Model Selection for Off-policy Evaluation: New Algorithms and Experimental Protocol

Pai Liu UIUC
Lingfeng Zhao Columbia University
Shivangi Agarwal IIIT Delhi
Jinghan Liu USTC
Audrey Huang UIUC
Philip Amortila UIUC
Nan Jiang* UIUC

Abstract

Holdout validation and hyperparameter tuning from data is a long-standing problem in offline reinforcement learning (RL). A standard framework is to use off-policy evaluation (OPE) methods to evaluate and select between different policies, but OPE methods either incur exponential variance (e.g., importance sampling) or have hyperparameters of their own (e.g., FQE and model-based). We focus on model selection for OPE itself, which is even more under-investigated. Concretely, we select among candidate value functions ("model-free") or dynamics models ("model-based") to best assess the performance of a target policy. We develop: (1) new model-free and model-based selectors with theoretical guarantees, and (2) a new experimental protocol for empirically evaluating them. Compared to the model-free protocol in prior works, our new protocol allows for more stable generation and better control of candidate value functions in an optimization-free manner, and evaluation of model-free and model-based methods alike. We exemplify the protocol on Gym-Hopper, and find that our new model-free selector, LSTD-Tournament, demonstrates promising empirical performance.

1 Introduction

Offline reinforcement learning (RL) is a promising paradigm for applying RL to important application domains where perfect simulators are not available and we must learn from data [34, 23]. Despite the significant progress made in devising more performant *training* algorithms, how to perform holdout validation and model selection—an indispensable component of any practical machine learning pipeline—remains an open problem and has hindered the deployment of RL in real-life scenarios. Concretely, after multiple training algorithms (or instances of the same algorithm with different hyperparameter settings) have produced candidate policies, the *primary* task (which contrasts the *secondary* task which we focus on) is to select a good policy from these candidates, much like how we select a good classifier/regressor in supervised learning. To do so, we may estimate the performance (i.e., expected return) of each policy, and select the one with the highest estimated return.

Unfortunately, estimating the performance of a new *target* policy based on data collected from a different *behavior* policy is a highly challenging task, known as *off-policy evaluation* (OPE). Popular OPE algorithms can be roughly divided into two categories, each with their own critical weaknesses: the first is importance sampling [49, 22, 55], which has elegant unbiasedness guarantees but suffers variance that is *exponential* in the horizon, limiting applicability beyond short-horizon settings such as contextual bandits [36]. The second category includes algorithms such as Fitted-Q Evaluation (FQE) [14, 31, 47], marginalized importance sampling [38, 44, 57], and model-based approaches

*Correspondence to: nanjiang@illinois.edu.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

https://doi.org/10.52202/085713-1430

[61], which avoid the exponential variance; unfortunately, this comes at the cost of introducing their own hyperparameters (choice of neural architecture, learning rates, etc.). While prior works have reported the effectiveness of these methods [47], they also leave a chicken-and-egg problem: **if these algorithms tune the hyperparameters of training, who tunes *their* hyperparameters?**

In this work, we make progress on the latter problem, namely model selection for OPE algorithms themselves. We lay out and study a dichotomy of two settings: in the **model-based** setting, evaluation algorithms build dynamics models to evaluate a target policy. Given the uncertainty of hyperparameters in model learning, we assume that multiple candidate models are given, and the task is to select one that best evaluates the performance of the target policy. In the **model-free** setting, evaluation algorithms only output *value functions* which we select from; see Figure 1 (left) for a visualization of the pipeline. Notably, these selection algorithms should be *hyperparameter-free themselves*, to avoid further chicken-and-egg problems. Our contributions are:

1. **New model-free selector (Section 3):** We propose a new selection algorithm (or simply *selector*), LSTD-Tournament, for selecting between candidate value functions by approximately checking whether the function satisfies the Bellman equation. The key technical difficulty here is the infamous ***double sampling*** problem [7, 53, 10], which typically requires additional function approximation (hence hyperparameters) to bypass. Our derivation builds on BVFT [62, 64], which is the only existing selector that addresses double sampling in a theoretically rigorous manner without additional function approximation assumptions. Our new selector enjoys better statistical rates ($n^{-1/2}$ vs. $n^{-1/4}$) and empirically outperforms BVFT and other baselines.
2. **New model-based selectors (Section 4):** When comparing candidate models, popular losses in model-based RL (e.g., ℓ_2 loss on next-state prediction) often exhibit biases under stochastic transitions [21]. Instead, we propose novel estimators with theoretical guarantees, including novel adaptation of previous model-free selectors that require additional assumptions, where the assumptions are automatically satisfied using additional information in the model set [6, 65].
3. **New experiment protocol (Section 5)**: To empirically evaluate the selection algorithms, prior works often use FQE to prepare candidate Q-functions [64, 46], which suffer from unstable training[2] and lack of control in the quality of the candidate functions. We propose a new experiment protocol, where the candidate value functions are induced from variations of the groundtruth environment; see Figure 1 (right) for an illustration. This bypasses the caveats of FQE and allows for the computation of Q-values in an *optimization-free* and controllable manner. Moreover, the protocol can also be used to evaluate and compare estimators for the model-based setting. Implementation-wise, we use ***lazy evaluation*** and Monte-Carlo roll-outs to generate the needed Q-values. Combined with parallelization and caching, we reduce the computational cost and make the evaluation of new algorithms easier.
4. **Preliminary experiments (Section 6)**: We instantiate the protocol in Gym Hopper [9] and demonstrate the various ways in which we can evaluate and understand different selectors.

2 Preliminaries and Model Selection Problem

Markov Decision Process (MDP). An MDP is specified by $(\mathcal{S}, \mathcal{A}, P, R, \gamma, d_0)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $P : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ is the transition dynamics, $R : \mathcal{S} \times \mathcal{A} \to [0, R_{\max}]$ is the reward function, $\gamma \in [0, 1)$ is the discount factor, and d_0 is the initial state distribution. A policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$ induces a distribution over random trajectories, generated as $s_0 \sim d_0$, $a_t \sim \pi(\cdot|s_t)$, $r_t = R(s_t, a_t)$, $s_{t+1} \sim P(\cdot|s_t, a_t)$. We use $\Pr_\pi[\cdot]$ and $\mathbb{E}_\pi[\cdot]$ to denote such a distribution and the expectation thereof. The performance of a policy is defined as $J(\pi) := \mathbb{E}_\pi[\sum_{t=0}^{\infty} \gamma^t r_t]$, which is in the range of $[0, V_{\max}]$ where $V_{\max} := R_{\max}/(1 - \gamma)$.

Value Function and Bellman Operator. The Q-function $Q^\pi \in \mathbb{R}^{\mathcal{S}\times\mathcal{A}}$ is the fixed point of $\mathcal{T}^\pi : \mathbb{R}^{\mathcal{S}\times\mathcal{A}} \to \mathbb{R}^{\mathcal{S}\times\mathcal{A}}$, i.e., $Q^\pi = \mathcal{T}^\pi Q^\pi$, where for any $f \in \mathbb{R}^{\mathcal{S}\times\mathcal{A}}$, $(\mathcal{T}^\pi f)(s, a) := R(s, a) + \gamma \mathbb{E}_{s\sim P(\cdot|s,a)}[f(s', \pi)]$. We use the shorthand $f(s', \pi)$ for $\mathbb{E}_{a'\sim\pi(\cdot|s')}[f(s', a')]$.

Off-policy Evaluation (OPE). OPE is about estimating the performance of a given *target* policy π in the real environment denoted as $M^\star = (\mathcal{S}, \mathcal{A}, P^\star, R, \gamma, d_0)$, namely $J_{M^\star}(\pi)$, using an offline dataset $\mathcal{D}$ sampled from a behavior policy π_b. For simplicity, from now on we may drop the $M^\star$

[2]For example, our preliminary investigation has found that FQE often diverges with CQL-trained policies [27], which is echoed by Nie et al. [46] in personal communications.

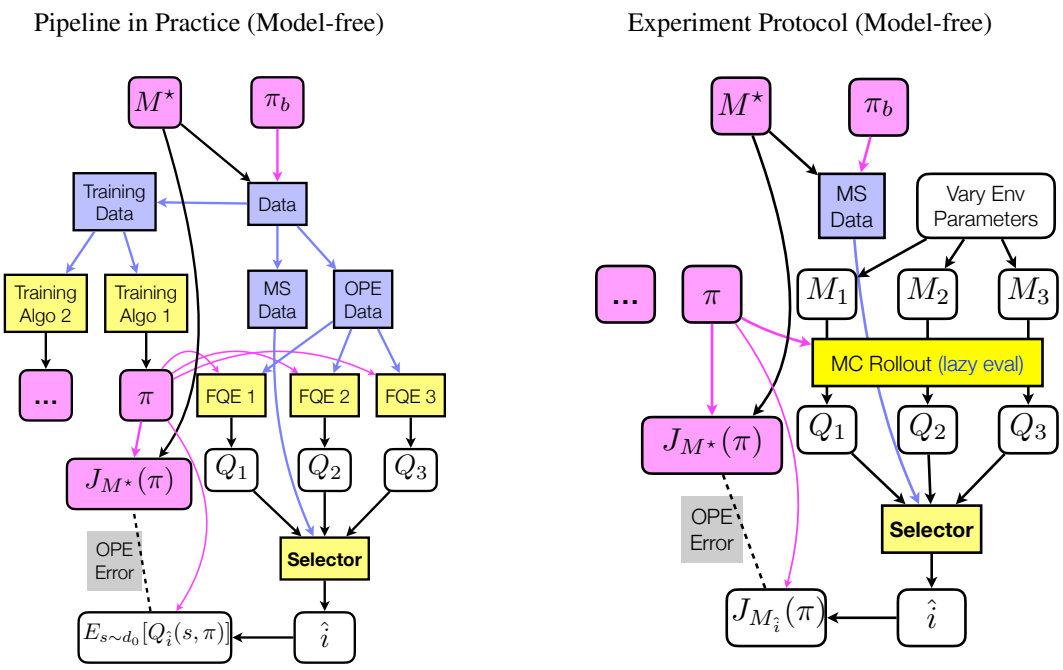

Figure 1: An illustration of the pipeline in practice that motivates our research (**left**) and our proposed experimental protocol (**right**), both for the model-free setting; the pipeline for the model-based setting is analogous and not visualized. **Left:** Training algorithms are run on training data to generate candidate policies, and choosing among them is the *primary model-selection (MS) task* (see Section 1) which is not the focus of our work. These policies (e.g., π) become the target policies in OPE (e.g., π), since accurate OPE can help solve the primary MS problem. Different FQE instances (e.g., with different hyperparameters, such as neural architectures) are used to approximate Q^π, producing $\{Q_i\}$. The **selector** takes MS data and $\{Q_i\}$ as input and choose one of them to estimate $J(\pi) \equiv J_{M^\star}(\pi)$. **Right:** Illustration of our protocol for an experiment *unit* (Section 5). The target policies π can be learned from separate training data but can also be produced in other ways, such as training on inaccurate models M_i, which can be realistic for practical scenarios with inaccurate simulators. $\{M_i\}$ is prepared by varying environment parameters. Monte-Carlo rollouts are used to generate the Q-values for the data points in the MS data, which avoids potentially unstable optimization in the OPE pipeline; this is the source of stability and controllability compared to the prior protocol that mimics the practical pipeline. For further discussion on the limitation of our protocol and the trade-offs, see Section 7.

in the subscript when referring to properties of $M^\star$, e.g., $J(\pi) \equiv J_{M^\star}(\pi)$, $Q^\pi \equiv Q^\pi_{M^\star}$, etc. As a standard simplification, our theoretical derivation assumes that the dataset $\mathcal{D}$ consists of n i.i.d. tuples (s, a, r, s') generated as $(s, a) \sim \mu$, $r = R(s, a)$, $s' \sim P^\star(\cdot|s, a)$. We use $\mathbb{E}_\mu[\cdot]$ to denote the true expectation under the data distribution, and $\mathbb{E}_\mathcal{D}[\cdot]$ denotes the empirical approximation from $\mathcal{D}$.

Model Selection. We assume that there are multiple OPE *instances* (e.g. OPE algorithms with different hyperparameters) that estimate $J(\pi)$, and our goal is to choose among them based on the offline dataset $\mathcal{D}$; see Figure 1 for a visualization. Our setup is agnostic w.r.t. the details of the OPE instances, and views them only through the Q-functions or the dynamics models they produce.[3] Concretely, two settings are considered:

- **Model-free:** Each OPE instance outputs a Q-function. The validation task is to select $\hat{Q}$ from the candidate Q-functions $\mathcal{Q} := \{Q_i\}_{i\in[m]}$,[4] such that $\mathbb{E}_{s\sim d_0}[\hat{Q}(s, \pi)] \approx J(\pi)$. We are interested in the regime where at least one of the OPE instances produces a reasonable approximation of Q^π, i.e., $\exists i, Q_i \approx Q^\pi$, while the quality of other candidates can be arbitrarily poor. We make

[3]For alternative formulations such as selecting a function *class* for the OPE algorithm, see Appendix A.

[4]In practical scenarios, $\{Q_i\}_{i\in[m]}$ and $\{M_i\}_{i\in[m]}$ may be learned from data, and we assume $\mathcal{D}$ is a holdout dataset independent of the data used for producing $\{M_i\}_{i\in[m]}$ and $\{Q_i\}_{i\in[m]}$.

https://doi.org/10.52202/085713-1430

the simplification assumption $Q^\pi \in \mathcal{Q}$ for theoretical derivations, and handling misspecification ($Q^\pi \notin \mathcal{Q}$) is routine in recent RL theory and omitted for presentation purposes [3, 4]. We will also test our algorithms empirically in the misspecified case (Section 6.2).

- **Model-based:** Each OPE instance outputs an MDP M_i and uses $J_{M_i}(\pi)$ as an estimate of $J(\pi)$. W.l.o.g. we assume M_i only differs from $M^\star$ in the transition P_i, as handling different reward functions is straightforward. The task is to select $\hat{M}$ from $\mathcal{M} := \{M_i\}_{i\in[m]}$, such that $J_{\hat{M}}(\pi) \approx J(\pi)$. Similar to the model-free case, we assume $M^\star \in \mathcal{M}$ in the derivations.

The model-free algorithms have wider applicability, especially when we lack structural knowledge of the dynamics. A model-free algorithm can always be applied to the model-based setting: any $\{M_1, \dots, M_m\}$ induces a Q-function class $\{Q^\pi_{M_1}, \dots, Q^\pi_{M_m}\}$ which can be fed into a model-free algorithm (as we demonstrate in our protocol; see Section 5), and $M^\star \in \mathcal{M}$ implies $Q^\pi = Q^\pi_{M^\star} \in \mathcal{Q}$.

3 New Model-Free Selector

In this section we introduce our new model-free selector, LSTD-Tournament. To start, we review the difficulties in model-free selection and the idea behind BVFT [62, 64] which we build on.

3.1 Challenges of Model-free Selection and BVFT

To select Q^π from $\mathcal{Q} = \{Q_1, \dots, Q_m\}$, perhaps the most natural idea is to check how much each candidate function Q_i violates the Bellman equation $Q^\pi = \mathcal{T}^\pi Q^\pi$, and choose the function that minimizes such a violation. This motivates the Bellman error (or residual) objective:

$$\mathbb{E}_\mu[(Q_i - \mathcal{T}^\pi Q_i)^2]. \tag{1}$$

Unfortunately, this loss cannot be estimated due to the infamous *double-sampling problem* [7, 53, 10], and the naïve estimation, which squares the TD error, is a biased estimation of the Bellman error (Eq.(1)) in stochastic environments:

$$\text{(TD-sq)} \qquad \mathbb{E}_\mu[(Q_i(s,a) - r - \gamma Q_i(s', \pi))^2]. \tag{2}$$

Common approaches to debiasing this objective involve additional "helper" classes (e.g., $\mathcal{G}$ in Section 4), making them not hyperparameter-free. See also Appendix A for a detailed discussion of how existing works tackle this difficulty under different formulations and often stronger assumptions.

BVFT. The idea behind BVFT [62] is to find an OPE algorithm for learning Q^π from a function class $\mathcal{F}$, such that to achieve polynomial sample-complexity guarantees, it suffices if $\mathcal{F}$ satisfies 2 assumptions: (1) Realizability, that $Q^\pi \in \mathcal{F}$. (2) Some *structural* (as opposed to *expressivity*) assumption on $\mathcal{F}$, e.g., smoothness, linearity, etc. Standard learning results in RL typically require stronger expressivity assumption than realizability, such as the widely adopted *Bellman-completeness* assumption ($\mathcal{T}^\pi f \in \mathcal{F}, \forall f \in \mathcal{F}$). However, exceptions exist, and BVFT shows that *they can be converted into a pairwise-comparison subroutine* for selecting between two candidates $\{Q_i, Q_j\}$, and extension to multiple candidates can be done via a tournament procedure. Crucially, we can use $\{Q_i, Q_j\}$ to **automatically create an $\mathcal{F}$ needed by the algorithm** without additional side information or prior knowledge. We will demonstrate such a process in the next subsection.

In short, BVFT provides a general recipe for converting a special kind of "base" OPE methods into selectors with favorable guarantees. Intuitively, the "base" method/analysis will determine the properties of the resulting selector. The "base" of BVFT is Q^π-irrelevant abstractions [35, 20], where the structural assumption on $\mathcal{F}$ is being piecewise-constant. Our novel insight is that for learning Q^π, there exists another algorithm, LSTDQ [29], which satisfies the needed criteria and has superior properties compared to Q^π-irrelevant abstractions, thus can induce better selectors than BVFT.

3.2 LSTD-Tournament

We now provide a theoretical analysis of LSTDQ (which is simplified from the literature [42, 48]), and show how to transform it into a selector via the BVFT recipe. In LSTDQ, we learn Q^π via linear function approximation, i.e., it is assumed that a feature map $\phi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$ is given, such that $Q^\pi(s,a) = \phi(s,a)^\top \theta^\star$, where $\theta^\star \in \mathbb{R}^d$ is the groundtruth linear coefficient. Equivalently, this asserts that the induced linear class, $\mathcal{F}_\phi := \{\phi(\cdot)^\top \theta : \theta \in \mathbb{R}^d\}$ satisfies realizability, $Q^\pi \in \mathcal{F}_\phi$.

https://doi.org/10.52202/085713-1430

LSTDQ provides a closed-form estimation of $\theta^\star$ by first estimating the following moment matrices:

$$\Sigma := \mathbb{E}_\mu[\phi(s,a)\phi(s,a)^\top], \quad \Sigma^{\text{cr}} := \mathbb{E}_\mu[\phi(s,a)\phi(s',\pi)^\top], \tag{3}$$

$$A := \Sigma - \gamma\Sigma^{\text{cr}}, \quad b := \mathbb{E}_\mu[\phi(s,a)r]. \tag{4}$$

As a simple algebraic fact, $A\theta^\star = b$. Therefore, when A is invertible, we immediately have that $\theta^\star = A^{-1}b$. The LSTDQ algorithm thus simply estimates A and b from data, denoted as $\widehat{A}$ and $\widehat{b}$, respectively, and estimate $\theta^\star$ as $\widehat{A}^{-1}\widehat{b}$. Alternatively, for any candidate θ, $\|A\theta - b\|_\infty$ can serve as a loss function that measures the violation of the equation $A\theta^\star = b$, which we can minimize over.Its finite-sample guarantee is given below. All proofs of the paper can be found in Appendix C.

Theorem 1. *Let $\Theta \subset \mathbb{R}^d$ be a set of parameters such that $\theta^\star \in \Theta$. Assume $\max_{s,a}\|\phi(s,a)\|_2 \le B_\phi$ and $\max_{\theta\in\Theta}\|\theta\|_2 \le 1$. Let $\widehat{\theta} := \arg\min_{\theta\in\Theta}\|\widehat{A}\theta - \widehat{b}\|_\infty$. Then, with probability at least $1-\delta$,*

$$\|Q^\pi - \hat{\phi}^\top\theta\|_\infty \le \frac{6\max\{R_{\max}, B_\phi\}^2}{\sigma_{\min}(A)}\sqrt{\frac{d\log(2d|\Theta|/\delta)}{n}}, \tag{5}$$

where $\sigma_{\min}(\cdot)$ is the smallest singular value.

Besides the realizability of Q^π, the guarantee also depends on the invertibility of A, which can be viewed as a coverage condition, since A changes with the data distribution μ [2, 3, 23]. In fact, in the on-policy setting, $\sigma_{\min}(A)$ can be shown to be lower-bounded away from 0 [42]; see Appendix C.5.

LSTD-Tournament. We are now ready to describe our new selector. Recall that we first deal with the case of two candidate functions, $\{Q_i, Q_j\}$, where $Q^\pi \in \{Q_i, Q_j\}$. To apply the LSTDQ algorithm and guarantee, all we need is to create the feature map ϕ such that Q^π is linearly realizable in ϕ. In the spirit of BVFT, we design the feature map as

$$\phi_{i,j}(s,a) := [Q_i(s,a), Q_j(s,a)]^\top. \tag{6}$$

The subscript "i,j" makes it clear that the feature is created based on Q_i and Q_j as candidates, and we will use similar conventions for all quantities induced from $\phi_{i,j}$, e.g., $A_{i,j}, b_{i,j}$, etc. Obviously, Q^π is linear in $\phi_{i,j}$ with $\theta^\star \in \{[1,0]^\top, [0,1]^\top\}$. Therefore, to choose between Q_i and Q_j, we can calculate the LSTDQ loss of $[1,0]^\top$ and $[0,1]^\top$ under feature $\phi_{i,j}$ and choose the one with smaller loss. For $\theta = [1,0]^\top$, we have $A_{i,j}\theta - b_{i,j} =$

$$\mathbb{E}_\mu\Big\{\begin{bmatrix}Q_i(s,a)\\ Q_j(s,a)\end{bmatrix}([Q_i(s,a), Q_j(s,a)] - \gamma[Q_i(s',\pi), Q_j(s',\pi)])\Big\}\begin{bmatrix}1\\0\end{bmatrix} \tag{7}$$

$$-\mathbb{E}_\mu\big[[Q_i(s,a), Q_j(s,a)]\cdot r\big]\begin{bmatrix}1\\0\end{bmatrix} = \mathbb{E}_\mu\left[\begin{bmatrix}Q_i(s,a)\\ Q_j(s,a)\end{bmatrix}(Q_i(s,a) - r - \gamma Q_i(s',\pi))\right]. \tag{8}$$

Taking the infinity-norm of the loss vector, we have

$$\|A_{i,j}\begin{bmatrix}1\\0\end{bmatrix} - b_{i,j}\|_\infty = \max_{k\in\{i,j\}}|\mathbb{E}_\mu[Q_k(s,a)(Q_i(s,a) - r - \gamma Q_i(s',\pi))]|. \tag{9}$$

The loss for $\theta = [0,1]^\top$ is similar, where Q_i is replaced by Q_j. Following BVFT, we can generalize the procedure to m candidate functions $\{Q_1, \ldots, Q_m\}$ by pairwise comparison and recording the worst-case loss, this leads to our final loss function:

$$\mathcal{L}(Q_i; \{Q_j\}_{j\in[m]}, \pi) := \max_{k\in[m]}|\mathbb{E}_\mu[Q_k(s,a)(Q_i(s,a) - r - \gamma Q_i(s',\pi))]|. \tag{10}$$

The actual algorithm replaces $\mathbb{E}_\mu$ with the empirical estimation from data, and chooses the Q_i that minimizes the loss. Building on Theorem 1, we have the following guarantee:

Theorem 2. *Given $Q^\pi := Q_{i^\star} \in \{Q_i\}_{i\in[m]}$, the $Q_{\widehat{i}}$ that minimizes the empirical estimation of $\mathcal{L}(Q_i; \{Q_j\}_{j\in[m]}, \pi)$ (Eq.(10)) satisfies that w.p. $\ge 1-\delta$,*

$$|J(\pi) - \mathbb{E}_{s\sim d_0}[Q_{\widehat{i}}(s,\pi)]| \le \max_{i\in[m]\setminus\{i^\star\}}\frac{24V_{\max}^3}{\sigma_{\min}(A_{i,i^\star})}\sqrt{\frac{\log(8m/\delta)}{n}}. \tag{11}$$

https://doi.org/10.52202/085713-1430

Comparison to BVFT [62]. BVFT's guarantee has a slow $n^{-1/4}$ rate for OPE [64, 19], whereas our method enjoys the standard $n^{-1/2}$ rate. The difference is due to an adaptive discretization step in BVFT, which also makes its implementation somewhat complicated as the resolution needs to be heuristically chosen. By comparison, the implementation of LSTD-Tournament is simple and straightforward. Both methods inherit the coverage assumptions from their base algorithms and are not directly comparable, and a nuanced discussion on this issue can be found in Appendix C.5.

Variants. A key step in the derivation is to design the linearly realizable feature of Eq.(6), but the design is not unique as any non-degenerate linear transformation would also suffice. For example, we can use $\phi_{i,j} = [Q_i/c_i, (Q_j - Q_i)/c_{j,i}]$; the "diff-of-value" term $Q_j - Q_i$ has shown improved numerical properties in practice [27, 12], and c_i, $c_{j,i}$ can normalize the discriminators to unit variance for further numerical stability; this will also be the version we use in the main-text experiments. Empirical comparison across these variants can be found in Appendix E.2.

4 New Model-Based Selectors

We now consider the model-based setting, i.e., choosing a model from $\{M_i\}_{i\in[m]}$ such that $J_{\hat{M}}(\pi) \approx J(\pi)$. This can be practically relevant when we have structural knowledge of the system dynamics and can build reasonable simulators, but simulators of complex real-world systems will likely have many design choices and knobs that cannot be set from prior knowledge alone. As alluded to at the end of Section 2, model-free methods can always be applied to the model-based setting by letting $\mathcal{Q} := \{Q^\pi_{M_i}\}_{i\in[m]}$. The question is: *are there methods that leverage the additional side information in $\mathcal{M}$ (that is not in $\mathcal{Q}$) to outperform model-free methods?*

We study this question by developing algorithms with provable guarantees that can only be run with $\mathcal{M}$ but not $\mathcal{Q}$ (which means they must be using additional information), and include them in the empirical comparisons in Section 6. To our surprise, however, these algorithms are outperformed by LSTD-Tournament, which is simpler, computationally more efficient, and more widely applicable. Despite this, our model-based development produces novel theoretical results and can be of independent interest, and also provides additional baselines for LSTD-Tournament in empirical comparison. We briefly describe the studied methods below, with details deferred to Appendix B.

1. **Naïve Baseline.** A natural method is to use the model-prediction loss: any model M is scored by $\mathbb{E}_{(s,a,s')\sim\mu,\tilde{s}\sim P(\cdot|s,a)}[\|s' - \tilde{s}\|]$, where P is the dynamics of M, and $\|\cdot\|$ is some norm (e.g., ℓ_2 norm) that measures the distance between states. Despite its wide use in the model-based RL literature, the loss has a crucial caveat that **it exhibits a bias towards more deterministic systems**, which is problematic when the groundtruth environment is stochastic. We will see empirical evidence of this in Section 6 (see "mb_naive" in Figures 3 and 4).

2. **Regression-based Selector (Appendix B.1).** Recall from Section 3 that a main difficulty in directly estimating the Bellman error is the lack of access to $\mathcal{T}^\pi$. Antos et al. [6] suggest that $\mathcal{T}^\pi f$ for any f can be learned via regression $\mathcal{T}^\pi f \in \arg\min_{g\in\mathcal{G}} \mathbb{E}_\mu[(g(s,a) - r - \gamma f(s',\pi))^2]$, provided that $\mathcal{T}^\pi f \in \mathcal{G}$. Zitovsky et al. [65] apply this idea to the model selection problem and uses a user-provided $\mathcal{G}$, which requires additional hyperparameters as a model-free selector. However, in Appendix B.1 we show that the $\mathcal{G}$ class can be constructed *automatically* from $\mathcal{M}$, making it a strong baseline (due to its use of additional information in $\mathcal{M}$) for our LSTD-Tournament in the empirical comparison ("mb_Zitovsky et al." and "mb_Antos et al" in Figure 4).

3. **Sign-flip Selector (Appendix B.2).** We also develop a novel selector that takes the spirit of regression-based selector, but replace its squared loss with absolute value. It comes with new theoretical guarantee (Theorem 4), and is implemented in the experiments ("mb_sign_flip").

5 A Model-Based Experiment Protocol

Given the new selectors, we would like to evaluate and compare them empirically. However, as alluded to in the introduction, current experiment protocols have various caveats and make it difficult to evaluate the estimators in well-controlled settings. In this section, we describe a novel model-based experiment protocol, which can be used to evaluate both model-based and model-free selectors.

5.1 The Protocol

Our protocol consists of experiment textitunits defined by the following elements (see Figure 1R):

1. Groundtruth model $M^\star$.
2. Candidate model list $\mathcal{M} = \{M_i\}_{i\in[m]}$.
3. Behavior policy π_b and offline sample size n.
4. Target policies $\Pi = \{\pi_1, \ldots, \pi_l\}$.

Given the specification of a unit, we will draw a dataset of size n from $M^\star$ using behavior policy π_b. For each target policy $\pi \in \Pi$, we apply different selectors to choose a model $M \in \mathcal{M}$ to evaluate π. Model-free algorithms will access M only through its Q-function, Q^π_M, effectively choosing from the set $\mathcal{Q} = \{Q^\pi_M : M \in \mathcal{M}\}$. Finally, the prediction error $|J_M(\pi) - J_{M^\star}(\pi)|$ is recorded and averaged over the target policies in Π. Moreover, we may gather results from multiple units that share the same $M^\star$ but differ in $\mathcal{M}$ and/or the behavior policy to investigate issues such as robustness to misspecification and data coverage, as we will demonstrate in the next section.

Lazy Evaluation of Q-values via Monte Carlo. While the pipeline is conceptually straightforward, practically accessing the Q-function Q^π_M is nontrivial: we could run TD-style algorithms in M to learn Q^π_M, but that invokes a separate RL algorithm that may require additional tuning and verification, and it can be difficult to control the quality of the learned function.

Our innovation here is to note that, for all the model-free algorithms we are interested in evaluating, **they all access Q^π_M exclusively through the value of $Q^\pi_M(s,a)$ and $Q^\pi_M(s',\pi)$ for (s,a,r,s') in the offline dataset $\mathcal{D}$**. That is, given n data points in $\mathcal{D}$, we only need to know $2n$ scalar values about Q^π_M. Therefore, we propose to directly compute these values without explicitly representing Q^π_M, and each value can be easily estimated by averaging over multiple Monte-Carlo rollouts [50], i.e., $Q^\pi_M(s,a) = \mathbb{E}_\pi[\sum_{t=0}^\infty \gamma^t r_t | s_0 = s, a_0 = a]$ can be approximated by rolling out multiple trajectories starting from (s,a) and taking actions according to π. For the model-based estimators proposed in Section 4, we need access to quantities in the form of $(\mathcal{T}^\pi_{M_j} Q^\pi_{M_i})(s,a)$ (see Appendix B). This value can also be obtained by Monte-Carlo simulation: (1) start in (s,a) and simulate one step in M_j, then (2) switch to M_i, simulate from step 2 onwards and rollout the rest of the trajectory.

5.2 Computational Efficiency

Despite not involving neural-net optimization, the experiment can still be computationally intensive due to rolling out a large number of trajectories. In our code, we incorporate the following measures to reduce the computational cost:

Q-caching. The most intensive computation is to roll-out Monte-Carlo trajectories for Q-value estimation; the cost of running the actual selection algorithms is often much lower and negligible. Hence, we generate these MC Q-estimates and save them to files, and retrieve them during selection. This makes it efficient to experiment with new selection algorithms or add extra baselines, and also enables fast experiment that involves a subset of the candidate models (see Section 6.2).

Bootstrapping. To account for the randomness of $\mathcal{D}$, we use bootstrapping to sample (with replacement) multiple datasets, and report an algorithm's mean performance across these bootstrapped samples with 95% confidence intervals. Using bootstrapping maximally reuses the cached Q-values and avoids the high computational costs of sampling multiple datasets and performing Q-caching in each of them, which is unavoidable if we were to repeat each experiment verbatim multiple times.

6 Exemplification of the Protocol

In this section we instantiate our protocol in the Gym Hopper environment to demonstrate its utility, while also providing preliminary empirical results for our algorithms. Our code is available at `https://github.com/Coder-PAI/2025_neurips_model_selection_rl.git`.

6.1 Experiment Setup and Main Results

Our experiments will be based on the *Hopper-v4* environment [9]. To create a variety of environments, we add different levels of stochastic noise in the transitions and change the gravity constant (see

 https://doi.org/10.52202/085713-1430

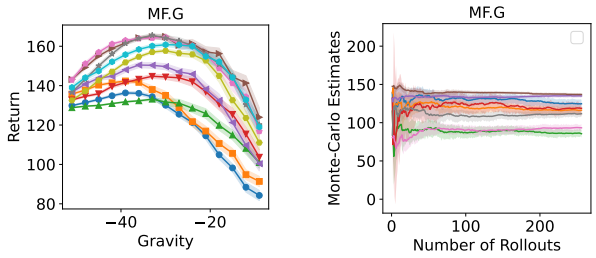

Figure 2: **Left:** $J_M(\pi)$ in $M \in \mathcal{M}_{\mathbf{g}}$ (cf. Section 6.1) for different target policies. **Right:** Convergence of Monte-Carlo estimates of $J(\pi)$. Each curve corresponds to a target policy.

Appendix D.1). Each environment is then parameterized by the gravity constant $\mathbf{g}$ and noise level $\mathbf{n}$. We consider arrays of such environments as the set of candidate simulator $\mathcal{M}$: in most of our results, we consider a "gravity grid" (denoted using **MF.G** in the figures) $\mathcal{M}_{\mathbf{g}} := \{M^0_{\mathbf{g}} \dots, M^{14}_{\mathbf{g}}\}$ (fixed noise level, varying gravity constants from -51 to -9) and a "noise grid" (**MF.N**) $\mathcal{M}_{\mathbf{n}} := \{M^0_{\mathbf{n}} \dots, M^{14}_{\mathbf{n}}\}$ (fixed gravity constant, varying noise level from 10 to 100). Each array contains 15 environments, though some subsequent results may only involve a subset of them (Section 6.2). Some of these simulators will also be treated as groundtruth environment $M^\star$, which determines the groundtruth performance of target policies and produces the offline dataset $\mathcal{D}$.

Behavior and Target Policies. We create 15 target policies by running DDPG [37] in one of the environments and take checkpoints. For each $M^\star$, the behavior policy is the randomized version of one of the target policies; see Appendix D.2 for details. A dataset is collected by sampling trajectories until $n = 3200$ transition tuples are obtained. As a sanity check, we plot $J_M(\pi)$ for $\pi \in \Pi_{\mathbf{g}}$ and $M \in \mathcal{M}_{\mathbf{g}}$ in Figure 2. As can be shown in the figure, the target policies have different performances, and also vary in a nontrivial manner w.r.t. the gravity constant $\mathbf{g}$. It is important to perform such a sanity check to avoid degenerate settings, such as $J_M(\pi)$ varies little across $M \in \mathcal{M}$ (then even a random selection will be accurate) or across $\pi \in \Pi$.

Number of Rollouts. We then decide the two important parameters for estimating the Q-value, the number of Monte-Carlo rollouts l and the horizon (i.e., trajectory length) H. For horizon, we set $H = 1024$ which is substantially longer than typically observed trajectories from the target policies. For l, we plot the convergence of $J_M(\pi)$ estimation and choose $l = 128$ accordingly (see Figure 2R).

Compared Methods. We compare our methods with baselines, including TD-square (Eq.(2)), naïve model-based (Eq.(12)), BVFT [64], and "average Bellman error" $|E_{\mathcal{D}}[Q_i(s,a) - r - \gamma Q_i(s', \pi)]|$ [25], which can be viewed as our LSTD-Tournament but with a trivial constant discriminator. The model-based methods in Section 4 require MC rollouts for $\{T^\pi_{M_j} Q^\pi_{M_i} : i, j \in [m]\}$, which requires $O(m^2)$ computational complexity. Therefore, we first compare other selectors (mostly model-free) in Figure 3 with $m = 15$; the relatively large number of candidate simulators will also enable the

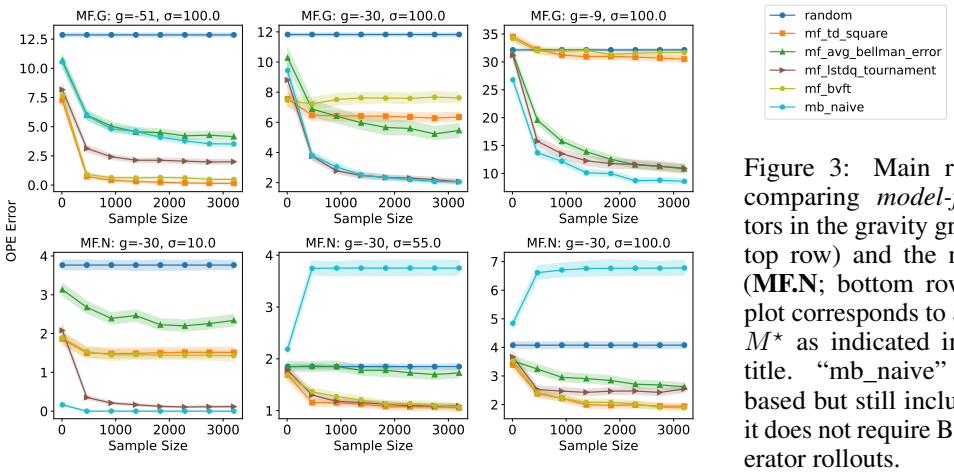

Figure 3: Main results for comparing *model-free* selectors in the gravity grid (**MF.G**; top row) and the noise grid (**MF.N**; bottom row). Each plot corresponds to a different $M^\star$ as indicated in the plot title. "mb_naive" is model-based but still included since it does not require Bellman operator rollouts.

https://doi.org/10.52202/085713-1430

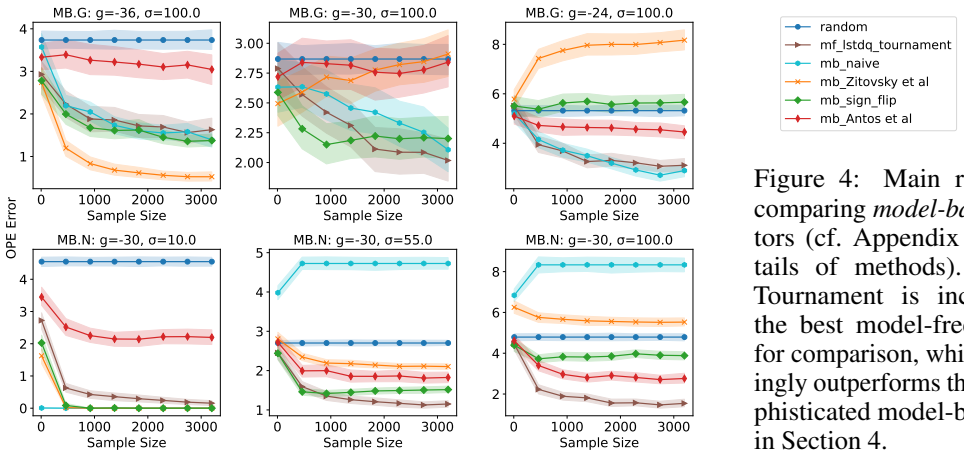

Figure 4: Main results for comparing *model-based* selectors (cf. Appendix B for details of methods). LSTD-Tournament is included as the best model-free selector for comparison, which surprisingly outperforms the more sophisticated model-based ones in Section 4.

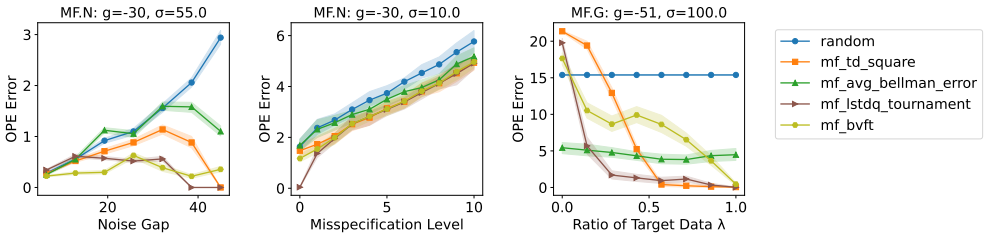

Figure 5: **Left:** OPE error vs. simulator gaps. **Middle:** OPE error vs. misspecification. **Right:** OPE error vs. data coverage.

later subgrid studies in Section 6.2. We then perform a separate experiment with $m = 5$ for the model-based selectors (Figure 4).

Main Results. Figure 3 shows the main model-free results. Our LSTD-Tournament method demonstrates strong and reliable performance. Note that while some methods sometimes outperform it, they suffer catastrophic performances when the true environment changes. For example, the naïve model-based method performs poorly in high-noise environment ($\sigma = 55.0$ and 100.0) when the candidate models have varying degrees of stochasticity (**MF.N**), as predicted by theory (Section 4). BVFT's performance mostly coincides with TD-sq, which is a possible degeneration predicted by [64]. This is particularly plausible when the number of data points n is not large enough to allow for meaningful discretization and partition of the state space required by the method.

Figure 4 shows the result on smaller candidate sets (**MB.G** and **MB.N**; see Appendix D.2), where we implement the three model-based selectors in Section 4 whose computational complexities grow quadratically with $|\mathcal{M}|$. Our expectation was that (1) these algorithms should address the double-sampling issue and will outperform naïve model-based when the latter fails catastrophically, and (2) by having access to more information, model-based should outperform model-free algorithms under realizability. While the first prediction is largely verified, we are surprised to find that the second prediction went wrong, and our LSTD-Tournament method is more robust and generally outperforms the more complicated model-based selectors.

6.2 Subgrid Studies: Gaps and Misspecifications

We now demonstrate how to extract additional insights from the Q-values cached earlier. Due to space limit we are only able to show representative results in Figure 5, and more comprehensive results can be found in Appendix E.

Gaps. We investigate an intellectually interesting question: is the selection problem easier if the candidate simulators are very similar to each other, or when they are very different? We argue that the answer is **neither**, and an intermediate difference (or *gap*) is the most challenging: if the

https://doi.org/10.52202/085713-1430

simulators are too similar, their $J_M(\pi)$ predictions will all be close to $J_{M^\star}(\pi)$ since $M \approx M^\star$, and any selection algorithm will perform well; if the simulators are too dissimilar, it should be easy to tell them apart, which also makes the task easy.

To empirically test this, we let $M^\star = M^7_{\mathbf{n}}$ and run the experiments with different 3-subsets of $\mathcal{M}_{\mathbf{n}}$, including $\{6, 7, 8\}$ (least gap), $\{5, 7, 9\}$, ..., $\{0, 7, 14\}$ (largest gap). Since the needed Q-values have already been cached in the main experiments, we can skip caching and directly run the selection algorithms. We plot the prediction error as a function of gap size in Figure 5L, and observe the down-U curves (except for trivial methods such as random) as expected from earlier intuition.

Misspecification. Similarly, we study the effect of misspecification, i.e., $M^\star \notin \mathcal{M}$. For example, we can take $M^\star = M^0_\sigma$, and consider different subsets of $\mathcal{M}_{\mathbf{n}}$: 0–4 (realizable), 1–5 (low misspecification), ..., 10–14 (high misspecification). Figure 5M plots OPE error vs. misspecification level, where we expect to observe potential difference in the sensitivity to misspecification. The actual result is not that interesting given similar increasing trends for all methods.

6.3 Data Coverage

In the previous subsection, we have seen how multiple experiment units that only differ in $\mathcal{M}$ can provide useful insights. Here we show that we can also probe the methods' sensitivity to data coverage by looking at experiment units that only differ in the dataset $\mathcal{D}$. In Figure 5R, we take a previous experiment setting ($\mathcal{M}_{\mathbf{g}}$) and isolate a particular target policy π; then, we create two datasets: (1) $\mathcal{D}_\pi$ sampled using π; (2) $\mathcal{D}_{\text{off}}$ sampled using a policy that is created to be very different from the target policies and offer very little coverage (see Appendix D.2). Then, we run the algorithm with λ fraction of data from $\mathcal{D}_\pi$ combined with $(1 - \lambda)$ from $\mathcal{D}_{\text{off}}$; as predicted by theory, most methods perform better with better coverage (large λ), and performance degrades as λ goes to 0.

7 Limitations, Future Directions, and Conclusion

We conclude the paper with a discussion of the limitations of our work and potential future directions.

Realism of Candidate Q-functions. While our proposed experimental protocol offers significant advantages in controllability and stability (c.f. Footnote 2), a key limitation lies in the realism of the candidate Q-functions it generates. In practice, these functions will likely come from learning algorithms (e.g., TD/FQE), whose errors may differ from the structured errors induced by our protocol (e.g., varying gravity *uniformly* across states). Varying environment parameters in more complex, state-dependent ways could potentially generate more realistic error patterns, presenting an interesting avenue for future investigation. That said, the prior protocol using FQE may also face realism challenges, just in a different way: in practice, we carefully tune the optimization of OPE algorithms for each target policy to produce reasonable candidates, which can be infeasible in empirical benchmarking given the sheer number of policies we are working with, resulting in poorer candidates. Neither approach perfectly mirrors reality. In addition, our protocol offers controllable quality that enables targeted studies (like the gap experiments in Section 6.2), so for more comprehensive empirical studies, comparing results from both protocols is advisable.

Fundamental Theory of LSTDQ. As discussed in Appendix C.5, our theoretical analysis of LSTD-Tournament reveals limitations and open questions for the coverage assumption made by LSTDQ ($\sigma_{\min}(A)$ in Theorem 1), which is standard in the literature and inherited by LSTD-Tournament. Such conditions differ significantly from the more standard concentrability coefficients (e.g., C^π_∞ in Theorems 3 and 4), and the lack of satisfactory understanding of such a classic algorithm calls for future investigation. We hope to study this question in the future, and any progress would directly improve the guarantees and understanding of LSTD-Tournament.

As another potential future direction, LSTDQ can be viewed as an application of *instrumental-variable* regression in value-function estimation [8, 11], where the left ϕ in the moments of Eq.(3) plays the role of an instrument. While such a choice is standard and natural, it will be interesting to explore alternative choices of instruments and examine if they can lead to improved theoretical guarantees and practical performance in the model-selection problem.

Acknowledgements

Nan Jiang acknowledges funding support from NSF CNS-2112471, NSF CAREER IIS-2141781, Google Scholar Award, and Sloan Fellowship.

References

[1] Alekh Agarwal, Sham Kakade, Akshay Krishnamurthy, and Wen Sun. Flambe: Structural complexity and representation learning of low rank mdps. *arXiv preprint arXiv:2006.10814*, 2020.

[2] Philip Amortila, Nan Jiang, and Tengyang Xie. A variant of the wang-foster-kakade lower bound for the discounted setting. *arXiv preprint arXiv:2011.01075*, 2020.

[3] Philip Amortila, Nan Jiang, and Csaba Szepesvári. The optimal approximation factors in misspecified off-policy value function estimation. In *International Conference on Machine Learning*, pages 768–790. PMLR, 2023.

[4] Philip Amortila, Tongyi Cao, and Akshay Krishnamurthy. Mitigating covariate shift in misspecified regression with applications to reinforcement learning. *arXiv preprint arXiv:2401.12216*, 2024.

[5] Philip Amortila, Dylan J Foster, Nan Jiang, Akshay Krishnamurthy, and Zakaria Mhammedi. Reinforcement learning under latent dynamics: Toward statistical and algorithmic modularity. *arXiv preprint arXiv:2410.17904*, 2024.

[6] András Antos, Csaba Szepesvári, and Rémi Munos. Learning near-optimal policies with Bellman-residual minimization based fitted policy iteration and a single sample path. *Machine Learning*, 2008.

[7] Leemon Baird. Residual algorithms: Reinforcement learning with function approximation. In *Machine Learning Proceedings 1995*, pages 30–37. Elsevier, 1995.

[8] Steven J Bradtke and Andrew G Barto. Linear least-squares algorithms for temporal difference learning. *Machine learning*, 22(1):33–57, 1996.

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

[10] Jinglin Chen and Nan Jiang. Information-theoretic considerations in batch reinforcement learning. In *Proceedings of the 36th International Conference on Machine Learning*, pages 1042–1051, 2019.

[11] Yutian Chen, Liyuan Xu, Caglar Gulcehre, Tom Le Paine, Arthur Gretton, Nando De Freitas, and Arnaud Doucet. On instrumental variable regression for deep offline policy evaluation. *Journal of Machine Learning Research*, 23(302):1–40, 2022.

[12] Ching-An Cheng, Tengyang Xie, Nan Jiang, and Alekh Agarwal. Adversarially trained actor critic for offline reinforcement learning. *International Conference on Machine Learning*, 2022.

[13] Google Deepmind. Mujoco documentation. URL `https://mujoco.readthedocs.io/en/stable/computation/index.html`.

[14] Damien Ernst, Pierre Geurts, and Louis Wehenkel. Tree-based batch mode reinforcement learning. *Journal of Machine Learning Research*, 6:503–556, 2005.

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

[16] Scott Fujimoto, David Meger, Doina Precup, Ofir Nachum, and Shixiang Shane Gu. Why should i trust you, bellman? the bellman error is a poor replacement for value error. *arXiv preprint arXiv:2201.12417*, 2022.

[17] David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018.

 https://doi.org/10.52202/085713-1430

[18] Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

[19] Zeyu Jia, Alexander Rakhlin, Ayush Sekhari, and Chen-Yu Wei. Offline reinforcement learning: Role of state aggregation and trajectory data. *arXiv preprint arXiv:2403.17091*, 2024.

[20] Nan Jiang. *CS 598: Notes on State Abstractions*. University of Illinois at Urbana-Champaign, 2018. `http://nanjiang.cs.illinois.edu/files/cs598/note4.pdf`.

[21] Nan Jiang. A note on loss functions and error compounding in model-based reinforcement learning. *arXiv preprint arXiv:2404.09946*, 2024.

[22] Nan Jiang and Lihong Li. Doubly Robust Off-policy Value Evaluation for Reinforcement Learning. In *Proceedings of the 33rd International Conference on Machine Learning*, volume 48, pages 652–661, 2016.

[23] Nan Jiang and Tengyang Xie. Offline reinforcement learning in large state spaces: Algorithms and guarantees. 2024. `https://nanjiang.cs.illinois.edu/files/STS_Special_Issue_Offline_RL.pdf`.

[24] Nan Jiang, Alex Kulesza, and Satinder Singh. Abstraction Selection in Model-based Reinforcement Learning. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 179–188, 2015.

[25] Nan Jiang, Akshay Krishnamurthy, Alekh Agarwal, John Langford, and Robert E Schapire. Contextual decision processes with low Bellman rank are PAC-learnable. In *International Conference on Machine Learning*, 2017.

[26] Haruka Kiyohara, Ren Kishimoto, Kosuke Kawakami, Ken Kobayashi, Kazuhide Nakata, and Yuta Saito. Scope-rl: A python library for offline reinforcement learning and off-policy evaluation. *arXiv preprint arXiv:2311.18206*, 2023.

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

[28] Aviral Kumar, Anikait Singh, Stephen Tian, Chelsea Finn, and Sergey Levine. A workflow for offline model-free robotic reinforcement learning. *arXiv preprint arXiv:2109.10813*, 2021.

[29] Michail G Lagoudakis and Ronald Parr. Least-squares policy iteration. *The Journal of Machine Learning Research*, 4:1107–1149, 2003.

[30] Alessandro Lazaric, Mohammad Ghavamzadeh, and Rémi Munos. Finite-sample analysis of least-squares policy iteration. *The Journal of Machine Learning Research*, 13(1):3041–3074, 2012.

[31] Hoang Le, Cameron Voloshin, and Yisong Yue. Batch policy learning under constraints. In *International Conference on Machine Learning*, pages 3703–3712, 2019.

[32] Jonathan Lee, George Tucker, Ofir Nachum, and Bo Dai. Model selection in batch policy optimization. In *International Conference on Machine Learning*, pages 12542–12569. PMLR, 2022.

[33] Jonathan N Lee, George Tucker, Ofir Nachum, Bo Dai, and Emma Brunskill. Oracle inequalities for model selection in offline reinforcement learning. *Advances in Neural Information Processing Systems*, 35:28194–28207, 2022.

[34] Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

[35] Lihong Li, Thomas J Walsh, and Michael L Littman. Towards a unified theory of state abstraction for MDPs. In *Proceedings of the 9th International Symposium on Artificial Intelligence and Mathematics*, pages 531–539, 2006.

[36] Lihong Li, Wei Chu, John Langford, and Xuanhui Wang. Unbiased Offline Evaluation of Contextual-bandit-based News Article Recommendation Algorithms. In *Proceedings of the 4th International Conference on Web Search and Data Mining*, pages 297–306, 2011.

[37] Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Manfred Otto Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *CoRR*, abs/1509.02971, 2015. URL `https://api.semanticscholar.org/CorpusID:16326763`.

[38] 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*, pages 5356–5366, 2018.

[39] Qinghua Liu, Praneeth Netrapalli, Csaba Szepesvari, and Chi Jin. Optimistic mle: A generic model-based algorithm for partially observable sequential decision making. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, pages 363–376, 2023.

[40] Vincent Liu, Prabhat Nagarajan, Andrew Patterson, and Martha White. When is offline policy selection sample efficient for reinforcement learning? *arXiv preprint arXiv:2312.02355*, 2023.

[41] Kohei Miyaguchi. Almost hyperparameter-free hyperparameter selection framework for offline policy evaluation. In *AAAI Conference on Artificial Intelligence*, 2022.

[42] Wenlong Mou, Ashwin Pananjady, and Martin J Wainwright. Optimal oracle inequalities for projected fixed-point equations, with applications to policy evaluation. *Mathematics of Operations Research*, 48(4):2308–2336, 2023.

[43] Alfred Müller. Integral probability metrics and their generating classes of functions. *Advances in Applied Probability*, 1997.

[44] Ofir Nachum, Yinlam Chow, Bo Dai, and Lihong Li. Dualdice: Behavior-agnostic estimation of discounted stationary distribution corrections. *Advances in Neural Information Processing Systems*, 32, 2019.

[45] Anusha Nagabandi, Gregory Kahn, Ronald S Fearing, and Sergey Levine. Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 7559–7566. IEEE, 2018.

[46] Allen Nie, Yannis Flet-Berliac, Deon Jordan, William Steenbergen, and Emma Brunskill. Data-efficient pipeline for offline reinforcement learning with limited data. *Advances in Neural Information Processing Systems*, 35:14810–14823, 2022.

[47] 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.

[48] Juan C Perdomo, Akshay Krishnamurthy, Peter Bartlett, and Sham Kakade. A complete characterization of linear estimators for offline policy evaluation. *Journal of Machine Learning Research*, 24(284):1–50, 2023.

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

[50] Touqir Sajed, Wesley Chung, and Martha White. High-confidence error estimates for learned value functions. *arXiv preprint arXiv:1808.09127*, 2018.

[51] Yi Su, Pavithra Srinath, and Akshay Krishnamurthy. Adaptive estimator selection for off-policy evaluation. In *International Conference on Machine Learning*, pages 9196–9205. PMLR, 2020.

[52] Wen Sun, Nan Jiang, Akshay Krishnamurthy, Alekh Agarwal, and John Langford. Model-based RL in Contextual Decision Processes: PAC bounds and Exponential Improvements over Model-free Approaches. In *Conference on Learning Theory*, 2019.

 https://doi.org/10.52202/085713-1430

[53] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[54] Shengpu Tang and Jenna Wiens. Model selection for offline reinforcement learning: Practical considerations for healthcare settings. In *Machine Learning for Healthcare Conference*, pages 2–35. PMLR, 2021.

[55] Philip Thomas and Emma Brunskill. Data-Efficient Off-Policy Policy Evaluation for Reinforcement Learning. In *Proceedings of the 33rd International Conference on Machine Learning*, 2016.

[56] Takuma Udagawa, Haruka Kiyohara, Yusuke Narita, Yuta Saito, and Kei Tateno. Policy-adaptive estimator selection for off-policy evaluation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 10025–10033, 2023.

[57] Masatoshi Uehara, Jiawei Huang, and Nan Jiang. Minimax Weight and Q-Function Learning for Off-Policy Evaluation. In *Proceedings of the 37th International Conference on Machine Learning*, pages 1023–1032, 2020.

[58] Masatoshi Uehara, Xuezhou Zhang, and Wen Sun. Representation learning for online and offline rl in low-rank mdps. *arXiv preprint arXiv:2110.04652*, 2021.

[59] Claas A Voelcker, Arash Ahmadian, Romina Abachi, Igor Gilitschenski, and Amir-massoud Farahmand. λ-ac: Learning latent decision-aware models for reinforcement learning in continuous state-spaces. *arXiv preprint arXiv:2306.17366*, 2023.

[60] Cameron Voloshin, Hoang M Le, Nan Jiang, and Yisong Yue. Empirical study of off-policy policy evaluation for reinforcement learning. *arXiv preprint arXiv:1911.06854*, 2019.

[61] Cameron Voloshin, Nan Jiang, and Yisong Yue. Minimax model learning. In *International Conference on Artificial Intelligence and Statistics*, pages 1612–1620. PMLR, 2021.

[62] Tengyang Xie and Nan Jiang. Batch value-function approximation with only realizability. In *International Conference on Machine Learning*, pages 11404–11413. PMLR, 2021.

[63] Tengyang Xie, Ching-An Cheng, Nan Jiang, Paul Mineiro, and Alekh Agarwal. Bellman-consistent pessimism for offline reinforcement learning. *arXiv preprint arXiv:2106.06926*, 2021.

[64] Siyuan Zhang and Nan Jiang. Towards hyperparameter-free policy selection for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 34:12864–12875, 2021.

[65] Joshua P Zitovsky, Daniel De Marchi, Rishabh Agarwal, and Michael Rene Kosorok. Revisiting bellman errors for offline model selection. In *International Conference on Machine Learning*, pages 43369–43406. PMLR, 2023.

A Other Related Works

Adaptivity Guarantees for Offline RL Model Selection. Our theoretical guarantees state that the selected candidate function/model provides a $J(\pi)$ estimate that is close to the groundtruth, as if the algorithm knew which candidate is correct. This can be viewed as a form of adaptivity guarantees, which is the goal of several theoretical works on model selection. Su et al. [51] and Udagawa et al. [56] study adaptive model selection for OPE, but their approaches crucially rely on the importance sampling estimator, which we do not consider due to the focus on long-horizon tasks. Lee et al. [33] (who build on and improve upon the earlier works of Farahmand and Szepesvári [15] and Jiang et al. [24]) study model-selection of value functions, with a focus on the double-sampling issue and its relationship with Bellman completeness; these are also the very core consideration in our theoretical derivations. However, their approach treats the OPE instances (such as FQE) in a less black-box manner, and the goal is to select the "right" function *class* for the OPE algorithm (instead of directly selecting a final output function) from a nested series of classes, one of which is assumed to be Bellman-complete [6, 10] and have low generalization errors.[5] This makes their approach less widely applicable than ours, and we cannot empirically evaluate their algorithm in our protocol as a consequence. Nevertheless, their results provide an interesting alternative and valuable insights to the model-selection problem, which are "complementary" to the BVFT line of work [62, 64] that we further develop. Similarly, Miyaguchi [41] considers the selection of Bellman operators, which can be instantiated as regression using different function classes, making their approach similar to Lee et al. [33] in spirit.

Other Works on Model Selection. Apart from the above works and those already discussed in the main text, most related works are not concerned about new selection algorithms with theoretical guarantees or experiment protocol for OPE model selection (see Voloshin et al. [60], Kiyohara et al. [26] for experiment protocol and benchmarks of OPE itself), so their focus is different and often provides insights complementary to our work. For example, Nie et al. [46] discuss data splitting in offline model selection; this is a question we avoid by assuming a fixed holdout dataset for OPE model selection. Tang and Wiens [54] compare importance sampling methods and FQE and conclude that FQE is more effective (which echos with Paine et al. [47]), but does not provide provable methods for selecting the hyperparameters of FQE, especially the choice of function approximation. Kumar et al. [28] provide a pipeline for offline RL that includes hyperparameter selection as a component, but the recommendations are heuristics specialized to algorithms such as CQL [27].

Debate on Bellman Error as a Proxy. Most of the selectors we consider estimate some variants of the Bellman error. Regarding this, Fujimoto et al. [16] challenge the idea of using Bellman errors for model selection due to their surrogacy and poor correlation with the actual objective. Despite the valid criticisms, there are no clear alternatives that address the pain points of Bellman errors, and the poor performance is often due to lack of data coverage, which makes the task fundamentally difficult for any algorithm. We still believe that Bellman-error-like objectives (defined in a broad sense, which includes our LSTD-Tournament) are promising for model selection, and the improvement on OPE error (which is what we report in experiments) is the right goal to pursue instead of correlation (which we know could be poor due to the surrogate nature of Bellman errors).

Data Coverage in Model Selection. As mentioned above and demonstrated in our experiments, the lack of data coverage is a key factor that determines the difficulty of the selection tasks. Lee et al. [32] propose feature selection algorithms for offline contextual bandits that account for the different coverage effects of candidate features, but it is unclear how to extend the ideas to MDPs. On a related note, ideas from offline RL training, such as version-space-based pessimism [63], can also be incorporated in our method. This will unlikely improve the accuracy of OPE itself, but may be helpful if we measure performance by how OPE can eventually lead to successful selection of performant policies, which we leave for future investigation.

Experimental Protocol. The closest to our experimental protocol is the work of Sajed et al. [50] who also uses Monte-Carlo rollouts to produce unbiased estimates of value functions on individual states. Other than this point, their work is largely orthogonal to ours as they focus on establishing high-confidence bounds for the estimated value, assuming the simulator is the groundtruth environment,

[5] Given the relationship between Bellman-completeness and bisimulation abstractions [10, Proposition 9], their setting is a natural generalization of Jiang et al. [24]'s setting of selecting a bisimulation from a nested series of state abstractions.

https://doi.org/10.52202/085713-1430

whereas we draw Monte-Carlo trajectories from different simulators to facilitate the model-selection problem.

B Details of Model-based Selectors

Here we expand Section 4 on the model-based setting, i.e., choosing a model from $\{M_i\}_{i\in[m]}$. This is a practical scenario when we have structural knowledge of the system dynamics and can build reasonable simulators, but simulators of complex real-world systems will likely have many design choices and knobs that cannot be set from prior knowledge alone. In some sense, the task is not very different from system identification in control and model learning in model-based RL, except that (1) we focus on a finite and small number of plausible models, instead of a rich and continuous hypothesis class, and (2) the ultimate goal is to perform accurate OPE, and learning the model is only an intermediate step.

Existing Methods. Given the close relation to model learning, a natural approach is to simply minimize the model prediction loss [21]: a candidate model M is scored by

$$\mathbb{E}_{(s,a,s')\sim\mu,\tilde{s}\sim P(\cdot|s,a)}[d(s',\tilde{s})], \tag{12}$$

where s' is in the data and generated according to the real dynamics $P^\star(\cdot|s,a)$, and $\tilde{s}$ is generated from the candidate model M's dynamics P. $d(\cdot,\cdot)$ is a distance metric that measures the difference between states.

Despite its wide use and simplicity [17, 45, 18], the method has major caveats: first, the distance metric $d(\cdot,\cdot)$ is a design choice. When the state is represented as a real-valued vector, it is natural to use the ℓ_2 distance as $d(\cdot,\cdot)$, which changes if we simply normalize/rescale the coordinates. Second, the metric is biased for stochastic environments as discussed in prior works [21, 59], which we also demonstrate in the experiment section (Section 6); essentially this is a version of the double-sampling issue but for the model-based setting [5]. As a minimal counterexample, suppose $\mathcal{S}\subset\mathbb{R}^2$ and we focus on the transition distribution from a particular (s,a) pair where groundtruth is uniform over 4 points $(\pm1,\pm1)$. The expected loss of the groundtruth model itself, as in Eq.(12), is $1+\sqrt{2}/2$; this is higher than a wrong model that always predicts 0 deterministically, which yields a loss of $\sqrt{2}$.

There are alternative methods that address these issues. For example, in the theoretical literature, MLE losses are commonly used, i.e., $\mathbb{E}_\mu[\log P(s'|s,a)]$ [1, 58, 39], which avoids $d(\cdot,\cdot)$ and works properly for stochastic MDPs by effectively measuring the KL divergence between $P^\star(\cdot|s,a)$ and $P(\cdot|s,a)$. Unfortunately, most complex simulators do not provide explicit probabilities $P(s'|s,a)$, making it difficult to use in practice. Moreover, when the support of $P^\star(\cdot|s,a)$ is not fully covered by $P(\cdot|s,a)$, the loss can become degenerate.

To address these issues, we propose to estimate the Bellman error $\mathbb{E}_\mu[(Q_i-\mathcal{T}^\pi Q_i)^2]$, where $Q_i:=Q^\pi_{M_i}$. As discussed earlier, this objective suffers the double-sampling issue in the model-free setting, which we show can be addressed when we have access to candidate models $\{M_1,\dots,M_m\}$ that contains the true dynamics $M^\star$. Moreover, the Bellman error $|Q^\pi_{M_i}(s,a)-(\mathcal{T}^\pi Q^\pi_{M_i})(s,a)|=$

$$\gamma|\mathbb{E}_{s'\sim P^\star(\cdot|s,a)}[Q_i(s',\pi)]-\mathbb{E}_{s'\sim P_i(\cdot|s,a)}[Q_i(s',\pi)]|,$$

which can be viewed as an IPM loss [43] that measures the divergence between $P^\star(\cdot|s,a)$ and $P(\cdot|s,a)$ under $Q_i(\cdot,\pi)$ as a discriminator. IPM is also a popular choice of model learning objective in theory [52, 61], and the Bellman error provides a natural discriminator relevant for the ultimate task of interest, namely OPE.

B.1 Regression-based Selector

Recall that the difficulty in estimating the Bellman error $\mathbb{E}_\mu[(Q_i-\mathcal{T}^\pi Q_i)^2]$ is the uncertainty in $\mathcal{T}^\pi$. To overcome this, we leverage the following observation from Antos et al. [6], where for any $f:\mathcal{S}\times\mathcal{A}\to\mathbb{R}$,

$$\mathcal{T}^\pi f\in\operatorname*{arg\,min}_{g:\mathcal{S}\times\mathcal{A}\to\mathbb{R}}\mathbb{E}_\mu[(g(s,a)-r-\gamma f(s',\pi))^2], \tag{13}$$

which shows that we can estimate $\mathcal{T}^\pi Q_i$ by solving a sample-based version of the above regression problem with $f=Q_i$. Statistically, however, we cannot afford to minimize the objective over

all possible functions g; we can only search over a limited set $\mathcal{G}_i$ that ideally captures the target $\mathcal{T}^\pi Q_i$. Crucially, in the model-based setting we can generate such a set directly from the candidates $\{M_i\}_{i\in[m]}$:

Proposition 1. *Let $\mathcal{G}_i := \{\mathcal{T}^\pi_{M_j} Q_i : j \in [m]\}$. Then if $M^\star \in \{M_i\}_{i\in[m]}$, it follows that $\mathcal{T}^\pi Q_i = \mathcal{T}^\pi_{M^\star} Q_i \in \mathcal{G}_i$.*

The constructed $\mathcal{G}_i$ ensures that regression is statistically tractable given its small cardinality, $|\mathcal{G}_i| = m$. To select Q_i, we choose Q_i with the smallest loss defined as follows:

1. $\widehat{g}_i := \arg\min_{g\in\mathcal{G}_i} \mathbb{E}_{\mathcal{D}}[(g(s,a) - r - \gamma Q_i(s',\pi))^2]$.
2. The loss of Q_i is $\mathbb{E}_{\mathcal{D}}[(\widehat{g}_i(s,a) - Q_i(s,a))^2]$.

The 2nd step follows from Zitovsky et al. [65]. Alternatively, we can also use the min value of Eq.(13) (instead of the argmin function) to correct for the bias in TD-squared (Eq.(2)) [6]; see Liu et al. [40] for another related variant. These approaches share similar theoretical guarantees under standard analyses [62, 63], and we only state the guarantee for the Zitovsky et al. [65] version below, but will include both in the experiments ("mb_Zitovsky et al." and "mb_Antos et al" in Figure 4).

Theorem 3. *Let $\mathcal{C}^\pi := \mathbb{E}_\pi\left[\frac{d^\pi(s,a)}{\mu(s,a)}\right]$. For $Q_{\widehat{i}}$ that minimizes $\mathbb{E}_{\mathcal{D}}[(\widehat{g}_i(s,a) - Q_i(s,a))^2]$ we have w.p. $\geq 1-\delta$,*

$$J(\pi) - \mathbb{E}_{d_0}\left[Q_{\widehat{i}}(s,\pi)\right] \leq \frac{V_{\max}}{1-\gamma}\sqrt{\frac{152\cdot\mathcal{C}^\pi\cdot\log\left(\frac{4m}{\delta}\right)}{n}}. \tag{14}$$

B.2 Sign-flip Average Bellman Error

We now present another selector that leverages the information of $\mathcal{G}_i = \{\mathcal{T}^\pi_{M_j} : j \in [m]\}$ in a different manner. Instead of measuring the squared Bellman error, we can also measure the absolute error, which can be written as (some (s,a) argument to functions are omitted):

$$\begin{aligned}
&\mathbb{E}_\mu[|Q_i - \mathcal{T}^\pi_{M^\star} Q_i|] \\
&= \mathbb{E}_\mu[\text{sgn}(Q_i(s,a) - (\mathcal{T}^\pi Q_i)(s,a))(Q_i - \mathcal{T}^\pi Q_i)] \\
&= \mathbb{E}_\mu[\text{sgn}(Q_i - \mathcal{T}^\pi Q_i)(Q_i(s,a) - r - \gamma Q_i(s',\pi))] \\
&\leq \max_{g\in\mathcal{G}_i} \mathbb{E}_\mu[\text{sgn}(Q_i - g)(Q_i(s,a) - r - \gamma Q_i(s',\pi))].
\end{aligned} \tag{15}$$

Here, the $\mathcal{G}_i$ from Proposition 1 induces a set of sign functions $\text{sgn}(Q_i - g)$, which includes $Q_i - \mathcal{T}^\pi Q_i$, that will negate any negative TD errors. The guarantee is as follows:

Theorem 4. *Let $Q_{\widehat{i}}$ be the minimizer of the empirical estimate of Eq.*(15)*, and $\mathcal{C}^\pi_\infty := \max_{s,a} \frac{d^\pi(s,a)}{\mu(s,a)}$. W.p. $\geq 1-\delta$,*

$$J(\pi) - \mathbb{E}_{d_0}\left[Q_{\widehat{i}}(s,\pi)\right] \leq 4\cdot\mathcal{C}^\pi_\infty\cdot V_{\max}\sqrt{\frac{\log(2m/\delta)}{n}}.$$

C Proofs

C.1 Proof of Theorem 1

Proof. Define the following loss vectors,

$$\ell(\theta) := A\theta - b \in \mathbb{R}^d,$$
$$\widehat{\ell}(\theta) := \widehat{A}\theta - \widehat{b} \in \mathbb{R}^d$$

and recall that we select as the estimator

$$\widehat{\theta} := \arg\min_{\theta\in\Theta} \|\widehat{\ell}(\theta)\|_\infty.$$

https://doi.org/10.52202/085713-1430

Since $\theta^\star = A^{-1}b$, we can write the desired bound as a function of $\ell(\theta)$ as follows,

$$\begin{aligned}
\|Q^\pi(\cdot) - \phi^\top(\cdot)\widehat{\theta}\|_\infty &= \|\phi^\top(\cdot)(\widehat{\theta} - \theta^\star)\|_\infty \\
&= \|\phi^\top(\cdot)A^{-1}(A\widehat{\theta} - b)\|_\infty \\
&= \|\phi^\top(\cdot)A^{-1}\ell(\widehat{\theta})\|_\infty \\
&= \max_{s,a}|\phi^\top(s,a)A^{-1}\ell(\widehat{\theta})| \\
&\leq \left(\max_{s,a}\left\|\phi^\top(s,a)\right\|_2\right)\cdot\left\|A^{-1}\right\|_2\cdot\|\ell(\widehat{\theta})\|_2 \\
&\leq \sqrt{d}B_\phi\cdot\left\|A^{-1}\right\|_2\cdot\|\ell(\widehat{\theta})\|_\infty
\end{aligned}$$

Next, we control the $\ell(\widehat{\theta})$ term. In the sequel we will establish via concentration that

$$\|\ell(\theta) - \widehat{\ell}(\theta)\|_\infty \leq \varepsilon_{\mathsf{stat}} := 3\cdot\max\{R_{\max}, B_\phi\}^2\cdot\sqrt{\frac{\log(2d|\Theta|\delta^{-1})}{n}},\ \forall\theta\in\Theta. \tag{16}$$

Then, we have that

$$\begin{aligned}
\|\ell(\widehat{\theta})\|_\infty &\leq \|\widehat{\ell}(\widehat{\theta})\|_\infty + \varepsilon_{\mathsf{stat}} \\
&\leq \|\widehat{\ell}(\theta^\star)\|_\infty + \varepsilon_{\mathsf{stat}} \\
&\leq \|\ell(\theta^\star)\|_\infty + 2\cdot\varepsilon_{\mathsf{stat}} \\
&= 2\cdot\varepsilon_{\mathsf{stat}},
\end{aligned}$$

where we recall that $\widehat{\theta} = \arg\min_{\theta\in\Theta}\|\widehat{\ell}(\theta)\|_\infty$ in the second inequality, and that $A\theta^\star = b$ in the last line. Combining the above, we obtain

$$\begin{aligned}
\|Q^\pi - \phi^\top\widehat{\theta}\|_\infty &\leq 2\sqrt{d}B_\phi\cdot\left\|A^{-1}\right\|_2\cdot\varepsilon_{\mathsf{stat}} \\
&= 6\sqrt{d}\cdot\left\|A^{-1}\right\|_2\cdot\max\{R_{\max}, B_\phi\}^2\cdot\sqrt{\frac{\log(2d|\Theta|\delta^{-1})}{n}},
\end{aligned}$$

as desired. We now establish the concentration result of Equation (16).

Concentration results. For $j\in[d]$, let $\phi_j(s,a)\in\mathbb{R}$ refer to the j'th entry of the vector. For any (s,a,s') and θ, define

$$B^\pi(s,a,s';\theta) := \phi^\top(s,a)\theta - \gamma\phi^\top(s',\pi)\theta - r(s,a)$$

Recall that $\|\phi(s,a)\|_2\leq B_\phi$ for all (s,a) and that $\|\theta\|_2\leq B_\Theta$ for all $\theta\in\Theta$. We have that, for all $j\in[d]$, $\theta\in\Theta$, and $s,a\in\mathcal{S}\times\mathcal{A}$ we have that $\phi_j(s,a)B^\pi(s,a,s';\theta)$ is bounded, since:

$$\begin{aligned}
&\phi_j(s,a)\big(\phi^\top(s,a)\theta - \gamma\phi^\top(s',\pi)\theta - r(s,a)\big) \\
&\leq \|\phi(s,a)\|_\infty(\|\phi(s,a)\|_2\|\theta\|_2 + \gamma\|\phi(s',\pi)\|_2\|\theta\|_2 + R_{\max}) \\
&\leq \max_{s,a}\|\phi(s,a)\|_2\left(\max_{s,a}\|\phi(s,a)\|_2\|\theta\|_2 + \gamma\max_{s,a}\|\phi(s,a)\|_2\|\theta\|_2 + R_{\max}\right) \\
&\leq (1+\gamma)B_\phi^2 + R_{\max}B_\phi \\
&\leq 3\max\{B_\phi, R_{\max}\}^2.
\end{aligned}$$

Thus, from Hoeffding's inequality and a union bound, we have that for all $j\in[d]$ and $\theta\in\Theta$:

$$\left|\mathbb{E}_\mu\big[\phi^j(s,a)B^\pi(s,a,s';\theta)\big] - \widehat{\mathbb{E}}_\mu\big[\phi^j(s,a)B^\pi(s,a,s';\theta)\big]\right| \leq 3\max\{B_\phi, R_{\max}\}^2\sqrt{\frac{2\log(d|\Theta|\delta^{-1})}{n}} = \varepsilon_{\mathsf{stat}},$$

with probability at least $1-\delta$. As a result, we can write

$$\begin{aligned}
\left\|\ell(\theta) - \widehat{\ell}(\theta)\right\|_\infty &= \left\|\mathbb{E}_\mu\big[\phi(s,a)\big(\phi^\top(s,a)\theta - \gamma\phi^\top(s',\pi)\theta - r(s,a)\big)\big] - \widehat{\mathbb{E}}_\mu\big[\phi(s,a)\big(\phi^\top(s,a)\theta - \gamma\phi^\top(s',\pi)\theta - r(s,a)\big)\big]\right\|_\infty \\
&= \left\|\mathbb{E}_\mu[\phi(s,a)B^\pi(s,a,s';\theta)] - \widehat{\mathbb{E}}_\mu[\phi(s,a)B^\pi(s,a,s';\theta)]\right\|_\infty \\
&\leq \|\mathbf{1}\cdot\varepsilon_{\mathsf{stat}}\|_\infty \\
&\leq \varepsilon_{\mathsf{stat}}.
\end{aligned}$$

This concludes the proof. □

C.2 Proof of Theorem 2

Proof. We first note that the proposed algorithm is equivalent to the following tournament procedure:

- $\forall i \in [m], j \neq i$:
  - Define $\phi_{i,j}(s,a) := [Q_i(s,a), Q_j(s,a)]^\top$ and associated $\hat{A}_{i,j}$ matrix and $\hat{b}_{i,j}$ vector (Eq. 4)
  - Define $\hat{\ell}_{i,j} = \hat{A}_{i,j} e_1 - \hat{b}_{i,j} \in \mathbb{R}^2$
- Pick $\arg\min_{i \in [m]} \max_{j \neq i} \|\hat{\ell}_{i,j}\|_\infty$

Let $i^\star \in [m]$ denote the index of Q^π in the enumeration of $\mathcal{Q}$. We start with the upper bound

$$|J_{M^\star}(\pi) - \mathbb{E}_{s\sim d_0}[Q_{\hat{i}}(s,\pi)]| = |\mathbb{E}_{s\sim d_0}[Q^\pi(s,\pi)] - \mathbb{E}_{s\sim d_0}[Q_{\hat{i}}(s,\pi)]| \leq \|Q^\pi(\cdot) - Q_{\hat{i}}(\cdot)\|_\infty.$$

Let $\ell_{i,j} \coloneqq A_{i,j} e_1 - b_{i,j}$ denote the population loss. We recall the concentration result from Equation (16), which, for any fixed i and j, implies:

$$\|\ell_{i,j} - \widehat{\ell}_{i,j}\|_\infty \leq \varepsilon_{\mathsf{stat}} = 3 \cdot \max\{B_\phi, R_{\max}\}^2 \cdot \sqrt{\frac{\log(2d\delta^{-1})}{n}},$$

with probability at least $1-\delta$. This further implies $|\|\ell_{i,j}\|_\infty - \|\hat{\ell}_{i,j}\|_\infty| \leq \varepsilon_{\mathsf{stat}}$. Taking a union bound over all (i,j) where either i or j equal $i^\star$, this implies that

$$\|\ell_{i,j} - \widehat{\ell}_{i,j}\|_\infty \leq \varepsilon_{\mathsf{stat}} = 3\cdot\max\{B_\phi, R_{\max}\}^2 \cdot \sqrt{\frac{\log(4dm\delta^{-1})}{n}} \quad \forall (i,j) \in ([m] \times \{i^\star\}) \cup (\{i^\star\} \times [m])$$

If $\hat{i} = i^\star$ then we are done. If not, then there exists a comparison in the tournament where $i = \hat{i}$ and $j = i^\star$. For these features $\phi_{\hat{i},i^\star}(s,a) = [Q_{\hat{i}}(s,a), Q_{i^\star}(s,a)]^\top$, we have:

$$\begin{aligned}
\|Q^\pi(\cdot) - Q_{\hat{i}}(\cdot)\|_\infty &= \|\phi_{\hat{i},i^\star}^\top (e_2 - e_1)\|_\infty \\
&= \|\phi_{\hat{i},i^\star}^\top A_{\hat{i},i^\star}^{-1} A_{\hat{i},i^\star} (e_2 - e_1)\|_\infty \\
&= \max_{s,a} |\phi_{\hat{i},i^\star}^\top(s,a) A_{\hat{i},i^\star}^{-1} A_{\hat{i},i^\star}(e_2 - e_1)| \\
&\leq \left(\max_{s,a} \|\phi_{\hat{i},i^\star}(s,a)\|_2\right) \|A_{\hat{i},i^\star}^{-1}\|_2 \|A_{\hat{i},i^\star}(e_2 - e_1)\|_2 \\
&\leq \sqrt{d}\left(\max_{s,a} \|\phi_{\hat{i},i^\star}(s,a)\|_2\right) \|A_{\hat{i},i^\star}^{-1}\|_2 \|A_{\hat{i},i^\star}(e_2 - e_1)\|_\infty \\
&= \sqrt{d}\left(\max_{s,a} \|\phi_{\hat{i},i^\star}(s,a)\|_2\right) \|A_{\hat{i},i^\star}^{-1}\|_2 \|A_{\hat{i},i^\star} e_1 - b_{\hat{i},i^\star}\|_\infty \\
&= \sqrt{d}\left(\max_{s,a} \|\phi_{\hat{i},i^\star}(s,a)\|_2\right) \|A_{\hat{i},i^\star}^{-1}\|_2 \|\ell_{\hat{i},i^\star}\|_\infty \\
&\leq \sqrt{d}\left(\max_{s,a} \|\phi_{\hat{i},i^\star}(s,a)\|_2\right) \|A_{\hat{i},i^\star}^{-1}\|_2 \left(\|\hat{\ell}_{\hat{i},i^\star}\|_\infty + \varepsilon_{\mathsf{stat}}\right) \\
&\leq \sqrt{d}\left(\max_{s,a} \|\phi_{\hat{i},i^\star}(s,a)\|_2\right) \|A_{\hat{i},i^\star}^{-1}\|_2 \left(\max_{j \in [m]\setminus\{i^\star\}} \|\hat{\ell}_{\hat{i},j}\|_\infty + \varepsilon_{\mathsf{stat}}\right) \\
&\leq \sqrt{d}\left(\max_{s,a} \|\phi_{\hat{i},i^\star}(s,a)\|_2\right) \|A_{\hat{i},i^\star}^{-1}\|_2 \left(\max_{j \in [m]\setminus\{i^\star\}} \|\hat{\ell}_{i^\star,j}\|_\infty + \varepsilon_{\mathsf{stat}}\right) \\
&\leq \sqrt{d}\left(\max_{s,a} \|\phi_{\hat{i},i^\star}(s,a)\|_2\right) \|A_{\hat{i},i^\star}^{-1}\|_2 \left(\max_{j \in [m]\setminus\{i^\star\}} \|\ell_{i^\star,j}\|_\infty + 2\varepsilon_{\mathsf{stat}}\right) \\
&\leq 2\sqrt{d}\left(\max_{s,a} \|\phi_{\hat{i},i^\star}(s,a)\|_2\right) \|A_{\hat{i},i^\star}^{-1}\|_2 \varepsilon_{\mathsf{stat}}. \\
&\leq 2\sqrt{d}\left(\max_{s,a} \|\phi_{\hat{i},i^\star}(s,a)\|_2\right) \max_{i \in [m]\setminus\{i^\star\}} \frac{1}{\sigma_{\min}(A_{i,i^\star})} \varepsilon_{\mathsf{stat}}.
\end{aligned}$$

https://doi.org/10.52202/085713-1430

To conclude, we note that $d = 2$ in our application and that $\max_{s,a} \|\phi_{i,j}(s,a)\|_2^2 = Q_i^2(s,a) + Q_j^2(s,a) \leq 2V_{\max}^2$. Plugging in the value for $\varepsilon_{\mathsf{stat}}$, this gives a final bound of

$$\begin{aligned}|J_{M^\star}(\pi) - \mathbb{E}_{s\sim d_0}[Q_{\hat{i}}(s,\pi)]| &\leq 4V_{\max} \max_{i\in[m]\setminus\{i^\star\}} \frac{1}{\sigma_{\min}(A_{i,i^\star})}\varepsilon_{\mathsf{stat}} \\ &= 24V_{\max}^3 \max_{i\in[m]\setminus\{i^\star\}} \frac{1}{\sigma_{\min}(A_{i,i^\star})}\sqrt{\frac{\log(4dm\delta^{-1})}{n}}.\end{aligned}$$

□

C.3 Proof of Theorem 3

We bound

$$\begin{aligned}J(\pi) - \mathbb{E}_{d_0}\left[\widehat{Q}(s,\pi)\right] &= \mathbb{E}_{d_0,\pi}\left[Q^\pi(s,a) - \widehat{Q}(s,a)\right] \\ &= \frac{1}{1-\gamma}\,\mathbb{E}_{d^\pi}\left[Q^\pi(s,a) - \gamma Q^\pi(s',\pi) - \widehat{Q}(s,a) - \gamma\widehat{Q}(s',\pi)\right] \\ &= \frac{1}{1-\gamma}\,\mathbb{E}_{d^\pi}\left[Q^\pi(s,a) - [\mathcal{T}^\pi Q^\pi](s,a) - \widehat{Q}(s,a) + \left[\mathcal{T}^\pi\widehat{Q}\right](s,a)\right] \\ &= \frac{1}{1-\gamma}\,\mathbb{E}_{d^\pi}\left[\left[\mathcal{T}^\pi\widehat{Q}\right](s,a) - \widehat{Q}(s,a)\right] \\ &\leq \frac{1}{1-\gamma}\sqrt{\mathcal{C}^\pi \cdot \mathbb{E}_\mu\left[\left(\left[\mathcal{T}^\pi\widehat{Q}\right](s,a) - \widehat{Q}(s,a)\right)^2\right]}\end{aligned}$$

where the second line follows from Bellman flow. Now we consider the term under the square root, and let $\widehat{g}_{\widehat{Q}} = \arg\min_{g\in\mathcal{G}_{\widehat{Q}}} \widehat{\ell}(g,\widehat{Q})$.

$$\mathbb{E}_\mu\left[\left(\left[\mathcal{T}^\pi\widehat{Q}\right](s,a) - \widehat{Q}(s,a)\right)^2\right] \leq 2\cdot\underbrace{\mathbb{E}_\mu\left[\left(\left[\mathcal{T}^\pi\widehat{Q}\right](s,a) - \widehat{g}_{\widehat{Q}}(s,a)\right)\right]^2}_{\text{(T1)}} + 2\cdot\underbrace{\mathbb{E}_\mu\left[\left(\widehat{g}_{\widehat{Q}}(s,a) - \widehat{Q}(s,a)\right)^2\right]}_{\text{(T2)}}$$

We consider each term above individually. $(T1)$ is the regression error between $\widehat{g}_Q$ and the population regression solution $\mathcal{T}^\pi Q$, which we can control using well-established bounds. The second term $(T2)$ measure how close the Q-value is to its estimated Bellman backup. To bound these two terms we utilize the following results. The first controls the error between the squared-loss minimizer $\widehat{g}_Q$ and the population solution $\mathcal{T}^\pi Q$, and is adapted from [62].

Lemma 1 (Lemma 9 from [62]). *Suppose that we have $|g|_\infty \leq V_{\max}$ for all $g \in \mathcal{G}_Q$ and $Q \in \mathcal{Q}$, and define*

$$\widehat{g}_Q := \arg\min_{g\in\mathcal{G}_Q} \mathbb{E}_{\mathcal{D}}\left[\left(g(s,a) - r - \gamma Q(s',\pi)\right)^2\right].$$

Then with probability at least $1-\delta$, for all $i \in [m]$ we have

$$\mathbb{E}_\mu\left[\left(\widehat{g}_Q(s,a) - [\mathcal{T}^\pi Q](s,a)\right)^2\right] \leq \frac{16V_{\max}^2\log\left(\frac{2m}{\delta}\right)}{n} := \varepsilon_{\mathsf{reg}}^2.$$

The second controls the error of estimating the objective for choosing $\widehat{i}$ from finite samples, and a proof is included at the end of this section.

Lemma 2 (Objective estimation error). *Suppose that we have $\|g\|_\infty \leq V_{\max}$ for all $g \in \mathcal{G}_Q$ and $Q \in \mathcal{Q}$. Then with probability at least $1-\delta$, for all $g \in \mathcal{G}_Q$ and $Q \in \mathcal{Q}$ we have*

$$\max\left\{\frac{1}{2}\cdot\mathbb{E}_\mu\left[\left(g(s,a) - Q(s,a)\right)^2\right] - \mathbb{E}_{\mathcal{D}}\left[\left(g(s,a) - Q(s,a)\right)^2\right],\right.$$
$$\left.\mathbb{E}_{\mathcal{D}}\left[\left(g(s,a) - Q(s,a)\right)^2\right] - \frac{3}{2}\cdot\mathbb{E}_\mu\left[\left(g(s,a) - Q(s,a)\right)^2\right]\right\} \leq \frac{3V_{\max}^2\log\left(\frac{2m}{\delta}\right)}{n} := \varepsilon_{\mathsf{obj}}.$$

Using Lemma 1, we directly obtain that with probability at $1-\delta$,

$$(\mathrm{T1}) \leq \varepsilon_{\mathsf{reg}}^2.$$

By leveraging Lemma 2, we have that with probability at least $1-\delta$,

$$\begin{aligned}
(\mathrm{T2}) &= \mathbb{E}_\mu\left[\left(\widehat{g}_{\widehat{Q}}(s,a) - \widehat{Q}(s,a)\right)^2\right] \\
&\leq 2\cdot\varepsilon_{\mathsf{obj}} + 2\cdot\mathbb{E}_{\mathcal{D}}\left[\left(\widehat{g}_{\widehat{Q}}(s,a) - \widehat{Q}(s,a)\right)^2\right] \\
&\leq 2\cdot\varepsilon_{\mathsf{obj}} + 2\cdot\mathbb{E}_{\mathcal{D}}\left[\left(\widehat{g}_{Q^\pi}(s,a) - Q^\pi(s,a)\right)^2\right] \\
&\leq 4\cdot\varepsilon_{\mathsf{obj}} + 3\cdot\mathbb{E}_{\mu}\left[\left(\widehat{g}_{Q^\pi}(s,a) - Q^\pi(s,a)\right)^2\right] \\
&= 4\cdot\varepsilon_{\mathsf{obj}} + 3\cdot\mathbb{E}_{\mu}\left[\left(\widehat{g}_{Q^\pi}(s,a) - [\mathcal{T}^\pi Q^\pi](s,a)\right)^2\right] \\
&\leq 4\cdot\varepsilon_{\mathsf{obj}} + 3\cdot\varepsilon_{\mathsf{reg}}^2
\end{aligned}$$

where in the first inequality we apply Lemma 2 (by lower bounding the LHS with the first expression in the $\max$); in the second we use the Q-value realizability assumption $Q^\pi \in \mathcal{Q}$ with the fact that $\widehat{Q}$ is the minimizer of the empirical objective; and in the third we again apply Lemma 2 (now lower bounding the LHS with the second expression in the $\max$). Then we use the identity that $Q^\pi = \mathcal{T}^\pi Q^\pi$, and apply the squared-loss regression guarantee. The bounds for (T1) and (T2) mean that

$$\mathbb{E}_\mu\left[\left(\left[\mathcal{T}^\pi\widehat{Q}\right](s,a) - \widehat{Q}(s,a)\right)^2\right] \leq 8\big(\varepsilon_{\mathsf{obj}} + \varepsilon_{\mathsf{reg}}^2\big),$$

resulting in the final estimation bound of

$$\begin{aligned}
J(\pi) - \mathbb{E}_{d_0}\left[\widehat{Q}(s,\pi)\right] &\leq \frac{1}{1-\gamma}\sqrt{\mathcal{C}^\pi\cdot\mathbb{E}_\mu\left[\left(\left[\mathcal{T}^\pi\widehat{Q}\right](s,a) - \widehat{Q}(s,a)\right)^2\right]} \\
&\leq \frac{1}{1-\gamma}\sqrt{8\cdot\mathcal{C}^\pi\cdot\big(\varepsilon_{\mathsf{obj}} + \varepsilon_{\mathsf{reg}}^2\big)}, \\
&= \frac{V_{\max}}{1-\gamma}\sqrt{\frac{152\cdot\mathcal{C}^\pi\cdot\log\big(\frac{2m}{\delta}\big)}{n}},
\end{aligned}$$

which holds with probability at least $1-2\delta$.

Proof of Lemma 2. Observe that the random variable $(g(s,a)-Q(s,a))^2 \in [-V_{\max}^2, V_{\max}^2]$, and

$$\begin{aligned}
\mathbb{V}_\mu\left[(g(s,a)-Q(s,a))^2\right] &\leq \mathbb{E}_\mu\left[(g(s,a)-Q(s,a))^4\right] \\
&\leq V_{\max}^2\cdot\mathbb{E}_\mu\left[(g(s,a)-Q(s,a))^2\right].
\end{aligned}$$

Then, applying Bernstein's inequality with union bound, we have that, for any $g\in\mathcal{G}_Q$ and $Q\in\mathcal{Q}$ with probability at least $1-\delta$,

$$\begin{aligned}
&\left|\mathbb{E}_\mu\left[(g(s,a)-Q(s,a))^2\right] - \mathbb{E}_{\mathcal{D}}\left[(g(s,a)-Q(s,a))^2\right]\right| \\
&\leq \sqrt{\frac{4\mathbb{V}_\mu\left[(g(s,a)-Q(s,a))^2\right]\log\big(\frac{2m}{\delta}\big)}{n}} + \frac{V_{\max}^2\log\big(\frac{2m}{\delta}\big)}{n} \\
&\leq \sqrt{\frac{4V_{\max}^2\mathbb{E}_\mu\left[(g(s,a)-Q(s,a))^2\right]\log\big(\frac{2m}{\delta}\big)}{n}} + \frac{V_{\max}^2\log\big(\frac{2m}{\delta}\big)}{n} \\
&\leq \frac{\mathbb{E}_\mu\left[(g(s,a)-Q(s,a))^2\right]}{2} + \frac{3V_{\max}^2\log\big(\frac{2m}{\delta}\big)}{n}.
\end{aligned}$$

https://doi.org/10.52202/085713-1430

Expanding the absolute value on the LHS and rearranging, this then implies that

$$\frac{1}{2} \cdot \mathbb{E}_\mu\Big[\big(g(s,a) - Q(s,a)\big)^2\Big] \leq \mathbb{E}_\mathcal{D}\Big[\big(g(s,a) - Q(s,a)\big)^2\Big] + \frac{3V_{\max}^2 \log\left(\frac{2m}{\delta}\right)}{n},$$
$$\mathbb{E}_\mathcal{D}\Big[\big(g(s,a) - Q(s,a)\big)^2\Big] \leq \frac{3}{2} \cdot \mathbb{E}_\mu\Big[\big(g(s,a) - Q(s,a)\big)^2\Big] + \frac{3V_{\max}^2 \log\left(\frac{2m}{\delta}\right)}{n}.$$

Combining these statements completes the proof. □

C.4 Proof of Theorem 4

We bound

$$\begin{aligned}
J(\pi) - \mathbb{E}_{d_0}\Big[\widehat{Q}(s,\pi)\Big] &= \mathbb{E}_{d_0,\pi}\Big[Q^\pi(s,a) - \widehat{Q}(s,a)\Big] \\
&= \frac{1}{1-\gamma}\,\mathbb{E}_{d^\pi}\Big[Q^\pi(s,a) - \gamma Q^\pi(s',\pi) - \widehat{Q}(s,a) - \gamma\widehat{Q}(s',\pi)\Big] \\
&= \frac{1}{1-\gamma}\,\mathbb{E}_{d^\pi}\Big[Q^\pi(s,a) - [\mathcal{T}^\pi Q^\pi](s,a) - \widehat{Q}(s,a) + \Big[\mathcal{T}^\pi\widehat{Q}\Big](s,a)\Big] \\
&= \frac{1}{1-\gamma}\,\mathbb{E}_{d^\pi}\Big[\Big[\mathcal{T}^\pi\widehat{Q}\Big](s,a) - \widehat{Q}(s,a)\Big] \\
&\leq \frac{\mathcal{C}^\pi_\infty}{1-\gamma} \cdot \mathbb{E}_\mu\Big[\Big|\Big[\mathcal{T}^\pi\widehat{Q}\Big](s,a) - \widehat{Q}(s,a)\Big|\Big] \\
&\leq \max_{g \in \mathcal{G}_{\widehat{Q}}} \mathbb{E}_\mu\Big[\operatorname{sgn}\Big(\widehat{Q}(s,a) - g(s,a)\Big)\Big(\widehat{Q}(s,a) - r - \gamma\widehat{Q}(s',\pi)\Big)\Big]
\end{aligned}$$

By assumption, $\max_{q\in\mathcal{Q}}\|q\|_\infty \leq V_{\max}$, and similarly $\max_{g\in\mathcal{G}_Q}\|g\|_\infty \leq V_{\max}$ for all $Q \in \mathcal{Q}$. Then for any $Q \in \mathcal{Q}$ and $g \in \mathcal{G}_Q$ and $(s,a) \in \mathcal{S}\times\mathcal{A}$ and $r \in [0, R_{\max}]$,

$$\operatorname{sgn}(Q(s,a) - g(s,a))(Q(s,a) - r - \gamma Q(s',\pi)) \in [-V_{\max}, V_{\max}],$$

and, using Hoeffding's inequality, we have for all $Q \in \mathcal{Q}$ and $g \in \mathcal{G}_Q$ that, with probability at least $1-\delta$,

$$\begin{aligned}
&\Big|\,\mathbb{E}_\mu\Big[\operatorname{sgn}\Big(\widehat{Q}(s,a) - g(s,a)\Big)\Big(\widehat{Q}(s,a) - r - \gamma\widehat{Q}(s',\pi)\Big)\Big] \\
&\quad - \mathbb{E}_\mathcal{D}\Big[\operatorname{sgn}\Big(\widehat{Q}(s,a) - g(s,a)\Big)\Big(\widehat{Q}(s,a) - r - \gamma\widehat{Q}(s',\pi)\Big)\Big]\Big| \leq 2V_{\max}\sqrt{\frac{\log\left(\frac{2m}{\delta}\right)}{n}} := \varepsilon_{\mathsf{obj}}.
\end{aligned}$$

Then using this concentration in the last line of the previous block,

$$\begin{aligned}
J(\pi) - \mathbb{E}_{d_0}\Big[\widehat{Q}(s,\pi)\Big] &\leq \max_{g\in\mathcal{G}_{\widehat{Q}}} \mathbb{E}_\mathcal{D}\Big[\operatorname{sgn}\Big(\widehat{Q}(s,a) - g(s,a)\Big)\Big(\widehat{Q}(s,a) - r - \gamma\widehat{Q}(s',\pi)\Big)\Big] + \varepsilon_{\mathsf{obj}} \\
&\leq \max_{g\in\mathcal{G}_{Q^\pi}} \mathbb{E}_\mathcal{D}[\operatorname{sgn}(Q^\pi(s,a) - g(s,a))(Q^\pi(s,a) - r - \gamma Q^\pi(s',\pi))] + \varepsilon_{\mathsf{obj}} \\
&\leq \max_{g\in\mathcal{G}_{Q^\pi}} \mathbb{E}_\mu[\operatorname{sgn}(Q^\pi(s,a) - g(s,a))(Q^\pi(s,a) - r - \gamma Q^\pi(s',\pi))] + 2\cdot\varepsilon_{\mathsf{obj}} \\
&= \max_{g\in\mathcal{G}_{Q^\pi}} \mathbb{E}_\mu[\operatorname{sgn}(Q^\pi(s,a) - g(s,a))(Q^\pi(s,a) - [\mathcal{T}^\pi Q^\pi](s,a))] + 2\cdot\varepsilon_{\mathsf{obj}} \\
&= \max_{g\in\mathcal{G}_{Q^\pi}} \mathbb{E}_\mu[\operatorname{sgn}(Q^\pi(s,a) - g(s,a))(Q^\pi(s,a) - Q^\pi(s,a))] + 2\cdot\varepsilon_{\mathsf{obj}} \\
&= 2\cdot\varepsilon_{\mathsf{obj}},
\end{aligned}$$

where in the first and third inequalities we apply the above concentration inequality, and in the second inequality we use the fact that $\widehat{Q}$ is the minimizer of the empirical objective, i.e.,

$$\widehat{Q} = \arg\min_{Q\in\mathcal{Q}} \max_{g\in\mathcal{G}_Q} \mathbb{E}_\mathcal{D}[\operatorname{sgn}(Q(s,a) - g(s,a))(Q(s,a) - r - Q(s',\pi))].$$

Combining the above inequalities, we obtain the theorem statement,

$$J(\pi) - \mathbb{E}_{d_0}\Big[\widehat{Q}(s,\pi)\Big] \leq 2\cdot\mathcal{C}^\pi_\infty\cdot\varepsilon_{\mathsf{obj}}.$$

C.5 Further Discussion on Comparison to BVFT and the Coverage Assumptions

Besides the difference in rates, the guarantees for LSTD-Tournament and BVFT differ in their coverage assumptions, which also differ from those for the model-based selectors (Theorems 3 and 4). In fact, Theorems 3 and 4 use the standard "concentrability coefficient" $\mathcal{C}^{\pi}_{\infty}$ that is widely adopted in offline RL theory [6, 10]. In contrast, LSTD-Tournament's coverage parameter is $\max_{i\in[m]\setminus\{i^\star\}} \frac{1}{\sigma_{\min}(A_{i,i^\star})}$, which is a highly specialized notion of coverage inherited from that of LSTDQ ($\sigma_{\min}(A)$ from Theorem 1). Similarly, the coverage parameter of BVFT is also a notion of coverage (called "aggregated concentrability" [62, 19]) specialized to state abstractions. As far as we know there is no definitive relationship between the coverage parameters of LSTD-Tournament and BVFT and neither dominates the other. Furthermore, these non-standard, highly algorithm-specific coverage definitions are a common situation for algorithms that only require realizability (instead of Bellman-completeness) as their expressivity condition, as discussed in Jiang and Xie [23].

Problems in $\sigma_{\min}(A)$ as LSTDQ's Coverage Assumption. As mentioned above, the coverage condition of LSTD-Tournament is inherited from its "base" algorithm, i.e., the $\sigma_{\min}(A)$ term in LSTDQ. Despite its wide use in the analyses of LSTDQ algorithms [30, 3, 48], we discover several problems with this definition that warrants further investigation. First, the quantity is not invariant to reparameterization, in the sense that if we scale the linear features ϕ with a constant (and scale θ accordingly), $\sigma_{\min}(A)$ will also change despite that the estimation problem essentially remains the same. Second, if ϕ has linearly dependent coordinates, $\sigma_{\min}(A)$ can be 0 while this says nothing about the quality of data (which could have perfect coverage); this is usually not a concern in LSTDQ analysis since we can always pre-process the features, but for its application in LSTD-Tournament the lack of linear independence can happen if we have nearly identical candidate functions $Q_i \approx Q_j$. Last but not least, we lack good understanding of when $\sigma_{\min}(A)$ is well-behaved outside very special cases; for example, it is well-known that fully on-policy data (in the sense that μ is invariant to the transition dynamics under π) implies lower-bounded $\sigma_{\min}(A)$. However, as soon as μ is off-policy, there lacks general characterization of what kind of μ helps lower-bound $\sigma_{\min}(A)$. For example, when concentrability coefficient $\mathcal{C}^{\pi}_{\infty}$ is well bounded, it is unclear whether that implies lower-bounded $\sigma_{\min}(A)$.

In this project we have explored ways to mitigate these issues, such as defining coverage as $\sigma_{\min}(\Sigma)/\sigma_{\min}(A)$ to avoid the linear-dependence issue. However, none of the resolutions we tried solve all problems elegantly. This question is, in fact, quite orthogonal to the central message of our work and of separate interest in the fundamental theory of RL, and speaks to how algorithms like LSTDQ are seriously under-investigated. In fact, coverage in state abstractions has been similarly under-investigated and has only seen interesting progress recently [19]. We wish to investigate the question for LSTDQ in the future, and any progress there can be directly inherited to improve the analysis of LSTD-Tournament.

D Experiment Details

D.1 Environment Setup: Noise and State Resetting

State Resetting. Monte-Carlo rollouts for Q-value estimation rely on the ability to (re)set the simulator to a particular state from the offline dataset. To the best of our knowledge, Mujoco environment does not natively support state resetting, and assigning values to the observation vector does not really change the underlying state. However, state resetting can still be implemented by manually assigning the values of the position vector `qpos` and the velocity vector `qvel`.

Noise. As mentioned in Section 6, we add noise to Hopper to create more challenging stochastic environments and create model selection tasks where candidate simulators have different levels of stochasticity. Here we provide the details about how we inject randomness into the deterministic dynamics of Hopper. Mujoco engine realizes one-step transition by leveraging `mjData.{ctrl, qfrc_applied, xfrc_applied}` objects [13], where `mjData.ctrl` corresponds to the action taken by our agent, and `mjData.{qfrc_applied, xfrc_applied}` are the user-defined perturbations in the joint space and Cartesian coordinates, respectively. To inject randomness into the transition at a noise level of σ, we first sample an isotropic Gaussian noise with variance σ^2 as the stochastic force in `mjData.xfrc[:3]` upon each transition, which jointly determines the next state with the input action `mjData.ctrl`, leaving the joint data `mjData.qfrc` intact.

 https://doi.org/10.52202/085713-1430

D.2 Experiment Settings

MF/MB.G/N. The settings of different experiments are summarized in Table 1. We first run DDPG in the environment of $g = -30, \sigma = 32$, and obtain 15 deterministic policies $\{\pi_{0:14}\}$ from the checkpoints. The first 10 are used as target policies in MF.G/N experiments, and MB.G/N use fewer due to the high computational cost. For the main results (Section 6.1), the choice of $M^\star$ is usually the two ends plus the middle point of the grid ($\mathcal{M}_{\mathbf{g}}$ or $\mathcal{M}_{\mathbf{n}}$). The corresponding behavior policy is an epsilon-greedy version of one of the target policies, denoted as π_i^ϵ, which takes the deterministic action of $\pi_i(s)$ with probability 0.7, and add a unit-variance Gaussian noise to $\pi_i(s)$ with the remaining 0.3 probability.

MF/MB.Off.G/N. In the above setup, the behavior and the target policies all stem from the same DDPG training procedure. While these policies still have significant differences (see Figure 2L), the distribution shift is relatively mild. For the data coverage experiments (Section 6.3), we prepare a different set of behavior policies that intentionally offer poor coverage: these policies, denoted as π_i^{poor}, are obtained by running DDPG with a different neural architecture (than the one used for generating $\pi_{0:14}$) in a different environment of $g = -60, \sigma = 100$. We also provide the parallel of our main experiments in Figures 3 and 4 under these behavior policies with poor coverage in Appendix E.1.

MF.T.G. This experiment is for data coverage (Section 6.3), where $\mathcal{D}$ is a mixture of two datasets, one sampled from π_7 (which is the sole target policy being considered) and one from π_i^{poor} that has poor coverage. They are mixed together under different ratios as explained in Section 6.3.

	Gravity g	**Noise Level σ**	**Groundtruth Model $M^\star$ and Behavior Policy π_b**	**Target Policies Π**
MF.G	LIN$(-51, -9, 15)$	100	$\{(M_i, \pi_i^\epsilon), i \in \{0, 7, 14\}\}$	$\{\pi_{0:9}\}$
MF.N	-30	LIN$(10, 100, 15)$	$\{(M_i, \pi_i^\epsilon), i \in \{0, 7, 14\}\}$	$\{\pi_{0:9}\}$
MB.G	LIN$(-36, -24, 5)$	100	$\{(M_i, \pi_i^\epsilon), i \in \{0, 2, 4\}\}$	$\{\pi_{0:5}\}$
MB.N	-30	LIN$(10, 100, 5)$	$\{(M_i, \pi_i^\epsilon), i \in \{0, 2, 4\}\}$	$\{\pi_{0:5}\}$
MF.OFF.G	LIN$(-51, -9, 15)$	100	$\{(M_i, \pi_i^{\text{poor}}), i \in \{0, 7, 14\}\}$	$\{\pi_{0:9}\}$
MF.OFF.N	-30	LIN$(10, 100, 15)$	$\{(M_i, \pi_i^{\text{poor}}), i \in \{0, 7, 14\}\}$	$\{\pi_{0:9}\}$
MF.T.G	LIN$(-51, -9, 15)$	100	$\{(M_i, \pi_8 \,\&\, \pi_i^{\text{poor}}), i \in \{0, 7, 14\}\}$	$\{\pi_8\}$

Table 1: Details of experiment settings. LIN(a, b, n) (per numpy convention) refers to the arithmetic sequence with n elements, starting from a and ending in b (e.g. LIN$(0, 1, 6) = \{0, 0.2, 0.4, 0.6, 0.8, 1.0\}$).

D.3 Computational Resources

We here provide a brief discussion of the amount of computation used to produce the results. The major cost is in Q-caching, i.e., rolling out trajectories. The cost consists of two parts: environment simulation steps (CPU-intensive) and neural-net inference for policy calls (GPU-intensive). The main experiment on **MF.G**, took nearly a week on a 4090 PC. The runtimes of **MF.N**, **MB.G**, **MB.N** are comparable to **MF.G**.

E Additional Experiment Results

Subgrid Studies. Figures 7 and 6 show more complete results for investigating the sensitivity to misspecification and gaps in Section 6.2 across 4 settings (good/poor coverage and gravity/noise grid).

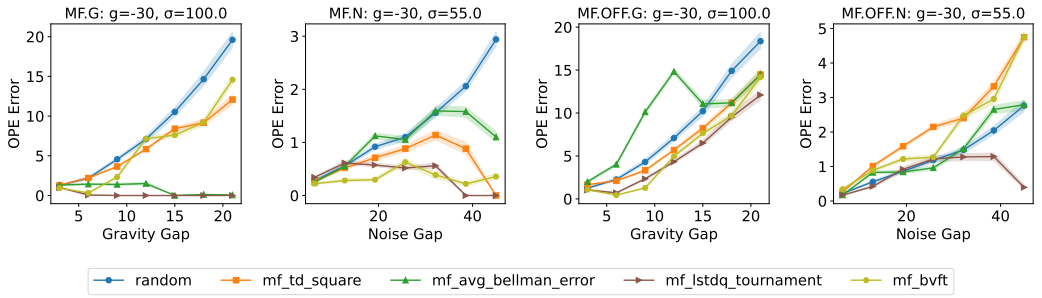

Figure 6: Subgrid studies for gaps. Plot **MF.N** is identical to Figure 5L.

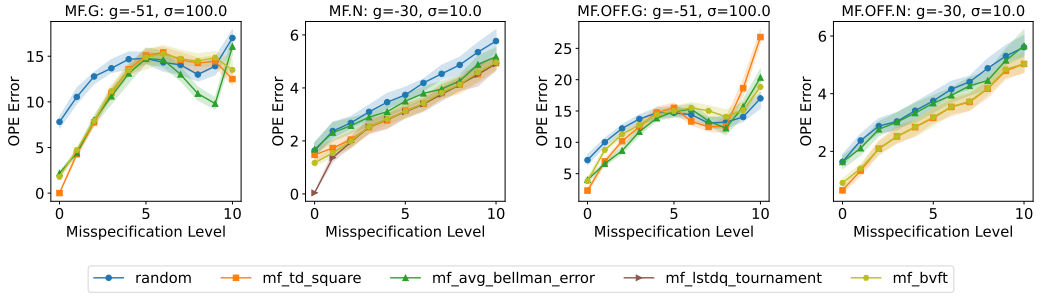

Figure 7: Subgrid studies for misspecification. Plot **MF.N** is identical to Figure 5M.

Data Coverage. Figure 8 shows more complete results for the data coverage experiment in Section 6.3, including more choices of $M^\star$.

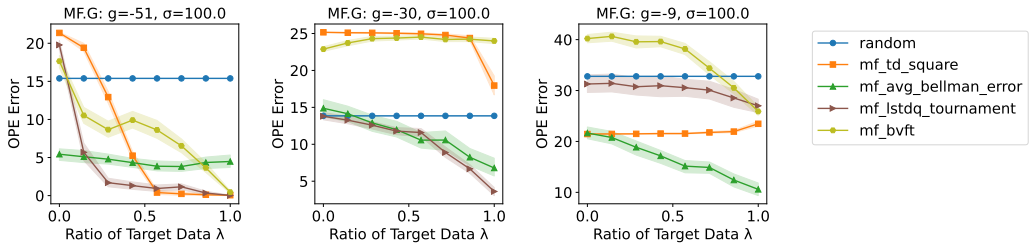

Figure 8: Data coverage results. The left figure is identical to Figure 5L.

E.1 Poor Coverage Results

We now show the counterpart of our model-free main results (Figure 3) under behavior policies that offer poor coverage. This makes the problem very challenging and no single algorithm have strong performance across the board. For example, naïve model-based demonstrate strong performance in **MF.OFF.G** (top row of Figure 9) and resilience to poor coverage, while still suffers catastrophic failures in **MF.OFF.N**. While LSTD-Tournament generally is more reliable than other methods, it also has worse-than-random performance in one of the environments in **MF.OFF.G**.

https://doi.org/10.52202/085713-1430

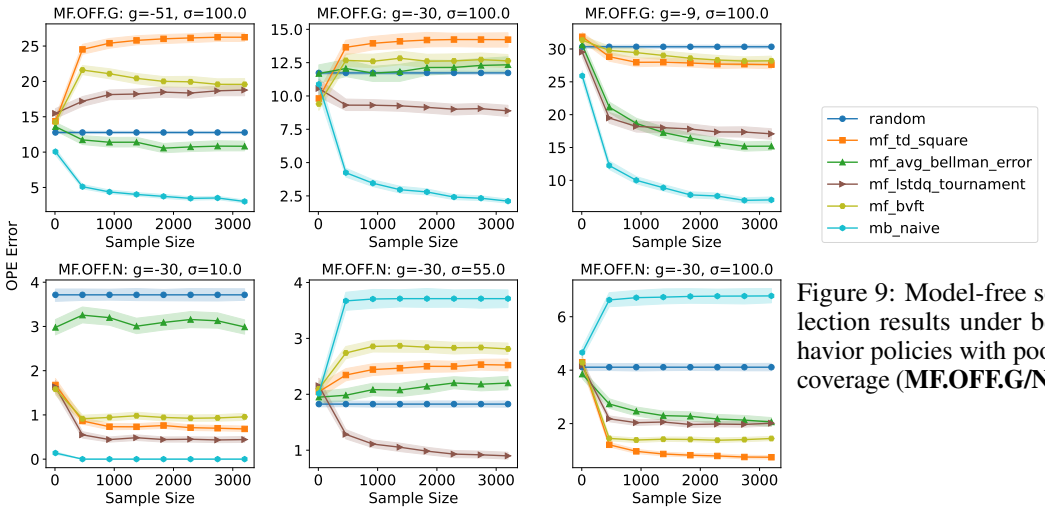

Figure 9: Model-free selection results under behavior policies with poor coverage (**MF.OFF.G/N**).

E.2 LSTDQ Family

As mentioned at the end of Section 3, our LSTD-Tournament can have several variants depending on how we design and transform the linear features. Here we compare 3 of them in Figure 10. The LSTD-Tournament method in all other figures corresponds to the "normalized_diff" version.

- **Vanilla:** $\phi_{i,j} = [Q_i, Q_j]$.
- **Normalized:** $\phi_{i,j} = [Q_i/c_i, Q_j/c_j]$, where $c_i = \sqrt{\mathbb{V}_{(s,a)\sim\mu}[Q_i(s,a)]}$ normalizes the discriminators to unit variance on the data distribution. In practice these variance parameters are estimated from data.
- **Normalized_diff:** $\phi_{i,j} = [Q_i/c_i, (Q_j - Q_i)/c_{j,i}]$, where c_i and $c_{j,i}$ performs normalization in the same way as above.

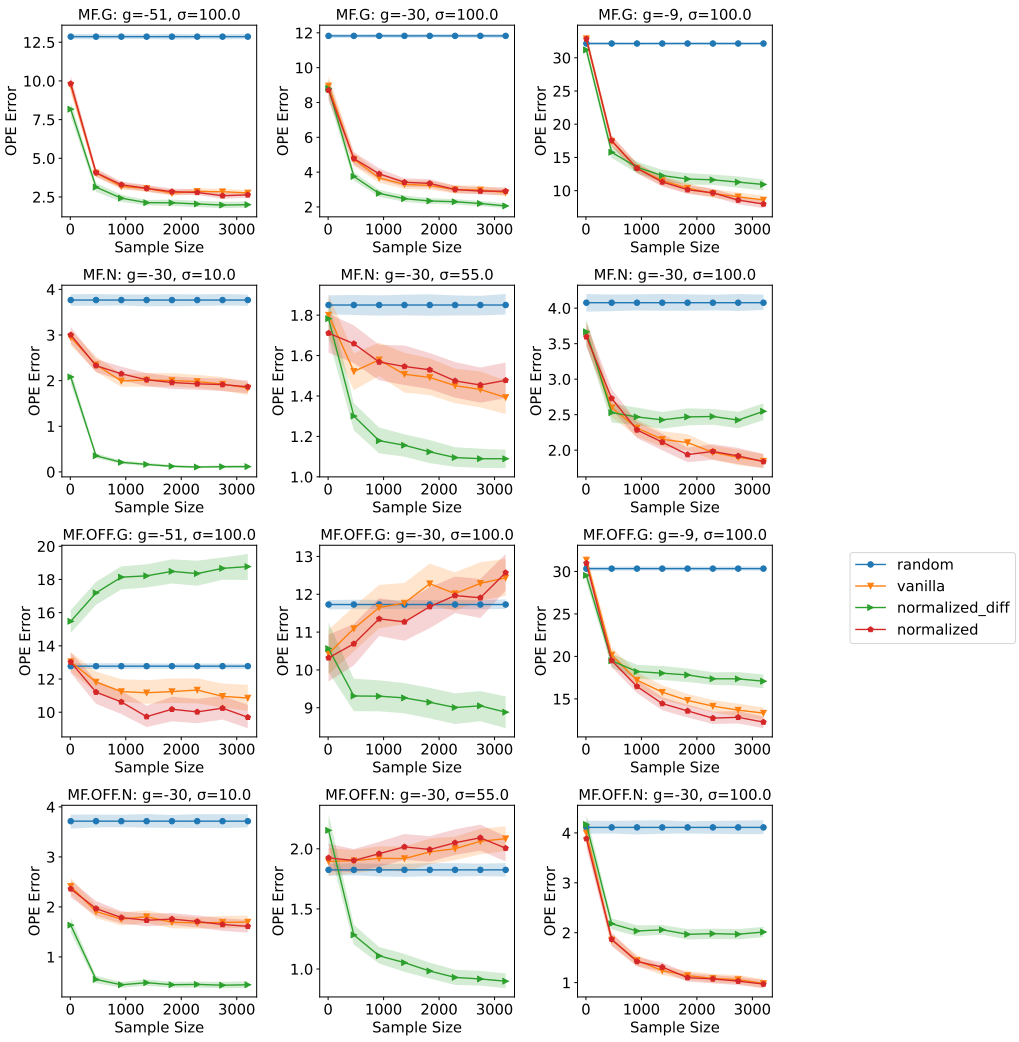

Figure 10: Comparison of variants of LSTD-Tournament.

https://doi.org/10.52202/085713-1430

E.3 $O(1)$ Rollouts

In our experiment design, we use a fairly significant number of rollouts $l = 128$ to ensure relatively accurate estimation of the Q-values. However, for the average Bellman error and the LSTD-Tournament algorithms, they enjoy convergence even when l is a constant. For example, consider the average Bellman error:

$$\mathbb{E}_{\mathcal{D}}[Q_i(s,a) - r - \gamma Q_i(s', \pi)],$$

which is an estimation of $\mathbb{E}_{\mu}[Q_i(s,a) - r - \gamma Q_i(s', \pi)]$. Thanks to its linearity in Q_i, replacing Q_i with its few-rollout (or even single-rollout) Monte-Carlo estimates will leave the unbiasedness of the estimator intact, and Hoeffding's inequality implies convergence as the sample size $n = |\mathcal{D}|$ increases, even when l stays as a constant, which is an advantage compared to other methods. That said, in practice, having a relatively large l can still be useful as it reduces the variance of each individual random variable that we average across $\mathcal{D}$, and the effect can be significant when n is relatively small.

A similar but slightly more subtle version of this property also holds for LSTD-Tournament. Take the vanilla version in Section 3 as example, we need to estimate

$$\mathbb{E}_{\mathcal{D}}[Q_j(s,a)(Q_i(s,a) - r - \gamma Q_i(s', \pi))].$$

Again, we can replace Q_j and Q_i with their Monte-Carlo estimates, as long as the Monte-Carlo trajectories for Q_i and Q_j are independent. This naturally holds in our implementation when $j \neq i$, but is violated when $j = i$ since $Q_j(s,a)$ and $Q_i(s,a)$ will share the same set of random rollouts, leading to biases. A straightforward resolution is to divide the Monte-Carlo rolllouts into two sets, and $Q_j(s,a)$ and $Q_i(s,a)$ can use different sets when $j \neq i$. We empirically test this procedure in Figure 11, where the OPE errors of average Bellman error and different variants of LSTD-Tournament are plotted against the number of rollouts in each set (i.e., $l/2$). While a small number of rollouts can sometimes lead to reasonable performance, more rollouts are often useful in providing further variance reduction and hence more accurate estimations.

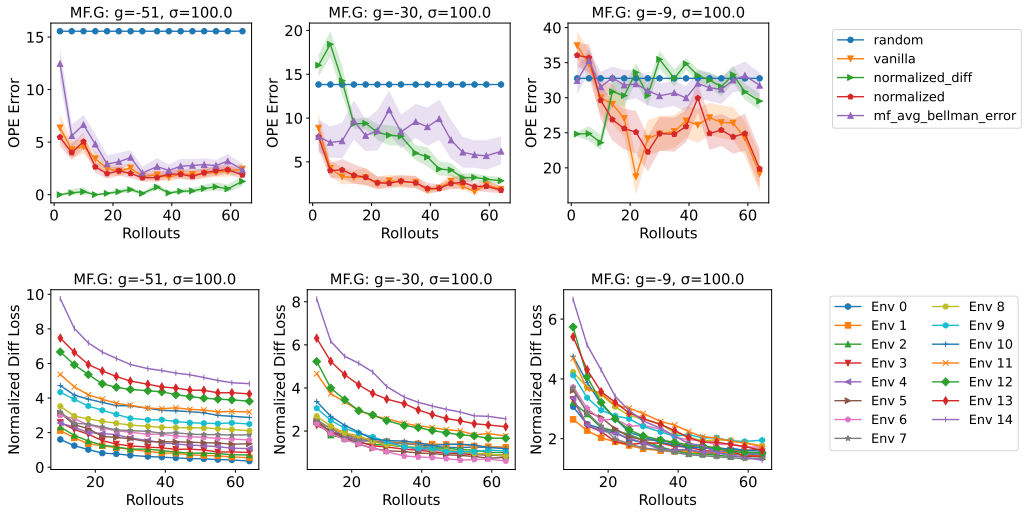

Figure 11: The effect of small rollouts in LSTD-Tournament methods. Sample size is fixed at $n = 3200$ and only l (the number of rollouts) varies. Since the rollouts are divided into two separate sets to ensure independence, each Q-value is estimated using $l/2$ rollouts in this experiment, which is shown on the x-axes. The top row shows the OPE error (i.e., final performance), whereas the bottom row shows the convergence of loss estimates.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We claim to provide new algorithms (Sections 3 and 4) with theoretical guarantees (Theorems 2, 3, and 4) and experimental protocols (Section 5) that is instantiated to provide empirical results (Section 6).

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations are discussed in Section 7.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

https://doi.org/10.52202/085713-1430

Answer: [Yes]

Justification: The standard assumptions that are often treated as part of the problem setup (e.g., OPE with i.i.d. data, realizable Q-function/model set) are stated in Section 2. Other than those, our theorems only require data coverage assumptions, which are manifested as the coverage parameters that explicitly appear in the bounds. All proofs are provided in Appendix C.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Additional experiment details can be found in Appendix D.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The code is uploaded to the supplementary files. The data can be generated from the simulators and the generation code is included.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Our results are not about standard training but model selection in evaluation, and all compared algorithms are optimization-free and in closed form, so the above aspects mostly do not apply. The closest is how the environments are set up and how the behavior/target policies are created, which can be found in Appendix D.1 and D.2. Our new algorithm LSTD-Tournament has a few variants (and the choice can be viewed as a form of hyperparameter), which are tested and compared in Appendix E.2.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We use bootstrapping to calculate error bars that reflect standard errors in the "Bootstrapping" paragraph of Section 5.2.

   Guidelines:

   - The answer NA means that the paper does not include experiments.

https://doi.org/10.52202/085713-1430

- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The computational resources used are discussed in Appendix D.3.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our work is foundational, focusing on theoretical development and simulation results. The topic of study, OPE, aims at providing estimation of new policies before deployment, which is beneficial to the safe use of ML in applications. Overall, we do not think the work presents any ethical concerns that are worth highlighting.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [No]

Justification: Our work is foundational, focusing on theoretical development and simulation results. The topic of study, OPE, aims at providing estimation of new policies before deployment, which is beneficial to the safe use of ML in applications. Overall, we do not think a discussion of broader impacts in the paper is necessary.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: The paper poses no such risks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [NA]

Justification: We used the Gym Hopper environment, which is cited at the end of Section 1.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.

https://doi.org/10.52202/085713-1430

- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Readme is included in the code zip file.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

https://doi.org/10.52202/085713-1430