Off-Policy Evaluation for Large Action Spaces via Embeddings

Yuta Saito [1] **Thorsten Joachims** [1]

Abstract

Off-policy evaluation (OPE) in contextual bandits has seen rapid adoption in real-world systems, since it enables offline evaluation of new policies using only historic log data. Unfortunately, when the number of actions is large, existing OPE estimators – most of which are based on inverse propensity score weighting – degrade severely and can suffer from extreme bias and variance. This foils the use of OPE in many applications from recommender systems to language models. To overcome this issue, we propose a new OPE estimator that leverages *marginalized* importance weights when *action embeddings* provide structure in the action space. We characterize the bias, variance, and mean squared error of the proposed estimator and analyze the conditions under which the action embedding provides statistical benefits over conventional estimators. In addition to the theoretical analysis, we find that the empirical performance improvement can be substantial, enabling reliable OPE even when existing estimators collapse due to a large number of actions.

1. Introduction

Many intelligent systems (e.g., recommender systems, voice assistants, search engines) interact with the environment through a *contextual bandit* process where a policy observes a context, takes an action, and obtains a reward. Logs of these interactions provide valuable data for *off-policy evaluation* (OPE), which aims to accurately evaluate the performance of new policies without ever deploying them in the field. OPE is of great practical relevance, as it helps avoid costly online A/B tests and can also act as subroutines for batch policy learning (Dudík et al., 2014; Su et al., 2020a). However, OPE is challenging, since the logs contain only partial-information feedback – specifically the reward of the chosen action, but not the counterfactual rewards of all the other actions a different policy might choose.

When the action space is small, recent advances in the design of OPE estimators have led to a number of reliable methods with good theoretical guarantees (Dudík et al., 2014; Swaminathan & Joachims, 2015a; Wang et al., 2017; Farajtabar et al., 2018; Su et al., 2019; 2020a; Metelli et al., 2021). Unfortunately, these estimators can degrade severely when the number of available actions is large. Large action spaces are prevalent in many potential applications of OPE, such as recommender systems where policies have to handle thousands or millions of items (e.g., movies, songs, products). In such a situation, the existing estimators based on *inverse propensity score* (IPS) weighting (Horvitz & Thompson, 1952) can incur high bias and variance, and as a result, be impractical for OPE. First, a large action space makes it challenging for the logging policy to have common support with the target policies, and IPS is biased under support deficiency (Sachdeva et al., 2020). Second, a large number of actions typically leads to high variance of IPS due to large importance weights. To illustrate, we find in our experiments that the variance and mean squared error of IPS inflate by a factor of over 300 when the number of actions increases from 10 to 5000 given a fixed sample size. While doubly robust (DR) estimators can somewhat reduce the variance by introducing a reward estimator as a control variate (Dudík et al., 2014), they do not address the fundamental issues that come with large action spaces.

To overcome the limitations of the existing estimators when the action space is large, we leverage additional information about the actions in the form of *action embeddings*. There are many cases where we have access to such prior information. For example, movies are characterized by auxiliary information such as genres (e.g., adventure, science fiction, documentary), director, or actors. We should then be able to utilize these supplemental data to infer the value of actions under-explored by the logging policy, potentially achieving much more accurate policy evaluation than the existing estimators. We first provide the conditions under which action embeddings provide another path for unbiased OPE, even with support deficient actions. We then propose the *Marginalized IPS* (MIPS) estimator, which uses the *marginal* distribution of action embeddings, rather than actual actions, to define a new type of importance weights. We

[1]Department of Computer Science, Cornell University, Ithaca, NY, USA. Correspondence to: Yuta Saito <ys552@cornell.edu>, Thorsten Joachims <tj@cs.cornell.edu>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

show that MIPS is unbiased under an alternative condition, which states that the action embeddings should mediate every causal effect of the action on the reward. Moreover, we show that MIPS has a lower variance than IPS, especially when there is a large number of actions, and thus the vanilla importance weights have a high variance. We also characterize the gain in MSE provided by MIPS, which implies an interesting bias-variance trade-off with respect to the quality of the action embeddings. Including many embedding dimensions captures the causal effect better, leading to a smaller bias of MIPS. In contrast, using only a subset of the embedding dimensions reduces the variance more. We thus propose a strategy to intentionally violate the assumption about the action embeddings by discarding less relevant embedding dimensions for achieving a better MSE at the cost of introducing some bias. Comprehensive experiments on synthetic and real-world bandit data verify the theoretical findings, indicating that MIPS can provide an effective bias-variance trade-off in the presence of many actions.

2. Off-Policy Evaluation

We follow the general contextual bandit setup, and an extensive discussion of related work is given in Appendix A. Let $x \in \mathcal{X} \subseteq \mathbb{R}^{d_x}$ be a d_x-dimensional context vector drawn i.i.d. from an unknown distribution $p(x)$. Given context x, a possibly stochastic *policy* $\pi(a|x)$ chooses action a from a finite action space denoted as $\mathcal{A}$. The reward $r \in [0, r_{\max}]$ is then sampled from an unknown conditional distribution $p(r|x,a)$. We measure the effectiveness of a policy π through its *value*

$$V(\pi) := \mathbb{E}_{p(x)\pi(a|x)p(r|x,a)}[r] = \mathbb{E}_{p(x)\pi(a|x)}[q(x,a)], \tag{1}$$

where $q(x,a) := \mathbb{E}[r|x,a]$ denotes the expected reward given context x and action a.

In OPE, we are given logged bandit data collected by a logging policy π_0. Specifically, let $\mathcal{D} := \{(x_i, a_i, r_i)\}_{i=1}^n$ be a sample of logged bandit data containing n independent observations drawn from the logging policy as $(x,a,r) \sim p(x)\pi_0(a|x)p(r|x,a)$. We aim to develop an estimator $\hat{V}$ for the value of a target policy π (which is different from π_0) using only the logged data in $\mathcal{D}$. The accuracy of $\hat{V}$ is quantified by its mean squared error (MSE)

$$\begin{aligned}\mathrm{MSE}(\hat{V}(\pi)) :&= \mathbb{E}_{\mathcal{D}}\Big[\big(V(\pi) - \hat{V}(\pi;\mathcal{D})\big)^2\Big] \\ &= \mathrm{Bias}(\hat{V}(\pi))^2 + \mathbb{V}_{\mathcal{D}}\big[\hat{V}(\pi;\mathcal{D})\big],\end{aligned}$$

where $\mathbb{E}_{\mathcal{D}}[\cdot]$ takes the expectation over the logged data and

$$\begin{aligned}\mathrm{Bias}(\hat{V}(\pi)) &:= \mathbb{E}_{\mathcal{D}}[\hat{V}(\pi;\mathcal{D})] - V(\pi), \\ \mathbb{V}_{\mathcal{D}}\big[\hat{V}(\pi;\mathcal{D})\big] &:= \mathbb{E}_{\mathcal{D}}\Big[\big(\hat{V}(\pi;\mathcal{D}) - \mathbb{E}_{\mathcal{D}}[\hat{V}(\pi;\mathcal{D})]\big)^2\Big].\end{aligned}$$

In the following theoretical analysis, we focus on the IPS estimator, since most advanced OPE estimators are based on IPS weighting (Dudík et al., 2014; Wang et al., 2017; Su et al., 2019; 2020a; Metelli et al., 2021). IPS estimates the value of π by re-weighting the observed rewards as follows.

$$\hat{V}_{\mathrm{IPS}}(\pi;\mathcal{D}) := \frac{1}{n}\sum_{i=1}^n \frac{\pi(a_i|x_i)}{\pi_0(a_i|x_i)} r_i = \frac{1}{n}\sum_{i=1}^n w(x_i,a_i) r_i$$

where $w(x,a) := \pi(a|x)/\pi_0(a|x)$ is called the *(vanilla) importance weight*.

This estimator is unbiased (i.e., $\mathbb{E}_{\mathcal{D}}[\hat{V}_{\mathrm{IPS}}(\pi;\mathcal{D})] = V(\pi)$) under the following common support assumption.

Assumption 2.1. (Common Support) The logging policy π_0 is said to have common support for policy π if $\pi(a|x) > 0 \rightarrow \pi_0(a|x) > 0$ for all $a \in \mathcal{A}$ and $x \in \mathcal{X}$.

The unbiasedness of IPS is desirable, making this simple re-weighting technique so popular. However, IPS can still be highly biased, particularly when the action space is large. Sachdeva et al. (2020) indicate that IPS has the following bias when Assumption 2.1 is not true.

$$\big|\mathrm{Bias}(\hat{V}_{\mathrm{IPS}}(\pi))\big| = \mathbb{E}_{p(x)}\left[\sum_{a \in \mathcal{U}_0(x,\pi_0)} \pi(a|x) q(x,a)\right],$$

where $\mathcal{U}_0(x,\pi_0) := \{a \in \mathcal{A} \mid \pi_0(a|x) = 0\}$ is the set of unsupported or deficient actions for context x under π_0. Note that $\mathcal{U}_0(x,\pi_0)$ can be large especially when $\mathcal{A}$ is large. This bias is due to the fact that the logged dataset $\mathcal{D}$ does not contain any information about the unsupported actions.

Another critical issue of IPS is that its variance can be large, which is given as follows (Dudík et al., 2014).

$$\begin{aligned}n\mathbb{V}_{\mathcal{D}}\big[\hat{V}_{\mathrm{IPS}}(\pi;\mathcal{D})\big] &= \mathbb{E}_{p(x)\pi_0(a|x)}[w(x,a)^2\sigma^2(x,a)] \\ &\quad + \mathbb{V}_{p(x)}\left[\mathbb{E}_{\pi_0(a|x)}[w(x,a)q(x,a)]\right] \\ &\quad + \mathbb{E}_{p(x)}\left[\mathbb{V}_{\pi_0(a|x)}[w(x,a)q(x,a)]\right],\end{aligned} \tag{2}$$

where $\sigma^2(x,a) := \mathbb{V}[r|x,a]$. The variance consists of three terms. The first term reflects the randomness in the rewards. The second term represents the variance due to the randomness over the contexts. The final term is the penalty arising from the use of IPS weighting, and it is proportional to the weights and the true expected reward. The variance contributed by the first and third terms can be extremely large when the weights $w(x,a)$ have a wide range, which occurs when π assigns large probabilities to actions that have low probability under π_0. The latter can be expected when the action space $\mathcal{A}$ is large and the logging policy π_0 aims to have *universal support* (i.e., $\pi_0(a|x) > 0$ for all a and x). Swaminathan et al. (2017) also point out that the variance of IPS grows linearly with $w(x,a)$, which can be $\Omega(|\mathcal{A}|)$.

This variance issue can be lessened by incorporating a reward estimator $\hat{q}(x,a) \approx q(x,a)$ as a control variate, resulting in the DR estimator (Dudík et al., 2014). DR often improves the MSE of IPS due to its variance reduction property. However, DR still suffers when the number of actions is large, and it can experience substantial performance deterioration as we demonstrate in our experiments.

3. The Marginalized IPS Estimator

The following proposes a new estimator that circumvents the challenges of IPS for large action spaces. Our approach is to bring additional structure into the estimation problem, providing a path forward despite the minimax optimality of IPS and DR. In particular, IPS and DR achieve the minimax optimal MSE of at most $\mathcal{O}(n^{-1}(\mathbb{E}_{\pi_0}[w(x,a)^2\sigma^2(x,a) + w(x,a)^2 r_{\max}^2]))$, which means that they are impossible to improve upon in the worst case beyond constant factors (Wang et al., 2017; Swaminathan et al., 2017), unless we bring in additional structure.

Our key idea for overcoming the limits of IPS and DR is to assume the existence of *action embeddings* as prior information. The intuition is that this can help the estimator transfer information between similar actions. More formally, suppose we are given a d_e-dimensional *action embedding* $e \in \mathcal{E} \subseteq \mathbb{R}^{d_e}$ for each action a, where we merely assume that the embedding is drawn i.i.d. from some unknown distribution $p(e|x,a)$. The simplest example is to construct action embeddings using predefined category information (e.g., product category). Then, the embedding distribution is independent of the context and it is deterministic given the action. Our framework is also applicable to the most general case of continuous, stochastic, and context-dependent action embeddings. For example, product prices may be generated by a personalized pricing algorithm running behind the system. In this case, the embedding is continuous, depends on the user context, and can be stochastic if there is some randomness in the pricing algorithm.

Using the action embeddings, we now refine the definition of the policy value as:

$$V(\pi) = \mathbb{E}_{p(x)\pi(a|x)p(e|x,a)p(r|x,a,e)}[r].$$

Note here that $q(x,a) = \mathbb{E}_{p(e|x,a)}[q(x,a,e)]$ where $q(x,a,e) := \mathbb{E}[r|x,a,e]$, so the above refinement does not contradict the original definition given in Eq. (1).

A logged bandit dataset now contains action embeddings for each observation in $\mathcal{D} = \{(x_i, a_i, e_i, r_i)\}_{i=1}^n$, where each tuple is generated by the logging policy as $(x,a,e,r) \sim p(x)\pi_0(a|x)p(e|x,a)p(r|x,a,e)$. Our strategy is to leverage this additional information for achieving a more accurate OPE for large action spaces.

To motivate our approach, we introduce two properties characterizing an action embedding.

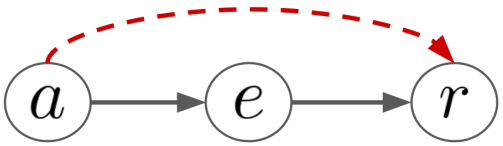

Figure 1. Causal Graph Consistent with Assumption 3.2

Note: Grey arrows indicate the existence of causal effect of the tail variable on the head variable. The dashed red arrow is a direct causal effect that is ruled out by Assumption 3.2.

Assumption 3.1. (Common Embedding Support) The logging policy π_0 is said to have common embedding support for policy π if $p(e|x,\pi) > 0 \rightarrow p(e|x,\pi_0) > 0$ for all $e \in \mathcal{E}$ and $x \in \mathcal{X}$, where $p(e|x,\pi) := \sum_{a \in \mathcal{A}} p(e|x,a)\pi(a|x)$ is the *marginal* distribution over the action embedding space given context x and policy π.

Assumption 3.1 is analogous to Assumption 2.1, but requires only the common support with respect to the action embedding space, which can be substantially more compact than the action space itself. Indeed, Assumption 3.1 is weaker than common support of IPS (Assumption 2.1).[1] Next, we characterize the expressiveness of the embedding in the ideal case, but we will relax this assumption later.

Assumption 3.2. (No Direct Effect) Action a has no direct effect on the reward r, i.e., $a \perp r \mid x, e$.

As illustrated in Figure 1, Assumption 3.2 requires that every possible effect of a on r be fully mediated by the observed embedding e. For now, we rely on the validity of Assumption 3.2, as it is convenient for introducing the proposed estimator. However, we later show that it is often beneficial to strategically discard some embedding dimensions and violate the assumption to achieve a better MSE.

We start the derivation of our new estimator with the observation that Assumption 3.2 gives us another path to unbiased estimation of the policy value without Assumption 2.1.

Proposition 3.3. *Under Assumption 3.2, we have*

$$V(\pi) = \mathbb{E}_{p(x)p(e|x,\pi)p(r|x,e)}[r]$$

See Appendix B.1 for the proof.

Proposition 3.3 provides another expression of the policy value *without* explicitly relying on the action variable a. This new expression naturally leads to the following *marginalized inverse propensity score* (MIPS) estimator,

[1] First, if Assumption 2.1 is true, Assumption 3.1 is also true because $p(e|x,a)$ remains the same for the target and logging policies. Table 1 will provide a counterexample for the opposite statement (i.e., Assumption 3.1 does not imply Assumption 2.1).

Table 1. A toy example illustrating the benefits of marginal importance weights

	$\pi_0(a\|x_1)$	$\pi(a\|x_1)$	$w(x_1,a)$
a_1	0.0	0.2	N/A
a_2	0.2	0.8	4.0
a_3	0.8	0.0	0.0

	$p(e_1\|a)$	$p(e_2\|a)$	$p(e_3\|a)$
a_1	0.25	0.25	0.5
a_2	0.5	0.25	0.25
a_3	0.25	0.5	0.25

	$p(e\|x_1,\pi_0)$	$p(e\|x_1,\pi)$	$w(x_1,e)$
e_1	0.3	0.45	1.5
e_2	0.45	0.25	0.55
e_3	0.25	0.3	1.2

which is our main proposal.

$$\hat{V}_{\mathrm{MIPS}}(\pi;\mathcal{D}) := \frac{1}{n}\sum_{i=1}^{n}\frac{p(e_i|x_i,\pi)}{p(e_i|x_i,\pi_0)}r_i = \frac{1}{n}\sum_{i=1}^{n} w(x_i,e_i)r_i,$$

where $w(x,e) := p(e|x,\pi)/p(e|x,\pi_0)$ is the *marginal importance weight* defined with respect to the marginal distribution over the action embedding space.

To obtain an intuition for the benefits of MIPS, we provide a toy example in Table 1 with $\mathcal{X}=\{x_1\}$, $\mathcal{A}=\{a_1,a_2,a_3\}$, and $\mathcal{E}=\{e_1,e_2,e_3\}$ (a special case of our formulation with a discrete embedding space). The left table describes the logging and target policies with respect to $\mathcal{A}$ and implies that Assumption 2.1 is violated ($\pi_0(a_1|x_1)=0.0$). The middle table describes the conditional distribution of the action embedding e given action a (e.g., probability of a movie a belonging to a genre e). The right table describes the marginal distributions over $\mathcal{E}$, which are calculable from the other two tables. By considering the marginal distribution, Assumption 3.1 is ensured in the right table, even if Assumption 2.1 is not true in the left table. Moreover, the maximum importance weight is smaller for the right table ($\max_{e\in\mathcal{E}} w(x_1,e) < \max_{a\in\mathcal{A}} w(x_1,a)$), which may contribute to a variance reduction of the resulting estimator.

Below, we formally analyze the key statistical properties of MIPS and compare them with those of IPS, including the realistic case where Assumption 3.2 is violated.

3.1. Theoretical Analysis

First, the following proposition shows that MIPS is unbiased under assumptions different from those of IPS.

Proposition 3.4. *Under Assumptions 3.1 and 3.2, MIPS is unbiased, i.e.,* $\mathbb{E}_{\mathcal{D}}[\hat{V}_{\mathrm{MIPS}}(\pi;\mathcal{D})] = V(\pi)$ *for any* π. *See Appendix B.2 for the proof.*

Proposition 3.4 states that, even when π_0 fails to provide common support over $\mathcal{A}$ such that IPS is biased, MIPS can still be unbiased if π_0 provides common support over $\mathcal{E}$ (Assumption 3.1) and e fully captures the causal effect of a on r (Assumption 3.2).

Having multiple estimators that enable unbiased OPE under different assumptions is in itself desirable, as we can choose the appropriate estimator depending on the data generating process. However, it is also helpful to understand *how violations of the assumptions influence the bias of the resulting estimator*. In particular, for MIPS, it is difficult to verify whether Assumption 3.2 is true in practice. The following theorem characterizes the bias of MIPS.

Theorem 3.5. *(Bias of MIPS) If Assumption 3.1 is true, but Assumption 3.2 is violated, MIPS has the following bias.*

$$\begin{aligned}&\mathrm{Bias}(\hat{V}_{\mathrm{MIPS}}(\pi))\\ &= \mathbb{E}_{p(x)p(e|x,\pi_0)}\Bigg[\sum_{a<b}\pi_0(a|x,e)\pi_0(b|x,e)\\ &\qquad\qquad\times(q(x,a,e)-q(x,b,e))\\ &\qquad\qquad\times(w(x,b)-w(x,a))\Bigg],\end{aligned}$$

where $a,b\in\mathcal{A}$. *See Appendix B.3 for the proof.*

Theorem 3.5 suggests that three factors contribute to the bias of MIPS when Assumption 3.2 is violated. The first factor is the *predictivity of the action embeddings with respect to the actual actions*. When action a is predictable given context x and embedding e, $\pi_0(a|x,e)$ is close to zero or one (deterministic), meaning that $\pi_0(a|x,e)\pi_0(b|x,e)$ is close to zero. This suggests that even if Assumption 3.2 is violated, action embeddings that identify the actions well still enable a nearly unbiased estimation of MIPS. The second factor is the *amount of direct effect of the action on the reward*, which is quantified by $q(x,a,e)-q(x,b,e)$. When the direct effect of a on r is small, $q(x,a,e)-q(x,b,e)$ also becomes small and so is the bias of MIPS. In an ideal situation where Assumption 3.2 is satisfied, we have $q(x,a,e)=q(x,b,e)=q(x,e)$, thus MIPS is unbiased, which is consistent with Proposition 3.4. Note that the first two factors suggest that, to reduce the bias, the action embeddings should be *informative* so that they are either predictive of the actions or mediate a large amount of the causal effect. The final factor is the *similarity between logging and target policies* quantified by $w(x,a)-w(x,b)$. When Assumption 3.2 is satisfied, MIPS is unbiased for any target policy, however, Theorem 3.5 suggests that if the assumption is not true, MIPS produces a larger bias for target policies dissimilar from the logging policy.[2]

[2]When $\pi=\pi_0$, the bias is zero regardless of the other factors as $w(x,a)=w(x,b)=1$, meaning that *on-policy estimation* is always unbiased, which is quite intuitive.

Next, we analyze the variance of MIPS, which we show is never worse than that of IPS and can be substantially lower.

Theorem 3.6. *(Variance Reduction of MIPS) Under Assumptions 2.1, 3.1, and 3.2, we have*

$$n\left(\mathbb{V}_{\mathcal{D}}[\hat{V}_{\mathrm{IPS}}(\pi;\mathcal{D})]-\mathbb{V}_{\mathcal{D}}[\hat{V}_{\mathrm{MIPS}}(\pi;\mathcal{D})]\right) = \mathbb{E}_{p(x)p(e|x,\pi_0)}\left[\mathbb{E}_{p(r|x,e)}\left[r^2\right]\mathbb{V}_{\pi_0(a|x,e)}\left[w(x,a)\right]\right],$$

which is non-negative. Note that the variance reduction is also lower bounded by zero even when Assumption 3.2 is not true. See Appendix B.4 for the proof.

There are two factors that affect the amount of variance reduction. The first factor is the second moment of the reward with respect to $p(r|x,e)$. This term becomes large when, for example, the reward is noisy even after conditioning on the action embedding e. The second factor is the variance of $w(x,a)$ with respect to the conditional distribution $\pi_0(a|x,e)$, which becomes large when **(i)** $w(x,a)$ has a wide range or **(ii)** there remain large variations in a even after conditioning on action embedding e so that $\pi_0(a|x,e)$ remains stochastic. Therefore, MIPS becomes increasingly favorable compared to IPS for larger action spaces where the variance of $w(x,a)$ becomes larger. Moreover, to obtain a large variance reduction, the action embedding should ideally not be unnecessarily predictive of the actions.

Finally, the next theorem describes the gain in MSE we can obtain from MIPS when Assumption 3.2 is violated.

Theorem 3.7. *(MSE Gain of MIPS) Under Assumptions 2.1 and 3.1, we have*

$$n\left(\mathrm{MSE}(\hat{V}_{\mathrm{IPS}}(\pi))-\mathrm{MSE}(\hat{V}_{\mathrm{MIPS}}(\pi))\right) = \mathbb{E}_{x,a,e\sim\pi_0}\left[\left(w(x,a)^2-w(x,e)^2\right)\cdot\mathbb{E}_{p(r|x,a,e)}[r^2]\right] + 2V(\pi)\mathrm{Bias}(\hat{V}_{\mathrm{MIPS}}(\pi)) + (1-n)\mathrm{Bias}(\hat{V}_{\mathrm{MIPS}}(\pi))^2.$$

See Appendix B.5 for the proof.

Note that IPS can have some bias when Assumption 2.1 is not true, possibly producing a greater MSE gain for MIPS.

3.2. Data-Driven Embedding Selection

The analysis in the previous section implies a clear bias-variance trade-off with respect to the *quality of the action embeddings*. Specifically, Theorem 3.5 suggests that the action embeddings should be as *informative* as possible to reduce the bias when Assumption 3.2 is violated. On the other hand, Theorem 3.6 suggests that the action embeddings should be as *coarse* as possible to gain a greater variance reduction. Theorem 3.7 summarizes the bias-variance trade-off in terms of MSE.

A possible criticism to MIPS is Assumption 3.2, as it is hard to verify whether this assumption is satisfied using only the observed logged data. However, the above discussion about the bias-variance trade-off implies that it might be effective to strategically violate Assumption 3.2 by discarding some embedding dimensions. This *action embedding selection* can lead to a large variance reduction at the cost of introducing some bias, possibly improving the MSE of MIPS. To implement the action embedding selection, we can adapt the estimator selection method called SLOPE proposed in Su et al. (2020b) and Tucker & Lee (2021). SLOPE is based on Lepski's principle for bandwidth selection in nonparametric statistics (Lepski & Spokoiny, 1997) and is used to tune the hyperparameters of OPE estimators. A benefit of SLOPE is that it avoids estimating the bias of the estimator, which is as difficult as OPE. Appendix C describes how to apply SLOPE to the action embedding selection in our setup, and Section 4 evaluates its benefit empirically.

3.3. Estimating the Marginal Importance Weights

When using MIPS, we might have to estimate $w(x,e)$ depending on how the embeddings are given. A simple approach to this is to utilize the following transformation.

$$w(x,e)=\mathbb{E}_{\pi_0(a|x,e)}\left[w(x,a)\right]. \tag{3}$$

Eq. (3) implies that we need an estimate of $\pi_0(a|x,e)$, which we compute by regressing a on (x,e). We can then estimate $w(x,e)$ as $\hat{w}(x,e)=\mathbb{E}_{\hat{\pi}_0(a|x,e)}\left[w(x,a)\right]$.[3] This procedure is easy to implement and tractable, even when the embedding space is high-dimensional and continuous. Note that, even if there are some deficient actions, we can directly estimate $w(x,e)$ by solving density ratio estimation as binary classification as done in Sondhi et al. (2020).

4. Empirical Evaluation

We first evaluate MIPS on synthetic data to identify the situations where it enables a more accurate OPE. Second, we validate real-world applicability on data from an online fashion store. Our experiments are conducted using the *OpenBandit-Pipeline* (OBP)[4], an open-source software for OPE provided by Saito et al. (2020). Our experiment implementation is available at https://github.com/usaito/icml2022-mips.

4.1. Synthetic Data

For the first set of experiments, we create synthetic data to be able to compare the estimates to the ground-truth value of the target policies. To create the data, we sample 10-dimensional context vectors x from the standard normal distribution. We also sample d_e-dimensional categorical action embedding $e\in\mathcal{E}$ from the following conditional

[3] Appendix B.7 describes the bias and variance of MIPS with estimated marginal importance weights $\hat{w}(x,e)$.

[4] https://github.com/st-tech/zr-obp

distribution given action a.

$$p(e \mid a) = \prod_{k=1}^{d_e} \frac{\exp(\alpha_{a,e_k})}{\sum_{e' \in \mathcal{E}_k} \exp(\alpha_{a,e'})}, \tag{4}$$

which is independent of the context x in the synthetic experiment. $\{\alpha_{a,e_k}\}$ is a set of parameters sampled independently from the standard normal distribution. Each dimension of $\mathcal{E}$ has a cardinality of 10, i.e., $\mathcal{E}_k = \{1, 2, \ldots, 10\}$. We then synthesize the expected reward as

$$q(x, e) = \sum_{k=1}^{d_e} \eta_k \cdot \left(x^\top M x_{e_k} + \theta_x^\top x + \theta_e^\top x_{e_k} \right), \tag{5}$$

where M, θ_x, and θ_e are parameter matrices or vectors to define the expected reward. These parameters are sampled from a uniform distribution with range $[-1, 1]$. x_{e_k} is a context vector corresponding to the k-th dimension of the action embedding, which is unobserved to the estimators. η_k specifies the importance of the k-th dimension of the action embedding, which is sampled from Dirichlet distribution so that $\sum_{k=1}^{d_e} \eta_k = 1$. Note that if we observe all dimensions of $\mathcal{E}$, then $q(x, e) = q(x, a, e)$. On the other hand, $q(x, e) \neq q(x, a, e)$, if there are some missing dimensions, which means that Assumption 3.2 is violated.

We synthesize the logging policy π_0 by applying the softmax function to $q(x, a) = \mathbb{E}_{p(e|a)}[q(x, e)]$ as

$$\pi_0(a \mid x) = \frac{\exp(\beta \cdot q(x, a))}{\sum_{a' \in \mathcal{A}} \exp(\beta \cdot q(x, a'))}, \tag{6}$$

where β is a parameter that controls the optimality and entropy of the logging policy. A large positive value of β leads to a near-deterministic and well-performing logging policy, while lower values make the logging policy increasingly worse. In the main text, we use $\beta = -1$, and additional results for other values of β can be found in Appendix D.2.

In contrast, the target policy π is defined as

$$\pi(a \mid x) = (1 - \epsilon) \cdot \mathbb{I}\{a = \arg\max_{a' \in \mathcal{A}} q(x, a')\} + \epsilon / |\mathcal{A}|,$$

where the noise $\epsilon \in [0, 1]$ controls the quality of π. In the main text, we set $\epsilon = 0.05$, which produces a near-optimal and near-deterministic target policy. We share additional results for other values of ϵ in Appendix D.2.

To summarize, we first sample context x and define the expected reward $q(x, e)$ as in Eq. (5). We then sample discrete action a from π_0 based on Eq. (6). Given action a, we sample categorical action embedding e based on Eq. (4). Finally, we sample the reward from a normal distribution with mean $q(x, e)$ and standard deviation $\sigma = 2.5$. Iterating this procedure n times generates logged data $\mathcal{D}$ with n independent copies of (x, a, e, r).

4.1.1. Baselines

We compare our estimator with Direct Method (DM), IPS, and DR.[5] We use the Random Forest (Breiman, 2001) implemented in *scikit-learn* (Pedregosa et al., 2011) along with 2-fold cross-fitting (Newey & Robins, 2018) to obtain $\hat{q}(x, e)$ for DR and DM. We use the Logistic Regression of *scikit-learn* to estimate $\hat{\pi}_0(a|x, e)$ for MIPS. We also report the results of MIPS with the true importance weights as "MIPS (true)". MIPS (true) provides the best performance we could achieve by improving the procedure for estimating the importance weights of MIPS.

4.1.2. Results

The following reports and discusses the MSE, squared bias, and variance of the estimators computed over 100 different sets of logged data replicated with different seeds.

How does MIPS perform with varying numbers of actions? First, we evaluate the estimators' performance when we vary the number of actions from 10 to 5000. The sample size is fixed at $n = 10000$. Figure 2 shows how the number of actions affects the estimators' MSE (both on linear- and log-scale). We observe that MIPS provides substantial improvements over IPS and DR particularly for larger action sets. More specifically, when $|\mathcal{A}| = 10$, $\frac{\mathrm{MSE}(\hat{V}_{\mathrm{IPS}})}{\mathrm{MSE}(\hat{V}_{\mathrm{MIPS}})} = 1.38$, while $\frac{\mathrm{MSE}(\hat{V}_{\mathrm{IPS}})}{\mathrm{MSE}(\hat{V}_{\mathrm{MIPS}})} = 12.38$ for $|\mathcal{A}| = 5000$, indicating a significant performance improvement of MIPS for larger action spaces as suggested in Theorem 3.6. MIPS is also consistently better than DM, which suffers from high bias. The figure also shows that MIPS (true) is even better than MIPS in large action sets, mostly due to the reduced bias when using the true marginal importance weights. This observation implies that there is room for further improvement in how to estimate the marginal importance weights.

How does MIPS perform with varying sample sizes? Next, we compare the estimators under varying numbers of samples ($n \in \{800, 1600, 3200, 6400, 12800, 25600\}$). The number of actions is fixed at $|\mathcal{A}| = 1000$. Figure 3 reports how the estimators' MSE changes with the size of logged bandit data. We can see that MIPS is appealing in particular for small sample sizes where it outperforms IPS and DR by a larger margin than in large sample regimes

[5] Appendix D.2 provides more comprehensive experiment results including Switch-DR (Wang et al., 2017), DR with Optimistic Shrinkage (DRos) (Su et al., 2020a), and DR-λ (Metelli et al., 2021) as additional baseline estimators. The additional experimental results suggest that all of these existing estimators based on IPS weighting experience significant accuracy deterioration with large action spaces due to either large bias or variance. Moreover, we observe that MIPS is more robust and outperforms all these baselines in a range of settings.

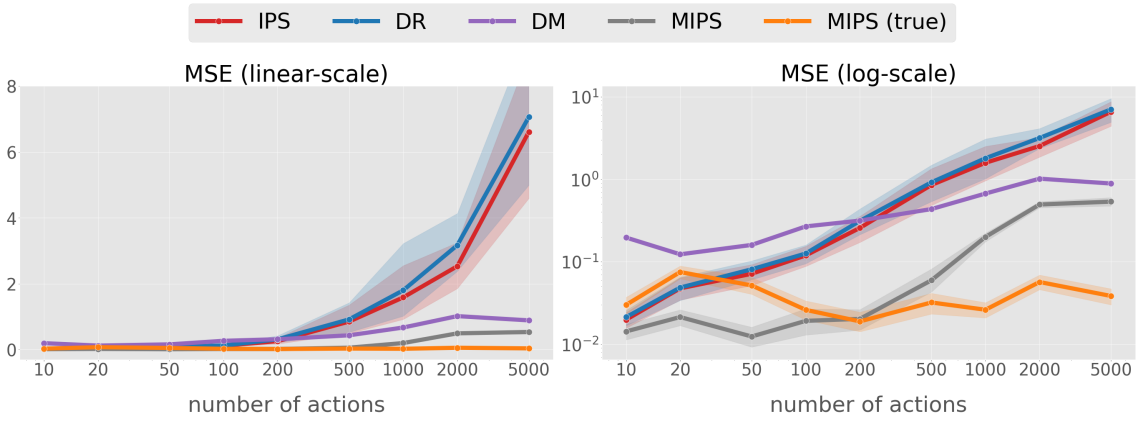

Figure 2. MSE (both on linear- and log-scale) with **varying number of actions**.

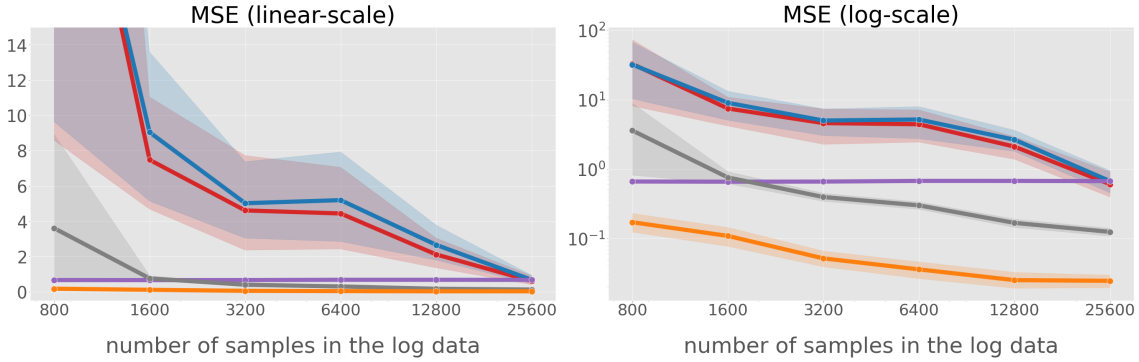

Figure 3. MSE (both on linear- and log-scale) with **varying number of samples**.

($\frac{\mathrm{MSE}(\hat{V}_{\mathrm{IPS}})}{\mathrm{MSE}(\hat{V}_{\mathrm{MIPS}})} = 9.10$ when $n = 800$, while $\frac{\mathrm{MSE}(\hat{V}_{\mathrm{IPS}})}{\mathrm{MSE}(\hat{V}_{\mathrm{MIPS}})} = 4.87$ when $n = 25600$). With the growing sample size, MIPS, IPS, and DR improve their MSE as their variance decreases. In contrast, the accuracy of DM does not change across different sample sizes, but it performs better than IPS and DR because they converge very slowly in the presence of many actions. In contrast, MIPS is better than DM except for $n = 800$, as the bias of MIPS is much smaller than that of DM. Moreover, MIPS becomes increasingly better than DM with the growing sample size, as the variance of MIPS decreases while DM remains highly biased.

How does MIPS perform with varying numbers of deficient actions? We also compare the estimators under varying numbers of deficient actions ($|\mathcal{U}_0| \in \{0, 100, 300, 500, 700, 900\}$) with a fixed action set ($|\mathcal{A}| = 1000$). Figure 4 shows how the number of deficient actions affects the estimators' MSE, squared bias, and variance. The results suggest that MIPS (true) is robust and not affected by the existence of deficient actions. In addition, MIPS is mostly better than DM, IPS, and DR even when there are many deficient actions. However, we also observe that the gap between MIPS and MIPS (true) increases for large numbers of deficient actions due to the bias in estimating the marginal importance weights. Note that the MSE of IPS and DR decreases with increasing number of deficient actions, because their variance becomes smaller with a smaller number of supported actions, even though their bias increases as suggested by Sachdeva et al. (2020).

How does MIPS perform when Assumption 3.2 is violated? Here, we evaluate the accuracy of MIPS when Assumption 3.2 is violated. To adjust the amount of violation, we modify the action embedding space and reduce the cardinality of each dimension of $\mathcal{E}$ to 2 (i.e., $\mathcal{E}_k = \{0, 1\}$), while we increase the number of dimensions to 20 ($d_e = 20$). This leads to $|\mathcal{E}| = 2^{20} = 1,048,576$, and we can now drop some dimensions to increase violation. In particular, when we observe all dimensions of $\mathcal{E}$, Assumption 3.2 is perfectly satisfied. However, when we withhold $\{0, 2, 4, \ldots, 18\}$ embedding dimensions, the assumption becomes increasingly invalid. When many dimensions are missing, the bias of MIPS is expected to increase as suggested in Theorem 3.5, potentially leading to a worse MSE.

Figure 5 shows how the MSE, squared bias, and variance of the estimators change with varying numbers of unobserved

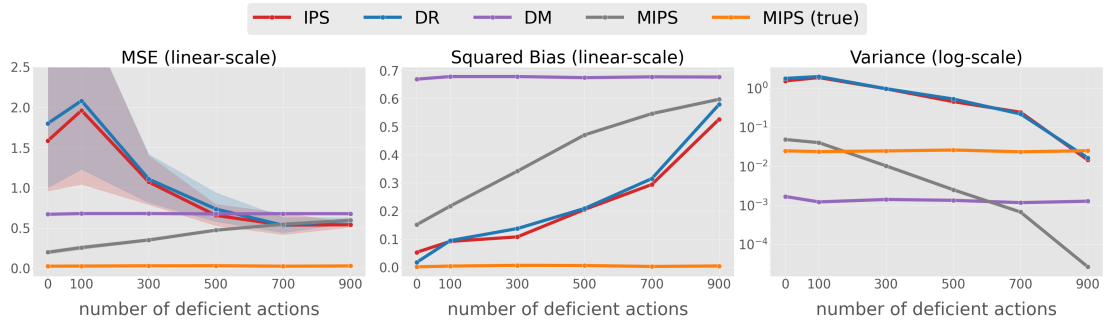

Figure 4. MSE, Squared Bias, and Variance with **varying number of deficient actions**.

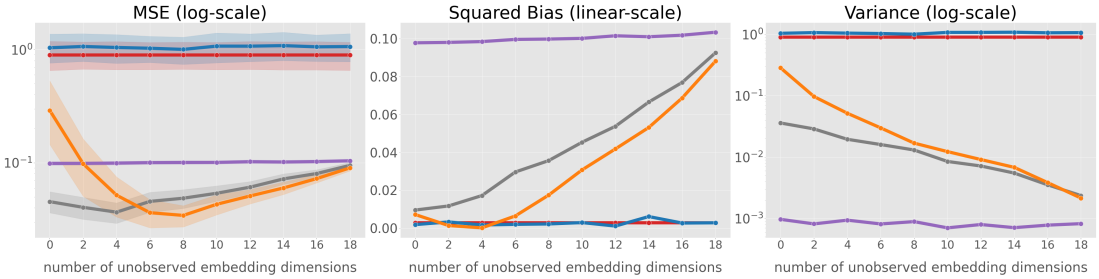

Figure 5. MSE, Squared Bias, and Variance with **varying number of unobserved dimensions in action embeddings**.

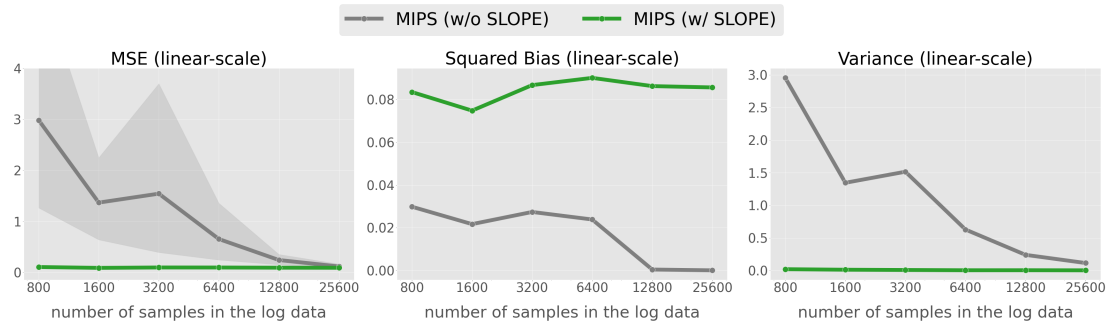

Figure 6. MSE, Squared Bias, and Variance of MIPS **w/ or w/o action embedding selection (SLOPE)**.

embedding dimensions. Somewhat surprisingly, we observe that MIPS and MIPS (true) perform better when there are some missing dimensions, even if it leads to the violated assumption. Specifically, the MSE of MIPS and MIPS (true) is minimized when there are 4 and 8 missing dimensions (out of 20), respectively. This phenomenon is due to the reduced variance. The third column of Figure 5 implies that the variance of MIPS and MIPS (true) decreases substantially with an increasing number of unobserved dimensions, while the bias increases with the violated assumption as expected. These observations suggest that MIPS can be highly effective despite the violated assumption.

How does data-driven embedding selection perform combined with MIPS? The previous section showed that there is a potential to improve the accuracy of MIPS by selecting a subset of dimensions for estimating the marginal importance weights. We now evaluate whether we can effectively address this embedding selection problem.

Figure 6 compares the MSE, squared bias, and variance of MIPS and MIPS with SLOPE (MIPS w/ SLOPE) using the same embedding space as in the previous section. Note that we vary the sample size n and fix $|\mathcal{A}| = 1000$. The results suggest that the data-driven embedding selection provides a substantial improvement in MSE for small sample sizes. As shown in the second and third columns in Figure 6, the embedding selection significantly reduces the variance at the cost of introducing some bias by strategically violating the assumption, which results in a better MSE.

Other benefits of MIPS. MIPS has additional benefits over the conventional estimators. In fact, in addition to the

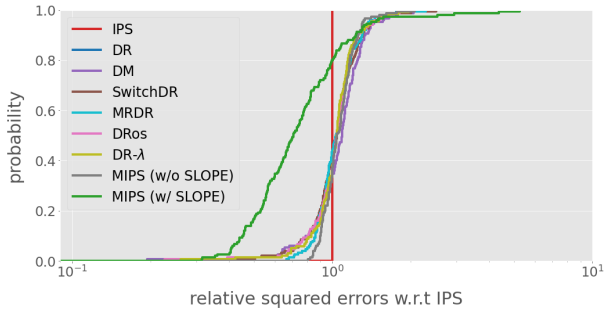

Figure 7. CDF of relative squared error w.r.t IPS.

case with many actions, IPS is also vulnerable when logging and target policies differ substantially and the reward is noisy (see Eq. (2)). Appendix D.2 empirically investigates the additional benefits of MIPS with varying logging/target policies and varying noise levels with a fixed action set. We observe that MIPS is substantially more robust to the changes in policies and added noise than IPS or DR, which provides further arguments for the applicability of MIPS.

4.2. Real-World Data

To assess the real-world applicability of MIPS, we now evaluate MIPS on real-world bandit data. In particular, we use the Open Bandit Dataset (OBD)[6] (Saito et al., 2020), a publicly available logged bandit dataset collected on a large-scale fashion e-commerce platform. We use 100,000 observations that are randomly sub-sampled from the "ALL" campaign of OBD. The dataset contains user contexts x, fashion items to recommend as action $a \in \mathcal{A}$ where $|\mathcal{A}| = 240$, and resulting clicks as reward $r \in \{0, 1\}$. OBD also includes 4-dimensional action embedding vectors such as hierarchical category information about the fashion items.

The dataset consists of two sets of logged bandit data collected by two different policies (uniform random and Thompson sampling) during an A/B test of these policies. We regard uniform random and Thompson sampling as logging and target policies, respectively, to perform an evaluation of OPE estimators. Appendix D.3 describes the detailed experimental procedure to evaluate the accuracy of the estimators on real-world bandit data.

Results. We evaluate MIPS (w/o SLOPE) and MIPS (w/ SLOPE) in comparison to DM, IPS, DR, Switch-DR, More Robust DR (Farajtabar et al., 2018), DRos, and DR-λ. We apply SLOPE to tune the built-in hyperparameters of Switch-DR, DRos, and DR-λ. Figure 7 compares the estimators by drawing the cumulative distribution function (CDF) of their squared errors estimated with 150 different bootstrapped samples of the logged data. Note that the squared errors are normalized by that of IPS. We find that MIPS (w/ SLOPE) outperforms IPS in about 80% of the simulation runs, while other estimators, including MIPS (w/o SLOPE), work similarly to IPS. This result demonstrates the real-world applicability of our estimator as well as the importance of implementing action embedding selection in practice. We report qualitatively similar results for other sample sizes (from 10,000 to 500,000) in Appendix D.3.

[6] https://research.zozo.com/data.html

5. Conclusion and Future Work

We explored the problem of OPE for large action spaces. In this setting, existing estimators based on IPS suffer from impractical variance, which limits their applicability. This problem is highly relevant for practical applications, as many real decision making problems such as recommender systems have to deal with a large number of discrete actions. To achieve an accurate OPE for large action spaces, we propose the MIPS estimator, which builds on the *marginal* importance weights computed with *action embeddings*. We characterize the important statistical properties of the proposed estimator and discuss when it is superior to the conventional ones. Extensive experiments demonstrate that MIPS provides a significant gain in MSE when the vanilla importance weights become large due to large action spaces, substantially outperforming IPS and related estimators.

Our work raises several interesting research questions. For example, this work assumes the existence of some predefined action embeddings and analyzes the resulting statistical properties of MIPS. Even though we discussed how to choose which embedding dimensions to use for OPE (Section 3.2), it would be intriguing to develop a more principled method to optimize or learn (possibly continuous) action embeddings from the logged data for further improving MIPS. Developing a method for accurately estimating the marginal importance weight would also be crucial to fill the gap between MIPS and MIPS (true) observed in our experiments. It would also be interesting to explore off-policy learning using action embeddings and possible applications of marginal importance weighting to other estimators that depend on the vanilla importance weight such as DR.

Acknowledgements

This research was supported in part by NSF Awards IIS-1901168 and IIS-2008139. Yuta Saito was supported by the Funai Overseas Scholarship. All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors.

References

Agrawal, R. The continuum-armed bandit problem. *SIAM journal on control and optimization*, 33(6):1926–1951, 1995.

Agrawal, S. and Goyal, N. Thompson sampling for contextual bandits with linear payoffs. In *International Conference on Machine Learning*, pp. 127–135. PMLR, 2013.

Athey, S., Chetty, R., Imbens, G. W., and Kang, H. The surrogate index: Combining short-term proxies to estimate long-term treatment effects more rapidly and precisely. Technical report, National Bureau of Economic Research, 2019.

Athey, S., Chetty, R., and Imbens, G. Combining experimental and observational data to estimate treatment effects on long term outcomes. *arXiv preprint arXiv:2006.09676*, 2020.

Borisov, A., Markov, I., De Rijke, M., and Serdyukov, P. A neural click model for web search. In *Proceedings of the 25th International Conference on World Wide Web*, pp. 531–541, 2016.

Breiman, L. Random forests. *Machine learning*, 45(1): 5–32, 2001.

Bubeck, S., Munos, R., Stoltz, G., and Szepesvári, C. X-armed bandits. *Journal of Machine Learning Research*, 12(5), 2011.

Chandak, Y., Theocharous, G., Kostas, J., Jordan, S., and Thomas, P. Learning action representations for reinforcement learning. In *International Conference on Machine Learning*, pp. 941–950. PMLR, 2019.

Chen, J. and Ritzwoller, D. M. Semiparametric estimation of long-term treatment effects. *arXiv preprint arXiv:2107.14405*, 2021.

Chu, W., Li, L., Reyzin, L., and Schapire, R. Contextual bandits with linear payoff functions. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pp. 208–214. JMLR Workshop and Conference Proceedings, 2011.

Chuklin, A., Markov, I., and Rijke, M. d. Click models for web search. *Synthesis lectures on information concepts, retrieval, and services*, 7(3):1–115, 2015.

Demirer, M., Syrgkanis, V., Lewis, G., and Chernozhukov, V. Semi-parametric efficient policy learning with continuous actions. *Advances in Neural Information Processing Systems*, 32, 2019.

Dudík, M., Erhan, D., Langford, J., and Li, L. Doubly robust policy evaluation and optimization. *Statistical Science*, 29(4):485–511, 2014.

Dulac-Arnold, G., Denoyer, L., Preux, P., and Gallinari, P. Fast reinforcement learning with large action sets using error-correcting output codes for mdp factorization. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 180–194. Springer, 2012.

Dulac-Arnold, G., Evans, R., van Hasselt, H., Sunehag, P., Lillicrap, T., Hunt, J., Mann, T., Weber, T., Degris, T., and Coppin, B. Deep reinforcement learning in large discrete action spaces. *arXiv preprint arXiv:1512.07679*, 2015.

Farajtabar, M., Chow, Y., and Ghavamzadeh, M. More robust doubly robust off-policy evaluation. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pp. 1447–1456. PMLR, 2018.

Guo, F., Liu, C., and Wang, Y. M. Efficient multiple-click models in web search. In *Proceedings of the 2nd ACM International Conference on Web Search and Data Mining*, pp. 124–131, 2009.

Guo, R., Zhao, X., Henderson, A., Hong, L., and Liu, H. Debiasing grid-based product search in e-commerce. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2852–2860, 2020.

Horvitz, D. G. and Thompson, D. J. A generalization of sampling without replacement from a finite universe. *Journal of the American statistical Association*, 47(260):663–685, 1952.

Jiang, N. and Li, L. Doubly robust off-policy value evaluation for reinforcement learning. In *Proceedings of the 33rd International Conference on Machine Learning*, volume 48, pp. 652–661. PMLR, 2016.

Kallus, N. and Mao, X. On the role of surrogates in the efficient estimation of treatment effects with limited outcome data. *arXiv preprint arXiv:2003.12408*, 2020.

Kallus, N. and Uehara, M. Double reinforcement learning for efficient off-policy evaluation in markov decision processes. *J. Mach. Learn. Res.*, 21:167–1, 2020.

Kallus, N. and Zhou, A. Policy evaluation and optimization with continuous treatments. In *International Conference on Artificial Intelligence and Statistics*, pp. 1243–1251. PMLR, 2018.

Kallus, N., Saito, Y., and Uehara, M. Optimal off-policy evaluation from multiple logging policies. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pp. 5247–5256. PMLR, 2021.

Kiyohara, H., Saito, Y., Matsuhiro, T., Narita, Y., Shimizu, N., and Yamamoto, Y. Doubly robust off-policy evaluation for ranking policies under the cascade behavior model. In *Proceedings of the 15th International Conference on Web Search and Data Mining*, 2022.

Kleinberg, R. Nearly tight bounds for the continuum-armed bandit problem. *Advances in Neural Information Processing Systems*, 17:697–704, 2004.

Kleinberg, R., Slivkins, A., and Upfal, E. Bandits and experts in metric spaces. *Journal of the ACM (JACM)*, 66 (4):1–77, 2019.

Krishnamurthy, A., Langford, J., Slivkins, A., and Zhang, C. Contextual bandits with continuous actions: Smoothing, zooming, and adapting. In *Conference on Learning Theory*, pp. 2025–2027. PMLR, 2019.

Lepski, O. V. and Spokoiny, V. G. Optimal pointwise adaptive methods in nonparametric estimation. *The Annals of Statistics*, pp. 2512–2546, 1997.

Li, S., Abbasi-Yadkori, Y., Kveton, B., Muthukrishnan, S., Vinay, V., and Wen, Z. Offline evaluation of ranking policies with click models. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1685–1694, 2018.

Liu, Q., Li, L., Tang, Z., and Zhou, D. Breaking the curse of horizon: infinite-horizon off-policy estimation. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 5361–5371, 2018.

Liu, Y., Bacon, P.-L., and Brunskill, E. Understanding the curse of horizon in off-policy evaluation via conditional importance sampling. In *International Conference on Machine Learning*, pp. 6184–6193. PMLR, 2020a.

Liu, Y., Swaminathan, A., Agarwal, A., and Brunskill, E. Off-policy policy gradient with stationary distribution correction. In *Uncertainty in Artificial Intelligence*, pp. 1180–1190. PMLR, 2020b.

Lopez, R., Dhillon, I. S., and Jordan, M. I. Learning from extreme bandit feedback. *Proc. Association for the Advancement of Artificial Intelligence*, 2021.

McInerney, J., Brost, B., Chandar, P., Mehrotra, R., and Carterette, B. Counterfactual evaluation of slate recommendations with sequential reward interactions. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1779–1788, 2020.

Metelli, A. M., Russo, A., and Restelli, M. Subgaussian and differentiable importance sampling for off-policy evaluation and learning. *Advances in Neural Information Processing Systems*, 34, 2021.

Narita, Y., Yasui, S., and Yata, K. Efficient counterfactual learning from bandit feedback. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 4634–4641, 2019.

Newey, W. K. and Robins, J. R. Cross-fitting and fast remainder rates for semiparametric estimation. *arXiv preprint arXiv:1801.09138*, 2018.

Pazis, J. and Parr, R. Generalized value functions for large action sets. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, pp. 1185–1192, 2011.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Édouard Duchesnay. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Sachdeva, N., Su, Y., and Joachims, T. Off-policy bandits with deficient support. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 965–975, 2020.

Saito, Y. Doubly robust estimator for ranking metrics with post-click conversions. In *14th ACM Conference on Recommender Systems*, pp. 92–100, 2020.

Saito, Y., Aihara, S., Matsutani, M., and Narita, Y. Open bandit dataset and pipeline: Towards realistic and reproducible off-policy evaluation. *arXiv preprint arXiv:2008.07146*, 2020.

Saito, Y., Udagawa, T., Kiyohara, H., Mogi, K., Narita, Y., and Tateno, K. Evaluating the robustness of off-policy evaluation. In *Proceedings of the 15th ACM Conference on Recommender Systems*, pp. 114–123, 2021.

Slivkins, A. Introduction to multi-armed bandits. *arXiv preprint arXiv:1904.07272*, 2019.

Sondhi, A., Arbour, D., and Dimmery, D. Balanced off-policy evaluation in general action spaces. In *International Conference on Artificial Intelligence and Statistics*, pp. 2413–2423. PMLR, 2020.

Su, Y., Wang, L., Santacatterina, M., and Joachims, T. Cab: Continuous adaptive blending for policy evaluation and learning. In *International Conference on Machine Learning*, volume 84, pp. 6005–6014, 2019.

Su, Y., Dimakopoulou, M., Krishnamurthy, A., and Dudík, M. Doubly robust off-policy evaluation with shrinkage. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pp. 9167–9176. PMLR, 2020a.

Su, Y., Srinath, P., and Krishnamurthy, A. Adaptive estimator selection for off-policy evaluation. In *International Conference on Machine Learning*, pp. 9196–9205. PMLR, 2020b.

Swaminathan, A. and Joachims, T. Batch learning from logged bandit feedback through counterfactual risk minimization. *The Journal of Machine Learning Research*, 16(1):1731–1755, 2015a.

Swaminathan, A. and Joachims, T. Counterfactual risk minimization: Learning from logged bandit feedback. In *International Conference on Machine Learning*, pp. 814–823. PMLR, 2015b.

Swaminathan, A. and Joachims, T. The self-normalized estimator for counterfactual learning. *Advances in Neural Information Processing Systems*, 28, 2015c.

Swaminathan, A., Krishnamurthy, A., Agarwal, A., Dudik, M., Langford, J., Jose, D., and Zitouni, I. Off-policy evaluation for slate recommendation. In *Advances in Neural Information Processing Systems*, volume 30, pp. 3632–3642, 2017.

Tennenholtz, G. and Mannor, S. The natural language of actions. In *International Conference on Machine Learning*, pp. 6196–6205. PMLR, 2019.

Thomas, P. and Brunskill, E. Data-efficient off-policy policy evaluation for reinforcement learning. In *Proceedings of the 33rd International Conference on Machine Learning*, volume 48, pp. 2139–2148. PMLR, 2016.

Thomas, P., Theocharous, G., and Ghavamzadeh, M. High confidence policy improvement. In *Proceedings of the 32th International Conference on Machine Learning*, pp. 2380–2388, 2015.

Tucker, G. and Lee, J. Improved estimator selection for off-policy evaluation. *Workshop on Reinforcement Learning Theory at the 38th International Conference on Machine Learning*, 2021.

Van Hasselt, H. and Wiering, M. A. Using continuous action spaces to solve discrete problems. In *2009 International Joint Conference on Neural Networks*, pp. 1149–1156. IEEE, 2009.

Vlassis, N., Chandrashekar, A., Gil, F. A., and Kallus, N. Control variates for slate off-policy evaluation. *Advances in Neural Information Processing Systems*, 34, 2021.

Voloshin, C., Le, H. M., Jiang, N., and Yue, Y. Empirical study of off-policy policy evaluation for reinforcement learning. *arXiv preprint arXiv:1911.06854*, 2019.

Wang, Y.-X., Agarwal, A., and Dudık, M. Optimal and adaptive off-policy evaluation in contextual bandits. In *International Conference on Machine Learning*, pp. 3589–3597. PMLR, 2017.

Xie, T., Ma, Y., and Wang, Y.-X. Towards optimal off-policy evaluation for reinforcement learning with marginalized importance sampling. In *Advances in Neural Information Processing Systems*, pp. 9665–9675, 2019.

A. Related Work

Off-Policy Evaluation: Off-policy evaluation of counterfactual policies has extensively been studied in both contextual bandits (Dudík et al., 2014; Wang et al., 2017; Liu et al., 2018; Farajtabar et al., 2018; Su et al., 2019; 2020a; Kallus et al., 2021; Metelli et al., 2021) and reinforcement learning (RL) (Jiang & Li, 2016; Thomas & Brunskill, 2016; Xie et al., 2019; Kallus & Uehara, 2020; Liu et al., 2020a). There are three main approaches in the literature. The first approach is DM, which estimates the policy value based on the estimated reward $\hat{q}$. DM has a lower variance than IPS, and is also proposed as an approach to deal with support deficient data (Sachdeva et al., 2020) where IPS is biased. A drawback is that it is susceptible to misspecification of the reward function. This misspecification issue is problematic, as the extent of misspecification cannot be easily evaluated for real-world data (Farajtabar et al., 2018; Voloshin et al., 2019). The second approach is IPS, which estimates the value of a policy by applying importance weighting to the observed reward. With some assumptions for identification such as common support, IPS is unbiased and consistent. However, IPS can suffer from high bias and variance when the action space is large. It can have a high bias when the logging policy fails to satisfy the common support condition, which is likely to occur for large action spaces (Sachdeva et al., 2020). Variance is also a critical issue especially when the action space is large, as the importance weights are likely to take larger values. The weight clipping (Swaminathan & Joachims, 2015b; Su et al., 2019; 2020a) and normalization (Swaminathan & Joachims, 2015c) are often used to address the variance issue, but they produce additional bias. Thus, DR has gained particular attention as the third approach. This estimator is a hybrid of the previous two approaches, and can achieve a lower bias than DM, and a lower variance than IPS (Dudík et al., 2014; Farajtabar et al., 2018). It can also achieve the lowest possible asymptotic variance, a property known as *efficiency* (Narita et al., 2019). Several recent works have extended DR to improve its performance with small samples (Wang et al., 2017; Su et al., 2020a) or under model misspecification (Farajtabar et al., 2018). Though there are a number of extensions of DR both in bandits (as described above) and RL (Jiang & Li, 2016; Thomas & Brunskill, 2016; Kallus & Uehara, 2020), none of them tackle the large discrete action space. Demirer et al. (2019) describe an estimator for finitely many possible actions as a special case of their main proposal, which is for continuous action spaces. However, this method is based on a linearity assumption of the reward function, which rarely holds in practice. Moreover, the bias arises from violating the assumption and the variance reduction due to the additional assumption are not analyzed. Kallus & Zhou (2018) formulate the problem of OPE for continuous action spaces and propose some estimators building on the kernel smoothing in nonparametric statistics. Specifically, kernel functions are used to infer the rewards among similar continuous actions where the bias-variance trade-off is controlled by a bandwidth hyperparameter. If every dimension of the action embedding space $\mathcal{E}$ is continuous, the continuous-action estimators of Kallus & Zhou (2018) might be applied to our setup under smoothness assumption. However, this naive application can suffer from the curse of dimensionality where the kernel smoothing performs dramatically worse as the number of embedding dimensions increases. In contrast, MIPS avoids the curse of dimensionality by estimating the marginal importance weights via supervised classification as in Section 3.3.

Note that there is an estimator called marginalized importance sampling in OPE of RL (Liu et al., 2018; Xie et al., 2019; Liu et al., 2020b). This method estimates the state marginal distribution and applies importance weighting with respect to this marginal distribution rather than the trajectory distribution. Although marginalization is a key trick of this estimator, it is aimed at resolving the curse of horizon, a problem specific to RL. In contrast, our approach utilizes the marginal distribution over action embeddings to deal with large action spaces. Applications of our estimator are not limited to RL.

Off-Policy Evaluation for Slate and Ranking Policies: Another line of work that shares the similar motivation to ours is OPE of slate or ranking policies (Swaminathan et al., 2017; Li et al., 2018; McInerney et al., 2020; Saito, 2020; Su et al., 2020a; Vlassis et al., 2021; Lopez et al., 2021; Kiyohara et al., 2022). In this setting, the estimators have to handle the combinatorial action space, which could be very large even if the number of unique actions is not. Therefore, some additional assumptions are imposed to make the combinatorial action space tractable. A primary problem setting in this direction is OPE for slate bandit policies, where it is assumed that only a single, slate-level reward is observed for each data. Swaminathan et al. (2017) tackle this setting by positing a linearity assumption on the reward function. The proposed pseudoinverse (PI) estimator was shown to provide an exponential gain in the sample complexity over IPS. Following this seminal work, Su et al. (2020a) extend their Doubly Robust with Optimistic Shrinkage, originally proposed for the general OPE problem, to the slate action case. Vlassis et al. (2021) improve the PI estimator by optimizing a set of control variates. Although PI is compelling, applications of this class of estimators are limited to the specific problem of slate bandits. On the other hand, our framework is more general and applicable not only to slate bandits, but also to other problem instances including OPE for ranking policies with observable slot-level rewards (described below) or general contextual bandits with large action spaces. In addition, all estimators for slate bandits rely on the linearity assumption, while our MIPS builds on a different assumption about the quality of the action embedding.

Another similar setting is OPE for ranking policies where it is assumed that the rewards for every slot in a ranking (slot-level rewards) are observable, a setting also known as semi-bandit feedback. PI and its variants discussed above are applicable to this setting, but McInerney et al. (2020) empirically verify that the PI estimators do not work well, as they do not utilize additional information about the slot-level rewards. To leverage slot-level rewards to further improve OPE, assumptions are made to capture different types of user behaviors to control the bias-variance trade-off in OPE. For example, Li et al. (2018) assume that users interact with items presented in different positions of a ranking totally independently. In contrast, McInerney et al. (2020) and Kiyohara et al. (2022) assume that users go down a ranking from top to bottom. These assumptions correspond to click models such as cascade model in information retrieval (Guo et al., 2009; Chuklin et al., 2015) and are useful in reducing the variance. However, whether these assumptions are reasonable depends highly on a ranking interface and real user behavior. If the assumption fails to capture real user behavior, this approach can produce unexpected bias. For example, the cascade model is only applicable when a ranking interface is vertical, however, real-world ranking interfaces are often more complex (Guo et al., 2020). Moreover, real-world user behaviors are often too diverse to model with a single, universal assumption (Borisov et al., 2016). In contrast, our approach is applicable to any ranking interfaces, once they are represented as action embeddings, without assuming any particular user behavior. Moreover, ours is more general in that its application is not limited to information retrieval and recommender systems, but includes robotics, education, conversational agents, or personalized medicine where click models are not applicable.

Reinforcement Learning for Large Action Spaces: Although we focus on OPE, there have been several attempts to enable high-performance policy learning for large action spaces. A typical approach is to factorize the action space into binary sub-spaces (Pazis & Parr, 2011; Dulac-Arnold et al., 2012). For example, Pazis & Parr (2011) represent each action with a binary format and train a value function for each bit. On the other hand, Van Hasselt & Wiering (2009) and Dulac-Arnold et al. (2015) assume the existence of continuous representations of discrete actions as prior knowledge. They perform policy gradients with the continuous actions and search the nearest discrete action. Similar to these works, we assume the existence of some predefined action embeddings and propose to use that prior information to enable an accurate OPE for large action spaces. We also analyze the bias-variance trade-off of the resulting estimator and relate it to the quality of the action embeddings. Some recent works also tackle how to learn useful action representations from only available data. Tennenholtz & Mannor (2019) achieve this by leveraging expert demonstrations, while Chandak et al. (2019) perform supervised learning to predict the state transitions and obtain action representations with no prior knowledge. Following these works, it may be valuable to develop an algorithm to optimize or learn (possibly continuous) action embeddings from the data to further improve OPE for large action spaces.

Multi-Armed Bandits with Side Information: There are two prominent approaches to deal with large or infinite action spaces in the *online* bandit literature (Krishnamurthy et al., 2019; Slivkins, 2019). The first one is the parametric approach such as linear or combinatorial bandits, which assumes that the expected reward can be represented as a parametric function of the action such as a linear function (Chu et al., 2011; Agrawal & Goyal, 2013). There is also a nonparametric approach, which typically makes much weaker assumptions about the rewards, e.g., Lipschitz assumptions. Lipschitz bandits have been studied to address large, structured action spaces such as the $[0, 1]$ interval, where the applications range from dynamic pricing to ad auction. A basic idea in this literature is that similar arms should have similar quality, as per Lipschitz-continuity or some corresponding assumptions on the structure of the action space. The Lipschitz assumption was introduced by Agrawal (1995) to the bandit setting. Kleinberg (2004) optimally solve this problem in the worst case. Kleinberg et al. (2019) and Bubeck et al. (2011) rely on the zooming algorithms, which gradually zoom in to the more promising regions of the action space to achieve data-dependent regret bounds. Further works extend this direction by relaxing the assumptions with various local definitions, as well as incorporating contexts into account, as surveyed in Section 4 of Slivkins (2019).

Causal Inference with Surrogates: From a statistical standpoint, causal inference with surrogates is also related (Athey et al., 2019; 2020; Kallus & Mao, 2020; Chen & Ritzwoller, 2021). Its aim is to identify and estimate the causal effect of some treatments (e.g., job training) on a *primary outcome*, which is unobservable without waiting for decades (e.g., lifetime earnings) (Athey et al., 2019). Instead of waiting for a long period to collect the data, these works assume the availability of *surrogate outcomes* such as test scores and college attendance rates, which could be observed in a much shorter period. In particular, Athey et al. (2019) build on what is called the surrogacy condition to identify the average treatment effect of treatments on the primary outcome. The surrogacy condition is analogous to Assumption 3.2 and states that there should not be any direct effect of treatments on the primary outcome. Although our formulation and assumptions share a similar structure, we would argue that our motivation is to enable an accurate OPE of decision making policies for large action spaces, which is quite different from identifying the average causal effect of binary treatments on a long-term outcome.

B. Proofs, Derivations, and Additional Analysis

B.1. Proof of Proposition 3.3

Proof.

$$
\begin{aligned}
V(\pi) &= \mathbb{E}_{p(x)\pi(a|x)p(e|x,a)}[q(x,a,e)] \\
&= \mathbb{E}_{p(x)\pi(a|x)p(e|x,a)}[q(x,e)] \qquad (7) \\
&= \mathbb{E}_{p(x)}\left[\sum_{a\in\mathcal{A}}\pi(a|x)\sum_{e\in\mathcal{E}}p(e|x,a)\cdot q(x,e)\right] \\
&= \mathbb{E}_{p(x)}\left[\sum_{e\in\mathcal{E}}q(x,e)\cdot\left(\sum_{a\in\mathcal{A}}\pi(a|x)\cdot p(e|x,a)\right)\right] \qquad (8) \\
&= \mathbb{E}_{p(x)}\left[\sum_{e\in\mathcal{E}}p(e|x,\pi)\cdot q(x,e)\right] \\
&= \mathbb{E}_{p(x)p(e|x,\pi)}[q(x,e)] \\
&= \mathbb{E}_{p(x)p(e|x,\pi)p(r|x,e)}[r]
\end{aligned}
$$

where we use Assumption 3.2 in Eq. (7) and $p(e|x,\pi)=\sum_{a\in\mathcal{A}}\pi(a|x)p(e|x,a)$ in Eq. (8). □

B.2. Proof of Proposition 3.4

Proof. From the linearity of expectation, we have $\mathbb{E}_{\mathcal{D}}[\hat{V}_{\mathrm{MIPS}}(\pi;\mathcal{D})]=\mathbb{E}_{p(x)\pi_0(a|x)p(e|x,a)p(r|x,a,e)}[w(x,e)r]$. Thus, we calculate only the expectation of $w(x,e)r$ (RHS of the equation) below.

$$
\begin{aligned}
&\mathbb{E}_{p(x)\pi_0(a|x)p(e|x,a)p(r|x,a,e)}[w(x,e)r] \\
&= \mathbb{E}_{p(x)\pi_0(a|x)p(e|x,a)}[w(x,e)\cdot q(x,a,e)] \\
&= \mathbb{E}_{p(x)\pi_0(a|x)p(e|x,a)}[w(x,e)\cdot q(x,e)] \qquad (9) \\
&= \mathbb{E}_{p(x)}\left[\sum_{a\in\mathcal{A}}\pi_0(a|x)\sum_{e\in\mathcal{E}}p(e|x,a)\frac{p(e|x,\pi)}{p(e|x,\pi_0)}q(x,e)\right] \\
&= \mathbb{E}_{p(x)}\left[\sum_{e\in\mathcal{E}}\frac{p(e|x,\pi)}{p(e|x,\pi_0)}\cdot q(x,e)\cdot\left(\sum_{a\in\mathcal{A}}p(e|x,a)\cdot\pi_0(a|x)\right)\right] \\
&= \mathbb{E}_{p(x)}\left[\sum_{e\in\mathcal{E}}\frac{p(e|x,\pi)}{p(e|x,\pi_0)}\cdot p(e|x,\pi_0)\cdot q(x,e)\right] \\
&= \mathbb{E}_{p(x)p(e|x,\pi)}[q(x,e)] \qquad (10) \\
&= \mathbb{E}_{p(x)p(e|x,\pi)p(r|x,e)}[r] \\
&= V(\pi)
\end{aligned}
$$

where we use Assumption 3.2 in Eq. (9) and $p(e|x,\pi_0)=\sum_{a\in\mathcal{A}}\pi_0(a|x)p(e|x,a)$ in Eq. (10). □

B.3. Proof of Theorem 3.5

To prove Theorem 3.5, we first state a lemma.

Lemma B.1. *For real-valued, bounded functions $f:\mathbb{N}\to\mathbb{R}, g:\mathbb{N}\to\mathbb{R}, h:\mathbb{N}\to\mathbb{R}$ where $\sum_{a\in[m]}g(a)=1$, we have*

$$
\sum_{a\in[m]}f(a)g(a)\Big(h(a)-\sum_{b\in[m]}g(b)h(b)\Big)=\sum_{a<b\leq m}g(a)g(b)(h(a)-h(b))(f(a)-f(b)) \qquad (11)
$$

Proof. We prove this lemma via induction. First, we show the $m = 2$ case below.

$$
\begin{aligned}
&f(1)g(1)\left(h(1) - (g(1)h(1) + g(2)h(2))\right) + f(2)g(2)\left(h(2) - (g(1)h(1) + g(2)h(2))\right) \\
&= f(1)g(1)h(1) - f(1)g(1)(g(1)h(1) + g(2)h(2)) + f(2)g(2)h(2) - f(2)g(2)(g(1)h(1) + g(2)h(2)) \\
&= f(1)g(1)h(1) - f(1)g(1)((1-g(2))h(1) + g(2)h(2)) + f(2)g(2)h(2) - f(2)g(2)(g(1)h(1) + (1-g(1))h(2)) \\
&= -f(1)g(1)(-g(2)h(1) + g(2)h(2)) - f(2)g(2)(g(1)h(1) - g(1)h(2)) \\
&= -f(1)g(1)g(2)(h(2) - h(1)) + f(2)g(1)g(2)(h(2) - h(1)) \\
&= g(1)g(2)(h(2) - h(1))(f(2) - f(1))
\end{aligned}
$$

Note that $g(1) + g(2) = 1$ from the statement.

Next, we assume Eq. (11) is true for the $m = k - 1$ case and show that it is also true for the $m = k$ case. First, note that

$$
\begin{aligned}
&\sum_{a<b\leq k} g(a)g(b)(h(a) - h(b))(f(a) - f(b)) \\
&= \sum_{a<b\leq k-1} g(a)g(b)(h(a) - h(b))(f(a) - f(b)) + \sum_{a\in[k-1]} g(a)g(k)(h(a) - h(k))(f(a) - f(k))
\end{aligned}
$$

Then, we have

$$
\begin{aligned}
&\sum_{a\in[k]} f(a)g(a)\Big(h(a) - \sum_{b\in[k]} g(b)h(b)\Big) \\
&= \sum_{a\in[k-1]} f(a)g(a)\Big(h(a) - \sum_{b\in[k]} g(b)h(b)\Big) + f(k)g(k)\Big(h(k) - \sum_{b\in[k]} g(b)h(b)\Big) \\
&= \sum_{a\in[k-1]} f(a)g(a)\left(\Big(h(a) - \sum_{b\in[k-1]} g(b)h(b)\Big) - g(k)h(k)\right) + f(k)g(k)h(k) - f(k)g(k)\sum_{a\in[k]} g(a)h(a) \\
&= \sum_{a\in[k-1]} f(a)g(a)\Big(h(a) - \sum_{b\in[k-1]} g(b)h(b)\Big) - g(k)h(k)\sum_{a\in[k-1]} f(a)g(a) + f(k)g(k)h(k) - f(k)g(k)\sum_{a\in[k]} g(a)h(a) \\
&= \sum_{a\in[k-1]} f(a)g(a)\Big(h(a) - \sum_{b\in[k-1]} g(b)h(b)\Big) \\
&\quad - g(k)h(k)\sum_{a\in[k-1]} f(a)g(a) + f(k)h(k)g(k) - f(k)g(k)\sum_{a\in[k-1]} g(a)h(a) - f(k)g(k)g(k)h(k) \\
&= \big(1-g(k)\big)\sum_{a\in[k-1]} f(a)\tilde{g}(a)\left(\big(1-g(k)\big)\Big(h(a) - \sum_{b\in[k-1]} \tilde{g}(b)h(b)\Big) + g(k)h(a)\right) \\
&\quad - g(k)h(k)\sum_{a\in[k-1]} f(a)g(a) + f(k)h(k)g(k) - f(k)g(k)\sum_{a\in[k-1]} g(a)h(a) - f(k)g(k)h(k)\left(1 - \sum_{a\in[k-1]} g(a)\right) \\
&= \big(1-g(k)\big)^2\sum_{a\in[k-1]} f(a)\tilde{g}(a)\Big(h(a) - \sum_{b\in[k-1]} \tilde{g}(b)h(b)\Big) \\
&\quad + g(k)\sum_{a\in[k-1]} f(a)g(a)h(a) - g(k)h(k)\sum_{a\in[k-1]} f(a)g(a) - f(k)g(k)\sum_{a\in[k-1]} g(a)h(a) + f(k)g(k)h(k)\sum_{a\in[k-1]} g(a)
\end{aligned}
\tag{12}
$$

where we use $g(k) = 1 - \sum_{a\in[k-1]} g(a)$ and define $\tilde{g}(a) := g(a)/(\sum_{a\in[k-1]} g(a)) = g(a)/(1-g(k))$.

The first term of Eq. (12) is the $m = k-1$ case, so we have the following from the assumption of induction.

$$\begin{aligned}\big(1-g(k)\big)^2 \sum_{a\in[k-1]} f(a)\tilde{g}(a)\Big(h(a) - \sum_{b\in[k-1]} \tilde{g}(b)h(b)\Big) &= \big(1-g(k)\big)^2 \sum_{a<b\le k-1} \tilde{g}(a)\tilde{g}(b)(h(a)-h(b))(f(a)-f(b)) \\ &= \sum_{a<b\le k-1} g(a)g(b)(h(a)-h(b))(f(a)-f(b))\end{aligned}$$

Note that $\sum_{a\in[k-1]} \tilde{g}(a) = 1$. Rearranging the remaining terms of Eq. (12) yields

$$\begin{aligned}&\sum_{a\in[k]} f(a)g(a)\Big(h(a) - \sum_{b\in[k]} g(b)h(b)\Big) \\ &= \sum_{a<b\le k-1} g(a)g(b)(h(a)-h(b))(f(a)-f(b)) + \sum_{a\in[k-1]} g(a)g(k)(h(a)-h(k))(f(a)-f(k))\end{aligned}$$

Implying that the $m = k$ case is true if the $m = k-1$ case is true. □

We then use the above Lemma to prove Theorem 3.5.

Proof.

$$\begin{aligned}\mathrm{Bias}(\hat{V}_{\mathrm{MIPS}}(\pi)) &= \mathbb{E}_{p(x)\pi_0(a|x)p(e|x,a)p(r|x,a,e)}[w(x,e)r] - V(\pi) \\ &= \mathbb{E}_{p(x)\pi_0(a|x)p(e|x,a)}[w(x,e)\cdot q(x,a,e)] - \mathbb{E}_{p(x)\pi(a|x)p(e|x,a)}[q(x,a,e)] \\ &= \mathbb{E}_{p(x)\pi_0(a|x)}\left[\sum_{e\in\mathcal{E}} p(e|x,a)\cdot w(x,e)\cdot q(x,a,e)\right] - \mathbb{E}_{p(x)\pi(a|x)}\left[\sum_{e\in\mathcal{E}} p(e|x,a)\cdot q(x,a,e)\right] \\ &= \mathbb{E}_{p(x)}\left[\sum_{a\in\mathcal{A}} \pi_0(a|x)\sum_{e\in\mathcal{E}} \frac{p(e|x,\pi_0)\cdot\pi_0(a|x,e)}{\pi_0(a|x)}\cdot w(x,e)\cdot q(x,a,e)\right] \\ &\quad - \mathbb{E}_{p(x)}\left[\sum_{a\in\mathcal{A}} \pi(a|x)\sum_{e\in\mathcal{E}} \frac{p(e|x,\pi_0)\cdot\pi_0(a|x,e)}{\pi_0(a|x)}\cdot q(x,a,e)\right] \qquad (13) \\ &= \mathbb{E}_{p(x)}\left[\sum_{e\in\mathcal{E}} p(e|x,\pi_0)\cdot w(x,e)\sum_{a\in\mathcal{A}} \pi_0(a|x,e)\cdot q(x,a,e)\right] \\ &\quad - \mathbb{E}_{p(x)}\left[\sum_{e\in\mathcal{E}} p(e|x,\pi_0)\sum_{a\in\mathcal{A}} w(x,a)\cdot\pi_0(a|x,e)\cdot q(x,a,e)\right] \\ &= \mathbb{E}_{p(x)p(e|x,\pi_0)}\left[w(x,e)\sum_{a\in\mathcal{A}} \pi_0(a|x,e)\cdot q(x,a,e)\right] \\ &\quad - \mathbb{E}_{p(x)p(e|x,\pi_0)}\left[\sum_{a\in\mathcal{A}} w(x,a)\cdot\pi_0(a|x,e)\cdot q(x,a,e)\right] \\ &= \mathbb{E}_{p(x)p(e|x,\pi_0)}\left[\sum_{a\in\mathcal{A}} w(x,a)\cdot\pi_0(a|x,e)\sum_{b\in\mathcal{A}} \pi_0(b|x,e)\cdot q(x,b,c)\right] \\ &\quad - \mathbb{E}_{p(x)p(e|x,\pi_0)}\left[\sum_{a\in\mathcal{A}} w(x,a)\cdot\pi_0(a|x,e)\cdot q(x,a,e)\right] \qquad (14) \\ &= \mathbb{E}_{p(x)p(e|x,\pi_0)}\left[\sum_{a\in\mathcal{A}} w(x,a)\cdot\pi_0(a|x,e)\cdot\left(\Big(\sum_{b\in\mathcal{A}} \pi_0(b|x,e)\cdot q(x,b,c)\Big) - q(x,a,e)\right)\right]\end{aligned}$$

where we use $p(e|x,a) = \frac{p(e|x,\pi_0)\pi_0(a|x,e)}{\pi_0(a|x)}$ in Eq. (13) and $w(x,e) = \mathbb{E}_{\pi_0(a|x,e)}[w(x,a)]$ in Eq. (14).

By applying Lemma A.1 to the last line (setting $f(a) = w(\cdot,a), g(a) = \pi_0(a|\cdot,\cdot), h(a) = q(\cdot,a,\cdot)$), we get the final expression of the bias. □

B.4. Proof of Theorem 3.6

Proof. Under Assumptions 2.1, 3.1, and 3.2, IPS and MIPS are both unbiased. Thus, the difference in their variance is attributed to the difference in their second moment, which is calculated below.

$$
\begin{aligned}
&\mathbb{V}_{p(x)\pi_0(a|x)p(e|x,a)p(r|x,a,e)}[w(x,a)r] - \mathbb{V}_{p(x)\pi_0(a|x)p(e|x,a)p(r|x,a,e)}[w(x,e)r] \\
&= \mathbb{E}_{p(x)\pi_0(a|x)p(e|x,a)p(r|x,a,e)}[w(x,a)^2 \cdot r^2] - \mathbb{E}_{p(x)\pi_0(a|x)p(e|x,a)p(r|x,a,e)}[w(x,e)^2 \cdot r^2] \\
&= \mathbb{E}_{p(x)\pi_0(a|x)p(e|x,a)} \left[w(x,a)^2 \cdot \mathbb{E}_{p(r|x,a,e)}[r^2] \right] - \mathbb{E}_{p(x)\pi_0(a|x)p(e|x,a)} \left[w(x,e)^2 \cdot \mathbb{E}_{p(r|x,a,e)}[r^2] \right] \\
&= \mathbb{E}_{p(x)\pi_0(a|x)p(e|x,a)} \left[\left(w(x,a)^2 - w(x,e)^2 \right) \cdot \mathbb{E}_{p(r|x,e)}[r^2] \right] \qquad (15) \\
&= \mathbb{E}_{p(x)} \left[\sum_{a \in \mathcal{A}} \pi_0(a|x) \sum_{e \in \mathcal{E}} p(e|x,a) \cdot \left(w(x,a)^2 - w(x,e)^2 \right) \cdot \mathbb{E}_{p(r|x,e)}[r^2] \right] \\
&= \mathbb{E}_{p(x)} \left[\sum_{a \in \mathcal{A}} \pi_0(a|x) \sum_{e \in \mathcal{E}} \frac{p(e|x,\pi_0) \cdot \pi_0(a|x,e)}{\pi_0(a|x)} \cdot \left(w(x,a)^2 - w(x,e)^2 \right) \cdot \mathbb{E}_{p(r|x,e)}[r^2] \right] \qquad (16) \\
&= \mathbb{E}_{p(x)} \left[\sum_{e \in \mathcal{E}} p(e|x,\pi_0) \cdot \mathbb{E}_{p(r|x,e)}[r^2] \sum_{a \in \mathcal{A}} \pi_0(a|x,e) \cdot \left(w(x,a)^2 - w(x,e)^2 \right) \right] \\
&= \mathbb{E}_{p(x)p(e|x,\pi_0)} \left[\mathbb{E}_{p(r|x,e)}[r^2] \cdot \left(\left(\sum_{a \in \mathcal{A}} \pi_0(a|x,e) \cdot w(x,a)^2 \right) - w(x,e)^2 \right) \right]
\end{aligned}
$$

where we use Assumption 3.2 in Eq. (15), $p(e|x,a) = \frac{p(e|x,\pi_0)\pi_0(a|x,e)}{\pi_0(a|x)}$ in Eq. (16). Here, we have

$$
\begin{aligned}
\left(\sum_{a \in \mathcal{A}} \pi_0(a|x,e) \cdot w(x,a)^2 \right) - w(x,e)^2 &= \left(\sum_{a \in \mathcal{A}} \pi_0(a|x,e) \cdot w(x,a)^2 \right) - \left(\sum_{a \in \mathcal{A}} \pi_0(a|x,e) \cdot w(x,a) \right)^2 \\
&= \mathbb{E}_{\pi_0(a|x,e)} \left[w(x,a)^2 \right] - \left(\mathbb{E}_{\pi_0(a|x,e)} \left[w(x,a) \right] \right)^2 \\
&= \mathbb{V}_{\pi_0(a|x,e)} \left[w(x,a) \right]
\end{aligned}
$$

where $w(x,e) = \mathbb{E}_{\pi_0(a|x,e)}[w(x,a)]$.

Therefore,

$$
\begin{aligned}
&\mathbb{E}_{p(x)\pi_0(a|x)p(e|x,a)p(r|x,a,e)}[w(x,a)^2 \cdot r^2] - \mathbb{E}_{p(x)\pi_0(a|x)p(e|x,a)p(r|x,a,e)}[w(x,e)^2 \cdot r^2] \\
&= \mathbb{E}_{p(x)p(e|x,\pi_0)} \left[\mathbb{E}_{p(r|x,e)}[r^2] \cdot \left(\left(\sum_{a \in \mathcal{A}} \pi_0(a|x,e) \cdot w(x,a)^2 \right) - w(x,e)^2 \right) \right] \\
&= \mathbb{E}_{p(x)p(e|x,\pi_0)} \left[\mathbb{E}_{p(r|x,e)}[r^2] \cdot \mathbb{V}_{\pi_0(a|x,e)} \left[w(x,a) \right] \right]
\end{aligned}
$$

Finally, as samples are independent, $n\mathbb{V}_{\mathcal{D}}[\hat{V}_{\mathrm{IPS}}(\pi;\mathcal{D})] = \mathbb{V}_{p(x)\pi_0(a|x)p(e|x,a)p(r|x,a,e)}[w(x,a)r]$ and $n\mathbb{V}_{\mathcal{D}}[\hat{V}_{\mathrm{MIPS}}(\pi;\mathcal{D})] = \mathbb{V}_{p(x)\pi_0(a|x)p(e|x,a)p(r|x,a,e)}[w(x,e)r]$. □

B.5. Proof of Theorem 3.7

Proof. First, we express the MSE gain of MIPS over the vanilla IPS with their bias and variance as follows.

$$
\mathrm{MSE}\big(\hat{V}_{\mathrm{IPS}}(\pi)\big) - \mathrm{MSE}\big(\hat{V}_{\mathrm{MIPS}}(\pi)\big) = \mathbb{V}_{\mathcal{D}}[\hat{V}_{\mathrm{IPS}}(\pi;\mathcal{D})] - \mathbb{V}_{\mathcal{D}}[\hat{V}_{\mathrm{MIPS}}(\pi;\mathcal{D})] - \mathrm{Bias}(\hat{V}_{\mathrm{MIPS}}(\pi))^2
$$

Since the samples are assumed to be independent, we can simply rescale the MSE gain as follows.

$$
n \left(\mathrm{MSE}\big(\hat{V}_{\mathrm{IPS}}(\pi)\big) - \mathrm{MSE}\big(\hat{V}_{\mathrm{MIPS}}(\pi)\big) \right) = \mathbb{V}_{x,a,r}[w(x,a)r] - \mathbb{V}_{x,e,r}[w(x,e)r] - n\mathrm{Bias}(\hat{V}_{\mathrm{MIPS}}(\pi))^2
$$

Below, we calculate the difference in variance.

$$
\begin{aligned}
&\mathbb{V}_{p(x)\pi_0(a|x)p(e|x,a)p(r|x,a,e)}[w(x,a)r] - \mathbb{V}_{p(x)\pi_0(a|x)p(e|x,a)p(r|x,a,e)}[w(x,e)r] \\
&= \mathbb{E}_{p(x)\pi_0(a|x)p(e|x,a)p(r|x,a,e)}[w(x,a)^2 \cdot r^2] - V(\pi)^2 \\
&\quad - \left(\mathbb{E}_{p(x)\pi_0(a|x)p(e|x,a)p(r|x,a,e)}[w(x,e)^2 \cdot r^2] - \left(V(\pi) + \mathrm{Bias}(\hat{V}_{\mathrm{MIPS}}(\pi)) \right)^2 \right) \\
&= \mathbb{E}_{p(x)\pi_0(a|x)p(e|x,a)} \left[\left(w(x,a)^2 - w(x,e)^2 \right) \cdot \mathbb{E}_{p(r|x,a,e)}[r^2] \right] - V(\pi)^2 + \left(V(\pi)^2 + 2V(\pi)\mathrm{Bias}(\hat{V}_{\mathrm{MIPS}}(\pi)) + \mathrm{Bias}(\hat{V}_{\mathrm{MIPS}}(\pi))^2 \right) \\
&= \mathbb{E}_{p(x)\pi_0(a|x)p(e|x,a)} \left[\left(w(x,a)^2 - w(x,e)^2 \right) \cdot \mathbb{E}_{p(r|x,a,e)}[r^2] \right] + 2V(\pi)\mathrm{Bias}(\hat{V}_{\mathrm{MIPS}}(\pi)) + \mathrm{Bias}(\hat{V}_{\mathrm{MIPS}}(\pi))^2
\end{aligned}
$$

Thus, we have

$$
\begin{aligned}
&n \left(\mathbb{V}_{\mathcal{D}}[\hat{V}_{\mathrm{IPS}}(\pi;\mathcal{D})] - \mathbb{V}_{\mathcal{D}}[\hat{V}_{\mathrm{MIPS}}(\pi;\mathcal{D})] - \mathrm{Bias}(\hat{V}_{\mathrm{MIPS}}(\pi))^2 \right) \\
&= \mathbb{E}_{p(x)\pi_0(a|x)p(e|x,a)} \left[\left(w(x,a)^2 - w(x,e)^2 \right) \cdot \mathbb{E}_{p(r|x,a,e)}[r^2] \right] + 2V(\pi)\mathrm{Bias}(\hat{V}_{\mathrm{MIPS}}(\pi)) + (1-n)\mathrm{Bias}(\hat{V}_{\mathrm{MIPS}}(\pi))^2
\end{aligned}
$$

□

The first term becomes large when the scale of the marginal importance weights is smaller than that of the vanilla importance weights. The second term becomes large when the value of π is large and MIPS overestimates it by a large margin. The third term can take a large negative value when the sample size is large and the bias of MIPS is large. This summarizes the bias-variance trade-off between the vanilla IPS and MIPS. When the sample size is small, the first and second terms in the MSE gain are dominant, and MIPS is more appealing due to its variance reduction property. However, as the sample size gets larger, the bias becomes dominant, and IPS is expected to overtake MIPS at some point. We would argue that, when the action space is large, the variance reduction of MIPS often provides the gain in MSE, as the variance components are more dominant, which is supported by our experiment.

B.6. Derivation of Eq. (3) in Section 3.3

$$
\begin{aligned}
w(x,e) &= \frac{p(e|x,\pi)}{p(e|x,\pi_0)} \\
&= \frac{\sum_{a\in\mathcal{A}} p(e|x,a)\cdot\pi(a|x)}{p(e|x,\pi_0)} \\
&= \frac{p(e|x,\pi_0)\sum_{a\in\mathcal{A}}(\pi_0(a|x,e)/\pi_0(a|x))\cdot\pi(a|x)}{p(e|x,\pi_0)} \\
&= \sum_{a\in\mathcal{A}} \pi_0(a|x,e)\frac{\pi(a|x)}{\pi_0(a|x)} \\
&= \mathbb{E}_{\pi_0(a|x,e)}\left[w(x,a)\right]
\end{aligned}
\tag{17}
$$

where we use $p(e|x,a) = \frac{p(e|x,\pi_0)\pi_0(a|x,e)}{\pi_0(a|x)}$ in Eq. (17).

B.7. Bias and Variance of MIPS with Estimated Marginal Importance Weights

Theorem B.2. *(Bias of MIPS with Estimated Marginal Importance Weights) If Assumption 3.1 is true, but Assumption 3.2 is violated, MIPS with the estimated marginal importance weight $\hat{w}(x,e)$ has the following bias.*

$$
\mathrm{Bias}(\hat{V}_{\mathrm{MIPS}}(\pi;\hat{w})) = \mathrm{Bias}(\hat{V}_{\mathrm{MIPS}}(\pi)) - \mathbb{E}_{p(x)p(e|x,\pi)}\left[\delta(x,e)q(x,\pi_0,e)\right],
$$

where $\hat{V}_{\mathrm{MIPS}}(\pi;\hat{w}) := n^{-1}\sum_{i=1}^{n}\hat{w}(x_i,e_i)r_i$, $\delta(x,e) := 1 - (\hat{w}(x,e)/w(x,e))$, *and* $q(x,\pi_0,e) := \sum_{a\in\mathcal{A}}\pi_0(a|x,e)\cdot q(x,a,e)$.

Proof.

$$\mathrm{Bias}(\hat{V}_{\mathrm{MIPS}}(\pi;\hat{w})) = \mathbb{E}_{p(x)\pi_0(a|x)p(e|x,a)p(r|x,a,e)}[\hat{w}(x,e)r] - V(\pi) \tag{18}$$

$$= \mathbb{E}_{p(x)\pi_0(a|x)p(e|x,a)p(r|x,a,e)}\left[(\hat{w}(x,e) - w(x,e)) \cdot r\right] + \mathrm{Bias}(\hat{V}_{\mathrm{MIPS}}(\pi)) \tag{19}$$

where we use $\mathbb{E}_{\mathcal{D}}[\hat{V}_{\mathrm{MIPS}}(\pi;\mathcal{D},\hat{w})] = \mathbb{E}_{p(x)\pi_0(a|x)p(e|x,a)p(r|x,a,e)}[\hat{w}(x,e)r]$ (as samples are assumed to be independent) in Eq. (18) and decompose the bias into the bias of MIPS with the true $w(x,e)$ and bias due to the estimation error of $\hat{w}(x,e)$ in Eq. (19). We know the bias of MIPS with the true weight from Theorem 3.5, so we calculate only the bias due to estimating the weight.

$$\begin{aligned}
&\mathbb{E}_{p(x)\pi_0(a|x)p(e|x,a)p(r|x,a,e)}[(\hat{w}(x,e) - w(x,e)) \cdot r] \\
&= \mathbb{E}_{p(x)\pi_0(a|x)p(e|x,a)}[(\hat{w}(x,e) - w(x,e)) \cdot q(x,a,e)] \\
&= \mathbb{E}_{p(x)}\left[\sum_{a\in\mathcal{A}} \pi_0(a|x) \sum_{e\in\mathcal{E}} p(e|x,a) \cdot (\hat{w}(x,e) - w(x,e)) \cdot q(x,a,e)\right] \\
&= \mathbb{E}_{p(x)}\left[\sum_{a\in\mathcal{A}} \pi_0(a|x) \sum_{e\in\mathcal{E}} \frac{p(e|x,\pi_0) \cdot \pi_0(a|x,e)}{\pi_0(a|x)} \cdot (\hat{w}(x,e) - w(x,e)) \cdot q(x,a,e)\right] \qquad (20) \\
&= \mathbb{E}_{p(x)}\left[\sum_{e\in\mathcal{E}} p(e|x,\pi_0) \cdot (\hat{w}(x,e) - w(x,e)) \sum_{a\in\mathcal{A}} \pi_0(a|x,e) \cdot q(x,a,e)\right] \\
&= -\mathbb{E}_{p(x)}\left[\sum_{e\in\mathcal{E}} p(e|x,\pi) \cdot \delta(x,e) \cdot q(x,\pi_0,e)\right] \qquad (21) \\
&= -\mathbb{E}_{p(x)p(e|x,\pi)}\left[\delta(x,e) \cdot q(x,\pi_0,e)\right]
\end{aligned}$$

where we use $p(e|x,a) = \frac{p(e|x,\pi_0)\pi_0(a|x,e)}{\pi_0(a|x)}$ in Eq. (20) and $q(x,\pi_0,e) = \sum_{a\in\mathcal{A}} \pi_0(a|x,e)q(x,a,e)$ in Eq. (21). □

Theorem B.3. *(Variance of MIPS with Estimated Marginal Importance Weights) Under Assumptions 3.1 and 3.2, we have*

$$\begin{aligned}
n\mathbb{V}_{\mathcal{D}}(\hat{V}_{\mathrm{MIPS}}(\pi;\mathcal{D},\hat{w})) = &\mathbb{E}_{p(x)p(e|x,\pi)}\left[(1-\delta(x,e))^2 w(x,e)\sigma^2(x,\pi_0,e)\right] \\
&+ \mathbb{E}_{p(x)}\left[\mathbb{V}_{\pi_0(a|x)p(e|x,a)}\left[\hat{w}(x,e)q(x,a,e)\right]\right] \\
&+ \mathbb{V}_{p(x)}\left[\mathbb{E}_{p(e|x,\pi)}\left[(1-\delta(x,e))q(x,\pi_0,e)\right]\right]
\end{aligned}$$

where $\delta(x,e) := 1-(\hat{w}(x,e)/w(x,e))$, $q(x,\pi_0,e) := \sum_{a\in\mathcal{A}} \pi_0(a|x,e)\cdot q(x,a,e)$, *and* $\sigma^2(x,\pi_0,e) := \sum_{a\in\mathcal{A}} \pi_0(a|x,e)\cdot \sigma^2(x,a,e)$.

Proof. Since the samples are assumed to be independent, we have

$$n\mathbb{V}_{\mathcal{D}}(\hat{V}_{\mathrm{MIPS}}(\pi;\mathcal{D},\hat{w})) = \mathbb{V}_{p(x)\pi_0(a|x)p(e|x,a)p(r|x,a,e)}\left[\hat{w}(x,e)r\right].$$

Below we apply the law of total variance twice to the RHS of the above equation.

$$\begin{aligned}
\mathbb{V}_{p(x)\pi_0(a|x)p(e|x,a)p(r|x,a,e)}\left[\hat{w}(x,e)r\right] &= \mathbb{E}_{p(x)\pi_0(a|x)p(e|x,a)}\left[\hat{w}(x,e)^2 \cdot \mathbb{V}_{p(r|x,a,e)}[r]\right] \\
&\quad + \mathbb{V}_{p(x)\pi_0(a|x)p(e|x,a)}\left[\hat{w}(x,e) \cdot \mathbb{E}_{p(r|x,a,e)}[r]\right] \\
&= \mathbb{E}_{p(x)\pi_0(a|x)p(e|x,a)}\left[\hat{w}(x,e)^2 \cdot \sigma^2(x,a,e)\right] \\
&\quad + \mathbb{V}_{p(x)\pi_0(a|x)p(e|x,a)}\left[\hat{w}(x,e) \cdot q(x,a,e)\right] \\
&= \mathbb{E}_{p(x)\pi_0(a|x)p(e|x,a)}\left[\hat{w}(x,e)^2 \cdot \sigma^2(x,a,e)\right] \\
&\quad + \mathbb{E}_{p(x)}\left[\mathbb{V}_{\pi_0(a|x)p(e|x,a)}\left[\hat{w}(x,e) \cdot q(x,a,e)\right]\right] \\
&\quad + \mathbb{V}_{p(x)}\left[\mathbb{E}_{\pi_0(a|x)p(e|x,a)}\left[\hat{w}(x,e) \cdot q(x,a,e)\right]\right] \\
&= \mathbb{E}_{p(x)p(e|x,\pi)}\left[(1-\delta(x,e))^2 \cdot w(x,e) \cdot \sigma^2(x,\pi_0,e)\right] \\
&\quad + \mathbb{E}_{p(x)}\left[\mathbb{V}_{\pi_0(a|x)p(e|x,a)}\left[\hat{w}(x,e) \cdot q(x,a,e)\right]\right] \\
&\quad + \mathbb{V}_{p(x)}\left[\mathbb{E}_{p(e|x,\pi)}\left[(1-\delta(x,e)) \cdot q(x,\pi_0,e)\right]\right] \qquad (22)
\end{aligned}$$

where we use $\mathbb{E}_{\pi_0(a|x)p(e|x,a)}[\hat{w}(x,e)^2\sigma^2(x,a,e)] = \mathbb{E}_{p(e|x,\pi)}[(1-\delta(x,e))^2 w(x,e)\sigma^2(x,\pi_0,e)]$ and $\mathbb{E}_{\pi_0(a|x)p(e|x,a)}[\hat{w}(x,e)q(x,a,e)] = \mathbb{E}_{p(e|x,\pi)}[(1-\delta(x,e))q(x,\pi_0,e)]$ in Eq. (22) □

B.8. Bias of MIPS with Deficient Embedding Support

Theorem B.4. *(Bias of MIPS with Deficient Embedding Support) If Assumption 3.2 is true, but Assumption 3.1 is violated, MIPS has the following bias.*

$$\left|\mathrm{Bias}(\hat{V}_{\mathrm{MIPS}}(\pi))\right| = \mathbb{E}_{p(x)}\left[\sum_{e\in\mathcal{U}_0^e(x,\pi_0)} p(e|x,\pi)q(x,e)\right],$$

where $\mathcal{U}_0^e(x,\pi_0) := \{e\in\mathcal{E} \mid p(e|x,\pi_0)=0\}$ is the space of unsupported embeddings for context x under π_0.

Proof. We follow Proposition 1 of Sachdeva et al. (2020) to derive the bias under deficient embedding support.

$$\begin{aligned}
\mathrm{Bias}(\hat{V}_{\mathrm{MIPS}}(\pi)) &= \mathbb{E}_{p(x)\pi_0(a|x)p(e|x,a)p(r|x,a,e)}[w(x,e)r] - V(\pi) \\
&= \mathbb{E}_{p(x)}\left[\sum_{e\in(\mathcal{U}_0^e(x,\pi_0))^c} w(x,e)q(x,e)\sum_{a\in\mathcal{A}}\pi_0(a|x)p(e|x,a)\right] - \mathbb{E}_{p(x)p(e|x,\pi)}[q(x,e)] \qquad (23)\\
&= \mathbb{E}_{p(x)}\left[\sum_{e\in(\mathcal{U}_0^e(x,\pi_0))^c} p(e|x,\pi)q(x,e) - \sum_{e\in\mathcal{E}} p(e|x,\pi)q(x,e)\right] \qquad (24)\\
&= -\mathbb{E}_{p(x)}\left[\sum_{e\in\mathcal{U}_0^e(x,\pi_0)} p(e|x,\pi)q(x,e)\right]
\end{aligned}$$

where Eq. (23) is due to Assumption 3.2 and Eq. (24) is from $p(e|x,a) = \frac{p(e|x,\pi_0)\pi_0(a|x,e)}{\pi_0(a|x)}$. □

C. Data-Driven Action Feature Selection Based on Tucker & Lee (2021) and Su et al. (2020b)

Wang et al. (2017) and Su et al. (2020a) describe a procedure for data-driven estimator selection, which is used to tune the built-in hyperparameters of their own estimators. However, their methods need to estimate the bias (or its loose upper bound as a proxy) of the estimator as a subroutine, which is as difficult as OPE itself. Su et al. (2020b) develop a generic data-driven method for estimator selection for OPE called SLOPE, which is based on Lepski's principle (Lepski & Spokoiny, 1997) and does not need a bias estimator. Tucker & Lee (2021) improve the theoretical analysis of Su et al. (2020b), resulting in a refined procedure called SLOPE++.

Given a finite set of estimators $\{\hat{V}_m\}_{m=1}^M$, which is often constructed by varying the value of hyperparameters, the estimator selection problem aims at identifying the estimator that minimizes some notion of estimation error such as the following absolute error with respect to a given target policy π.

$$m^* := \arg\min_{m\in[M]} \left|V(\pi) - \hat{V}_m(\pi;\mathcal{D})\right|,$$

where $\mathcal{D}$ is a given logged bandit dataset.

For solving this selection problem, SLOPE++ requires the following monotonicity assumption (SLOPE requires a slightly stronger assumption).

Assumption C.1. (Monotonicity)

1. $\mathrm{Bias}(\hat{V}_m) \leq \mathrm{Bias}(\hat{V}_{m+1}),\ \forall m\in[M]$
2. $\mathrm{CNF}(\hat{V}_{m+1}) \leq \mathrm{CNF}(\hat{V}_m),\ \forall m\in[M]$

where $\mathrm{CNF}(\hat{V})$ is a high probability bound on the deviation of $\hat{V}$, which requires that the following holds with a probability at least $1-\delta$.

$$\left|\mathbb{E}_{\mathcal{D}}\left[\hat{V}(\pi;\mathcal{D})\right]-\hat{V}(\pi;\mathcal{D})\right| \leq \mathrm{CNF}(\hat{V}),$$

which we can generally bound with high confidence using techniques such as concentration inequalities.

Based on this assumption, Tucker & Lee (2021) derive the following universal bound.

Theorem C.2. *(Theorem 1 of Tucker & Lee (2021)) Given $\delta > 0$, high confidence bound $\mathrm{CNF}(\hat{V}_m)$ on the deviations, and that we have ordered the candidate estimators such that $\mathrm{CNF}(\hat{V}_{m+1}) \leq \mathrm{CNF}(\hat{V}_m)$. Selecting the estimator as*

$$\hat{m} := \max\left\{m : \left|\hat{V}_m - \hat{V}_j\right| \leq \mathrm{CNF}(m) + (\sqrt{6}-1)\mathrm{CNF}(j), j < m\right\} \tag{25}$$

ensures that with probability at least $1-\delta$,

$$\left|\hat{V}_{\hat{m}} - \hat{V}_{m^*}\right| \leq (\sqrt{6}+3)\min_m\left(\max_{j\leq m}\mathrm{Bias}(j) + \mathrm{CNF}(m)\right).$$

Under Assumption C.1, the bound simplifies to

$$\left|\hat{V}_{\hat{m}} - \hat{V}_{m^*}\right| \leq (\sqrt{6}+3)\min_m\left(\mathrm{Bias}(m) + \mathrm{CNF}(m)\right).$$

In contrast, when the set of estimators is not ordered with respect to $\mathrm{CNF}(\cdot)$, we have a looser bound as below.

$$\left|\hat{V}_{\hat{m}} - \hat{V}_{m^*}\right| \leq (\sqrt{6}+3)\min_m\left(\max_{j\leq m}\mathrm{Bias}(j) + \max_{k\leq m}\mathrm{CNF}(k)\right).$$

Note that Tucker & Lee (2021) also provide the corresponding universal upper bound with respect to MSE in their Corollary 1.1.

We build on the selection procedure given in Eq. (25) to implement data-driven action feature selection. Specifically, in our case, the task is to identify which dimensions of the action embedding e we should use to minimize the MSE of the resulting MIPS as follows.

$$\min_{\mathcal{E}\subseteq\mathcal{V}} \mathrm{Bias}\big(\hat{V}_{\mathrm{MIPS}}(\pi;\mathcal{E})\big)^2 + \mathbb{V}_{\mathcal{D}}\big[\hat{V}_{\mathrm{MIPS}}\big(\pi;\mathcal{D},\mathcal{E}\big)\big]$$

where $\mathcal{V} := \{\mathcal{E}_1, \mathcal{E}_2, \ldots, \mathcal{E}_k\}$ is a set of available action features. Note that we make the dependence of MIPS on the action embedding space $\mathcal{E}$ explicit in the above formulation.

As described in Theorems 3.5, 3.6, and 3.7, we should use as many dimensions as possible to reduce the bias, while we should use as coarse information as possible to gain a large variance reduction. For identifying useful features to compute the marginal importance weights, we construct a set of estimators $\{\hat{V}_{\mathrm{MIPS}}(\pi;\mathcal{D},\mathcal{E})\}_{\mathcal{E}\subseteq\mathcal{V}}$ and simply apply Eq. (25). Note that when the number of embedding dimensions is not small, the brute-force search over all possible combinations of the embedding dimensions is not tractable. Thus, we sometime define the action embedding search space $\mathcal{V}$ via a greedy procedure to make the embedding selection tractable. In our experiments, we perform action embedding selection based on the greedy version of SLOPE++, and we estimate a high probability bound on the deviation ($\mathrm{CNF}(\hat{V})$) based on the Student's t distribution as done in Thomas et al. (2015). The MIPS estimator along with the exact and greedy versions of embedding dimension selection is now implemented in the OBP package.[7].

D. Experiment Details and Additional Results

D.1. Baseline Estimators

Below, we define and describe the baseline estimators in detail.

[7] https://github.com/st-tech/zr-obp

Direct Method (DM) DM is defined as follows.

$$\hat{V}_{\mathrm{DM}}(\pi; \mathcal{D}, \hat{q}) := \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_{\pi(a|x_i)}[\hat{q}(x_i, a)] = \frac{1}{n} \sum_{i=1}^{n} \sum_{a \in \mathcal{A}} \pi(a|x_i) \hat{q}(x_i, a),$$

where $\hat{q}(x, a)$ estimates $q(x, a)$ based on logged bandit data. The accuracy of DM depends on the quality of $\hat{q}(x, a)$. If $\hat{q}(x, a)$ is accurate, so is DM. However, if $\hat{q}(x, a)$ fails to estimate the expected reward accurately, the final estimator is no longer consistent. As discussed in Appendix A, the misspecification issue is challenging, as it cannot be easily detected from available data (Farajtabar et al., 2018; Voloshin et al., 2019). This is why DM is often described as a high bias estimator.

Doubly Robust (DR) (Dudík et al., 2014) DR is defined as follows.

$$\hat{V}_{\mathrm{DR}}(\pi; \mathcal{D}, \hat{q}) := \frac{1}{n} \sum_{i=1}^{n} \left\{ \mathbb{E}_{\pi(a|x_i)}[\hat{q}(x_i, a)] + w(x_i, a_i)(r_i - \hat{q}(x_i, a_i)) \right\},$$

which combines DM and IPS in a way to reduce the variance. More specifically, DR utilizes $\hat{q}$ as a control variate. If the expected reward is correctly specified, DR is *semiparametric efficient* meaning that it achieves the minimum possible asymptotic variance among regular estimators (Narita et al., 2019). A problem is that, if the expected reward is misspecified, this estimator can have a larger asymptotic MSE compared to IPS.

Switch Doubly Robust (Switch-DR) (Wang et al., 2017) Although DR generally reduces the variance of IPS and is also minimax optimal (Wang et al., 2017), it can still suffer from the variance issue in practice, particularly when the importance weights are large due to a weak overlap between target and logging policies. Switch-DR is introduced to further deal with the variance issue and is defined as follows.

$$\hat{V}_{\mathrm{SwitchDR}}(\pi; \mathcal{D}, \hat{q}, \lambda) := \frac{1}{n} \sum_{i=1}^{n} \left\{ \mathbb{E}_{\pi(a|x_i)}[\hat{q}(x_i, a)] + w(x_i, a_i) \mathbb{I}\{w(x_i, a_i) \leq \lambda\}(r_i - \hat{q}(x_i, a_i)) \right\},$$

where $\mathbb{I}\{\cdot\}$ is the indicator function and $\lambda \geq 0$ is a hyperparameter. When $\lambda = 0$, Switch-DR becomes DM, while $\lambda \to \infty$ leads to DR. Switch-DR is also minimax optimal when λ is appropriately set (Wang et al., 2017).

More Robust Doubly Robust (Farajtabar et al., 2018) MRDR uses an expected reward estimator ($\hat{q}_{\mathrm{MRDR}}$) derived by minimizing the variance of the resulting DR estimator. This estimator is defined as $\hat{V}_{\mathrm{MRDR}}(\pi; \mathcal{D}, \hat{q}_{\mathrm{MRDR}}) := \hat{V}_{\mathrm{DR}}(\pi; \mathcal{D}, \hat{q}_{\mathrm{MRDR}})$, where $\hat{q}_{\mathrm{MRDR}}$ is derived by minimizing the (empirical) variance objective: $\hat{q}_{\mathrm{MRDR}} \in \arg\min_{\hat{q} \in \mathcal{Q}} \mathbb{V}_n(\hat{V}_{\mathrm{DR}}(\pi; \mathcal{D}, \hat{q}))$, where $\mathcal{Q}$ is a function class for $\hat{q}$. When $\mathcal{Q}$ is well-specified, then $\hat{q}_{\mathrm{MRDR}} = q$. The main point is that, even if $\mathcal{Q}$ is misspecified, MRDR is still expected to perform reasonably well, as the target function is the resulting variance. To implement MRDR, we follow Farajtabar et al. (2018) and Su et al. (2020a), and derive $\hat{q}_{\mathrm{MRDR}}$ by minimizing the weighted squared loss with respect to the reward prediction on the logged data.

Doubly Robust with Optimistic Shrinkage (Su et al., 2020a) DRos is defined via minimizing an upper bound of the MSE and is defined as follows.

$$\hat{V}_{\mathrm{DRos}}(\pi; \mathcal{D}, \hat{q}, \lambda) := \frac{1}{n} \sum_{i=1}^{n} \left\{ \mathbb{E}_{\pi(a|x_i)}[\hat{q}(x_i, a)] + \frac{\lambda w(x_i, a_i)}{w(x_i, a_i)^2 + \lambda}(r_i - \hat{q}(x_i, a_i)) \right\},$$

where $\lambda \geq 0$ is a hyperparameter. When $\lambda = 0$, DRos is equal to DM, while $\lambda \to \infty$ makes DRos identical to DR. DRos is aimed at improving the small sample performance of DR, but is indeed biased due to the weight shrinkage.

DR-λ (Metelli et al., 2021) DR-λ is a recent estimator building on a "smooth shrinkage" of the importance weights to mitigate the heavy-tailed behavior of the previous estimators. This estimator is defined as follows.

$$\hat{V}_{\mathrm{DR}-\lambda}(\pi; \mathcal{D}, \hat{q}, \lambda) := \frac{1}{n} \sum_{i=1}^{n} \left\{ \mathbb{E}_{\pi(a|x_i)}[\hat{q}(x_i, a)] + \frac{w(x_i, a_i)}{1 - \lambda + \lambda w(x_i, a_i)}(r_i - \hat{q}(x_i, a_i)) \right\},$$

where $\lambda \in [0, 1]$ is a hyperparameter. Note that Metelli et al. (2021) define a more general weight, $((1 - \lambda) w(x, a)^s + \lambda)^{\frac{1}{s}}$, with an additional hyperparameter s. The above instance is a special case with $s = 1$, which is the main proposal of Metelli et al. (2021).

D.2. Additional Results on Synthetic Bandit Data

In this section, we explore two additional research questions regarding the estimators' performance for different logging/target policies and different levels of noise on the rewards. We demonstrate that MIPS works particularly better than other baselines when the target and logging policies differ greatly and the reward is noisy. After discussing the two research questions, we report detailed experimental results regarding the research questions addressed in the main text with additional baselines.

How does MIPS perform with varying logging and target policies? We compare the MSE, squared bias, and variance of the estimators (DM, IPS, DR, MIPS, and MIPS with the true weights) with varying logging and target policies. We can do this by varying the values of β and ϵ as described in Section 4. Note that we set $\beta = -1$ and $\epsilon = 0.05$ for all synthetic results in the main text.

First, Figure 8 reports the results with varying logging policies ($\beta \in \{-3, -2, -1, 0, 1, 2, 3\}$) and with a near-optimal/near-deterministic target policy defined by $\epsilon = 0.05$ (fixed). A large negative value of β leads to a worse logging policy, meaning that it creates a large discrepancy between logging and target policies in this setup. The left column of Figure 8 demonstrates that the MSEs of the estimators generally become larger for larger negative values of β as expected. Most notably, the MSEs of IPS and DR blow up for $\beta = -3, -2$ due to their inflated variance as suggested in the right column of the same figure. On the other hand, MIPS and MIPS (true) work robustly for a range of logging policies, suggesting the strong variance reduction for the case with a large discrepancy between policies. DM also suffers from a larger discrepancy between logging and target policies due to its increased bias caused by the extrapolation error issue.

Next, Figure 9 shows the results with varying target policies ($\epsilon = \{0.0, 0.2, 0.4, 0.6, 0.8, 1.0\}$) and with a logging policy slightly worse than uniform random defined by $\beta = -1$ (fixed). A larger value of ϵ introduces a larger entropy for the target policy, making it closer to the logging policy in this setup (an extreme case with $\epsilon = 1.0$ produces a uniform random target policy). On the other hand, $\epsilon = 0$ produces the optimal, deterministic target policy, which makes OPE harder given $\beta = -1$. The left column of Figure 9 suggests that all estimators perform worse for smaller values of ϵ as expected. IPS and DR perform worse as their variance increases with decreasing ϵ, while DM performs worse as it produces larger bias. The variance of MIPS also increases with decreasing ϵ, but it is often much smaller and robust than those of IPS and DR. Note that, for the uniform random target policy ($\epsilon = 1.0$), all estimators are very accurate and there is no significant difference among the estimators.

How does MIPS perform with varying noise levels? Next, we explore how the level of noise on the rewards affects the comparison of the estimators. To this end, we vary the noise level $\sigma \in \{0.5, 1.0, 1.5, \ldots, 4.0\}$ where σ is the standard deviation of the Gaussian noise, i.e., $r \sim \mathcal{N}(q(x,a), \sigma^2)$. As stated in the main text, the variance of IPS grows when the reward is noisy. Theorem 3.6 also implies that the variance reduction of MIPS becomes more appealing with the noisy rewards. Figure 10 empirically supports these claims. Specifically, IPS significantly exacerbates its MSE from 0.55 (when $\sigma = 0.5$) to 3.22 (when $\sigma = 4.0$). MIPS also struggles with noisy rewards, but the improvement of MIPS compared to IPS/DR becomes larger with the added noise. When the noise level is small ($\sigma = 0.5$), $\frac{\mathrm{MSE}(\hat{V}_{\mathrm{IPS}})}{\mathrm{MSE}(\hat{V}_{\mathrm{MIPS}})} = 2.97$, while $\frac{\mathrm{MSE}(\hat{V}_{\mathrm{IPS}})}{\mathrm{MSE}(\hat{V}_{\mathrm{MIPS}})} = 14.98$ when the noise is large ($\sigma = 4.0$). Different from IPS, DR, and MIPS, DM is not affected so much by the noise level and becomes increasingly better than IPS and DR in noisy environments. Nontheless, MIPS achieves much smaller MSE than DM even with noisy rewards.

Comparison with additional baselines across additional experimental conditions. We include additional baselines (Switch-DR, DRos, and DR-λ) described in Appendix D.1 to the empirical evaluations. Their built-in hyperparameters are tuned with SLOPE++ proposed by Tucker & Lee (2021), which slightly improves the original SLOPE of Su et al. (2020b). We use implementations of these advanced estimators provided by OBP (version 0.5.5). We evaluate the four research questions addressed in the main text with six different pairs of (β, ϵ). Figures 11-14 report the results with $\beta = -1$ and $\epsilon = 0.05$. Figures 15-18 report the results with $\beta = -1$ and $\epsilon = 0.8$. Figures 19-22 report the results with $\beta = 0$ and $\epsilon = 0.05$. Figures 23-26 report the results with $\beta = 0$ and $\epsilon = 0.8$. Figures 27-30 report the results with $\beta = 1$ and $\epsilon = 0.05$. Figures 31-34 report the results with $\beta = 1$ and $\epsilon = 0.8$.

In general, we observe results similar to those reported in the main text. Specifically, MIPS works better than all existing estimators, including the advanced ones, in a range of situations, in particular for small data and large action spaces. This result suggests that even the recent state-of-the-art estimators fail to deal with large action spaces. Regarding the additional baselines, Switch-DR, DRos, and DR-λ work similarly to DM. These estimators fail to improve their variance with the

growing sample sizes and become worse than IPS and DR in large sample regimes. This observation suggests that SLOPE++ avoids huge importance weights and favors low variance, but highly biased estimators in our setup. We indeed also tested the More Robust Doubly Robust (MRDR) estimator (Farajtabar et al., 2018), but find that MRDR suffers from its growing variance with a growing number of actions and works similarly to IPS and DR.

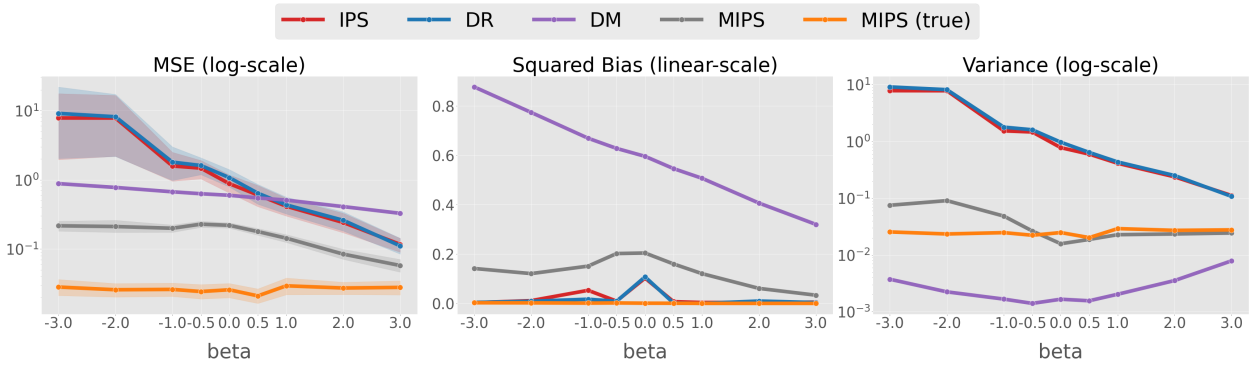

Figure 8. MSE, Squared Bias, and Variance with **varying logging policies** (β)

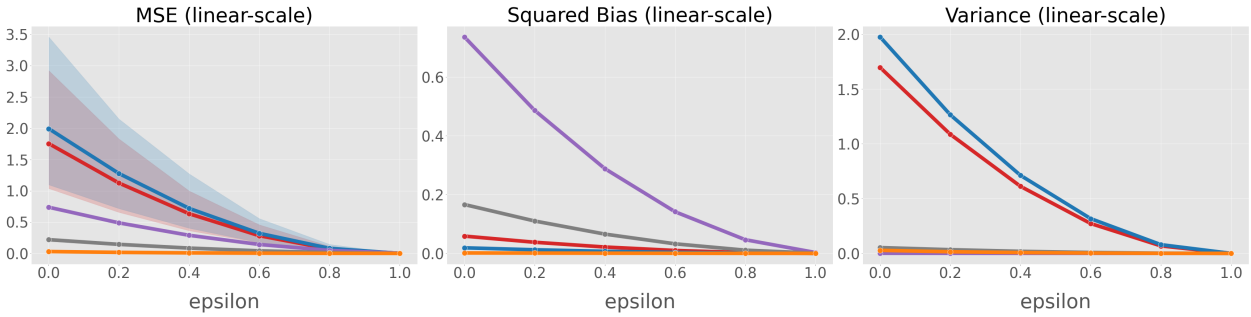

Figure 9. MSE, Squared Bias, and Variance with **varying target policies** (ϵ)

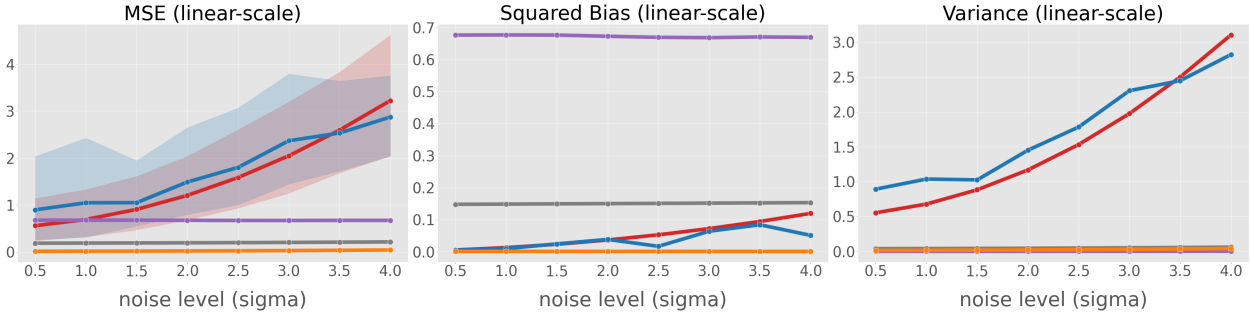

Figure 10. MSE, Squared Bias, and Variance with **varying noise levels** (σ)

Note: We set $n = 10,000$ and $|\mathcal{A}| = 1,000$. For Figure 8, we fix $\epsilon = 0.05, \sigma = 2.5$, for Figure 9, we fix $\beta = -1, \sigma = 2.5$, and for Figure 10, we fix $\epsilon = 0.05, \beta = -1$. The results are averaged over 100 different sets of synthetic logged data replicated with different random seeds. The shaded regions in the MSE plots represent the 95% confidence intervals estimated with bootstrap.

MSE	Squared Bias	Variance

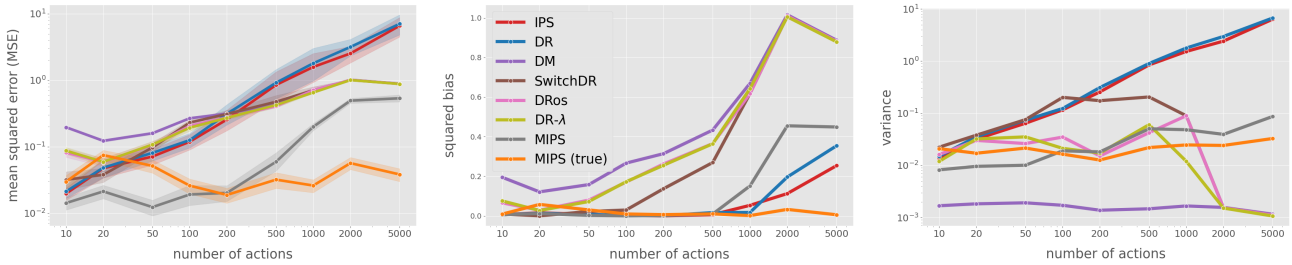

Figure 11. MSE, Squared Bias, and Variance with **varying number of actions**

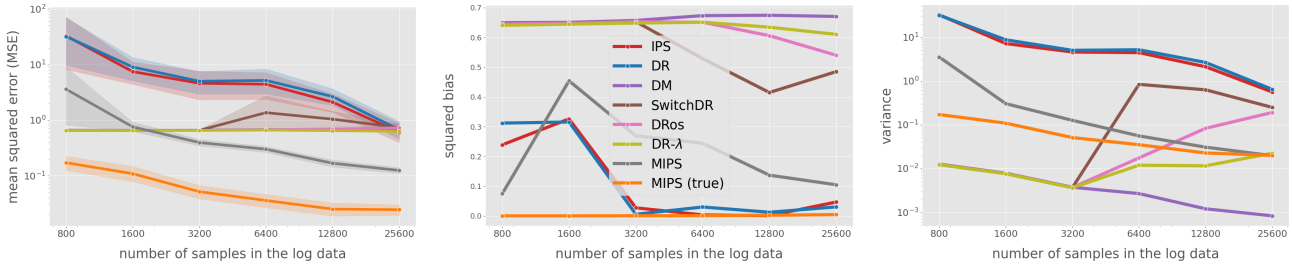

Figure 12. MSE, Squared Bias, and Variance with **varying sample size**

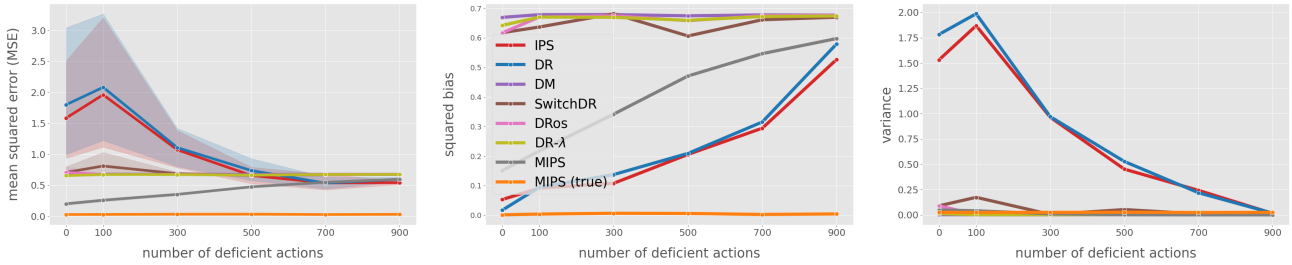

Figure 13. MSE, Squared Bias, and Variance **with varying number of deficient actions**

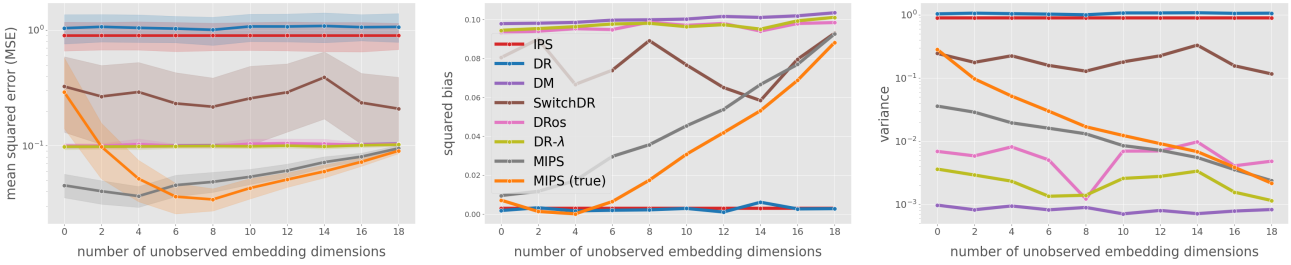

Figure 14. MSE, Squared Bias, and Variance with **varying number of unobserved dimensions in action embeddings**

Note: We set $\beta = -1$ and $\epsilon = 0.05$, which produce **logging policy slightly worse than uniform random** and **near-optimal/near-deterministic target policy**. The results are averaged over 100 different sets of synthetic logged data replicated with different random seeds. The shaded regions in the MSE plots represent the 95% confidence intervals estimated with bootstrap. The y-axis of MSE and Variance plots (the left and right columns) is reported on log-scale.

MSE	Squared Bias	Variance

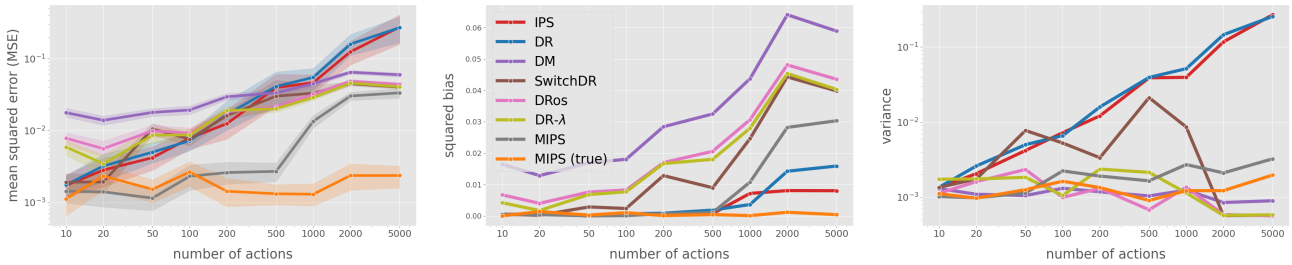

Figure 15. MSE, Squared Bias, and Variance with **varying number of actions**

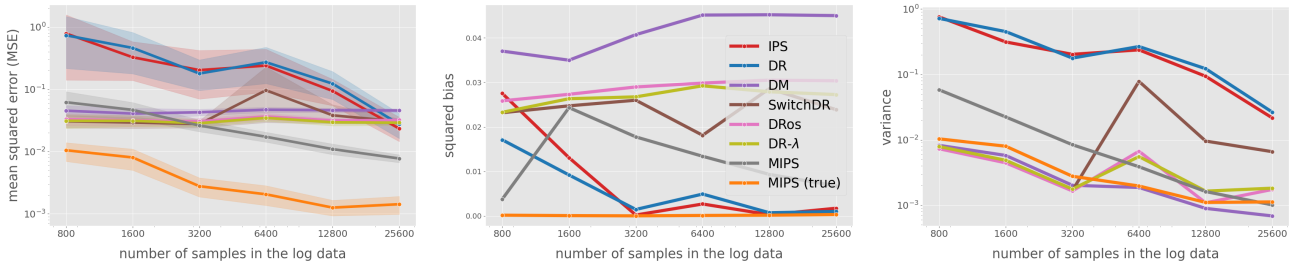

Figure 16. MSE, Squared Bias, and Variance with **varying sample size**

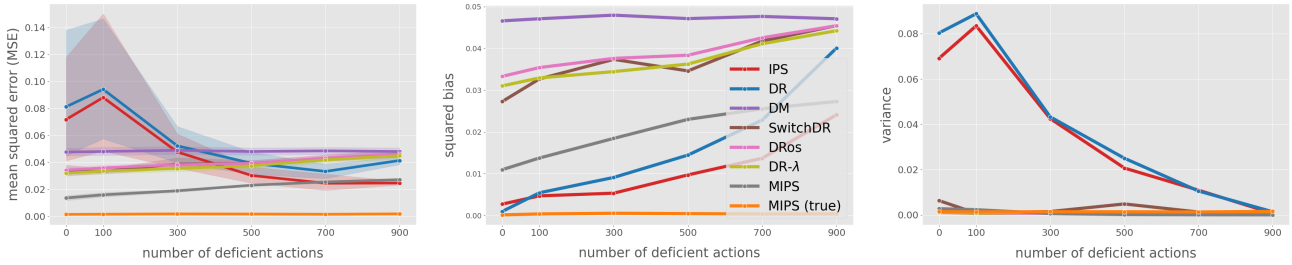

Figure 17. MSE, Squared Bias, and Variance **with varying number of deficient actions**

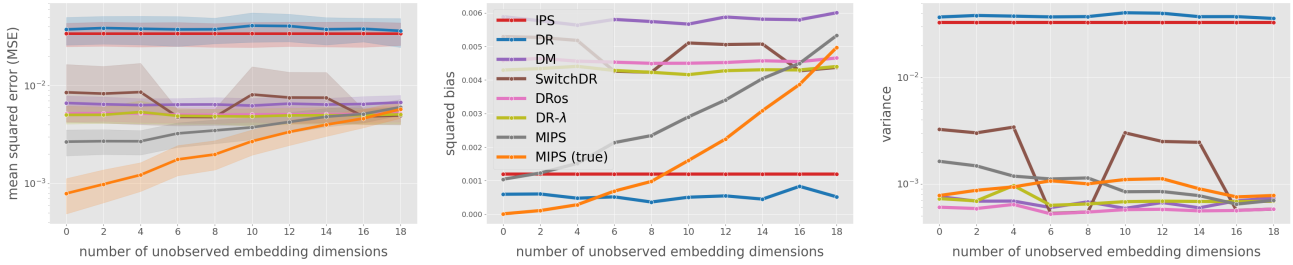

Figure 18. MSE, Squared Bias, and Variance with **varying number of unobserved dimensions in action embeddings**

Note: We set $\beta = -1$ and $\epsilon = 0.8$, which produce **logging policy slightly worse than uniform random** and **near-uniform target policy**. The results are averaged over 100 different sets of synthetic logged data replicated with different random seeds. The shaded regions in the MSE plots represent the 95% confidence intervals estimated with bootstrap. The y-axis of MSE and Variance plots (the left and right columns) is reported on log-scale.

MSE	Squared Bias	Variance

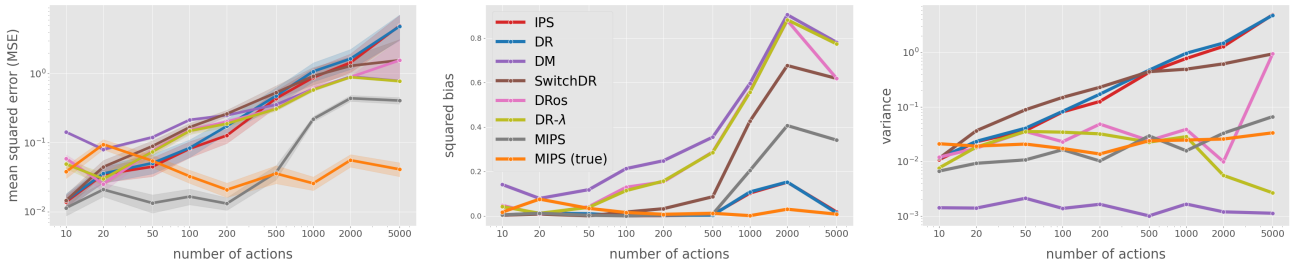

Figure 19. MSE, Squared Bias, and Variance with **varying number of actions**

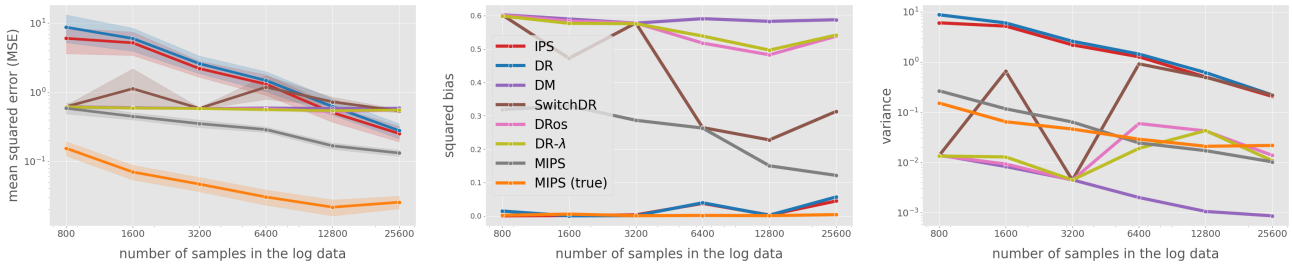

Figure 20. MSE, Squared Bias, and Variance with **varying sample size**

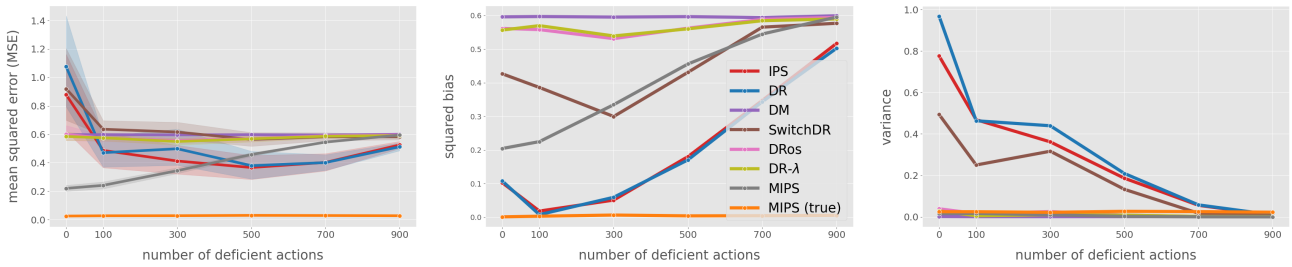

Figure 21. MSE, Squared Bias, and Variance **with varying number of deficient actions**

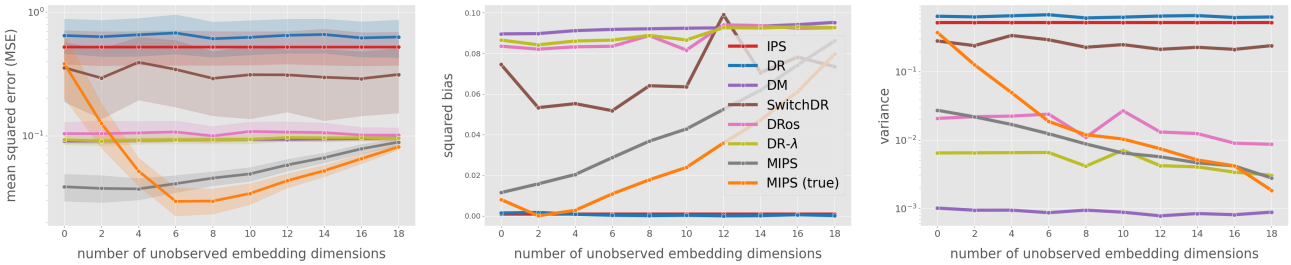

Figure 22. MSE, Squared Bias, and Variance with **varying number of unobserved dimensions in action embeddings**

Note: We set $\beta = 0$ and $\epsilon = 0.05$, which produce **uniform random logging policy** and **near-optimal/near-deterministic target policy**. The results are averaged over 100 different sets of synthetic logged data replicated with different random seeds. The shaded regions in the MSE plots represent the 95% confidence intervals estimated with bootstrap. The y-axis of MSE and Variance plots (the left and right columns) is reported on log-scale.

MSE	Squared Bias	Variance

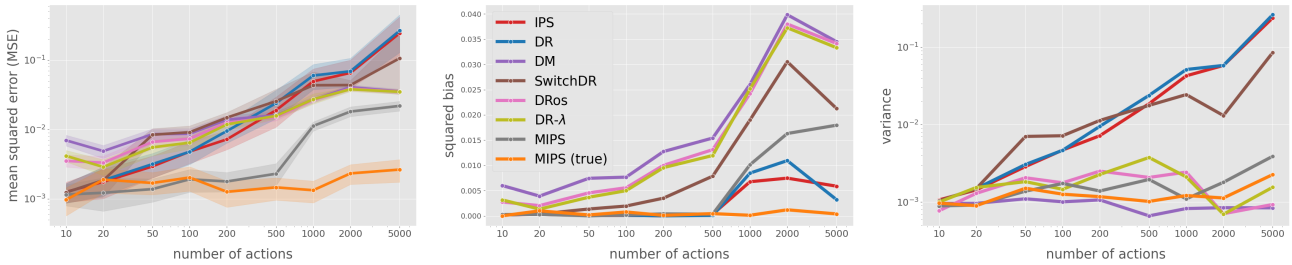

Figure 23. MSE, Squared Bias, and Variance with **varying number of actions**

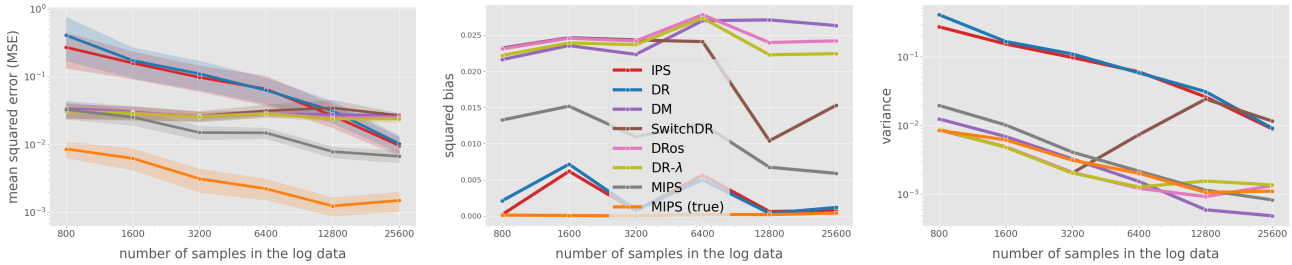

Figure 24. MSE, Squared Bias, and Variance with **varying sample size**

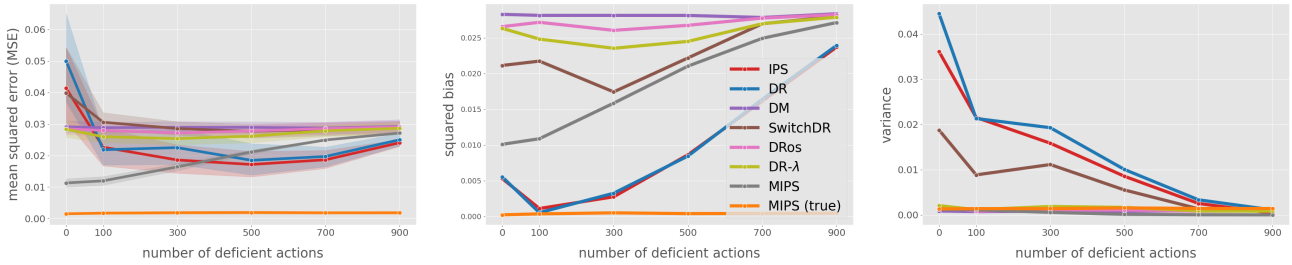

Figure 25. MSE, Squared Bias, and Variance **with varying number of deficient actions**

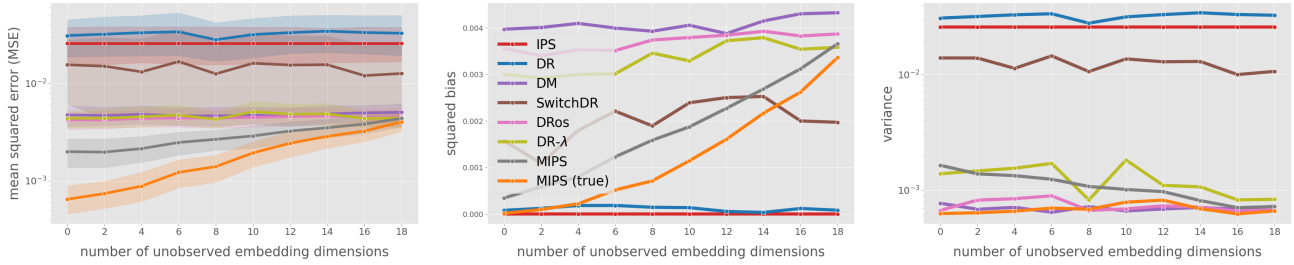

Figure 26. MSE, Squared Bias, and Variance with **varying number of unobserved dimensions in action embeddings**

Note: We set $\beta = 0$ and $\epsilon = 0.8$, which produce **uniform random logging policy** and **near-uniform target policy**. The results are averaged over 100 different sets of synthetic logged data replicated with different random seeds. The shaded regions in the MSE plots represent the 95% confidence intervals estimated with bootstrap. The y-axis of MSE and Variance plots (the left and right columns) is reported on log-scale.

MSE	Squared Bias	Variance

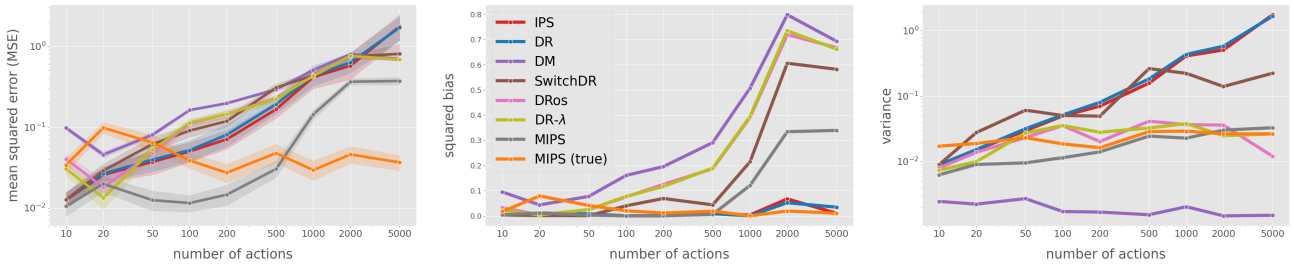

Figure 27. MSE, Squared Bias, and Variance with **varying number of actions**

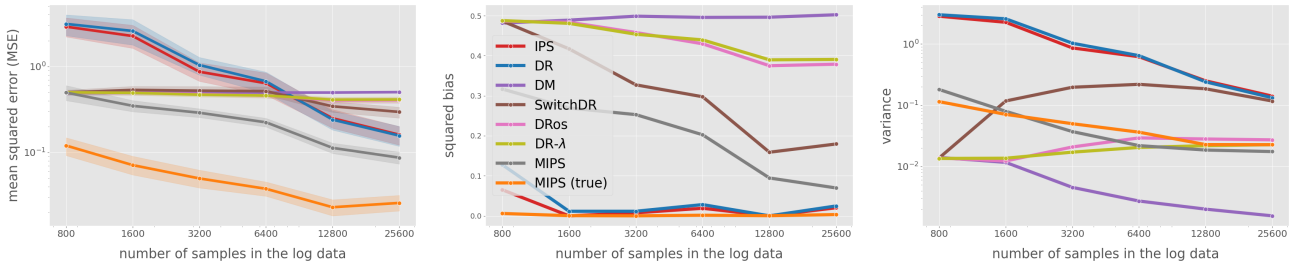

Figure 28. MSE, Squared Bias, and Variance with **varying sample size**

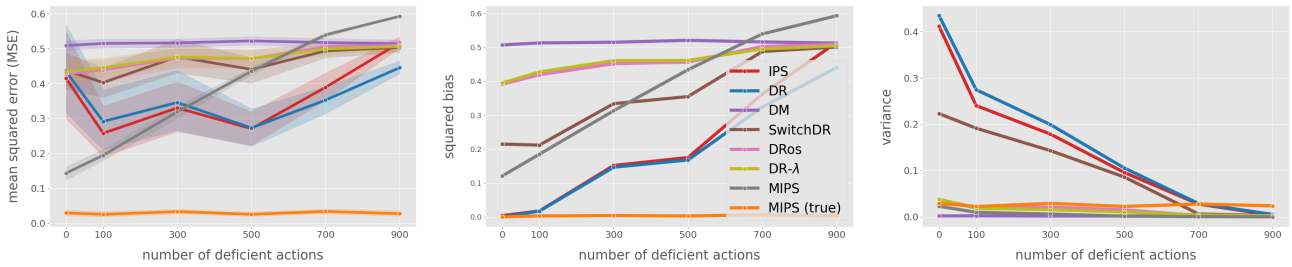

Figure 29. MSE, Squared Bias, and Variance **with varying number of deficient actions**

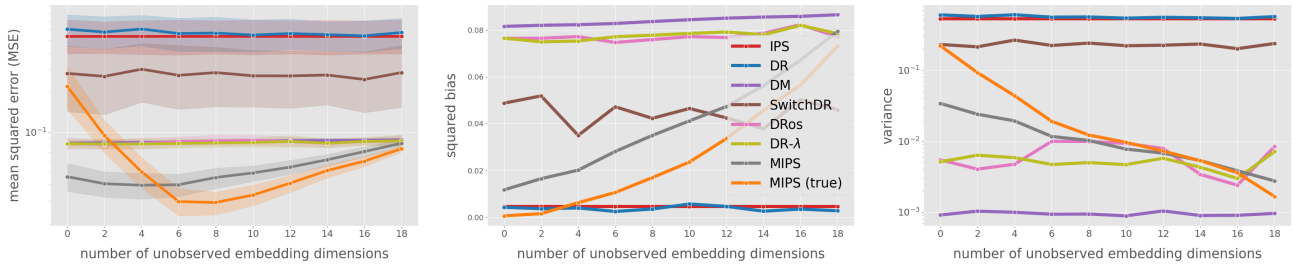

Figure 30. MSE, Squared Bias, and Variance with **varying number of unobserved dimensions in action embeddings**

Note: We set $\beta = 1$ and $\epsilon = 0.05$, which produce **logging policy slightly better than uniform random** and **near-optimal/near-deterministic target policy**. The results are averaged over 100 different sets of synthetic logged data replicated with different random seeds. The shaded regions in the MSE plots represent the 95% confidence intervals estimated with bootstrap. The y-axis of MSE and Variance plots (the left and right columns) is reported on log-scale.

MSE	Squared Bias	Variance

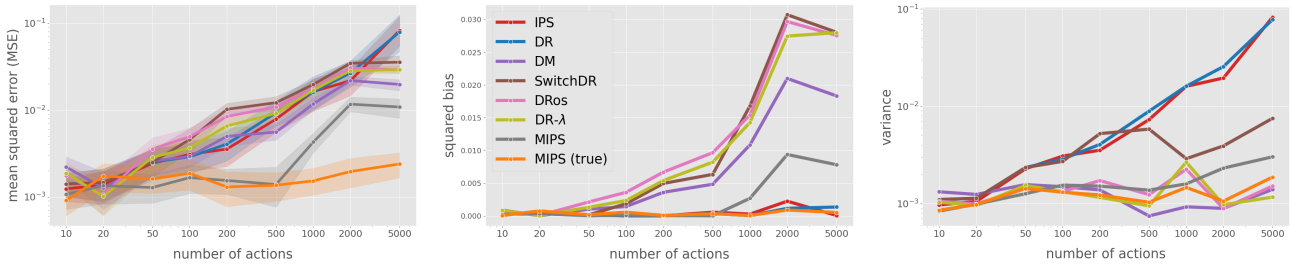

Figure 31. MSE, Squared Bias, and Variance with **varying number of actions**

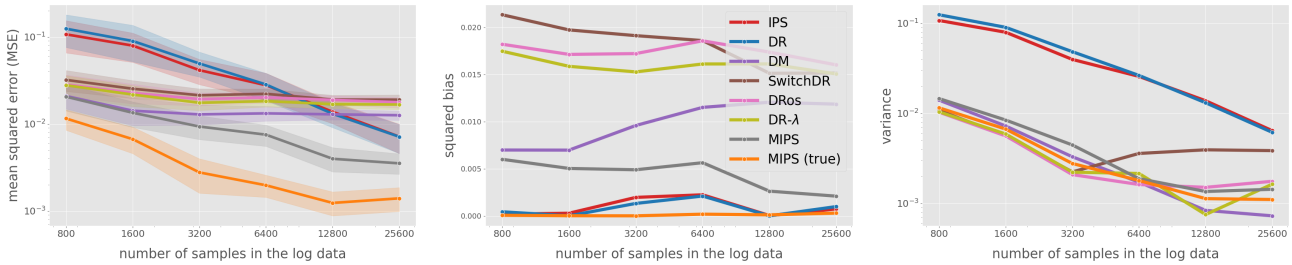

Figure 32. MSE, Squared Bias, and Variance with **varying sample size**

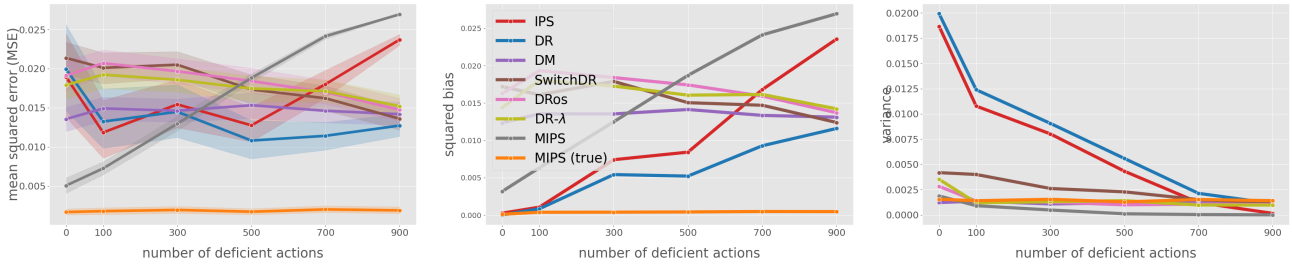

Figure 33. MSE, Squared Bias, and Variance **with varying number of deficient actions**

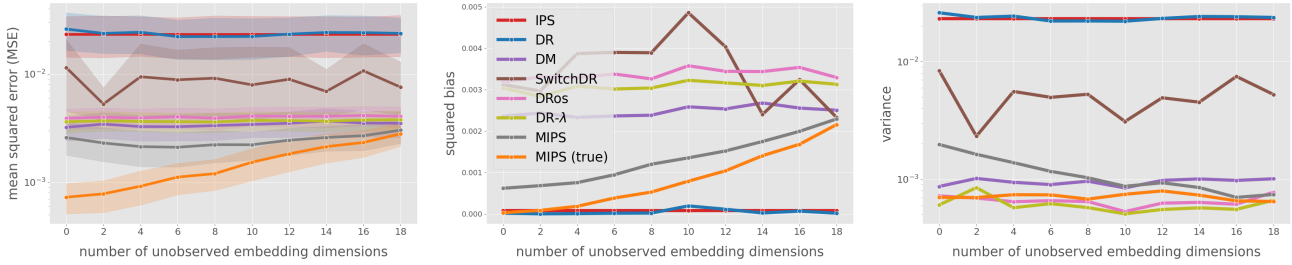

Figure 34. MSE, Squared Bias, and Variance with **varying number of unobserved dimensions in action embeddings**

Note: We set $\beta = 1$ and $\epsilon = 0.8$, which produce **logging policy slightly better than uniform random** and **near-uniform target policy**. The results are averaged over 100 different sets of synthetic logged data replicated with different random seeds. The shaded regions in the MSE plots represent the 95% confidence intervals estimated with bootstrap. The y-axis of MSE and Variance plots (the left and right columns) is reported on log-scale.

Algorithm 1 An Experimental Procedure to Evaluate an OPE Estimator with Real-World Bandit Data

Require: an estimator to be evaluated $\hat{V}$, target policy and corresponding logged bandit data $(\pi, \mathcal{D})$, logging policy and corresponding logged bandit data $(\pi_0, \mathcal{D}_0)$, sample size in OPE n, number of random seeds T
Ensure: empirical CDF of the squared error $(\hat{F}_Z)$
$\mathcal{Z} \leftarrow \emptyset$ (initialize set of results)
for $t = 1, 2, \ldots, T$ **do**
  $\mathcal{D}^*_{0,t} \leftarrow \mathrm{Bootstrap}(\mathcal{D}_0; n)$ *// randomly sample size n of bootstrapped samples*
  $z' \leftarrow \big(V_{\mathrm{on}}(\pi; \mathcal{D}) - \hat{V}(\pi; \mathcal{D}^*_{0,t})\big)^2 / \big(V_{\mathrm{on}}(\pi; \mathcal{D}) - \hat{V}_{\mathrm{IPS}}(\pi; \mathcal{D}^*_{0,t})\big)^2$ *// calculate the relative SE of $\hat{V}$ w.r.t IPS*
  $\mathcal{Z} \leftarrow \mathcal{Z} \cup \{z'\}$ *// store the result*
end for
Estimate CDF of relative SE (F_Z) based on $\mathcal{Z}$ (Eq. 26)

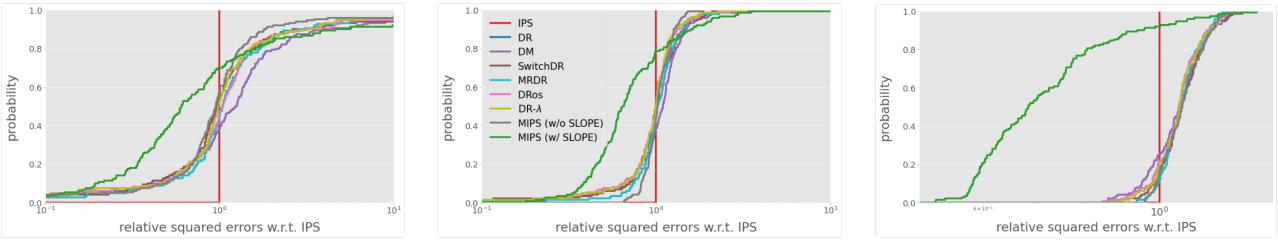

Figure 35. CDF of squared errors relative to IPS with different sample sizes (From left to right, $n = 10000, 50000, 500000$). CDFs are estimated with 150 different sets of bootstrapped logged bandit data. Note that the x-axis is reported on a log-scale.

D.3. Experimental Procedure to Evaluate OPE Estimators on Real-World Bandit Data

Following Saito et al. (2020; 2021), we empirically evaluate the accuracy of the estimators by leveraging two sources of logged bandit data collected by running two different policies denoted as π (regarded as target policy) and π_0 (regarded as logging policy). We let $\mathcal{D}$ denote a logged bandit dataset collected by π and $\mathcal{D}_0$ denote that collected by π_0. We then apply the following procedure to evaluate the accuracy of an OPE estimator $\hat{V}$.

1. Perform bootstrap sampling on $\mathcal{D}_0$ and construct $\mathcal{D}^*_0 := \{(x^*_i, a^*_i, r^*_i)\}_{i=1}^n$, which consists of size n of independently resampled data with replacement.
2. Estimate the policy value of π using $\mathcal{D}^*_0$ and OPE estimator $\hat{V}$. We represent a policy value estimated by $\hat{V}$ as $\hat{V}(\pi; \mathcal{D}^*_0)$.
3. Evaluate the estimation accuracy of $\hat{V}$ with the following *relative squared error w.r.t IPS* (rel-SE):
$$\textit{rel-SE}(\hat{V}; \mathcal{D}^*_0) := \big(V_{\mathrm{on}}(\pi; \mathcal{D}) - \hat{V}(\pi; \mathcal{D}^*_0)\big)^2 / \big(V_{\mathrm{on}}(\pi; \mathcal{D}) - \hat{V}_{\mathrm{IPS}}(\pi; \mathcal{D}^*_0)\big)^2,$$
where $\hat{V}_{\mathrm{on}}(\pi; \mathcal{D}) := |\mathcal{D}|^{-1} \sum_{(\cdot,\cdot,r_j)\in\mathcal{D}} r_j$ is the Monte-Carlo estimate of $V(\pi)$ based on on-policy data $\mathcal{D}$.
4. Repeat the above process T times with different random seeds, and estimate the CDF of the relative SE as follows.
$$\hat{F}_Z(z) := \frac{1}{T} \sum_{t=1}^{T} \mathbb{I}\big\{\textit{rel-SE}_t(\hat{V}; \mathcal{D}^*_{0,t}) \leq z\big\}, \tag{26}$$
where $\textit{rel-SE}(\hat{V}; \mathcal{D}^*_{0,t})$ is the relative SE of $\hat{V}$ computed with the t-th bootstrapped samples $\mathcal{D}^*_{0,t}$.

Algorithm 1 describes this experimental protocol for evaluating OPE estimators in detail. Figure 35 reports the results with real bandit data for varying numbers of logged data ($n = 10000, 50000, 500000$). Note that we use the Random Forest implemented in *scikit-learn* along with 2-fold cross-fitting (Newey & Robins, 2018) to obtain $\hat{q}(x, e)$ for the model-dependent estimators. We also use the Categorical Naive Bayes[8] to estimate $\hat{\pi}_0(a|x, e)$ for MIPS.

[8] https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.CategoricalNB.html

Note that we use OBD's "ALL" campaign, because it has the largest number of actions among three available campaigns. We also regard the same action presented at a different position in a recommendation interface as different actions. As OBD has 80 unique actions and 3 different positions in its recommendation interface, the resulting action space has the cardinality of $80 \times 3 = 240$.