OFFLINE POLICY SELECTION UNDER UNCERTAINTY

Anonymous authors
Paper under double-blind review

ABSTRACT

The presence of uncertainty in policy evaluation significantly complicates the process of policy ranking and selection in real-world settings. We formally consider *offline policy selection* as learning preferences over a set of policy prospects given a fixed experience dataset. While one can select or rank policies based on point estimates of their policy values or high-confidence intervals, access to the full distribution over one's belief of the policy value enables more flexible selection algorithms under a wider range of downstream evaluation metrics. We propose *BayesDICE* for estimating this belief distribution in terms of posteriors of distribution correction ratios derived from stochastic constraints (as opposed to explicit likelihood, which is not available). Empirically, BayesDICE is highly competitive to existing state-of-the-art approaches in confidence interval estimation. More importantly, we show how the belief distribution estimated by BayesDICE may be used to rank policies with respect to any arbitrary downstream policy selection metric, and we empirically demonstrate that this selection procedure significantly outperforms existing approaches, such as ranking policies according to mean or high-confidence lower bound value estimates.

1 INTRODUCTION

Off-policy evaluation (OPE) (Precup et al., 2000) in the context of reinforcement learning (RL) is often motivated as a way to mitigate risk in practical applications where deploying a policy might incur significant cost or safety concerns (Thomas et al., 2015a). Indeed, by providing methods to estimate the value of a *target policy* solely from a static *offline* dataset of logged experience in the environment, OPE can help practitioners determine whether a target policy is or is not safe and worthwhile to deploy. Still, in many practical applications the ability to accurately estimate the online value of a specific policy is less of a concern than the ability to select or rank a set of policies (one of which may be the currently deployed policy). This problem, related to but subtly different from OPE, is *offline policy selection* (Doroudi et al., 2017; Paine et al., 2020; Kuzborskij et al., 2020), and it often arises in practice. For example, in recommendation systems, a practitioner may have a large number of policies trained offline using various hyperparameters, while cost and safety constraints only allow a few of those policies to be deployed as live experiments. Which policies should be chosen to form the small subset that will be evaluated online?

This and similar questions are closely related to OPE, and indeed, the original motivations for OPE were arguably with offline policy selection in mind (Precup et al., 2000; Jiang, 2017), the idea being that one can use estimates of the value of a set of policies to rank and then select from this set. Accordingly, there is a rich literature of approaches for computing point estimates of the value of the policy (Dudík et al., 2011; Bottou et al., 2013; Jiang & Li, 2015; Thomas & Brunskill, 2016; Nachum et al., 2019; Zhang et al., 2020; Uehara & Jiang, 2020; Kallus & Uehara, 2020; Yang et al., 2020). Because the offline dataset is finite and collected under a logging policy that may be different from the target policy, prior OPE methods also estimate high-confidence lower and upper bounds on a target policy's value (Thomas et al., 2015a; Kuzborskij et al., 2020; Bottou et al., 2013; Hanna et al., 2016; Feng et al., 2020; Dai et al., 2020; Kostrikov & Nachum, 2020). These existing approaches may be readily applied to our recommendation systems example, by using either mean or lower-confidence bound estimates on each candidate policy to rank the set and picking the top few to deploy online.

However, this naïve approach ignores crucial differences between the problem setting of OPE and the downstream evaluation criteria a practitioner prioritizes. For example, when choosing a few policies out of a large number of available policies, a recommendation systems practitioner may have a number of objectives in mind: The practitioner may strive to ensure that the policy with the overall highest groundtruth value is within the small subset of selected policies (akin to top-k precision). Or, in scenarios where the practitioner is sensitive to large differences in achieved value, a more relevant downstream metric may be the difference between the largest groundtruth value within the k selected policies compared to the groundtruth of the best possible policy overall (akin to top-k regret). With these or other potential offline policy selection metrics, it is far from obvious that ranking according to OPE estimates is ideal (Doroudi et al., 2017).

The diversity of potential downstream metrics in offline policy selection presents a challenge to any algorithm that yields a point estimate for each policy. Any one approach to computing point estimates will necessarily be sub-optimal for some policy selection criteria. To circumvent this challenge, we propose to compute a *belief distribution* over groundtruth values for each policy. Specifically, with the posteriors for the distribution over value for each policy calculated, one can use a straightforward procedure that takes estimation uncertainty into account to rank the policy candidates according to arbitrarily complicated downstream metrics. While this belief distribution approach to offline policy selection is attractive, it also presents its own challenge: how should one estimate a distribution over a policy's value in the pure offline setting?

In this work, we propose *Bayesian Distribution Correction Estimation (BayesDICE)* for off-policy estimation of a belief distribution over a policy's value. BayesDICE works by estimating posteriors over correction ratios for each state-action pair (correcting for the distribution shift between the off-policy data and the target policy's on-policy distribution). A belief distribution of the policy's value may then be estimated by averaging these correction distributions over the offline dataset, weighted by rewards. In this way, BayesDICE builds on top of the state-of-the-art DICE point estimators (Nachum et al., 2019; Zhang et al., 2020; Yang et al., 2020), while uniquely leveraging posterior regularization to satisfy chance constraints in a Markov decision process (MDP). As a preliminary experiment, we show that BayesDICE is highly competitive to existing frequentist approaches when applied to confidence interval estimation. More importantly, we demonstrate BayesDICE's application in offline policy selection under different utility measures on a variety of discrete and continuous RL tasks. Among other findings, our policy selection experiments suggest that, while the conventional wisdom focuses on using lower bound estimates to select policies (due to safety concerns) (Kuzborskij et al., 2020), policy ranking based on the lower bound estimates does not always lead to lower (top-k) regret. Furthermore, when other metrics of policy selection are considered, such as top-k precision, being able to sample from the posterior enables significantly better policy selection than only having access to the mean or confidence bounds of the estimated policy values.

2 Preliminaries

We consider an infinite-horizon Markov decision process (MDP) (Puterman, 1994) denoted as $\mathcal{M} = \langle S, A, R, T, \mu_0, \gamma \rangle$, which consists of a state space, an action space, a deterministic reward function,[1] a transition probability function, an initial state distribution, and a discount factor $\gamma \in (0, 1]$. In this setting, a policy $\pi(a_t|s_t)$ interacts with the environment starting at $s_0 \sim \mu_0$ and receives a scalar reward $r_t = R(s_t, a_t)$ as the environment transitions into a new state $s_{t+1} \sim T(s_t, a_t)$ at each timestep t. The value of a policy is defined as

$$\rho(\pi) := (1 - \gamma) \mathbb{E}_{s_0, a_t, s_t} \left[\sum_{t=0}^{\infty} \gamma^t r_t \right]. \tag{1}$$

2.1 Offline policy selection

We formalize the *offline policy selection* problem as providing a ranking $\mathcal{O} \in \mathrm{Perm}([1, N])$ over a set of *candidate* policies $\{\pi_i\}_{i=1}^N$ given only a *fixed* dataset $\mathcal{D} = \{x^{(j)} := (s_0^{(j)}, s^{(j)}, a^{(j)}, r^{(j)}, s'^{(j)})\}_{j=1}^n$ where $s_0^{(j)} \sim \mu_0$, $(s^{(j)}, a^{(j)}) \sim d^{\mathcal{D}}$ are samples of an unknown distribution $d^{\mathcal{D}}$, $r^{(j)} = R(s^{(j)}, a^{(j)})$, and $s'^{(j)} \sim T(s^{(j)}, a^{(j)})$.[2] One approach to the offline policy selection problem is to first characterize the *value* of each policy (Eq. 1, also known as the normalized per-step reward) via OPE under some *utility* function $u(\pi)$ that leverages a point estimate (or

[1] For simplicity, we restrict our analysis to deterministic rewards, and extending our methods to stochastic reward scenarios is straightforward.

[2] This tuple-based representation of the dataset is for notational and theoretical convenience, following Dai et al. (2020); Kostrikov & Nachum (2020), among others. In practice, the dataset is usually presented as finite-length trajectories $\{(s_0^{(j)}, a_0^{(j)}, r_0^{(j)}, s_1^{(j)}, \dots)\}_{j=1}^m$, and this can be processed into a dataset of finite samples from μ_0 and from $d^{\mathcal{D}} \times R \times T$. For mathematical simplicity, we assume that the dataset is sampled i.i.d. This

lower bound) of the policy value; i.e.,

$$\mathcal{O} \leftarrow \mathrm{ArgSortDescending}(\{u(\pi_i)\}_{i=1}^N).$$

2.2 Selection evaluation

A proposed ranking $\mathcal{O}$ will eventually be evaluated according to how well its policy ordering aligns with the policies' groundtruth values. In this section, we elaborate on potential forms of this evaluation score.

To this end, let us denote the groundtruth distribution of returns of policy π_i by $Z(\cdot|\pi_i)$. In other words, $Z(\cdot|\pi_i)$ is a distribution over $\mathbb{R}$ such that

$$z \sim Z(\cdot|\pi_i) \equiv \left[z := (1-\gamma) \sum_{t=0}^{\infty} \gamma^t \cdot R(s_t, a_t) \ ; \ s_0 \sim \mu_0, a_t \sim \pi_i(s_t), s_{t+1} \sim T(s_t, a_t) \right]. \quad (2)$$

Note that $\mathbb{E}_{Z(\cdot|\pi_i)}[z] = \rho(\pi_i)$.

As part of the offline policy selection problem, we are given a *ranking score* $\mathcal{S}$ that is a function of a proposed ranking $\mathcal{O}$ and groundtruth policy statistics $\{Z(\cdot|\pi_i)\}_{i=1}^N$. The ranking score $\mathcal{S}$ can take on many forms and is application specific; *e.g.*,

- **top-k precision**: This is an *ordinal* ranking score. The ranking score considers the top k policies in terms of groundtruth means $\rho(\pi_i)$ and returns the proportion of these which appear in the top k spots of $\mathcal{O}$.
- **top-k accuracy**: Another ordinal ranking score, this score considers the top-k policies in sorted order in terms of groundtruth means $\rho(\pi_i)$ and returns the proportion of these which appear in the same ordinal location in $\mathcal{O}$.
- **top-k correlation**: Another ordinal ranking score, this represents the Pearson correlation coefficient between the ranking of top-k policies in sorted order in terms of groundtruth means $\rho(\pi_i)$ and the truly best top-k policies.
- **top-k regret**: This is a *cardinal* ranking score. This score respresents the difference in groundtruth means $\rho(\pi_i)$ between the overall best policy – *i.e.*, $\max_i \rho(\pi_i)$ – and the best policy among the top-k ranked policies – *i.e.*, $\max_{i\in[1,k]} \rho(\pi_{\mathcal{O}[k]})$.
- **Beyond expected return**: One may define the above ranking scores in terms of statistics of $Z(\cdot|\pi_i)$ other than the groundtruth means $\rho(\pi_i)$. For example, in safety-critical applications, one may be concerned with the variance of the policy return. Accordingly, one may define CVaR analogues to top-k precision and regret.

For simplicity, we will restrict our attention to ranking scores which only depend on the average return of π_i. To this end, we will use $\overline{\rho}_i$ as shorthand for $\rho(\pi_i)$ and assume that the ranking score $\mathcal{S}$ is a function of $\mathcal{O}$ and $\{\overline{\rho}_i\}_{i=1}^N$.

2.3 Ranking score simulation from the posterior

It is not clear whether ranking according to vanilla OPE (either mean or confidence based) is ideal for any of the ranking scores above, including, for example, top-1 regret in the presence of uncertainty. However, if one has access to an approximate belief distribution over the policy's values, there is a simple sampling-based approach that can be used to find a near-optimal ranking (optimality depending on how accurate the belief distribution is) with respect to an arbitrary specified downstream ranking score, and we elaborate on this procedure here.

First, note that if we have access to the true groundtruth policy values $\{\overline{\rho}_i\}_{i=1}^N$, and the ranking score function $\mathcal{S}$, we can calculate the score value of *any* ranking $\mathcal{O}$ and find the ranking $\mathcal{O}^*$ that optimizes this score. However, we are limited to a finite offline dataset and the full return distributions are unknown. In this offline setting, we propose to instead compute a belief distribution $q(\{\overline{\rho}_i\}_{i=1}^N)$, and then we can optimize over the expected ranking score $\mathbb{E}_q\left[\mathcal{S}(\mathcal{O}, \{\overline{\rho}_i\}_{i=1}^N)\right]$ as shown in Algorithm 1. This algorithm simulates realizations of the groundtruth values $\{\overline{\rho}_i\}_{i=1}^N$ by sampling from the belief distribution $q(\{\overline{\rho}_i\}_{i=1}^N)$, and in this way estimates the expected realized ranking score $\mathcal{S}$ over all

is a common assumption in the OPE literature (Uehara & Jiang, 2020) and may be relaxed in some cases by assuming a fast mixing time (Nachum et al., 2019).

possible rankings $\mathcal{O}$. As we will show empirically, matching the selection process (the $\mathcal{S}$ used in Algorithm 1) to the downstream ranking score naturally leads to improved performance. The question now becomes how to effectively learn a belief distribution over $\{\overline{\rho}_i\}_{i=1}^N$.

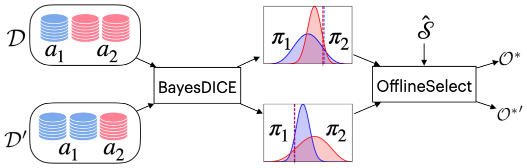

Figure 1: The belief distributions of $\overline{\rho}_1$ and $\overline{\rho}_2$ depend on the uncertainty induced from the finite offline data ($\mathcal{D}$ and $\mathcal{D}'$). A user might prefer π_2 only if $p(\overline{\rho}_2 < \overline{\rho}_1) < \xi$ (a choice of $\mathcal{S}$). OPE based on mean point estimates would select π_2 in either case as $\overline{\rho}_2$ has the greater mean. Sampling from the posterior belief in `OfflineSelect` allows simulation of any ranking score under $\mathcal{S}$, aligning policy selection with the user's choice of $\mathcal{S}$.

Algorithm 1 OfflineSelect

```
Inputs Posteriors $q(\{\overline{\rho}_i\}_{i=1}^N)$, ranking score $\hat{\mathcal{S}}$
Initialize $\mathcal{O}^*$; $L^*$        ▷ Track best score
for $\mathcal{O}$ in Perm([1, ..., N]) do
    $L = 0$
    for $j = 1$ to $n$ do
        sample $\{\hat{\rho}_i^{(j)}\}_{i=1}^N \sim q(\{\overline{\rho}_i\}_{i=1}^N)$
        ▷ Sum up sample scores
        $L = L + \hat{\mathcal{S}}(\{\hat{\rho}_i^{(j)}\}_{i=1}^N, \mathcal{O})$
    end for
    if $L < L^*$ then
        ▷ Update best ranking/score
        $L^* = L$; $\mathcal{O}^* = \mathcal{O}$
    end if
end for
return $\mathcal{O}^*, L^*$
```

3 BayesDICE

To learn a belief distribution over $\{\overline{\rho}_i\}_{i=1}^N$, we pursue a Bayesian approach to infer an approximate posterior distribution given prior beliefs. While model-based Bayesian approaches exist (*e.g.*, (Deisenroth & Rasmussen, 2011) and variants (Parmas et al., 2018)), they typically suffer from compounding error, so a model-free approach is preferable. However, Bayesian inference is challenging in this model-free scenario because the likelihood function is not easy to compute, as it is defined over infinite horizon returns.

Therefore, we first investigate several approaches to representing policy value, before identifying a novel posterior estimator that is computationally attractive and can support a broad range of ranking scores for downstream tasks.

3.1 Policy Ranking Score Representation

In practice, the downstream task of ranking or selecting policy candidates might require more than the value expectation, but also other properties of the policy value distribution. To ensure that the necessary distribution properties are computable, we first consider the class of ranking scores we would like to support:

- **Offline**: Since we focus on ranking policies given only *offline* data, the ranking score should not depend on on-policy samples.
- **Flexible**: Since the downstream task may utilize different ranking scores, the representation of the policy value should be sufficient to support their efficient computation.

With these considerations in mind, we review ways to represent the value of a policy π. Define $Q^\pi(s,a) = \mathbb{E}\left[\sum_{t=0}^\infty \gamma^t R(s_t,a_t) | s_0 = s, a_0 = a\right]$ and

$$d^\pi(s,a) = (1-\gamma)\sum_{t=0}^\infty \gamma^t d_t^\pi(s,a), \quad \text{with}$$
$$d_t^\pi(s,a) = \mathbf{P}\left(s_t = s, a_t = a | s_0 \sim \mu_0, \forall i < t, a_i \sim \pi(\cdot|s_i), s_{i+1} \sim T(\cdot|s_i,a_i)\right),$$

which are the *state-action* values and *stationary visitations* of π. These satisfy the recursions

$$Q^\pi(s,a) = R(s,a) + \gamma \cdot \mathcal{P}^\pi Q^\pi(s,a), \text{ where } \mathcal{P}^\pi Q(s,a) := \mathbb{E}_{s'\sim T(s,a), a'\sim\pi(s')}[Q(s',a')]; \tag{3}$$
$$d^\pi(s,a) = (1-\gamma)\mu_0(s)\pi(a|s) + \gamma \cdot \mathcal{P}_*^\pi d^\pi(s,a), \text{ where } \mathcal{P}_*^\pi d(s,a) := \pi(a|s)\sum_{\tilde{s},\tilde{a}} T(s|\tilde{s},\tilde{a})d(\tilde{s},\tilde{a}). \tag{4}$$

From these identities, the policy value can be expressed in two equivalent ways:

$$\rho(\pi) = (1-\gamma) \cdot \mathbb{E}_{\substack{a_0\sim\pi(s_0)\\ s_0\sim\mu_0}}[Q^\pi(s_0,a_0)] \tag{5}$$
$$= \mathbb{E}_{(s,a)\sim d^\pi}[r(s,a)]. \tag{6}$$

Current OPE methods are generally based on one of the representations (1), (5) or (6). For example, importance sampling (IS) estimators (Precup et al., 2000; Murphy et al., 2001; Dudík et al., 2011) are based on (1); LSTDQ (Lagoudakis & Parr, 2003) is a representative algorithm for fitting Q^π and thus based on (5); the recent DICE algorithms (Nachum & Dai, 2020; Yang et al., 2020) estimate the stationary density ratio $\zeta\left(s,a\right) := \frac{d^\pi(s,a)}{d^\mathcal{D}}$ so that $\rho\left(\pi\right) = \mathbb{E}_{d^\mathcal{D}}\left[\zeta \cdot r\right]$, and are thus based on (6).

Among the three strategies, the third is the most promising in our scenario. First, IS suffers from an exponential growth in variance (Liu et al., 2018) and further requires knowledge of the behavior policy. In contrast, the functions Q^π and d^π are duals (Nachum & Dai, 2020; Yang et al., 2020), and share common *behavior-agnostic* and minimax properties (Uehara & Jiang, 2020), However, estimation of Q^π assumes a ranking score with a linear dependence on $R\left(s,a\right)$, and therefore, even if we estimate Q^π accurately, it is still impossible to evaluate ranking scores that involve $\left(1-\gamma\right)\mathbb{E}\left[\sum_{t=0}^\infty \gamma^t \sigma(r_t)\right]$ such that $\sigma(\cdot) : \mathbb{R} \rightarrow \mathbb{R}$ is a nonlinear function (unless one learns a different Q function for each possible ranking score, which may be computationally expensive). By contrast, ranking scores with such nonlinear components can be easily computed from the stationary density ratio as $\mathbb{E}_{d^\mathcal{D}}\left[\zeta \cdot \sigma\left(r\right)\right]$. Given these considerations, the estimator via stationary density ratio satisfies both requirements: it enjoys statistical advantages in the offline setting and is flexible for downstream ranking score calculation. Therefore, we focus on a Bayesian estimator for ζ^π next.

3.2 Stationary Ratio Posterior Estimation

Recall that to apply a simple Bayesian approach to infer the posterior of ζ^π, one requires a $\log$-likelihood function, but such a quantity is not readily calculable in our scenario from the given data. Therefore, we develop an alternative, computationally tractable approach by considering an optimization view of Bayesian inference under a chance constraint, which allows us to derive the posterior over a set of stochastic equations.

Let $f\left(\cdot\right)$ denote a non-negative convex function with $f(0)$ achieving the minimum 0, *e.g.*, $f(x) = x^\top x$. Also let $\Delta_d\left(s,a\right) := (1-\gamma)\mu_0(s)\pi(a|s) + \gamma \cdot \mathcal{P}_*^\pi d(s,a) - d\left(s,a\right)$. Starting with (5) we reduce the $\left|S\right|\left|A\right|$ many constraints for the stationary distribution of π to a single feature-vector-based constraint for ζ:

$$\Delta_d\left(s,a\right) = 0, \quad \forall(s,a) \in S \times A \Rightarrow \left\langle \phi, \Delta_d \right\rangle = 0 \tag{7}$$

$$\Rightarrow \quad f\left(\left\langle \phi, \Delta_d \right\rangle\right) = 0 \Rightarrow \max_{\beta \in \mathcal{H}_\phi} \beta^\top \left\langle \phi, \Delta_d \right\rangle - f^*\left(\beta\right) = 0 \tag{8}$$

$$\Rightarrow \quad \max_{\beta \in \mathcal{H}_\phi} \mathbb{E}_{d^\mathcal{D}}\left[\zeta\left(s,a\right) \cdot \beta^\top\left(\gamma\phi(s',a') - \phi\left(s,a\right)\right)\right] + \left(1-\gamma\right)\mathbb{E}_{\mu_0\pi}\left[\beta^\top\phi\right] - f^*\left(\beta\right) = 0, \tag{9}$$

where $\mathcal{H}_\phi$ denotes the bounded Hilbert space with the feature mappings ϕ, $d^\mathcal{D}$ denotes the distribution generating the empirical experience, and we have used Fenchel duality in the middle step. The function $\phi\left(\cdot,\cdot\right) : S \times A \rightarrow \mathbb{R}^m$ is a feature mapping, with m possibly infinite. Then the condition $\left\langle \phi, \Delta_d \right\rangle = 0$ can be understood as matching the two distributions $(1-\gamma)\mu_0(s)\pi(a|s) + \gamma \cdot \mathcal{P}_*^\pi d(s,a)$ and $d\left(s,a\right)$ in terms of their embeddings (Smola et al., 2007), which is a generalization of the approximation methods in (De Farias & Van Roy, 2003; Lakshminarayanan et al., 2017). In particular, when $\left|S\right|\left|A\right|$ is finite and we set $\phi(s,a) = \delta_{s,a}$, where $\delta_{s,a} \in \left\{0,1\right\}^{|S||A|}$ is an indicator vector with a single 1 at position $\left(s,a\right)$ and 0 otherwise, we are matching the distributions pointwise. The feature map $\phi\left(s,a\right)$ can also be set to general reproducing kernel $k\left((s,a),\cdot\right) \in \mathbb{R}^\infty$. As long as the kernel $k\left(\cdot,\cdot\right)$ is characteristic, the embeddings will match if and only if the distributions are identical almost surely (Sriperumbudur et al., 2011).

Given that the experience was collected by some other means, *i.e.*, $\mathcal{D} \sim d^\mathcal{D}$, the constraint for ζ in (7) might not hold exactly. Therefore, we consider a feasible set $\zeta \in \left\{\zeta : \ell\left(\zeta, \mathcal{D}\right) \leqslant \epsilon\right\}$ where

$$\ell\left(\zeta,\mathcal{D}\right) := \max_{\beta\in\mathcal{H}_\phi} \hat{\mathbb{E}}_\mathcal{D}\left[\zeta\left(s,a\right) \cdot \beta^\top\left(\gamma\phi(s',a') - \phi\left(s,a\right)\right) - f^*\left(\beta\right)\right] + \left(1-\gamma\right)\mathbb{E}_{\mu_0\pi}\left[\beta^\top\phi\right]. \tag{10}$$

Note that $\ell\left(\zeta\right) \geqslant 0$ since $\mathcal{H}_\phi$ is symmetric. We expect the posterior of ζ, $q\left(\zeta\right)$, to concentrate most of its mass on this set and balance the prior. Formally, this means

$$\min_q \; KL\left(q||p\right) - \lambda\xi, \quad \text{s.t. } \mathbb{P}_q\left(\ell\left(\zeta\right) \leqslant \epsilon\right) \geqslant \xi, \tag{11}$$

where the chance constraint considers the probability of the feasibility of ζ under the posterior. This formulation can be equivalently rewritten as

$$\min_q \qquad KL\left(q||p\right) - \lambda\mathbb{P}_q\left(\ell\left(\zeta\right) \leqslant \epsilon\right) \tag{12}$$

Then, by applying Markov's inequality, *i.e.*, $\mathbb{P}_q\left(\ell\left(\zeta\right) \leqslant \epsilon\right) = 1 - \mathbb{P}_q\left(\ell\left(\zeta\right) \geqslant \epsilon\right) \geqslant 1 - \frac{\mathbb{E}_q[\ell(\zeta)]}{\epsilon}$, we can obtain an upper bound on (12) as

$$\min_q \; KL\left(q||p\right) + \frac{\lambda}{\epsilon}\mathbb{E}_q\left[\ell(\zeta, \mathcal{D})\right] \tag{13}$$

$$= \min_{q(\zeta)} \max_{q(\beta|\zeta)} KL\left(q||p\right) + \frac{\lambda}{\epsilon}\mathbb{E}_{q(\zeta)q(\beta|\zeta)}\Big[\hat{\mathbb{E}}_{\mathcal{D}}\left[\zeta\left(s,a\right) \cdot \beta^\top\left(\gamma\phi(s',a') - \phi\left(s,a\right)\right) - f^*\left(\beta\right)\right] + \left(1-\gamma\right)\mathbb{E}_{\mu_0\pi}\left[\beta^\top\phi\right]\Big], \tag{14}$$

where the equality follows by interchangeability (Shapiro et al., 2014; Dai et al., 2017). We amortize the optimization for β w.r.t. each ζ to a distribution $q\left(\beta|\zeta\right)$ to reduce the computational effort.

Due to the space limitation, we postpone the discussion about the important properties of BayesDICE, including the parametrization of the posteriors, the variants of BayesDICE for undiscounted MDP and alternatives of the log-likelihoods, and the connections to the vanilla Bayesian stochastic processes, to Appendix A. Please refer the details there.

Finally, note that with the posterior approximation for ζ_i, denoting the estimate for candidate policy i, we can draw posterior samples of $\bar{\rho}_i$ by drawing a sample $\zeta_i \sim q(\zeta_i)$ and computing $\hat{\rho}_i = \frac{1}{n}\sum_{(s,a,r)\in\mathcal{D}} \zeta_i(s,a)r$. This defines a posterior distribution over $\bar{\rho}_i$ and we further assume that the distributions are independent for each policy, so $q(\{\bar{\rho}_i\}_{i=1}^N) = \prod_i q(\bar{\rho}_i)$. This defines the necessary inputs for `OfflineSelect` to determine a ranking of the candidate policies.

4 Related work

We categorize the relevant related work into three categories: offline policy selection, off-policy evaluation, and Bayesian inference for policy evaluation.

Offline policy selection The decision making problem we formalize as offline policy selection is a member of a set of problems in RL referred to as *model selection*. Previously, this term has been used to refer to state abstraction selection (Jiang, 2017; Jiang et al., 2015) as well as learning algorithm and feature selection (Foster et al., 2019; Pacchiano et al., 2020). More relevant to our proposed notion of policy selection are a number of previous works which use model selection to refer to the problem of choosing a near-optimal Q-function from a set of candidate approximation functions (Fard & Pineau, 2010; Farahmand & Szepesvári, 2011; Irpan et al., 2019; Xie & Jiang, 2020). In this case, the evaluation metric is typically defined as the L_∞ norm of difference of Q versus the state-action value function of the optimal policy Q^*. While one can relate this evaluation metric to the sub-optimality (i.e., regret) of the policy induced by the Q-function, we argue that our proposed policy selection problem is both more general – since we allow for the use of policy evaluation metrics other than sub-optimality – and more practically relevant – since in many practical applications, the policy may not be expressible as the argmax of a Q-function. Lastly, the offline policy selection problem we describe is arguably a formalization of the problem approached in Paine et al. (2020) and referred to as *hyperparameter selection*. In contrast to this previous work, we not only formalize the decision problem, but also propose a method to directly optimize the policy selection evaluation metric. Offline policy selection has also been studied by Doroudi et al. (2017), which considers what properties a point estimator should have in order for it to yield good rankings in terms of a notion of ranking score referred to as *fairness*.

Off-policy evaluation Off-policy evaluation (OPE) is a highly active area of research. While the original motivation for OPE was in the pursuit of policy selection (Precup et al., 2000; Jiang, 2017), the field has historically almost exclusively focused on the related but distinct problem of estimating the online value (accumulated rewards) of a single target policy. In addition to a plethora of techniques for providing point estimates of this groundtruth value (Dudík et al., 2011; Bottou et al., 2013; Jiang & Li, 2015; Thomas & Brunskill, 2016; Kallus & Uehara, 2020; Nachum et al., 2019; Zhang et al., 2020; Yang et al., 2020), there is also a growing body of literature that uses frequentist principles to derive high-confidence lower bounds for the value of a policy (Bottou et al., 2013; Thomas et al., 2015b; Hanna et al., 2016; Kuzborskij et al., 2020; Feng et al., 2020; Dai et al., 2020; Kostrikov & Nachum, 2020). As our results demonstrate, ranking or selecting policies based on either their estimated mean or lower confidence bounds can at times be sub-optimal, depending on the evaluation criteria.

Bayesian inference for policy evaluation Our proposed method for policy selection relies on Bayesian principles to estimate a posterior distribution over the groundtruth policy value. While many Bayesian-inspired methods have been proposed for policy optimization (Deisenroth & Rasmussen, 2011; Parmas et al., 2018), especially in the context of exploration (Houthooft et al., 2016; Dearden et al., 2013; Kolter & Ng, 2009), relatively few have been proposed for policy evaluation. In one instance, Fard & Pineau (2010) derive PAC-Bayesian bounds on estimates of the Bellman error of a candidate Q-value function. In contrast to this work, we use our BayesDICE algorithm to estimate a distribution over target policy value, and this distribution allows us to directly optimize arbitrary downstream policy selection metrics.

5 Experiments

We empirically evaluate the performance of BayesDICE on confidence interval estimation (which can be used for policy selection) and offline policy selection under linear and neural network posterior parametrizations on tabular – Bandit, Taxi (Dietterich, 1998), FrozenLake (Brockman et al., 2016) – and continuous-control – Reacher (Brockman et al., 2016) – tasks. As we show below, BayesDICE outperforms existing methods for confidence interval estimation, producing accurate coverage while maintaining tight interval width, suggesting that BayesDICE achieves accurate posterior estimation, being robust to approximation errors and potentially misaligned Bayesian priors in practice. Moreover, in offline policy selection settings, matching the selection algorithm (Algorithm 1) to the ranking score (enabled by the estimating the posterior) shows clear advantages over ranking based on point estimates or confidence intervals on a variety of ranking scores. See Appendix C for additional results and implementation details.

5.1 Confidence interval estimation

Before applying BayesDICE to policy selection, we evaluate the BayesDICE approximate posterior by computing the accuracy of the confidence intervals it produces. We compare BayesDICE against a known set of confidence interval estimators based on concentration inequalities. To compute these baselines, we first use weighted (*i.e.*, self-normalized) per-step importance sampling (Thomas & Brunskill, 2016) to compute a policy value estimate for each logged trajectory. These trajectories provide a finite sample of value estimates. We use self-normalized importance sampling since it has been found to yield better empirical results in MDPs despite being biased (Liu et al., 2018; Nachum et al., 2019); for Bandit results without self-normalization, see Figure 5 in Appendix C. We then use empirical *Bernstein's* inequality (Thomas et al., 2015b), bias-corrected *bootstrap* (Thomas et al., 2015a), and *Student's t-test* to derive lower and upper high-confidence bounds on these estimates. We further consider Bayesian Deep Q-Networks (BDQN) (Azizzadenesheli et al., 2018) with an average empirical reward prior in the function approximation setting, which applies Bayesian linear regression to the last layer of a deep Q-network to learn a distribution of Q-values. Both BayesDICE and BDQN output a distribution of parameters, from which we conduct Monte Carlo sampling and use the resulting samples to compute a confidence interval at a given confidence level.

We plot the empirical coverage and interval width at different confidence levels in Figure 2. To compute the empirical *interval coverage*, we conduct 200 trials with randomly sampled datasets. The interval coverage is the proportion of the 200 intervals that contains the true value of the target policy. The *interval log-width* is the median of the log width of the 200 intervals. As shown in Figure 2, BayesDICE's coverage closely follows the intended coverage (black dotted line), while maintaining narrow interval width across all tasks considered. This suggests that BayesDICE's posterior estimation is highly accurate, being robust to approximation errors and potentially misaligned Bayesian priors in practice.

5.2 Policy selection

Next, we demonstrate the benefit of matching the policy selection criteria to the ranking score in offline policy selection. Our evaluation is based on a variety of cardinal and ordinal ranking scores defined in Section 2.2. We begin by considering the use of Algorithm 1 with BayesDICE-approximated posteriors. By keeping the BayesDICE posterior fixed, we focus our evaluation on the performance of Algorithm 1. We plot the groundtruth performance of this procedure applied to Bandit and Reacher in Figure 3. These figures compare using different $\hat{\mathcal{S}}$ to rank the policies according to Algorithm 1 across different downstream ranking scores $\mathcal{S}$. We find that aligning the criteria $\hat{\mathcal{S}}$ used in Algorithm 1 with the downstream ranking score $\mathcal{S}$ is empirically the best approach ($\hat{\mathcal{S}} = \mathcal{S}$).

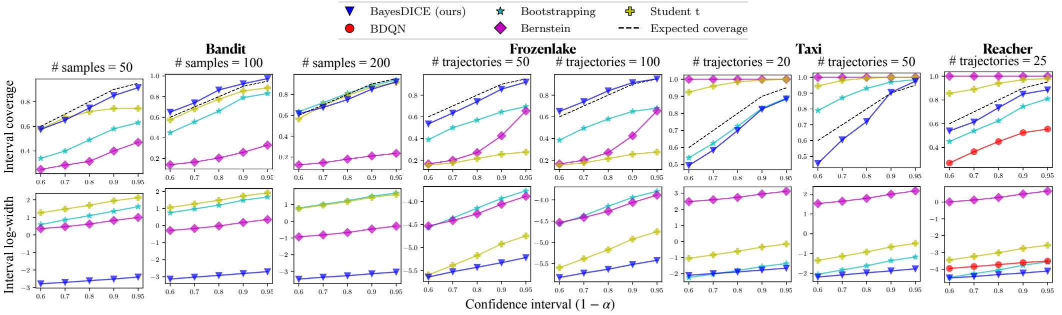

Figure 2: Confidence interval estimation on Bandit, FrozenLake, Taxi, and Reacher. The y-axis shows the empirical coverage and median log-interval width across 200 trials. BayesDICE exhibits near true coverage while maintaining narrow interval width, suggesting an accurate posterior approximation.

In contrast, using point estimates such as Mean or Mean $\pm$ Std can yield much worse downstream performance. We also see that in the Bandit setting, where we can analytically compute the Bayes-optimal ranking, using aligned ranking scores in conjunction with BayesDICE-approximated posteriors achieves near-optimal performance.

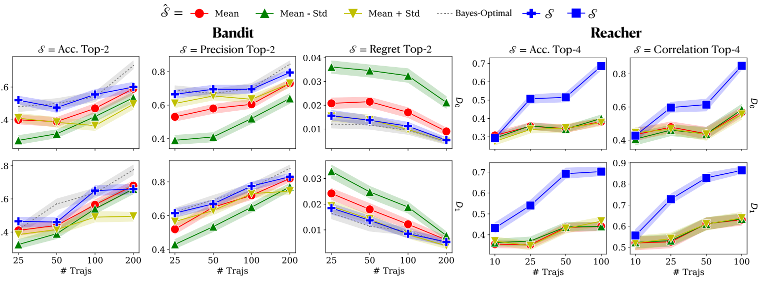

Figure 3: Policy selection using top-k ranking scores compared to mean/confidence ranking approaches on two-armed Bandit and Reacher. In these experiments, we fix the posterior to the one approximated by BayesDICE and evaluate different $\hat{\mathcal{S}}$ used in Algorithm 1 to compute a policy ranking. We find that using $\hat{\mathcal{S}} = \mathcal{S}$ (*i.e.*, aligning the ranking score in posterior simulation with the groundtruth evaluation) results in better performance than simple point estimates. Interestingly, the lower-bound point estimate almost always performs worse than the mean or the upper bound.

Having established BayesDICE's ability to compute accurate posterior distributions as well as the benefit of appropriately aligning the ranking score used in Algorithm 1, we compare BayesDICE to state-of-the-art OPE methods in policy selection. In these experiments, we use Algorithm 1 with posteriors approximated by BayesDICE and $\hat{\mathcal{S}} = \mathcal{S}$. We compare the use of BayesDICE in this way to ranking via point estimates of DualDICE (Nachum et al., 2019) and other confidence-interval estimation methods introduced in Section 5.1. We present results in Figure 4, in terms of top-k regret and correlation on bandit and reacher across different sample sizes and behavior data. BayesDICE outperforms other methods on both tasks. See additional ranking results in Appendix C.

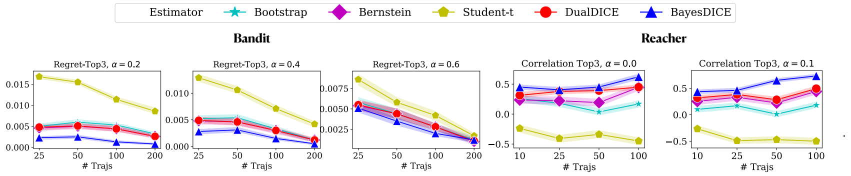

Figure 4: Policy selection evaluation under correlation and regret at top-k in two-armed Bandit (left) and Reacher (right) compared to other methods using point estimate (DualDICE) or high-confidence lower bounds. Please see Appendix C for more results with respect to other downstream metrics.

6 Conclusion

In this paper, we formally defined the offline policy selection problem, and proposed BayesDICE to first estimate posterior distributions of policy values before using a simulation-based procedure to compute an optimal policy ranking. Empirically, BayesDICE not only provides accurate belief distribution estimation, but also shows excellent performance in policy selection tasks.

REFERENCES

Kamyar Azizzadenesheli, Emma Brunskill, and Animashree Anandkumar. Efficient exploration through bayesian deep q-networks. In *2018 Information Theory and Applications Workshop (ITA)*, pp. 1–9. IEEE, 2018.

Aharon Ben-Tal, Laurent El Ghaoui, and Arkadi Nemirovski. *Robust optimization*. Princeton University Press, 2009.

Léon Bottou, Jonas Peters, Joaquin Quiñonero-Candela, Denis X. Charles, D. Max Chickering, Elon Portugaly, Dipankar Ray, Patrice Simard, and Ed Snelson. Counterfactual reasoning and learning systems: The example of computational advertising. *Journal of Machine Learning Research*, 14 (65):3207–3260, 2013. URL http://jmlr.org/papers/v14/bottou13a.html.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. OpenAI Gym. *arXiv preprint arXiv:1606.01540*, 2016.

Bo Dai, Niao He, Hanjun Dai, and Le Song. Provable bayesian inference via particle mirror descent. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, pp. 985–994, 2016.

Bo Dai, Niao He, Yunpeng Pan, Byron Boots, and Le Song. Learning from conditional distributions via dual embeddings. In *Artificial Intelligence and Statistics*, pp. 1458–1467, 2017.

Bo Dai, Ofir Nachum, Yinlam Chow, Lihong Li, Csaba Szepesvári, and Dale Schuurmans. Coindice: Off-policy confidence interval estimation. In *Advances in Neural Information Processing Systems*, 2020.

Daniela Pucci De Farias and Benjamin Van Roy. The linear programming approach to approximate dynamic programming. *Operations research*, 51(6):850–865, 2003.

Richard Dearden, Nir Friedman, and David Andre. Model-based bayesian exploration. *arXiv preprint arXiv:1301.6690*, 2013.

Marc Deisenroth and Carl E Rasmussen. Pilco: A model-based and data-efficient approach to policy search. In *Proceedings of the 28th International Conference on machine learning (ICML-11)*, pp. 465–472, 2011.

Thomas G Dietterich. The maxq method for hierarchical reinforcement learning. In *ICML*, volume 98, pp. 118–126. Citeseer, 1998.

Shayan Doroudi, Philip S Thomas, and Emma Brunskill. Importance sampling for fair policy selection. *Grantee Submission*, 2017.

Miroslav Dudík, John Langford, and Lihong Li. Doubly robust policy evaluation and learning. *arXiv preprint arXiv:1103.4601*, 2011.

Yaakov Engel, Shie Mannor, and Ron Meir. Bayes meets bellman: The gaussian process approach to temporal difference learning. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pp. 154–161, 2003.

Amir-massoud Farahmand and Csaba Szepesvári. Model selection in reinforcement learning. *Machine learning*, 85(3):299–332, 2011.

Mahdi M Fard and Joelle Pineau. Pac-bayesian model selection for reinforcement learning. In *Advances in Neural Information Processing Systems*, pp. 1624–1632, 2010.

Yihao Feng, Tongzheng Ren, Ziyang Tang, and Qiang Liu. Accountable off-policy evaluation with kernel bellman statistics. *arXiv preprint arXiv:2008.06668*, 2020.

Dylan J Foster, Akshay Krishnamurthy, and Haipeng Luo. Model selection for contextual bandits. In *Advances in Neural Information Processing Systems*, pp. 14741–14752, 2019.

Mohammad Ghavamzadeh, Shie Mannor, Joelle Pineau, and Aviv Tamar. Bayesian reinforcement learning: A survey. *arXiv preprint arXiv:1609.04436*, 2016.

Josiah P Hanna, Peter Stone, and Scott Niekum. Bootstrapping with models: Confidence intervals for off-policy evaluation. *arXiv preprint arXiv:1606.06126*, 2016.

Rein Houthooft, Xi Chen, Yan Duan, John Schulman, Filip De Turck, and Pieter Abbeel. Vime: Variational information maximizing exploration. In *Advances in Neural Information Processing Systems*, pp. 1109–1117, 2016.

Alexander Irpan, Kanishka Rao, Konstantinos Bousmalis, Chris Harris, Julian Ibarz, and Sergey Levine. Off-policy evaluation via off-policy classification. In *Advances in Neural Information Processing Systems*, pp. 5437–5448, 2019.

Nan Jiang. *A Theory of Model Selection in Reinforcement Learning*. PhD thesis, 2017.

Nan Jiang and Lihong Li. Doubly robust off-policy value evaluation for reinforcement learning. *arXiv preprint arXiv:1511.03722*, 2015.

Nan Jiang, Alex Kulesza, and Satinder Singh. Abstraction selection in model-based reinforcement learning. In *International Conference on Machine Learning*, pp. 179–188, 2015.

Nathan Kallus and Masatoshi Uehara. Double reinforcement learning for efficient off-policy evaluation in markov decision processes. *Journal of Machine Learning Research*, 21(167):1–63, 2020.

J Zico Kolter and Andrew Y Ng. Near-bayesian exploration in polynomial time. In *Proceedings of the 26th annual international conference on machine learning*, pp. 513–520, 2009.

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

Ilja Kuzborskij, Claire Vernade, András György, and Csaba Szepesvári. Confident off-policy evaluation and selection through self-normalized importance weighting. *arXiv preprint arXiv:2006.10460*, 2020.

Michail G Lagoudakis and Ronald Parr. Least-squares policy iteration. *Journal of machine learning research*, 4(Dec):1107–1149, 2003.

Chandrashekar Lakshminarayanan, Shalabh Bhatnagar, and Csaba Szepesvári. A linearly relaxed approximate linear program for markov decision processes. *CoRR*, abs/1704.02544, 2017.

Qiang Liu, Lihong Li, Ziyang Tang, and Dengyong Zhou. Breaking the curse of horizon: Infinite-horizon off-policy estimation. In *Advances in Neural Information Processing Systems*, pp. 5356–5366, 2018.

S. Murphy, M. van der Laan, and J. Robins. Marginal mean models for dynamic regimes. *Journal of American Statistical Association*, 96(456):1410–1423, 2001.

Ofir Nachum and Bo Dai. Reinforcement learning via fenchel-rockafellar duality. *arXiv preprint arXiv:2001.01866*, 2020.

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*, pp. 2315–2325, 2019.

Arkadi Nemirovski and Alexander Shapiro. Convex approximations of chance constrained programs. *SIAM Journal on Optimization*, 17(4):969–996, 2007.

Brendan ODonoghue, Ian Osband, Remi Munos, and Volodymyr Mnih. The uncertainty bellman equation and exploration. In *International Conference on Machine Learning*, pp. 3836–3845, 2018.

Ian Osband, Benjamin Van Roy, Daniel J Russo, and Zheng Wen. Deep exploration via randomized value functions. *Journal of Machine Learning Research*, 20(124):1–62, 2019.

Aldo Pacchiano, My Phan, Yasin Abbasi-Yadkori, Anup Rao, Julian Zimmert, Tor Lattimore, and Csaba Szepesvari. Model selection in contextual stochastic bandit problems. *arXiv preprint arXiv:2003.01704*, 2020.

Tom Le Paine, Cosmin Paduraru, Andrea Michi, Caglar Gulcehre, Konrad Zolna, Alexander Novikov, Ziyu Wang, and Nando de Freitas. Hyperparameter selection for offline reinforcement learning. *arXiv preprint arXiv:2007.09055*, 2020.

Paavo Parmas, Carl Edward Rasmussen, Jan Peters, and Kenji Doya. Pipps: Flexible model-based policy search robust to the curse of chaos. In *International Conference on Machine Learning*, pp. 4065–4074. PMLR, 2018.

Doina Precup, Richard S. Sutton, and Satinder P. Singh. Eligibility traces for off-policy policy evaluation. In *Proceedings of the 17th International Conference on Machine Learning*, pp. 759–766, 2000.

Martin L Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., 1994.

Alexander Shapiro, Darinka Dentcheva, and Andrzej Ruszczyński. *Lectures on stochastic programming: modeling and theory*. SIAM, 2014.

Alex Smola, Arthur Gretton, Le Song, and Bernhard Schölkopf. A hilbert space embedding for distributions. In *International Conference on Algorithmic Learning Theory*, pp. 13–31. Springer, 2007.

Bharath K Sriperumbudur, Kenji Fukumizu, and Gert RG Lanckriet. Universality, characteristic kernels and rkhs embedding of measures. *Journal of Machine Learning Research*, 12(7), 2011.

Philip Thomas and Emma Brunskill. Data-efficient off-policy policy evaluation for reinforcement learning. In *International Conference on Machine Learning*, pp. 2139–2148, 2016.

Philip Thomas, Georgios Theocharous, and Mohammad Ghavamzadeh. High confidence policy improvement. In *International Conference on Machine Learning*, pp. 2380–2388, 2015a.

Philip S Thomas, Georgios Theocharous, and Mohammad Ghavamzadeh. High-confidence off-policy evaluation. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015b.

Masatoshi Uehara and Nan Jiang. Minimax weight and q-function learning for off-policy evaluation. 2020.

Tengyang Xie and Nan Jiang. Batch value-function approximation with only realizability, 2020.

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*, 2020.

Arnold Zellner. Optimal Information Processing and Bayes's Theorem. *The American Statistician*, 42(4), November 1988.

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

Jun Zhu, Ning Chen, and Eric P. Xing. Bayesian inference with posterior regularization and applications to infinite latent svms. *J. Mach. Learn. Res.*, 15(1):1799–1847, January 2014. ISSN 1532-4435.

Appendix

A More Discussions on BayesDICE

In this section, we provide more details about BayesDICE.

Remark (parametrization of $q(\zeta)$ and $q(\beta|\zeta)$): We parametrize both $q(\zeta)$ (and the resulting $q(\beta|\zeta)$) as Gaussians with the mean and variance approximated by a multi-layer perceptron (MLP), *i.e.*: $\zeta = \text{MLP}_w(s,a) + \sigma_{w'}\xi,\ \xi \sim \mathcal{N}(0,1)$. w and w' denote the parameters of the MLP.

Remark (connection to Bayesian inference for stochastic processes): Recall the posterior can be viewed as the solution to an optimization (Zellner, 1988; Zhu et al., 2014; Dai et al., 2016),

$$q(\zeta|\mathcal{D}) = \mathop{\text{argmin}}_{q\in\mathcal{P}}\ \langle q(\zeta), \log p(\zeta,\mathcal{D})\rangle + KL\left(q(\zeta)\,||p(\zeta)\right)$$

The (13) is equivalent to define the $\log$-likelihood proportion to $\ell(\zeta,\mathcal{D})$, which is a stochastic process, including Gaussian process ($\mathcal{GP}$) by setting $f^*(\beta) = \frac{1}{2}\beta^\top\beta$. Specifically, plug $f(\beta) = \frac{1}{2}\beta^\top\beta$ back into (13), we have $\beta^* = \hat{\mathbb{E}}_{\mathcal{D}}\left[\zeta(s,a)\cdot(\gamma\phi(s',a') - \phi(s,a))\right] + (1-\gamma)\,\mathbb{E}_{\mu_0\pi}[\phi]$, resulting the optimization

$$\min_q KL(q||p) + \frac{\lambda}{\epsilon}\mathbb{E}_q\mathbb{E}_{\mu_0\pi}\hat{\mathbb{E}}_{\mathcal{D}}\left[\zeta(s_1,a_1)^\top k\left((s_1,a_1,s_1',a_1'),(s_2,a_2,s_2',a_2')\right)\zeta(s_2,a_2)\right], \quad (15)$$

with the kernel $k(x_1,x_2) := \left(\gamma\phi(s_1',a_1') - \phi(s_1,a_1)\right)^\top\left(\gamma\phi(s_2',a_2') - \phi(s_2,a_2)\right) + (1-\gamma)^2\phi\left(s_1^0,a_1^0\right)^\top\phi\left(s_1^0,a_1^0\right) + 2(1-\gamma)\,\phi\left(s_1^0,a_1^0\right)^\top\left(\gamma\phi(s_2',a_2') - \phi(s_2,a_2)\right)$, which is a $\mathcal{GP}$. Obviously, with different choices of $f^*(\cdot)$, the BayesDICE framework is far beyond $\mathcal{GP}$.

Although the $\mathcal{GP}$ has been applied for RL (Engel et al., 2003; Ghavamzadeh et al., 2016; Azizzadenesheli et al., 2018), they all focus on prior on value function; while BayesDICE considers general stochastic processes likelihood, including $\mathcal{GP}$, for the stationary ratio modeling, which as we justified is more flexible for different selection criteria in downstream tasks.

Remark (auxilary constraints and undiscounted MDP): As Yang et al. (2020) suggested, the non-negative and normalization constraints are important for optimization. We exploit positive neuron to ensure the non-negativity of the mean of the $q(\zeta)$. For the normalization, we consider the chance constraints $\mathbb{P}\left(\left(\hat{\mathbb{E}}_{\mathcal{D}}(\zeta) - 1\right)^2 \leqslant \epsilon_1\right) \geqslant \xi_1$. By applying the same technique, it leads to extra term $\frac{\lambda_1}{\epsilon_1}\mathbb{E}_q\left[\max_{\alpha\in\mathbb{R}}\alpha\cdot\hat{\mathbb{E}}_{\mathcal{D}}[\zeta - 1]\right]$ in (13).

With the normalization condition introduced, the proposed BayesDICE is ready for undiscounted MDP by simply setting $\gamma = 1$ in (13) together with the above extra term for normalization.

Remark (variants of $\log$-likelihood): We apply the Markov's inequality to (12) for the upper bound (13). In fact, the optimization with chance constraint has rich literature (Ben-Tal et al., 2009), where plenty of surrogates can be derived with different safe approximation. For example, if the q is simple, one can directly calculate the CDF for the probability $\mathbb{P}_q(\ell(\zeta) \leqslant \epsilon)$; or one can also exploit different probability inequalities to derive other surrogates, *e.g.*, condition value-at-risk, *i.e.*,

$$\min_q\ KL(q||p) + \lambda\inf_t\left[t + \frac{1}{\epsilon}\mathbb{E}_q\left[\ell(\zeta) - t\right]\right]_+, \quad (16)$$

and Bernstein approximation (Nemirovski & Shapiro, 2007). These surrogates lead to better approximation to the chance probability $\mathbb{P}_q(\ell(\zeta) \leqslant \epsilon)$ with the extra cost in optimization.

B BayesDICE for Exploration vs. Exploitation Tradeoff

In main text, we mainly consider exploiting BayesDICE for estimating various ranking scores for both discounted MDP and undiscounted MDP. In fact, with the posterior of the stationary ratio computed, we can also apply it for better balance between exploration vs. exploitation for policy optimization.

Instead of selecting from a set of policy candidates, the policy optimization is considering all feasible policies and selecting optimistically. Specifically, the feasibility of the stationary state-action distribution can be characterized as

$$\sum_a d\left(s,a\right) = \left(1-\gamma\right)\mu_0 + \mathcal{P}_* d\left(s\right), \quad \forall s \in S, \tag{17}$$

where $\mathcal{P}_* d\left(s\right) := \sum_{\bar{s},\bar{a}} T\left(s|\bar{s},\bar{a}\right) d\left(\bar{s},\bar{a}\right)$. Apply the feature mapping for distribution matching, we obtain the constraint for $\zeta\cdot\pi$ with $\zeta\left(s,a\right) := \frac{d(s)}{d^{\mathcal{D}}(s,a)}$ as

$$\max_{\beta\in\mathcal{H}_\phi} \; \beta^\top \mathbb{E}_{d^{\mathcal{D}}}\left[\sum_a \left(\zeta\left(s,a\right)\pi\left(a|s\right)\right)\phi\left(s\right) - \gamma\left(\zeta\left(s,a\right)\pi\left(a|s\right)\right)\phi\left(s'\right)\right] + \left(1-\gamma\right)\mathbb{E}_{\mu_0}\left[\beta^\top\phi\right] - f^*\left(\beta\right) = 0. \tag{18}$$

Then, we have the posteriors for all valid policies should satisfies

$$\lambda\mathbb{P}_q\left(\ell\left(\zeta\cdot\pi,\mathcal{D}\right) \leqslant \epsilon\right) \geqslant \xi, \tag{19}$$

with $\ell\left(\zeta\cdot\pi,\mathcal{D}\right) := \max_{\beta\in\mathcal{H}_\phi} \; \beta^\top\hat{\mathbb{E}}_{\mathcal{D}}\left[\sum_a \left(\zeta\left(s,a\right)\pi\left(a|s\right)\right)\phi\left(s\right) - \gamma\left(\zeta\left(s,a\right)\pi\left(a|s\right)\right)\phi\left(s'\right)\right] + \left(1-\gamma\right)\mathbb{E}_{\mu_0}\left[\beta^\top\phi\right] - f^*\left(\beta\right)$. Meanwhile, we will select one posterior from among these posteriors of all valid policies optimistically, *i.e.*,

$$\max_{q(\zeta)q(\pi)} \quad \mathbb{E}_q\left[U\left(\tau,r,\mathcal{D}\right)\right] + \lambda_1\xi - \lambda_2 KL\left(q\left(\zeta\right)q\left(\pi\right)||p\left(\zeta,\pi\right)\right) \tag{20}$$

$$\text{s.t.} \quad \mathbb{P}_q\left(\ell\left(\zeta\cdot\pi,\mathcal{D}\right) \leqslant \epsilon\right) \geqslant \xi \tag{21}$$

where $\mathbb{E}_q\left[U\left(\tau,r,\mathcal{D}\right)\right]$ denotes the optimistic policy score to capture the upper bound of the policy value estimation. For example, the most widely used one is

$$\mathbb{E}_q\left[U\left(\tau,r,\mathcal{D}\right)\right] = \mathbb{E}_q\hat{\mathbb{E}}_{\mathcal{D}}\left[\tau\cdot r\right] + \lambda_u\mathbb{E}_q\left[\left(\hat{\mathbb{E}}_{\mathcal{D}}\left[\tau\cdot r\right] - \mathbb{E}_q\hat{\mathbb{E}}_{\mathcal{D}}\left[\tau\cdot r\right]\right)^2\right],$$

where the second term is the empirical variance and usually known as one kind of “exploration bonus”.

Then the whole algorithm is iterating between solving (20) and use the obtain policy collecting data into $\mathcal{D}$ in (20).

This Exploration-BayesDICE follows the same philosophy of Osband et al. (2019); ODonoghue et al. (2018) where the variance of posterior of the policy value is taken into account for exploration. However, there are several significant differences: **i),** the first and most different is the modeling object, Osband et al. (2019); ODonoghue et al. (2018) is updating with Q-function, while we are handling the dual representation; **ii),** BayesDICE is compatible with arbitary nonlinear function approximator, while Osband et al. (2019); ODonoghue et al. (2018) considers tabular or linear functions; **iii),** BayesDICE is considering infinite-horizon MDP, while Osband et al. (2019); ODonoghue et al. (2018) considers fixed finite-horizon case. Therefore, the exploration with BayesDICE pave the path for principle and practical exploration-vs-exploitation algorithm. The regret bound is out of the scope of this paper, and we leave for future work.

C Experiment details and additional results

C.1 Environments and policies.

Bandit. We create a Bernoulli two-armed bandit with binary rewards where α controls the proportion of optimal arm ($\alpha = 0$ and $\alpha = 1$ means never and always choosing the optimal arm respectively). Our selection experiments are based on 5 target policies with $\alpha = [0.75, 0.8, 0.85, 0.9, 0.95]$.

Reacher. We modify the Reacher task to be infinite horizon, and sample trajectories of length 100 in the behavior data. To obtain different behavior and target policies, We first train a deterministic policy from OpenAI Gym (Brockman et al., 2016) until convergence, and define various policies by converting the optimal policy into a Gaussian policy with optimal mean with standard deviation $0.4 - 0.3\alpha$. Our selection experiments are based on 5 target policies with $\alpha = [0.75, 0.8, 0.85, 0.9, 0.95]$.

C.2 Details of neural network implementation

We parametrize the distribution correction ratio as a Gaussian using a deep neural network for the continuous control task. Specifically, we use feed-forward networks with two hidden-layers of 64 neurons each and ReLU as the activation function. The networks are trained using the Adam optimizer ($\beta_1 = 0.99$, $\beta_2 = 0.999$) with batch size 2048.

C.3 Additional experimental results

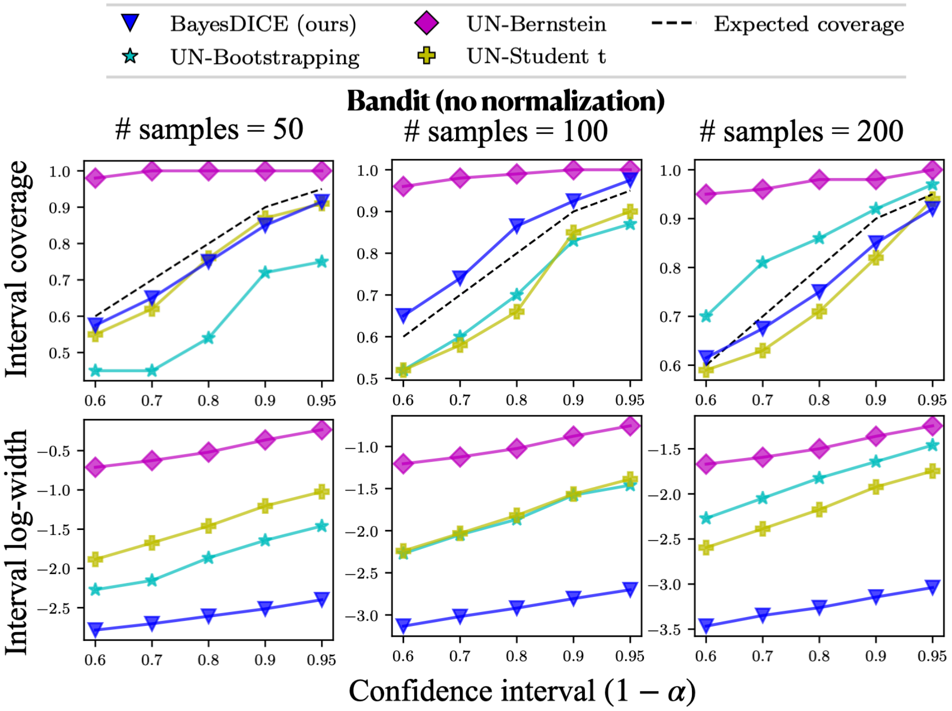

Figure 5: Confidence interval estimation on Bandit where the baselines are unnormalized (UN).

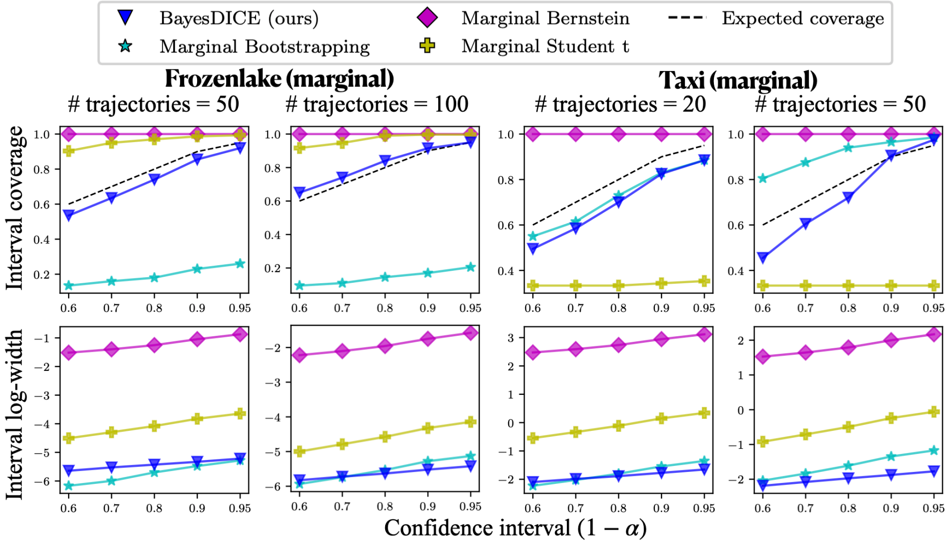

Figure 6: Confidence interval estimation with baselines computed from marginalized importance sampling.

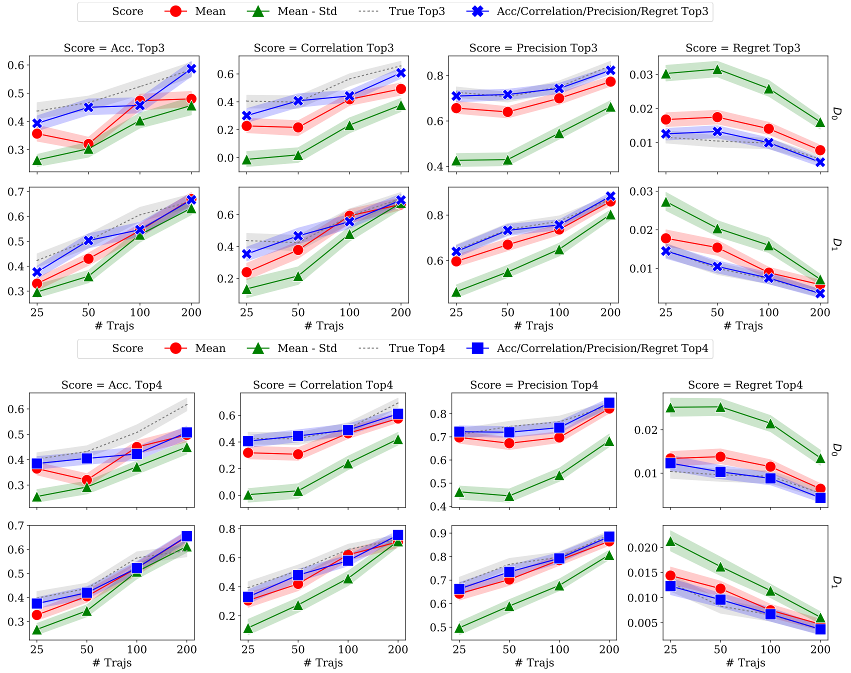

Figure 7: Additional k values for top-k ranking on bandit. Ranking results based on Algorithm 1 (blue lines) always perform better than using mean or high-confidence lower bound.

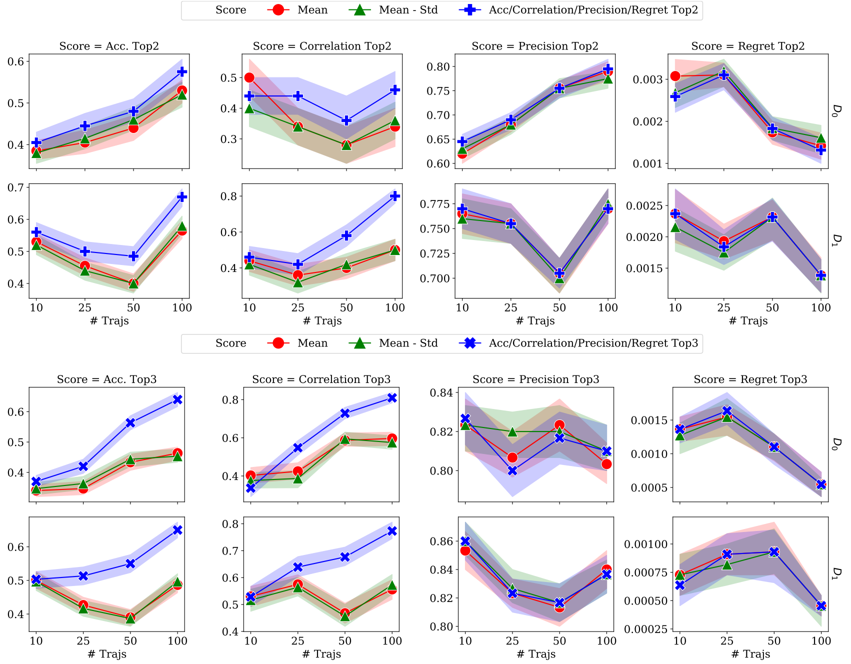

Figure 8: Additional k values for top-k ranking on reacher and additional scores (precision and regret). Ranking results based on Algorithm 1 (blue lines) generally perform much better than using mean or high-confidence lower bound for top-k accuracy and correlation. Precision and regret are similar between posterior samples and the mean/confidence bound based ranking.

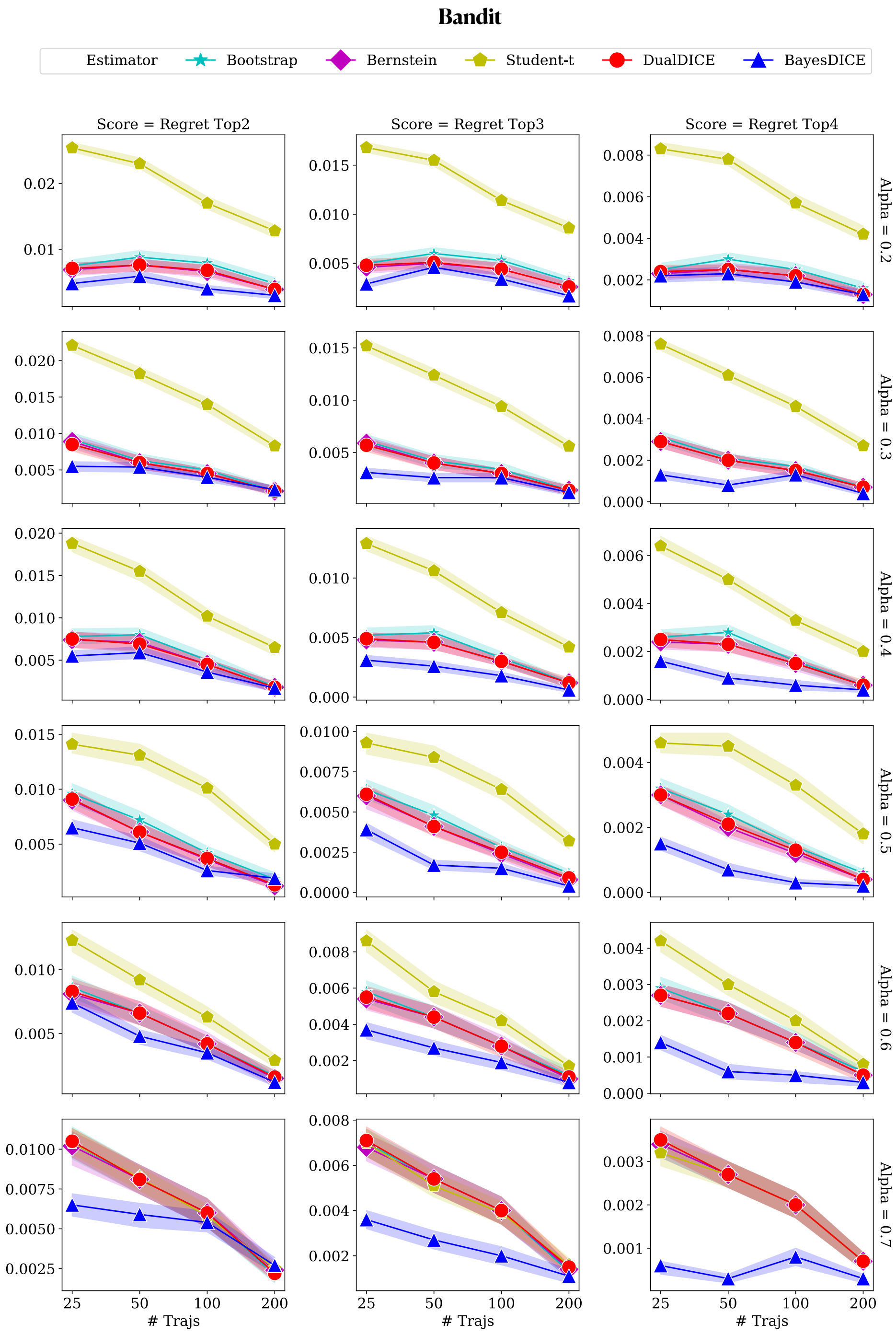

Figure 9: Improved regret using BayesDICE across all trajectory lengths, behavior data, and top-k values considered for the bandit task.

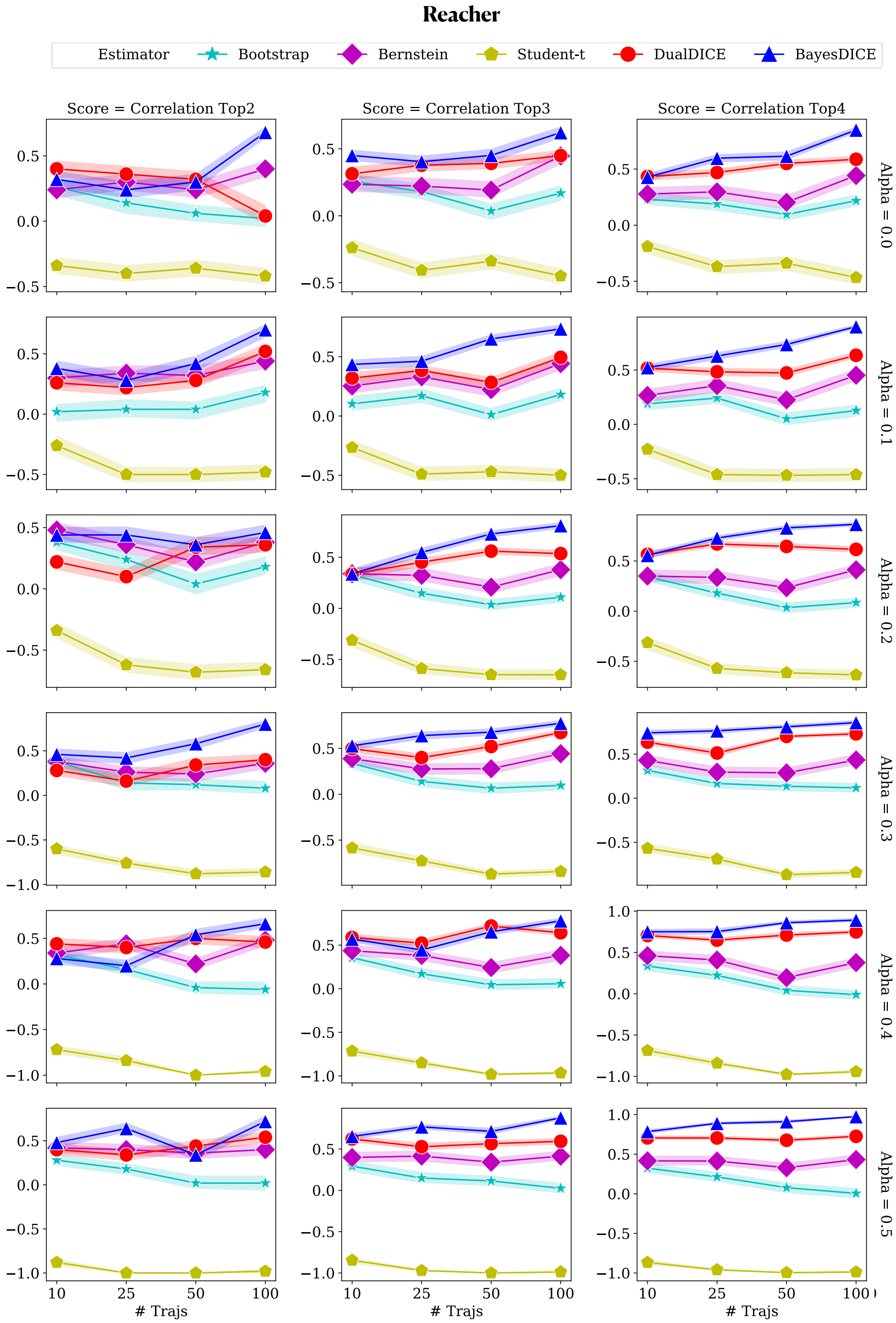

Figure 10: Improved correlation using BayesDICE across all trajectory lengths, behavior data, and top-k values considered for the reacher task.