x) dx \tag{16}$$

$$= \int_{\mathcal{X}} \gamma^*(x, x + (z - x)) dx = \int_{\mathcal{X}} \gamma^*(x, z) dx = q(z) \tag{17}$$

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

$$\mathcal{W}_1(p, q) = \inf_{\gamma \in \Gamma(p,q)} \mathbb{E}_{x',y' \sim \gamma} \left[\|x' - y'\|_2 \right] = \mathbb{E}_{x',y' \sim \gamma^*} \left[\|x' - y'\|_2 \right] = \mathbb{E}_{x' \sim p(x')} \mathbb{E}_{\boldsymbol{\omega} \sim \tilde{\gamma}^*(\boldsymbol{\omega}|x')} \left[\|\boldsymbol{\omega}\|_2 \right] \tag{18}$$

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_π-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_π-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.

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

□

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

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

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,

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

□

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*

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

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

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

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

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

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{s}_t,\boldsymbol{a}_t+\boldsymbol{\omega}_t)\|_2\right]\right] \\
&\qquad\qquad \text{(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{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] \\
&\qquad\qquad \text{(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] \\
&\qquad\qquad \text{(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

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

□

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

Proof of Theorem 3.5.

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

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

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

which allows us to write,

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

□

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

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

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 $\boldsymbol{K} = [k(\boldsymbol{x}_i, \boldsymbol{x}_j)]_{i,j\leq n}$. 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 γ_n with n for multiple different kernels.

Proof of Lemma 3.6. Let γ_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 $\forall\, \boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}$

$$|f_j(\boldsymbol{x}) - f_j(\boldsymbol{x}')| = |\langle f_j, k(\boldsymbol{x},\cdot) - k(\boldsymbol{x}',\cdot)\rangle_k| \leq \|f_j\|_k\, d_k(\boldsymbol{x}, \boldsymbol{x}') \tag{19}$$

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

$$\|\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{f}(\boldsymbol{x}')\|_2 = \sqrt{\sum_{j=1}^{d_s}(f_j(\boldsymbol{x}) - f_j(\boldsymbol{x}'))^2} \leq \sqrt{d_s B^2 d_k(\boldsymbol{x}, \boldsymbol{x}')^2} = \sqrt{d_s} B d_k(\boldsymbol{x}, \boldsymbol{x}') \leq \sqrt{d_s} B L_k \|\boldsymbol{x} - \boldsymbol{x}'\|_2;. \tag{20}$$

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:

$$\|\boldsymbol{\sigma}_n(\boldsymbol{x}) - \boldsymbol{\sigma}_n(\boldsymbol{x})\|_2 \leq \sqrt{d_s d_k^2(\boldsymbol{x}, \boldsymbol{x}')} \leq \sqrt{d_s} L_k \|\boldsymbol{x} - \boldsymbol{x}'\|_2 \tag{21}$$

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 compact support. Then the following holds for sequences $\{x_i\}_{i=0}^{n-1}$ sampled i.i.d. from $p(x)$,*

$$C_n^2 \mathop{\mathbb{E}}_{\boldsymbol{x}_{1:n}\sim p}\left[\sigma^2(x_n|\{x_i\}_{i=0}^{n-1})\right] \leq C_n^2 \mathop{\mathbb{E}}_{x_{1:n-1}\sim p}\left[\sigma^2(x_{n-1}|\{x_i\}_{i=0}^{n-2})\right]. \tag{22}$$

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 $p(\boldsymbol{x})$,*

$$
nC_n^2 \mathop{\mathbb{E}}_{\boldsymbol{x}_{1:n}\sim p}\left[\sigma^2(\boldsymbol{x}_n|\{\boldsymbol{x}_i\}_{i=0}^{n-1})\right] \leq C_n^2 \mathop{\mathbb{E}}_{\boldsymbol{x}_{1:n}\sim p}\left[\sum_{j=1}^{n}\sigma^2(\boldsymbol{x}_j|\{\boldsymbol{x}_i\}_{i=0}^{j-1})\right]. \tag{23}
$$

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 γ_n represents the maximum information gain (Srinivas et al. [2012], Cover and Thomas [2006]).

Proof. We prove by induction. For $n = 1$, Eq. 23 holds trivially. Now assume $n > 1$,

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

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 γ_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 γ_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 $n \to \infty$. □

Kernel	Bounds on γ_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 γ_n from [Vakili et al., 2021, Theorem 5.]

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

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

For kernels with a maximum information capacity $\gamma_n = \mathcal{O}(poly(\log(n))$ that grows at most polylogarithmically with n, we have that

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

Proof. From Lemma B.7

$$\mathop{\mathbb{E}}_{x_{1:n-1}\sim p}\left[C_n^2 \mathop{\mathbb{E}}_{x\sim p}\left[\sigma_n^2(x)\right]\right] \leq \frac{C_n^2\gamma_n}{n}. \tag{24}$$

Using the Markov inequality, we get

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

Let X denote $C_n^2 \mathop{\mathbb{E}}_{x\sim p}\left[\sigma_n^2(x)\right]$. Then we have for all $a > 0$,

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

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 β_n (Lemma 3.6), we have $\frac{C_n^2\gamma_n}{n} \propto \frac{\gamma_n^{T+1}}{n}$. By assumption, we have that $\gamma_n = \mathcal{O}(\text{poly}(\log(n))$, and, thus $\frac{\gamma_n^{T+1}}{n} = \mathcal{O}(\text{poly}(\log(n))^{T+1}/n) = \mathcal{O}(\text{poly}(\log(n))/n)$. Hence, $\frac{C_n^2\gamma_n}{n} \to 0$ for $n \to \infty$. For example, for the linear and RBF kernel, we have (see Table 1)

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

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

Now to recover the rate of convergence, let $v = \mathop{\mathbb{E}}_{x\sim p}[C_n\sigma_n(x)]$. 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, $\mathbb{E}[v] \leq \sqrt{\mathbb{E}[v^2]} \leq C_n\sqrt{\frac{\gamma_n}{n}}$. Now, we apply the Chebyshev inequality, i.e.,

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

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

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

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

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

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 σ 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,

$$J(\pi_e) \geq \tilde{J}(\pi_e) \geq J(\pi_e) - C_n \mathop{\mathbb{E}}_{\boldsymbol{s},\boldsymbol{a}\sim\rho^{\pi_e}}\left[\|\boldsymbol{\sigma}_n(\boldsymbol{s},\boldsymbol{a})\|_2\right]. \tag{25}$$

To prove consistency, we then only need to show that $C_n \mathop{\mathbb{E}}_{\boldsymbol{s},\boldsymbol{a}\sim\rho^{\pi_e}}\left[\|\boldsymbol{\sigma}_n(\boldsymbol{s},\boldsymbol{a})\|_2\right]$ goes to 0 for $n \to \infty$. Since the support of the behavioural policy's state-occupancy measure ρ^{π_b} is compact, and $\text{supp}(\rho^{\pi_e}) \subseteq \text{supp}(\rho^{\pi_b})$, 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,

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

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

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

	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].

Consider a sequence $\{\delta_n\}_{n\geq 0}$ such that $\lim_{n\to 0}\delta_n = 0$, and $\lim_{n\to\infty} d_s\left(1+\sqrt{d_s/\delta_n}\right)\sqrt{\gamma_n^{T+1}/n} = 0$ (e.g., $\delta_n = \gamma_n^{-1}$), and let $S_n = \sum_{i=1}^{d_s} C_n\sigma_{n,i}(\boldsymbol{s},\boldsymbol{a})$. Then we have for all $\epsilon > 0$

$$\sum_{n=0}^{\infty}\mathbb{P}\left(S_n > \epsilon\right) = \sum_{n=0}^{N^*(\epsilon)}\mathbb{P}\left(S_n > \epsilon\right) + \sum_{n=N^*(\epsilon)}^{\infty}\mathbb{P}\left(S_n > \epsilon\right)$$
$$\leq N^*(\epsilon) + \sum_{n=N^*(\epsilon)}^{\infty}\delta_n < \infty,$$

where, $N^*(\epsilon)$ is the smallest integer such that $d_s\left(1+\sqrt{d_s/\delta}\right)\sqrt{\gamma_n^{T+1}/n} \leq \epsilon$. This implies that

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

□

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.

$$\tilde{J}(\pi^*) := \max_{\pi}\min_{\boldsymbol{\eta}} J_{\tilde{p}_{\boldsymbol{\eta}}}(\pi)\,. \tag{26}$$

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.

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 (26) 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 26). 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.

References

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

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.

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, 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*, 2018.

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

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.

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

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.

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.

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.