High Confidence Off-Policy Evaluation with Models

Josiah P. Hanna
Department of Computer Science
The University of Texas at Austin
jphanna@cs.utexas.edu

Peter Stone
Department of Computer Science
The University of Texas at Austin
pstone@cs.utexas.edu

Scott Niekum
Department of Computer Science
The University of Texas at Austin
sniekum@cs.utexas.edu

Abstract

In many reinforcement learning applications executing a poor policy may be costly or even dangerous. Thus, it is desirable to determine confidence interval lower bounds on the performance of any given policy without executing said policy. Current methods for high confidence off-policy evaluation require a substantial amount of data to achieve a tight lower bound, while existing model-based methods only address the problem in discrete state spaces. We propose two bootstrapping approaches combined with learned MDP transition models in order to efficiently estimate lower confidence bounds on policy performance with limited data in both continuous and discrete state spaces. Since direct use of a model may introduce bias, we derive a theoretical upper bound on model bias when we estimate the model transitions with i.i.d. sampled trajectories. This bound can be used to guide selection between the two methods. Finally, we empirically validate the data-efficiency of our proposed methods across three domains and analyze the settings where one method is preferable to the other.

1 Introduction

As reinforcement learning (RL) methods find application in the real world it will be critical to establish the performance of policies with high confidence before they are executed. For example, deploying a poorly performing policy on a manufacturing robot may introduce inefficiency or in the worst case cause damage to the robot or humans working around it. It is insufficient to have a policy that is predicted to work well—we want to specify a lower bound on the policy’s performance that is correct with a pre-determined level of confidence. This problem is known as the high confidence off-policy evaluation problem. We propose data-efficient solutions to this problem.

Current methods for high confidence off-policy evaluation use importance sampling [12] with existing domain data [18]. Due to the large variance of importance sampled returns, these algorithms can require prohibitively large amounts of data to produce meaningful confidence bounds. The current state-of-the-art for high confidence off-policy evaluation involves combining bootstrap techniques [5] with importance sampling [18]. Unfortunately, even with bootstrapping, importance sampling based approaches still lack the data-efficiency needed for data-scarce settings such as robotics.

In order to improve the data-efficiency of high confidence off-policy evaluation, we investigate bootstrapping with learned models of the environment’s transition dynamics. We propose two new bootstrap methods which use models to lower the variance of off-policy value estimates. Our first method, Model-based Bootstrapping (MB-Bootstrap), directly uses a learned model for off-policy value estimation. Our second method, weighted doubly robust bootstrapping (WDR-Bootstrap), uses the recently proposed weighted doubly robust estimator [16] which uses a model to lower the variance
of importance sampling estimators without adding bias to the estimate. Since MB-Bootstrap may exhibit bias when the true model is outside the class of learned models, we derive an upper bound on model bias for models estimated from arbitrary distributions of trajectories. This bound characterizes the settings where MB-Bootstrap may have high bias and WDR-Bootstrap is more appropriate. We empirically evaluate both methods on three policy evaluation tasks and show these methods are far more data-efficient than existing importance sampling based approaches.

2 Preliminaries

2.1 Markov Decision Processes

We formalize our problem as a Markov decision process (MDP) defined as $⟨S, A, P, r, γ, p₀⟩$ where $S$ is a set of states, $A$ is a set of actions, $P : S × A × S → [0, 1]$ is a probability mass function defining state transitions, $r : S × A → [0, r_{max}]$ is a bounded and positive reward function, $γ ∈ [0, 1]$ is a discount factor, and $p₀$ is an initial state distribution. An agent takes actions by sampling from a policy $π : S × A → [0, 1]$ which is a probability mass function for actions conditioned on a given state. A policy can be deterministic in which case $π(a|s)$ is non-zero for only one $a$ in $s$.

A trajectory, $H$ of length $L$ is defined to be a state-action history, $S₁, A₁, S₂, ..., S_L, A_L$ where $S₁ ∼ p₀, A₁ ∼ π(⋅|S₁),$ and $Sᵢ₊₁ ∼ P(⋅|Sᵢ, Aᵢ)$. The return of a trajectory is $g(H) = ∑_{t=1}^{L} γ^{t−1} r(Sᵢ, Aᵢ)$. The policy, $π$, and transition dynamics, $P$, induce a distribution over trajectories, $p_π$. We also write $H ∼ π$ to denote a trajectory sampled by executing $π$. The expected discounted return of a policy, $π$, is defined as $V(π) := E_{H ∼ π} [g(H)]$. $V_π(s)$ is commonly used in the RL literature to represent the expected discounted return when in state $s$ at time $t$ and following policy $π$, $V_π(s_t) := E_{A_t ∼ π(⋅|s_t), S_{t+1} ∼ P(⋅|s_t, A_t)} [γ^{t−1} r(s_t, A_t) + V_π(S_{t+1})]$. Then $V(π) := E_{S₁ ∼ p₀} [V₁(π(S₁))].$

Given a set of $n$ trajectories, $D = \{H₁, ..., H_n\}$, where $Hᵢ ∼ p₀$ for some $p₀$, an evaluation policy, $π_e$, and a confidence level, $δ$, we propose two methods to find a confidence interval lower bound, $V_{−δ}(π_e)$, on $V(π_e)$ such that $V_{−δ}(π_e) < V(π_e)$ with probability $1 − δ$.

2.2 Importance Sampling

We define an off-policy estimator as any method for evaluating $V(π_e)$ using trajectories from a second policy, $π_b$. Importance sampling (IS) is one such method [12]. For a trajectory $H ∼ π_b$, where $H = \{S₁, A₁, ..., S_L, A_L\}$, define the importance weight up to time $t$ for policy $π_e$ as $ρ_t^H := ∏_{i=1}^{t} \frac{π_e(A_i|S_i)}{π_b(A_i|S_i)}$. Then the IS estimator of $V(π_e)$ with a trajectory, $H ∼ π_b$ is defined as $IS(π_e, H, π_b) := g(h)ρ_t^H$. A lower variance version of importance sampling is per-decision importance sampling, PDIS($π_e, H, π_b$) := $∑_{i=1}^{L} r(S_i, A_i)ρ_t^H$. We overload IS notation to define the batch IS estimator for a set of $n$ trajectories, $D$, so that $IS(D) := \frac{1}{n} ∑_{i=1}^{n} IS(π_e, H_i, π_b)$. The batch PDIS estimator is defined similarly.

The potential high variance of batch IS estimators can be reduced with weighted importance sampling (WIS) and per-decision weighted importance sampling (PDWIS). Define the weighted importance weight up to time $t$ for the $i^{th}$ trajectory as $w_t^{H_i} := \frac{ρ_t^{H_i}}{∑_{i=1}^{n} ρ_t^{H_i}}$. Then the WIS estimator is defined as: $WIS(D) := ∑_{i=1}^{n} g(H_i)w_t^{H_i}$. PDWIS is defined similarly to PDIS with $w_t^{H_i}$ replacing $ρ_t^{H_i}$.

IS and PDIS are unbiased but potentially high variance estimators. WIS and PDWIS have less variance than their unweighted counterparts but introduce bias. However, WIS and PDWIS are statistically consistent meaning that bias[WIS] → 0 as $n → ∞$. So with enough data, WIS and PDWIS will also converge to the correct off-policy estimate.

2.3 Bootstrapping

This section overviews bootstrapping as introduced by Efron [6]. In the next section we propose bootstrapping with learned models to estimate confidence intervals for off-policy estimates.

---

1For simplicity we define notation for discrete MDPs. Our results hold for continuous $S$ and $A$ by replacing summations with integrals and representing $P$ and $π$ as probability density functions.

2To be precise, IS and PDIS are unbiased provided $π_b(a|s) = 0 ⇒ π_e(a|s) = 0$. 

---

2
Consider a sample $X$ of $n$ random variables $X_i$ for $i = 1...n$ where we sample $X_i$ i.i.d. from a distribution $f(X)$. We can compute a sample estimate, $\hat{\theta}$ of a parameter, $\theta$, from the sample such that $\hat{\theta} = t(X)$. For example if $\theta$ is the population mean then $t(X) := \frac{1}{n} \sum_{i=1}^{n} X_i$. For a finite sample, we would like to specify the accuracy of $\hat{\theta}$ without placing restrictive assumptions on the sampling distribution of $\hat{\theta}$ (e.g. assuming $\hat{\theta}$ is distributed normally). Bootstrapping allows us to estimate the distribution of $\hat{\theta}$. Starting from a sample $X = \{X_1, ..., X_n\}$, we creates $B$ new samples, $\hat{X}^j = \{\hat{X}_1^j, ..., \hat{X}_n^j\}$, by sampling $\hat{X}_j^i$ from a bootstrap distribution, $\hat{f}$. We sample $\hat{f}$ by independently sampling $X_i$ from $X$ with replacement. For each $\hat{X}^j$ we compute $\hat{\theta}_j = t(\hat{X}^j)$. The distribution of the $\hat{\theta}_j$ approximates the distribution of $\hat{\theta}$ which allows us to compute sample confidence bounds. We recommend the original paper by Efron [5] for a deeper introduction to bootstrapping.

While bootstrapping has strong guarantees as $n \to \infty$, bootstrap confidence intervals lack finite sample guarantees. Using bootstrapping requires the assumption that the bootstrap distribution is representational of the distribution of the statistic of interest which may be false for a finite sample. Therefore, we characterize bootstrap confidence intervals as “semi-safe” due to this possibly false assumption. In contrast to lower bounds from concentration inequalities, bootstrapped lower bounds can be thought of as approximating the allowable $\delta$ error rate instead of upper bounding it. However, bootstrapping is considered safe enough for high risk medical predictions and in practice has a well established record of producing accurate confidence intervals [3]. In the context of policy evaluation, Thomas et al. established that bootstrap confidence intervals with IS estimators can provide accurate lower bounds in the high confidence off-policy evaluation setting [18]. The primary contribution of our work is to incorporate off-policy estimators of $V$ that use models into bootstrapping to decrease the data requirements needed to produce a tight lower bound.

### 3 Bootstrapping Policy Lower Bounds

In this section we propose model-based bootstrapping and weighted doubly robust bootstrapping for estimating confidence intervals on off-policy estimates. First, we present general pseudocode for computing a bootstrap confidence interval lower bound for any off-policy estimator (Algorithm 1).

**Algorithm 1 Bootstrap Confidence Interval**

| input $\pi_e, D, \delta, B$ | output $1 - \delta$ confidence interval lower bound on $V(\pi_e)$ |
|-------------------------------|---------------------------------------------------------------|
| 1: for all $i \in [1, B]$ do |
| 2: $\hat{D}_i = \{H_1^i, ..., H_n^i\}$ where $H_j^i \sim D$ |
| 3: $\hat{V}_i = \text{Off-PolicyEstimate}(\pi_e, \hat{D}_i)$ |
| 4: end for |
| 5: sort({$\hat{V}_i | i \in [1, B]$}) // Sort ascending |
| 6: $l \leftarrow \lceil \delta B \rceil$ |
| 7: Return $\hat{V}_l$ |

Define **Off-PolicyEstimate** to be any method that takes a data set of trajectories $D$ and a policy $\pi_e$ and returns a policy value estimate $\hat{V}(\pi_e)$ (i.e. an off-policy estimator). Algorithm 1 is a general Bootstrap Confidence Interval procedure in which $\hat{V}(\pi_e)$, as computed by **Off-PolicyEstimate**, is the statistic of interest ($\hat{\theta}$ in the proceeding section). Since we desire lower bounds on $V(\pi_e)$ we give pseudocode for a bootstrap lower bound. The method is equally applicable to upper bounds and two sided intervals.

Specifically, the bootstrap method we present is the percentile bootstrap for confidence levels [2]. A more sophisticated bootstrap approach is Bias Corrected and Accelerated bootstrapping (BCa) which adjusts for the skew of the distribution of $\hat{V}_i$. When combining IS with bootstrapping, BCa can correct for the heavy upper tailed distribution of IS returns [18].

#### 3.1 Direct Model-Based Off-Policy Evaluation

We now introduce our first algorithmic contribution—model-based bootstrapping (MB-Bootstrap). The model-based off-policy estimator, MB, computes $\hat{V}(\pi_e)$ by first using all trajectories in $D$ to build a model $\hat{M} = \langle S, A, \hat{P}, r, \gamma, \hat{p}_0 \rangle$ where $\hat{P}$ and $\hat{p}_0$ are estimated from data generated by the

---

3Specifically the resampling approach is the non-parametric form of $\hat{f}$. If we can assume a parameterized form, $f_0$, then we can estimate $\theta'$ from $X$ and use $f_{\theta'}(x)$ as the bootstrap distribution.

4If WIS is used for **Off-PolicyEstimate** then Algorithm 1 describes a simplified version of the bootstrapping method presented by [18].
behavior policy, \( \pi_b \).

Then we estimate \( \hat{V}(\pi_e) \) as the value of \( \pi_e \) acting in \( \hat{M} \). Algorithm 1 with the off-policy estimator MB as **Off-PolicyEstimate** defines MB-Bootstrap.

If a model can capture the true MDP’s dynamics or generalize well to unseen parts of the state-action space then MB estimates can have much lower variance than IS estimates. Thus we expect less variance in our \( \hat{V}_i \) estimates in Algorithm 1. However, models reduce variance at the cost of adding bias to the estimate. Model bias in MDPs can come from two sources:

1. When we lack data for a particular \((s, a)\) pair then we must make assumptions about how to estimate \( P(\cdot|s, a) \).
2. If we use function approximation, we assume the model class from which we select \( \hat{P} \) includes the true transition model, \( P \). When \( P \) is outside the chosen model class (e.g. fitting linear dynamics to a non-linear system) then MB(\( D \)) is biased because MB(\( D \)) \( V(\pi_e) \to b \) for some constant \( b \neq 0 \) as \( n \to \infty \).

The second source of bias is more problematic since even as \( n \to \infty \) the bootstrap model estimates will converge to a different value from \( V(\pi_e) \). In the next section we propose bootstrapping with the recently proposed weighted doubly-robust estimator in order to obtain data-efficient lower bounds in settings where model bias may be large. Later we will present a new theoretical upper bound on model bias when \( \hat{P} \) is learned from a dataset of i.i.d. trajectories. This bound characterizes MDPs and policies that are likely to produce high bias estimates.

### 3.2 Bootstrapped Weighted Doubly Robust

We also propose **weighted doubly robust bootstrapping** (WDR-bootstrap) with the recently proposed WDR off-policy estimator for settings where the MB estimator may exhibit high bias. The WDR estimator is based on per-decision weighted importance sampling (PDWIS) but uses a model to reduce variance in the estimate. The **doubly robust** (DR) estimator has its origins in bandit problems but was recently extended by Jiang and Li to finite horizon MDPs [7]. Thomas and Brunskill then extended DR to infinite horizon MDPs and combined with weighted IS weights to produce the weighted DR estimator [16]. The WDR estimator is defined as:

\[
\text{WDR}(D) := \text{PDWIS}(D) - \sum_{i=1}^{n} \sum_{t=0}^{L} \gamma^t (w_i^t \hat{Q}^\pi_e(S_i^t, A_i^t) - w_{i-1}^t \hat{V}^\pi_e(S_i^t))
\]

\( \hat{V}^\pi_e \) and \( \hat{Q}^\pi_e \) are model-based estimates of the state and state-action value functions (i.e. the value functions of the model). This estimator is able to incorporate a model yet remain free of model bias because the model value-functions only serve as a control variate which reduces the variance of the PDWIS estimate. We refer the reader to [16] and [7] for more in depth discussion of the WDR and DR estimators. Since WDR estimates of \( V(\pi_e) \) were shown to achieve lower mean squared error (MSE) than those of DR in several domains, we propose WDR-Bootstrap which uses WDR as **Off-PolicyEstimate** in Algorithm 1.

Although WDR is biased (since PDWIS is biased), the consistency property of WIS ensures that the bootstrap estimates of WDR-Bootstrap will converge to the correct estimate as \( n \) increases. Thus it is free of the MB out-of-class bias as \( n \to \infty \). Empirical results have shown that WDR can achieve lower MSE than MB in domains where the model converges to an incorrect model [16]. However, they also demonstrated situations where the MB evaluation is more efficient at achieving low MSE than WDR when the variance of the PDWIS weights is high. We empirically analyze the trade-offs when using these estimators with bootstrapping for high confidence off-policy evaluation.

Note that WDR-Bootstrap has three options for the model used to estimate the control variate for WDR: the model can be provided (for instance a domain simulator), the model can be estimated with all the trajectories and then this model be used with WDR to compute each \( \hat{V}_i \), or we can build a new model for every bootstrap data set, \( D_i \), and use it to compute WDR for \( D_i \). In our experiments we estimate a single model with all trajectories in \( D \) before bootstrapping. Then we use the value functions of this single model to compute the WDR estimate for each \( D_i \).

\[\text{In general, the reward function, } r, \text{ may be approximated as } \hat{r} \text{ and initial state distribution, } p_0, \text{ as } \hat{p}_0. \text{ Our theoretical results assume } r \text{ is known but make no assumptions about } p_0 \text{ and } \hat{p}_0.\]
4 Trajectory Based Model Bias

We now present a new theoretical upper bound on model bias to help characterize settings where MB-Bootstrap is likely to be unsuccessful. The bound is related to other model bias bounds in the literature and we discuss these in our survey of related work. Appendix A of the supplementary material provides the full derivation of the bound. For this section we introduce the additional assumption that $L$ is finite. All methods proposed in this paper are applicable to both finite and infinite horizon problems however the bias upper bound is currently only applicable to the finite horizon setting.

**Theorem 1** For any policies $\pi_e$ and $\pi_b$ let $p_{\pi_e}$ and $p_{\pi_b}$ be the distributions of trajectories induced by each policy. Then for an approximate model, $\hat{M}$, with transition probabilities estimated from i.i.d. trajectories $H \sim p_{\pi_b}$, the bias of $\hat{V}(\pi_e)$ is upper bounded by:

$$|\hat{V}(\pi_e) - V(\pi_e)| \leq 2L \cdot r_{max} \sqrt{2 \sup_h \rho_h^b (D_{KL}(p_{\pi_b}||\hat{p}_{\pi_e}) - D_{KL}(p_{\pi_b}||p_{\pi_b}))}$$

where $\rho_h^b$ is the importance weight of trajectory $h$ at step $L$, $\hat{p}_{\pi_e}$ is the distribution of trajectories induced by $\pi_e$ in $\hat{M}$, and $D_{KL}$ is the Kullback-Leibler (KL) divergence.

The KL-divergence comes from information theory and is frequently used as a distance measure between probability distributions. This result tells us that bias[MB] depends on how different the distribution of trajectories of $\pi_e$ under the model is from the distribution of trajectories seen when executing $\pi_b$ in the true MDP. Since most model building techniques (e.g. supervised learning algorithms, tabular methods) build the model from $(s_t, a_t, s_{t+1})$ transitions even if the transitions come from sampled trajectories (i.e. non i.i.d. transitions) we extend theorem 1 to be in terms of transitions:

**Corollary 1** For any policies $\pi_e$ and $\pi_b$ and an approximate model, $\hat{M}$, with transition probabilities, $P$, estimated with trajectories $H \sim p_{\pi_b}$, the bias of the approximate model’s estimate of $V(\pi_e)$, $\hat{V}(\pi_e)$, is upper bounded by:

$$|\hat{V}(\pi_e) - V(\pi_e)| \leq 2L \cdot r_{max} \sqrt{2k \cdot \mathbb{E}_{S \sim d_{\pi_e}, A \sim \pi_e} [D_{KL}(P(\cdot|S,A)||\hat{P}(\cdot|S,A))] + D_{KL}(p_0||\hat{p}_0)}$$

where $k = \sup_h \rho_h^b$ is the largest possible trajectory importance weight and $d_{\pi_e}$ is the distribution of states observed when executing $\pi_b$.

Since $P$ is unknown it is impossible to estimate the $D_{KL}$ terms in corollary 1. However, $D_{KL}$ can be upper bounded by two common supervised learning loss functions: negative log likelihood and cross-entropy. Thus we can express Corollary 1 in terms of either a regression loss function (for continuous MDPs) or a classification loss function (for discrete MDPs) and minimize the bound with observed $(s_t, a_t, s_{t+1})$ transitions. Both theorem 1 and corollary 1 can be extended to finite sample bounds using Hoeffding’s inequality (see Appendix A).

**Discussion** Corollary 1 allows us to compute the upper bound proposed in Theorem 1. However in practice the dependence on the largest importance weight makes the bound too loose to calculate and subtract off from the lower bound found by MB-Bootstrap. Instead we propose its use for determining settings where the MB estimator may exhibit high bias. Specifically a MB estimate of $V(\pi_e)$, where the model is estimated from trajectories generated by $\pi_b$, will have low bias when:

1. We build a model which obtains low training error under the negative log-likelihood or cross-entropy loss functions regardless of whether or not the model is in the same class of models as the true transition dynamics.
2. The probability of observing any trajectory when executing $\pi_e$ is similar to the probability of observing that trajectory when executing $\pi_b$.

5 Empirical Results

We now evaluate MB-Bootstrap and WDR-Bootstrap across three policy evaluation domains. In each domain, we estimate a 95% confidence interval lower bound ($\delta = 0.05$) with our proposed methods.
as well as the importance sampling BCa-bootstrap methods from Thomas et al. [18]. To the best of our knowledge, these IS methods are the current state-of-the-art for high confidence off-policy evaluation. For all methods we use $B = 2000$ bootstrap estimates, $\hat{V}_i$. Appendix B provides full domain and experimental details.

5.1 Experimental Domains

**Discrete Domains** The first domain is the ModelWin domain from Thomas and Brunskill [16]. ModelWin is a simple three state MDP in which an exact model can be learned quickly but the choice of $\pi_0$ and $\pi_b$ result in high variance IS weights. The second domain is the standard mountain car task from the RL literature [15]. In both of these domains we build tabular models (the mountain car state-action space is discretized) which cannot generalize from observed $(s,a)$ pairs. We compute the model state-action value function, $\hat{Q}$, and state value function, $\hat{V}$ with value-iteration for WDR. We use Monte Carlo rollouts to estimate $\hat{V}$ with MB.

**Continuous State-Action Spaces** Our final domain is a continuous two-dimensional CliffWorld (depicted in Figure 5.1). We build models in two ways: linear regression (converges to true model as $n \to \infty$) and regression over nonlinear polynomial basis functions. The first model class choice represents the ideal case (analogous to ModelWin) and the second is the case when the true dynamics cannot be represented by the learned model. Our results refer to MB-Bootstrap$_{LR}$ and MB-Bootstrap$_{PR}$ as the MB estimator using linear regression and polynomial regression respectively. Similarly, we evaluate WDR-Bootstrap$_{LR}$ and WDR-Bootstrap$_{PR}$. We use Monte Carlo evaluations to estimate the model value-functions for WDR.

5.2 Empirical Results

Figure 2 displays the average empirical 95% confidence interval lower bound found by each method in each domain. The ideal result is a lower bound, $V_{-,\delta}(\pi_e)$, that is as large as possible subject to $V_{-,\delta}(\pi_e) < V(\pi_e)$. Given that any unbiased method will achieve the ideal result as $n \to \infty$, our main point of comparison is which method gets closest the fastest. As a general trend we note that our proposed methods—MB-Bootstrap and WDR-Bootstrap—get closer to this ideal result with less data than all other methods. Figure 3 displays the empirical error rate for MB-Bootstrap and WDR-Bootstrap and shows that they approximate the allowable 5% error in each domain.

**Discrete Results** We see in Figure 2(a) that MB-Bootstrap and WDR-Bootstrap outperform the IS lower bounds in the ModelWin domain. WDR-Bootstrap is slightly less data-efficient than MB-Bootstrap which can be expected since it is impacted by the high variance of importance sampling. Nevertheless, incorporating a model into the estimate reduces variance enough to outperform all IS methods. While all IS methods and WDR approximate the required error rate, it is notable that after approximately 200 trajectories the MB-Bootstrap error rate is much lower than the required error rate yet Figure 2(a) shows the lower bound is no looser. The reason for this is that all bootstrap models eventually converge to the true MDP. Then all bootstrap models look approximately the same and variation in the $\hat{V}_i$ estimates of Algorithm 1 are only due the variance of the Monte Carlo estimates of MB. It is also notable that since bootstrapping only approximates the 5% allowable error rate all methods can do worse then 5% when data is extremely sparse (only two trajectories).

In mountain car (Figure 2(b), both of our methods (WDR-Bootstrap and MB-Bootstrap) outperform purely IS methods. The notable trend here is that both methods produce approximately the same average lower bound. Mountain car is a larger domain than ModelWin so MB requires more data to learn an accurate model. Therefore, even though MB will eventually converge to $V(\pi_e)$ it does so no faster than WDR which can produce good estimates without a good model.

**Continuous Results** In CliffWorld (Figure 2(c)), we first note that MB-Bootstrap$_{PR}$ quickly converges to a suboptimal lower bound. In practice an incorrect model may lead to a bound that is too high (positive bias) or too loose (negative bias). Here, MB-Bootstrap$_{PR}$ exhibits negative bias which means we converge to a bound that is too loose. More dangerous is positive bias which will make the method unsafe. The models learned for MB-Bootstrap$_{PR}$ have large training error compared to the
Figure 2: The average empirical lower bound across three domains. The ModelWin and Mountain Car plots use the top line legend. The CliffWorld plot uses the bottom line legend. Each plot displays the 95% lower bound on $V(\pi_e)$ computed by each method with varying amounts of trajectories. The ideal lower bound is just below the line labelled $V(\pi_e)$. Results demonstrate that the proposed model-based bootstrapping (MB-Bootstrap) and weighted doubly robust bootstrapping (WDR-Bootstrap) find a tighter lower bound with less data than previous bootstrapping with importance sampling methods. For clarity we omit IS in Mountain Car and IS, WIS and PDIS in CliffWorld. Error bars are for a 95% two-sided confidence interval.

Figure 3: The empirical error rate for model-based bootstrapping and weighted doubly robust bootstrapping for the model win and mountain car domains. We use our proposed methods to compute the lower bound $m$ times ($m = 500$ for ModelWin, $m = 400$ for Mountain Car) and observe how many times the lower bound is above the true $V(\pi_e)$. We see empirically that both methods correctly approximate the allowable 5% error rate for a 95% confidence interval lower bound. These figures use the top legend in Figure 2.

models learned for MB-Bootstrap$^{LR}$. Our theoretical results predicts that this setting is one where MB bias may be high.

The second notable trend is that WDR is also negatively impacted by the incorrect model. In Figure 2(c) we see that WDR-Bootstrap$^{LR}$ (correct model) starts at a tighter bound and increases from there. WDR-Bootstrap$^{PR}$ with an incorrect model actually performs worse than PDWIS until larger $n$. Using an incorrect model with WDR decreases the variance of the PDWIS term less than the correct model would but it should still have less variance and produce a tighter lower bound than PDWIS by itself. One possibility is that error in the estimate of the model value functions increases the variance of WDR. This result motivates investigating the effect of inaccurate model state-value and state-action-value functions on WDR as these functions are certain to have error in any continuous setting. Nevertheless, MB-Bootstrap$^{PR}$ eventually overcomes the poor model to find a tighter bound.

6 Discussion

We have proposed two bootstrapping methods that incorporate models to produce tight lower bounds on off-policy estimates. We now describe the advantages and disadvantages of each and make specific recommendations about their use in practice.

Our MB-Bootstrap method can be the most data-efficient method. This data-efficiency comes at a cost of potential bias where bias comes from 1) finite samples or 2) using the wrong model class to
represent $P$. Empirical results show when we have few trajectories, MB-Bootstrap correctly identifies model uncertainty from lack of data and produces looser confidence bounds. Our theoretical results show that the second form of bias cannot be avoided for some domains, some model-class choices, and some $\pi_e$ and $\pi_b$. However if our model-class can achieve low approximation error and the policies have similar action selection distributions than model bias will be low. The second restriction can be restated as, “do we have data to model the actions $\pi_e$ would take?” If the answer is no then WDR-Bootstrap is more appropriate.

Our proposed WDR-Bootstrap provides a low variance and low bias method of high confidence off-policy evaluation. These two properties allow WDR-Bootstrap to outperform other variants of IS and sometimes perform as well or better than MB-Bootstrap. In contrast to MB-Bootstrap, WDR-Bootstrap achieves data-efficient lower bounds while remaining free of model bias. A disadvantage of WDR-Bootstrap is that it requires the model’s value functions be known for all states and state-action pairs that occur in trajectories in $D$. In continuous state and action spaces this requires either function approximation or Monte Carlo evaluation. Either method can introduce error into the WDR estimate and reduce the power of the WDR control variate term to reduce variance.

Two special cases that occur in real world off-policy evaluation are deterministic policies and unknown $\pi_b$. When $\pi_e$ is deterministic, one should use MB-Bootstrap since the importance weights equal zero at any time step that $\pi_b$ chose action $a_t$ such that $\pi_e(A_t = a_t|S_t) = 0$. Deterministic $\pi_b$ are problematic for any method since they produce trajectories which lack a variety of action selection data. Unknown $\pi_b$ occur when we have domain trajectories but no knowledge of the policy that produced the trajectories. For example, a medical treatment domain could have data on treatments and outcomes but the doctor’s treatment selection policy be unknown. In this setting, importance sampling methods cannot be applied and MB-Bootstrap may be the only way to provide a confidence interval on a new policy.

7 Related Work

Concentration inequalities have been used with IS returns to lower bound off-policy estimates [17]. The concentration inequality approach is notable in that it produces a true probabilistic bound on the policy performance. Unfortunately, this approach requires prohibitive amounts of data and was shown to be far less data-efficient than bootstrapping with IS [18]. Jiang and Li evaluated the DR estimator for safe-policy improvement [7]. They compute confidence intervals similar to the Student’s $t$-Test confidence interval shown to be less data-efficient than bootstrapping [18].

Chow et al. used ideas from robust optimization to bound model bias caused by error in a discrete MDP’s transition function and showed that tight lower bounds can be proven on the performance of a policy [4]. These methods are also strongly related to robust MDPs [9, 11]. Other bounds on error in estimates of $V(\pi)$ with an inaccurate model have been introduced for discrete MDPs [14, 8]. In contrast, we present a bound on model bias for both continuous and discrete MDPs. Ross and Bagnell introduce a bound similar to Theorem 1 but assume that the model is built from transitions sampled i.i.d. from a given exploration distribution [13]. Since we bootstrap over trajectories their bound is inapplicable to our setting. Paduraru introduced tight model bias bounds for i.i.d. sampled transitions from general MDPs and i.i.d. trajectories from DAG (directed acyclic graph) MDPs [10]. We made no assumptions on the structure of the MDP when deriving our bound.

8 Conclusion and Future Work

We have introduced two novel methods—MB-Bootstrap and WDR-Bootstrap—that determine confidence intervals for off-policy evaluation. Empirically, our methods yielded superior data-efficiency and tighter lower bounds on $V(\pi_e)$ than state-of-the-art importance sampling based method. We also prove a new bound on the expected bias of MB when learning models that minimize error over a dataset $D$ of trajectories sampled from an arbitrary policy.

Our ongoing research agenda includes applying these techniques to high confidence evaluation of robot skills. Robots may exhibit complex, non-linear dynamics that are hard to model. At the same time robots may have high dimensional continuous state and action spaces which will increase the variance of IS estimators. These two challenges suggest robotics as a domain for the WDR-Bootstrap method—potentially using a simulator to provide the control variate for WDR.

A final direction for future work is to evaluate the MAGIC off-policy estimator from Thomas and Brunskill with bootstrapping [16]. MAGIC combines low variance, biased MB estimates with higher
variance, low bias WDR estimates. MAGIC as the Off-PolicyEstimate method in Algorithm 1 could be an alternative to selecting between MB and WDR but this idea needs empirical validation. In our work we kept WDR and MB separate in order to better understand their properties and potentially improve blending in the future. Our theoretical results may also provide insight into better methods for blending these estimators.

This work has taken place in the Personal Autonomous Robotics Lab (PeARL) and Learning Agents Research Group (LARG) at the Artificial Intelligence Laboratory, The University of Texas at Austin. PeARL research is supported in part by the National Science Foundation under grant IIS1208497. LARG research is supported in part by grants from the National Science Foundation (CNS-1350072, CNS-1305287), ONR (21C184-01), AFOSR (FA9550-14-1-0087). Peter Stone serves on the Board of Directors of Cogitai, Inc. The terms of this arrangement have been reviewed and approved by the University of Texas at Austin in accordance with its policy on objectivity in research.

References

[1] KB Athreya. Bootstrap of the mean in the infinite variance case. The Annals of Statistics, pages 724–731, 1987.
[2] James Carpenter and John Bithell. Bootstrap confidence intervals: when, which, what? a practical guide for medical statisticians. Statistics in Medicine, pages 1141–1164, 2000.
[3] Lloyd E Chambless, Aaron R Folsom, A Richey Sharrett, Paul Sorlie, David Couper, Moyo Szkelo, and F Javier Nieto. Coronary heart disease risk prediction in the atherosclerosis risk in communities (aric) study. Journal of clinical epidemiology, 56(9):880–890, 2003.
[4] Yinlam Chow, Marek Petrik, and Mohammad Ghavamzadeh. Robust policy optimization with baseline guarantees. arXiv preprint arXiv:1506.04514, 2015.
[5] B Efron et al. Bootstrap methods: Another look at the jackknife. The Annals of Statistics, 7(1):1–26, 1979.
[6] Bradley Efron. Better bootstrap confidence intervals. Journal of the American statistical Association, 82(397):171–185, 1987.
[7] Nan Jiang and Lihong Li. Doubly robust off-policy evaluation for reinforcement learning. arXiv preprint arXiv:1511.03722, 2015.
[8] Michael Kearns and Satinder Singh. Near-optimal reinforcement learning in polynomial time. Machine Learning, 49(2-3):209–232, 2002.
[9] Arnab Nilim and Laurent El Ghaoui. Robust control of markov decision processes with uncertain transition matrices. Operations Research, 53(5):780–798, 2005.
[10] Cosmin Paduraru. Off-policy Evaluation in Markov Decision Processes. PhD thesis, McGill University, 2013.
[11] Marek Petrik and Dharmashankar Subramanian. Raam: The benefits of robustness in approximating aggregated mdps in reinforcement learning. In Advances in Neural Information Processing Systems, pages 1979–1987, 2014.
[12] Doina Precup, Richard S. Sutton, and Satinder Singh. Eligibility traces for off-policy policy evaluation. In Proceedings of the 17th International Conference on Machine Learning, 2000.
[13] Stephane Ross and J. Andrew Bagnell. Agnostic system identification for model-based reinforcement learning. In 29th International Conference on Machine Learning, ICML, 2012.
[14] Alexander L Strehl, Lihong Li, and Michael L Littman. Reinforcement learning in finite mdps: Pac analysis. Journal of Machine Learning Research, 10:2413–2444, 2009.
[15] Richard S. Sutton and Andrew G. Barto. Reinforcement Learning: An Introduction. MIT Press, 1998.
[16] Phil Thomas and Emma Brunskill. Data-efficient off-policy policy evaluation for reinforcement learning. arXiv preprint arXiv:1604.00923, 2016.
[17] Philip S. Thomas, Georgios Theocharous, and Mohammad Ghavamzadeh. High confidence off-policy evaluation. In Association for the Advancement of Artificial Intelligence, AAAI, 2015.
[18] Philip S. Thomas, Georgios Theocharous, and Mohammad Ghavamzadeh. High confidence policy improvement. In Proceedings of the 32nd International Conference on Machine Learning, ICML, 2015.
[19] P.S. Thomas. Safe Reinforcement Learning. PhD thesis, University of Massachusetts Amherst, 2015.
Appendices

A Upper Bounds on Model Bias

This appendix proves Theorem 1 and Corollary 1 from the main text. It also includes two additional Lemmas and a Corollary that are useful in proving Theorem 1 and Corollary 1 (Corollary 2 in the supplemental material). All notation from the main paper carries over to this supplementary document.

First we introduce some additional definitions and notations:

- Define the maximum possible return as $g_{\text{max}} := L \cdot r_{\text{max}}$.
- Define the distribution of states at time $t$ when following policy $\pi$,
  \[ d^t_\pi(s) := \sum_{s_1} \sum_{a_1} \cdots \sum_{s_{t-1}} \sum_{a_{t-1}} p_0(s_1)P(s_2|s_1, a_1) \cdots P(s_t|s_{t-1}, a_{t-1}) \]
  with $d^0_\pi(s) := p_0(s)$.
- Define the distribution of states when following policy $\pi$, $d_\pi(s) := \frac{1}{L} \sum_{t=1}^L d^t_\pi(s)$.

We define notation for discrete states and actions. Results hold for continuous states and actions by replacing summations over states and actions with integrals and changing probability mass functions to probability density functions.

A.1 Model Bias when Evaluation and Behavior Policy are the Same

**Lemma 1** For any policy $\pi$, let $p_\pi$ be the distribution of trajectories generated by $\pi$ and $\hat{p}_\pi$ be the distribution of trajectories generated by $\pi$ in an approximate model, $\hat{M}$. Then the bias of an estimate, $\hat{V}(\pi)$, under $\hat{M}$ is upper bounded by:
\[ |V(\pi) - \hat{V}(\pi)| \leq 2L \cdot r_{\text{max}} \sqrt{2D_{KL}(p_\pi||\hat{p}_\pi)} \]
where $D_{KL}(p_\pi||\hat{p}_\pi)$ is the Kullback-Leibler (KL) divergence between probability distributions $p_\pi$ and $\hat{p}_\pi$.

Proof:
\[ |V(\pi) - \hat{V}(\pi)| = |\sum_h p_\pi(h)g(h) - \sum_h \hat{p}_\pi(h)g(h)| \]
Combining the summations and factoring out $g(h)$:
\[ = |\sum_h (p_\pi(h) - \hat{p}_\pi(h))g(h)| \]
Moving the absolute value around the difference between the probabilities can only increase the r.h.s.:
\[ \leq \sum_h |p_\pi(h) - \hat{p}_\pi(h)|g(h) \]
Since $\forall h \ g(h) \leq g_{\text{max}}$, we can replace $g(h)$ with the constant $g_{\text{max}}$:
\[ \leq \sum_h |p_\pi(h) - \hat{p}_\pi(h)|g_{\text{max}} \]
Factoring $g_{\text{max}}$ out of the summation, we use the definition of the total variation $(D_{TV}(p||q) = \frac{1}{2} \sum_x |p(x) - q(x)|)$ to get:
\[ = 2D_{TV}(p||\hat{p}_\pi)g_{\text{max}} \]
In the undiscounted case the largest return is $g_{\text{max}} = L \cdot r_{\text{max}}$. We replace $g_{\text{max}}$ with this expression:
\[ \leq 2L \cdot r_{\text{max}} D_{TV}(p_\pi||\hat{p}_\pi) \]
Combining with Pinker’s inequality $(D_{TV}(p||q) \leq \sqrt{2D_{KL}(p||q)})$ completes the proof.
A.2 Bounding the KL-divergence Between Trajectory Distributions

Lemma 2 For any policies \( \pi \) and \( \hat{\pi} \) and a model \( \hat{M} \) with initial state distribution \( \hat{p} \) and transition probabilities \( \hat{P} \), let \( p_r \) be the distribution of trajectories generated by \( \pi \) and \( \hat{p}_r \) be the distribution of trajectories generated by \( \hat{\pi} \) in \( \hat{M} \). Then the Kullback-Leibler (KL) divergence between \( p_r \) and \( \hat{p}_r \) is upper bounded by:

\[
D_{KL}(p_r || \hat{p}_r) \leq D_{KL}(p_0 || \hat{p}_0) + L \cdot E_{S \sim d_{\pi}}[D_{KL}(\pi(\cdot|S)||\hat{\pi}(\cdot|S))] + L \cdot E_{S \sim d_{\hat{\pi}}, A \sim \hat{\pi}(\cdot|S)}[D_{KL}(P(\cdot|S, A)||\hat{P}(\cdot|S, A))]
\]

Proof:

\[
D_{KL}(p_r || \hat{p}_r) = \sum_h p_r(h) \log \frac{p_r(h)}{\hat{p}_r(h)}
\]

\[
= \sum_{s_1} \sum_{a_1} \cdots \sum_{s_L} \sum_{a_L} p_0(s_1) \pi(a_1|s_1) \cdots P(s_L|s_{L-1}, a_{L-1}) \pi(a_L|s_L) \log \frac{p_0(s_1) \pi(a_1|s_1) \cdots P(s_L|s_{L-1}, a_{L-1}) \pi(a_L|s_L)}{\hat{p}_0(s_1) \hat{\pi}(a_1|s_1) \cdots \hat{P}(s_L|s_{L-1}, a_{L-1}) \hat{\pi}(a_L|s_L)}
\]

Using the logarithm property that \( \log(ab) = \log(a) + \log(b) \) and rearranging the summation gives us:

\[
= \sum_{s_1} p_0(s_1) \log \frac{p_0(s_1)}{\hat{p}_0(s_1)} + \sum_{a_1} \pi(a_1|s_1) \cdots \sum_{a_L} \pi(a_L|s_L) \log \frac{P(s_L|s_{L-1}, a_{L-1})}{\hat{P}(s_L|s_{L-1}, a_{L-1})}
\]

\[
+ \sum_{t=1}^L \sum_{s_t} p_0(s_t) \cdots \sum_{s_t} \pi(a_t|s_t) \log \frac{P(s_t|s_{t-1}, a_{t-1})}{\hat{P}(s_t|s_{t-1}, a_{t-1})} \sum_{a_t} \pi(a_t|s_t)
\]

\[
+ \sum_{t=1}^L \sum_{s_t} p_0(s_t) \cdots \sum_{s_t} \pi(a_t|s_t) \log \frac{P(s_t|s_{t-1}, a_{t-1})}{\hat{P}(s_t|s_{t-1}, a_{t-1})} \sum_{a_t} \pi(a_t|s_t)
\]

\[
= D_{KL}(p_0 || \hat{p}_0) + \sum_{t=1}^L E_{S \sim d_{\pi}}[D_{KL}(\pi(\cdot|S)||\hat{\pi}(\cdot|S))]
\]

\[
+ \sum_{t=2}^L E_{S \sim d_{\pi}}[D_{KL}(\pi(\cdot|S)||\hat{\pi}(\cdot|S))]
\]

\[
+ \sum_{t=1}^{L-1} E_{S \sim d_{\hat{\pi}}, A \sim \hat{\pi}(\cdot|S)}[D_{KL}(P(\cdot|S, A)||\hat{P}(\cdot|S, A))]
\]

Note that summations of the form \( \sum_{t=1}^L \sum_s d_n(s) f(s) \) can be rewritten as \( L \cdot \sum_s d_n(s) f(s) \) because \( \sum_{t=1}^L \sum_s d_n(s) f(s) = L \cdot \sum_s \sum_{t=1}^L d_n(s) f(s) = L \cdot \sum_s d_n(s) f(s) \):

\[
= D_{KL}(p_0 || \hat{p}_0) + L \cdot E_{S \sim d_{\pi}} [D_{KL}(\pi(\cdot|S)||\hat{\pi}(\cdot|S))] + L \cdot E_{S \sim d_{\hat{\pi}}, A \sim \hat{\pi}(\cdot|S)}[D_{KL}(P(\cdot|S, A)||\hat{P}(\cdot|S, A))]
\]

\[
- E_{S \sim d_{\hat{\pi}}, A \sim \hat{\pi}(\cdot|S)}[D_{KL}(P(\cdot|S, A)||\hat{P}(\cdot|S, A))]
\]

\[
\leq D_{KL}(p_0 || \hat{p}_0) + E_{S \sim d_{\pi}} [D_{KL}(P(\cdot|S, A)||\hat{P}(\cdot|S, A))] + E_{S \sim d_{\hat{\pi}}}[D_{KL}(\pi(\cdot|S)||\hat{\pi}(\cdot|S))]
\]
A.3 Minimizing Lemma 1 in terms of Supervised Learning Loss Functions

Corollary 2 For any policy \( \pi \), let \( P \) be the true transition probabilities and \( \hat{P} \) be the transition probabilities of an approximate model, \( M \). Then the bias of the approximate model’s estimate of \( V(\pi) \) is upper bounded by:

\[
|V(\pi) - \hat{V}(\pi)| = 2L \cdot r_{\text{max}} \sqrt{2L \cdot \mathbb{E}_{S \sim \pi \circ \cdot, A \sim \cdot \mid S}[D_{KL}(P(\cdot \mid S, A) \mid \hat{P}(\cdot \mid S, A)] + D_{KL}(p_0 || \hat{p}_0)}
\]

where \( D_{KL}(p || q) \) is the Kullback-Leibler (KL) divergence between distributions \( p \) and \( q \).

Corollary 1 follows directly from an application of Lemma 2 to Lemma 1.

We relate \( D_{KL} \) to two common supervised learning loss functions so that we can minimize Corollary 1 with \((S_t, A_t, S_{t+1})\) samples. \( D_{KL}(P || \hat{P}) = H[ \hat{P} ] - H[P] \) where \( H[P] \) and \( H[ \hat{P}, P] \) are entropy and cross-entropy respectively. \( H[P, \hat{P}] - H[P] \leq H[ \hat{P}, P] \) since entropy is always positive. So we can upper bound \( D_{KL} \) with the cross-entropy loss function. The cross-entropy loss function is equivalent to the expected negative log likelihood loss function: \( H( P(\cdot|s,a), \hat{P}(\cdot|s,a)) = \mathbb{E}_{S \sim P(\cdot|s,a)}[- \log \hat{P}(S'|s,a)] = \mathbb{E}_{S \sim P(\cdot|s,a)}[\text{nhl}(\hat{P}, s, a, S')] \) where \( \text{nhl}(P, s, a, s') := -\log(P(s'|s,a)). \) Thus our bound applies to maximum likelihood model learning.

A.4 Bounds in terms of behavior policy data

We want an upper bound when model built from trajectories sampled from an arbitrary distribution. The reasons for this are:

1. In general the model will be built with data generated by a different policy than the one we are evaluating.
2. When we bootstrap with the MB estimator models are built from trajectories sampled i.i.d. from the bootstrap distribution.

Theorem 1 meets this requirement.

Theorem 2 For any policies \( \pi_e \) and \( \pi_b \) let \( p_{\pi_e} \) and \( p_{\pi_b} \) be the distributions of trajectories induced by each policy. Then for an approximate model, \( M \), estimated with i.i.d. trajectories, \( H \sim p_{\pi_b} \), the bias of the estimate of \( V(\pi_e) \) with \( \hat{M} \), \( V(\pi_e) \), is upper bounded by:

\[
|\hat{V}(\pi_e) - V(\pi_e)| \leq 2L \cdot r_{\text{max}} \sqrt{2 \sup_h \rho^h_{\pi_b} (D_{KL}(p_{\pi_e} || \hat{p}_{\pi_e}) - D_{KL}(p_{\pi_b} || p_{\pi_e}))}
\]

where \( \rho^h_{\pi_b} \) is the importance weight of trajectory \( h \) at step \( L \). \( \hat{p}_{\pi_e} \) is the distribution of trajectories induced by \( \pi_e \) in \( M \), and \( D_{KL} \) is the Kullback-Leibler (KL) divergence.

We will make use of the following fact from Ross and Bagnell [13]:

Fact 1 \( \mathbb{E}_{X \sim p}[f(X)] \leq \sup_x \frac{p(x)}{q(x)} \mathbb{E}_{x \sim q}[f(x)] \)

This bound is derived from importance sampling the expectation under \( p \) and factoring the largest possible importance weight out of the expectation.

First recall that Lemma 1 states that:

\[
|\hat{V}(\pi_e) - V(\pi_e)| \leq 2L \cdot r_{\text{max}} \sqrt{2D_{KL}(p_{\pi_e} || \hat{p}_{\pi_e})}
\]

Our main goal is to bound the KL-divergence part of Lemma 1 (an expectation under trajectories from \( \pi_e \)) in terms of an expectation under trajectories from \( \pi_b \). From Fact 1:

\[
D_{KL}(p_{\pi_e} || \hat{p}_{\pi_e}) \leq \sup_h \frac{p_{\pi_b}(h)}{p_{\pi_e}(h)} \mathbb{E}_{H \sim p_{\pi_b}}[\log \frac{p_{\pi_b}(H)}{p_{\pi_e}(H)}]
\]

Consider the new expectation:

\[
\mathbb{E}_{H \sim p_{\pi_b}}[\log \frac{p_{\pi_b}(H)}{p_{\pi_e}(H)}] = \mathbb{E}_{H \sim p_{\pi_b}}[\log p_{\pi_b}(H) - \log p_{\pi_e}(H)]
\]

Separate the expectations and then apply the definition of the cross-entropy \( (H[p, q] = \mathbb{E}_{X \sim p}[- \log q(X)]) \):
Any of our theoretical results can be expressed as finite-sample bounds by applying Hoeffding’s inequality to as with Corollary 1, the same application of Hoeffding’s can be used for Lemmas 1 and 2, Corollary 1, and Theorem 1.

Finally, we apply the fact that $D_{KL}(p||q) = H[p, q] - H[p]$:

$$D_{KL}(p_{\pi_b}||\hat{p}_{\pi_e}) - D_{KL}(p_{\pi_b}||\pi_e)$$

Substituting $\sup_{p_{\pi_b}} \mathbb{E}_{H \sim p_{\pi_b}} [\log \frac{p_{\pi_b}}{p_{\pi_e}}]$ for $D_{KL}(p_{\pi_b}||\hat{p}_{\pi_e})$ in Lemma 1:

$$|\hat{V}(\pi_e) - V(\pi_e)| \leq 2L \cdot r_{max} \sqrt{2 \sup_{h} \frac{p_{\pi_b}(h)}{p_{\pi_e}(h)} \left[D_{KL}(p_{\pi_b}||\hat{p}_{\pi_e}) - D_{KL}(p_{\pi_b}||\pi_e)\right]}$$

Note that transition probabilities will cancel in $\sup_{h} \frac{p_{\pi_b}(h)}{p_{\pi_e}(h)}$ leaving us with $\sup_{h} \rho(h)$ and completing the proof.

A.5 Bounding Theorem 1 in terms of a Supervised Loss Function

As for Lemma 1 we would like to express Theorem 2 in terms of a supervised loss function.

**Corollary 3** For any policies $\pi_e$ and $\pi_b$ and an approximate model, $\hat{M}$, with transition probabilities, $\hat{P}$, estimated with trajectories $H \sim \pi_b$, the bias of the approximate model’s estimate of $V(\pi_e)$, $\hat{V}(\pi_e)$, is upper bounded by:

$$|\hat{V}(\pi_e) - V(\pi_e)| \leq 2L \cdot r_{max} \sqrt{2 \sup_{h} \frac{p_{\pi_b}(h)}{p_{\pi_e}(h)} \left[D_{KL}(P(\cdot||S, A)||\hat{P}(\cdot||S, A)) + D_{KL}(p_0||\hat{p}_0)\right]}$$

where $k = \sup_{h} \rho(h)$ is the largest possible trajectory importance weight and $d_{\pi_b}$ is the distribution of states observed when executing $\pi_b$.

Corollary 2 follows from an application of Lemma 2 to Theorem 1.

As with Corollary 1, $\mathbb{E}_{S \sim d_{\pi_b}, A \sim \pi_b} [D_{KL}(P(\cdot||S, A)||\hat{P}(\cdot||S, A))]$ can be expressed as the negative log likelihood or cross-entropy loss functions.

A.6 Finite Sample Bounds

Any of our theoretical results can be expressed as finite-sample bounds by applying Hoeffding’s inequality to bound the expectations in the bound. Since Corollary 2 (Corollary 1 in the main text) is the bound that can be computed as a function of training error we restate it here as a finite-sample bound:

**Corollary 4** For any policies $\pi_e$ and $\pi_b$ and an approximate model, $\hat{M}$, with transition probabilities, $\hat{P}$, estimated with transitions, $(s, a)$, from trajectories $H \sim \pi_b$, and after observing $m$ transitions then with probability $\alpha$, the bias of the approximate model’s estimate of $V(\pi_e)$, $\hat{V}(\pi_e)$, is upper bounded by:

$$|\hat{V}(\pi_e) - V(\pi_e)| \leq 2L \cdot r_{max} \sqrt{2k \cdot \frac{1}{m} \sum_{(s, a) \in D} [D_{KL}(P(\cdot||s, a)||\hat{P}(\cdot||s, a))] - \sqrt{\frac{\ln(\frac{1}{\alpha})}{2m} + D_{KL}(p_0||\hat{p}_0)}}$$

where $k = \sup_{h} \rho(h)$ is the largest possible trajectory importance weight and $d_{\pi_b}$ is the distribution of states observed when executing $\pi_b$.

The same application of Hoeffding’s can be used for Lemmas 1 and 2, Corollary 1, and Theorem 1.

B Additional Experimental Details

B.1 Extended Domain Descriptions

Here we provide additional domain details. Figure 4 illustrates the ModelWin and Cliffworld domains.
we do not use this domain. Building models with regression over nonlinear basis functions in CliffWorld is
analogous to ModelFail for illustrating a setting where models exhibit high bias. The final domain is a continuous two-dimensional cliff world domain (Figure 4(b)). An agent’s state is a four dimensional vector of horizontal and vertical position and velocity. Actions are acceleration values in the horizontal and vertical direction. The reward is negative and proportional to the agent’s distance to the goal magnetism: $r(s_t, a_t) = ||s_t - g||_1 + ||a_t||_1$. Not pictured: we also present empirical results in the canonical mountain car task.

**ModelWin**  ModelWin (Figure 4(a)) is a simple three state MDP that was developed by [16] to illustrate the settings where the MB estimator would outperform methods using IS. We replicate their work as described in Appendix D.2. of their paper. The small size of ModelWin means an exact model can be learned quickly and correctly with a tabular model. In this domain $\pi_e$ chooses action $a_1$ with probability 0.27 and $a_2$ with probability 0.73 in $s_1$. Policy $\pi_b$ does the opposite. These probabilities were arbitrarily chosen. This difference in policy action selection combined with a horizon of $L = 20$ results in IS methods having high variance. Therefore, the MB estimator has lower variance than methods that use IS. MB estimates $V(\pi_e)$ with Monte Carlo rollouts with simulated model trajectories. WDR uses value iteration to compute the state-value function and state-action-value function.

**Mountain Car**  Our second domain is the standard Mountain Car domain [15]. In this domain an agent attempts to drive an underpowered car up a hill. The car cannot drive straight up the hill and a successful policy must first move in reverse up another hill in order to gain momentum to reach its goal. States are discretized horizontal position and velocity and the agent may choose to accelerate left, right, or neither. At each time-step the reward is $-1$ except for in a terminal state when it is 0. We build models as done by Jiang and Li [7] where a lack of data for a $(s, a)$ pair causes a deterministic transition to $s$. Also, as in previous work on importance sampling, we shorten the horizon of the problem by holding action $a_t$ constant for 4 updates of the environment state $[7, 19]$. This modification changes the problem horizon to $L = 100$. Policy $\pi_a$ chooses actions uniformly at random and $\pi_e$ is a sub-optimal but better policy than $\pi_a$. MB estimates $V(\pi_e)$ with Monte Carlo rollouts with simulated model trajectories. WDR uses value iteration to compute the state-value function and state-action-value function.

**CliffWorld**  The final domain is a continuous two-dimensional cliffworld domain (Figure 4(b)). An agent’s state is a four dimensional vector of horizontal and vertical position and velocity. Actions are acceleration values in the horizontal and vertical direction. The reward is negative and proportional to the agent’s distance to the goal and magnitude of the actions taken, $r(s_t, a_t) = ||s_t - g||_1 + ||a_t||_1$. If the agent falls off a cliff it receives a large negative penalty. In this domain, we hand code a deterministic policy, $\pi_d$. Then the agent samples $\pi_e(s)$ by sampling from $\mathcal{N}(\pi_e(s), \Sigma)$. The behavior policy is the same except the Gaussian has greater variance. Domain dynamics are linear with additive Gaussian noise. These dynamics mean that the bootstrap models of MB-Bootstrap$^R$ and WDR-Bootstrap$^R$ will quickly converge to a correct model as the amount of data increases since they build models with linear regression. On the other hand, these dynamics mean that the models of MB-Bootstrap$^P$ and WDR-Bootstrap$^P$ will quickly converge to an incorrect model since they use regression over nonlinear polynomial basis functions (3$^{rd}$ order).

### B.2 Extended Empirical Procedure

For each domain we computed the lower bound for $n$ trajectories where $n$ varied logarithmically from 2 to 20000. The exception is CliffWorld which only went up to 6324 due to longer experiment run times. For each $n$ we generate a set of $n$ trajectories $m$ times and compute the lower bound with each method (e.g. MB, WDR, IS) on that set of trajectories. For ModelWin $m = 500$, Mountain Car $m = 400$, and CliffWorld $m = 50$. The
large number of trials is required for the empirical error rate calculations done in the ModelWin and Mountain Car domains. When plotting the average lower bound across methods, we only average valid lower bounds (i.e. $\hat{V}_{\pi_e}(\pi_e) \leq V(\pi_e)$). The reason for this is that invalid lower bounds raise the average which can make a method appear to produce a tighter average lower bound when it fact it has a higher error rate.

### B.3 Value Function Estimation in Continuous Domains

A challenge for WDR in continuous domains is that it must learn a model and the value functions for that model. We consider two methods for learning model value-functions. The first is simulating trajectories with $\pi_e$ in the model (i.e. $H \sim \hat{p}_{\pi_e}$) and using the sum of future returns at each time-step as a target for a learning the function $\hat{V}_{\pi_e}^t$. Specifically we create a data set of $\{(q_h^t | s_h^t, a_h^t)\}$ and $\{(v_h^t | s_h^t)\}$. These estimates can be used as targets for value-function regression (i.e. mapping state-action pairs or states to values). We attempted this approach using two-layer neural networks to represent the value functions but found the neural network approximation error negatively impacted WDR. Therefore, we instead used Monte Carlo estimates of $\hat{Q}_{\pi_e}^t(S, A)$ and $\hat{V}_{\pi_e}^t(S)$. In general, using Monte Carlo estimates with WDR is computationally prohibitive since we need two MC estimates for every state-action pair that occurs along trajectories in $D$. However, the nature of the domain (linear dynamics) and policies (mean action with additive Gaussian noise) allows an estimate of the value function under the model with a single rollout. When simulating trajectories with the model we take the expected action from $\pi_e(\cdot | S_t)$ and update the next state to the maximum likelihood state under the model transition function. We thus obtain the mean trajectory with a single rollout. If our cost function was linear we would have an unbiased estimate of $\hat{V}_{\pi_e}$. Since we have a non-linear cost function $\hat{V}_{\pi_e}$ and $\hat{Q}_{\pi_e}$ will be biased with only this single rollout. Nevertheless, in this domain the bias is significantly less than the error introduced by function approximation.

### C BCa Bootstrapping

This appendix contains the pseudocode for BCa Bootstrapping with an off-policy estimator (Algorithm 2). In our experiments only the IS methods used BCa for three reasons:

1. Thomas et al. used BCa when demonstrating that bootstrapping could outperform other confidence interval methods by an order of magnitude less data.
2. Importance sampled returns have high variance which can be problematic for a percentile bootstrap [1].
3. MB and WDR require more computation and the number of calls to **Off-PolicyEstimate** is $O(n)$ where $n$ is the number of trajectories in $D$. 


Algorithm 2 BCa Bootstrap Confidence Interval

**input** $\pi_e, D, \delta, B$

**output** $1 - \delta$ confidence interval lower bound on $V(\pi_e)$.

1: $\hat{\theta} \leftarrow \text{Off-PolicyEstimate}(\pi_e, D)$
2: for all $i \in [1, B]$ do
3:   $D_i = \{H_1^i, ..., H_n^i\}$ where $H_j^i \sim D$
4:   $\hat{\theta}_i \leftarrow \text{Off-PolicyEstimate}(\pi_e, D_i)$
5: end for
6: sort($\{\hat{\theta}_i | i \in [1, B]\}$)
7: $z_0 \leftarrow \Phi^{-1}(\frac{\#h_i < \hat{\theta}}{B})$
8: for all $i \in [1, n]$ do
9:   $y_i \leftarrow \frac{1}{n} \text{Off-PolicyEstimate}(\pi_e, D / \{H_i\}, \pi_b)$
10: end for
11: $\bar{y} \leftarrow \frac{1}{n} \sum_{i=1}^{n} y_i$
12: $\alpha \leftarrow \frac{\sum_{i=1}^{n} (\bar{y} - y_i)^3}{6 \sum_{i=1}^{n} (\bar{y} - y_i)^2}^{\frac{3}{2}}$
13: $z_L \leftarrow z_0 - \frac{\Phi^{-1}(1-\delta) - z_0}{1 + q(\Phi^{-1}(1-\delta) - z_0)}$
14: $Q \leftarrow (B + 1) \Phi(z_L)$
15: $l \leftarrow \min\{\lfloor Q \rfloor, B - 1\}$
16: Return $\hat{\theta} + \frac{\Phi^{-1}(\frac{Q}{B+1}) - \Phi^{-1}(\frac{l}{B+1})}{\Phi^{-1}(\frac{Q}{B+1}) - \Phi^{-1}(\frac{l}{B+1})}$