Uncertainty-Aware Instance Reweighting for Off-Policy Learning

Xiaoying Zhang[1] **Junpu Chen**[2] **Hongning Wang**[3] **Hong Xie**[4] **Yang Liu**[1]
John C.S. Lui[5] **Hang Li**[1]
[1]ByteDance Research [2]ChongQing University [3]Tsinghua University
[4] Chongqing Institute of Green and Intelligent Technology, Chinese Academy of Science
[5] The Chinese University of Hong Kong
{zhangxiaoying.xy,yang.liu01,lihang.lh}@bytedance.com
{jumpchan98,hongx87,wang.hongn}@gmail.com
cslui@cse.cuhk.edu.hk

Abstract

Off-policy learning, referring to the procedure of policy optimization with access only to logged feedback data, has shown importance in various real-world applications, such as search engines and recommender systems. While the ground-truth logging policy is usually unknown, previous work simply employs its estimated value for the off-policy learning, ignoring the negative impact from both high bias and high variance resulted from such an estimator. And such impact is often magnified on samples with small and inaccurately estimated logging probabilities. The contribution of this work is to explicitly model the uncertainty in the estimated logging policy, and propose an Uncertainty-aware Inverse Propensity Score estimator (UIPS) for improved off-policy learning, with a theoretical convergence guarantee. Experiment results on the synthetic and real-world recommendation datasets demonstrate that UIPS significantly improves the quality of the discovered policy, when compared against an extensive list of state-of-the-art baselines.

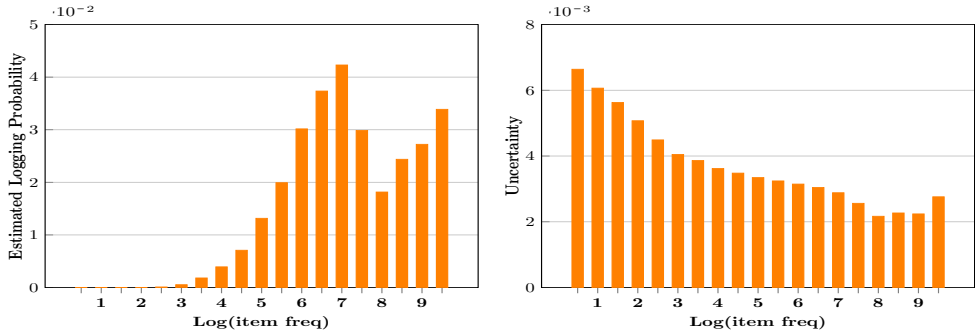
1 Introduction

In many real-world applications, including search engines [2], online advertisements [35], recommender systems [8, 22], only logged data is available for subsequent policy learning. For example, in recommender systems, various complex recommendation policies are optimized over logged user interactions (e.g., clicks or stay time) with items recommended by previous recommendation policies (referred to as the *logging policy*) [51, 14]. However, such logged data is often known to be biased, since the feedback on items where the logging policy did not take is unknown. This inevitably distorts the evaluation and optimization of a new policy when it differs from the logging policy.

Off-policy learning [41, 27] thus emerges as a preferred way to learn an improved policy only from the logged data, by addressing the mismatch between the learning and logging policies. One of the most commonly used off-policy learning methods is the Inverse Propensity Scoring (IPS) [8, 25], which assigns per-sample importance weight (i.e., propensity score) to the training objective on the logged data, so as to get an unbiased optimization objective in expectation. The importance weight in IPS is the probability ratio of taking an action between the learning and logging policies.

Unfortunately, the ground-truth logging policy is oftentimes unavailable to the learner in practice, due to reasons like legacy issues, i.e., it was not recorded in the data. Additionally, in specific situations like the healthcare domain [28] or two-stage recommender systems [8], access to the ground-truth logging policy is not feasible. One common treatment by many previous studies [35, 22, 8, 24] is to first estimate the logging policy using a supervised learning method (e.g., logistic regression,

37th Conference on Neural Information Processing Systems (NeurIPS 2023).



(a) Estimated Logging Probability (b) Uncertainty of Estimation

Figure 1: Estimated logging policy and its uncertainty under different item frequency on KuaiRec.

neural networks, etc.), and then employ the estimated logging policy for off-policy learning. In this work, we first show that such an approximation results in a biased estimator which is sensitive to data with small estimated logging probabilities. Worse still, small estimated logging probabilities usually suggest there are limited related samples in the logged data, whose estimations can have high uncertainties, i.e., being wrong with a high probability. Figure 1 shows a piece of empirical evidence from a large-scale recommendation benchmark KuaiRec dataset [12], where items with lower frequencies in the logged dataset have lower estimated logging probabilities (via a neural network estimator) and higher uncertainties at the same time. The high bias and variance caused by these samples can greatly hinder the performance of subsequent off-policy learning. We defer detailed discussions of this result in Section 2.

In this work, we explicitly take the uncertainty of the estimated logging policy into consideration and design an Uncertainty-aware Inverse Propensity Score estimator (UIPS) for off-policy learning. UIPS reweighs the propensity score of each logged sample to control its impact on policy optimization, and learns an improved policy by alternating between: (1) Find the optimal weight that makes the estimator as accurate as possible, based on the uncertainty of the estimated logging policy; (2) Improve the policy by optimizing the resulting objective function. The optimal weight for each sample is obtained by minimizing the upper bound of the mean squared error (MSE) to the ground-truth policy evaluation, with a closed-form solution. Furthermore, UIPS ensures that off-policy learning converges to a stationary point where the true policy gradient is zero; while convergence may not be guaranteed when directly using the estimated logging policy. Extensive experiments on a synthetic and three real-world recommendation datasets against a rich set of state-of-the-art baselines demonstrate the power of UIPS. All data and code can be found in `https://github.com/Xiaoyinggit/UIPS.git`.

2 Preliminary: off-policy learning

We focus on the standard contextual bandit setup to explain the key concepts in UIPS. Following the convention [16, 29, 36], let $\boldsymbol{x} \in \mathcal{X} \subseteq R^d$ be a d-dimensional context vector drawn from an unknown distribution $p(\boldsymbol{x})$. Each context is associated with a finite set of actions denoted by $\mathcal{A}$, where $|\mathcal{A}| > 1$. Let $\pi : \mathcal{A} \times \mathcal{X} \to [0, 1]$ denote a stochastic policy, such that $\pi(a|\boldsymbol{x})$ is the probability of selecting action a under context $\boldsymbol{x}$ and $\sum_{a \in \mathcal{A}} \pi(a|\boldsymbol{x}) = 1$. Under a given context $\boldsymbol{x}$, the reward $r_{\boldsymbol{x},a}$ is only observed when action a is chosen, i.e., bandit feedback. Without loss of generality, we assume $r_{\boldsymbol{x},a} \in [0, 1]$. Let $V(\pi)$ denote the expected reward of the policy π:

$$V(\pi) = \mathbb{E}_{\boldsymbol{x} \sim p(\boldsymbol{x}), a \sim \pi(a|\boldsymbol{x})}[r_{\boldsymbol{x},a}]. \tag{1}$$

We look for a policy $\pi(a|\boldsymbol{x})$ to maximize $V(\pi)$. In the rest, we denote $\mathbb{E}_{\boldsymbol{x} \sim p(\boldsymbol{x}), a \sim \pi(a|\boldsymbol{x})}[\,]$ as $\mathbb{E}_{\pi}[\,]$.

In off-policy learning, one can only access a set of logged feedback data $D := \{(\boldsymbol{x}_n, a_n, r_{\boldsymbol{x}_n,a_n})|n \in [N]\}$. Given $\boldsymbol{x}_n$, the action a_n was generated by a stochastic logging policy β, i.e., $a_n \sim \beta(a|\boldsymbol{x}_n)$, which is usually different from the learning policy $\pi(a|\boldsymbol{x})$ [24, 40, 8]. The actions $\{a_1, \ldots, a_N\}$ and their corresponding rewards $\{r_{\boldsymbol{x}_1,a_1}, \ldots, r_{\boldsymbol{x}_N,a_N}\}$ are generated independently given β. The main challenge is then to address the distributional discrepancy between $\beta(a|\boldsymbol{x})$ and $\pi(a|\boldsymbol{x})$, when optimizing $\pi(a|\boldsymbol{x})$ to maximize $V(\pi)$ with access only to the logged dataset D.

One of the most widely used methods to address the distribution shift between $\pi(a|\boldsymbol{x})$ and $\beta\ (a|\boldsymbol{x})$ is the Inverse Propensity Scoring (IPS) [8, 25]. One can easily get that:

$$V(\pi) = \mathsf{E}_{\beta}\ \left[\frac{\pi(a|\boldsymbol{x})}{\beta\ (a|\boldsymbol{x})} r_{\boldsymbol{x},a}\right],$$

yielding the following empirical estimator of $V(\pi)$:

$$\hat{V}_{\mathrm{IPS}}(\pi) = \frac{1}{N}\sum_{n=1}^{N} \frac{\pi(a_n|\boldsymbol{x}_n)}{\beta\ (a_n|\boldsymbol{x}_n)} r_{\boldsymbol{x}_n,a_n}, \tag{2}$$

where $\pi(a_n|\boldsymbol{x}_n)/\beta\ (a_n|\boldsymbol{x}_n)$ is referred to as the propensity score. Various algorithms can be readily used for policy optimization under $\hat{V}_{\mathrm{IPS}}(\pi)$, including value-based methods [33] and policy-based methods [19, 31, 42]. In this work, we adopt a well-known policy gradient algorithm, REINFORCE [42]. Assume the policy $\pi(a|\boldsymbol{x})$ is parameterized by $\boldsymbol{\vartheta}$, via the "log-trick", the gradient of $\hat{V}_{\mathrm{IPS}}(\pi_{\boldsymbol{\vartheta}})$ with respect to $\boldsymbol{\vartheta}$ can be readily derived as,

$$\nabla_{\boldsymbol{\vartheta}} \hat{V}_{\mathrm{IPS}}(\pi_{\boldsymbol{\vartheta}}) = \frac{1}{N}\sum_{n=1}^{N} \frac{\pi_{\boldsymbol{\vartheta}}(a_n|\boldsymbol{x}_n)}{\beta\ (a_n|\boldsymbol{x}_n)} r_{\boldsymbol{x}_n,a_n} \nabla_{\boldsymbol{\vartheta}} \log(\pi_{\boldsymbol{\vartheta}}(a_n|\boldsymbol{x}_n)).$$

Approximation with an unknown logging policy. In many real-world applications, the ground-truth logging probabilities, i.e., $\beta\ (a|\boldsymbol{x})$ of each observation $(\boldsymbol{x}, a)$ in D, are unknown. As a typical walk-around, previous work employs supervised learning methods such as logistic regression [30] and nerural networks [8] to estimate the logging policy, and replaces $\beta\ (a|\boldsymbol{x})$ with its estimated value $\hat{\beta}(a|\boldsymbol{x})$ to get the following BIPS estimator for policy learning:

$$\hat{V}_{\mathrm{BIPS}}(\pi_{\boldsymbol{\vartheta}}) = \frac{1}{N}\sum_{n=1}^{N} \frac{\pi_{\boldsymbol{\vartheta}}(a_n|\boldsymbol{x}_n)}{\hat{\beta}(a_n|\boldsymbol{x}_n)} r_{\boldsymbol{x}_n,a_n}. \tag{3}$$

However, as shown in the following proposition, inaccurate $\hat{\beta}(a|\boldsymbol{x})$ leads to high bias and variance in BIPS. Worse still, smaller and inaccurate $\hat{\beta}(a|\boldsymbol{x})$ further enlarges this bias and variance.

Proposition 2.1. *The bias and variance of $\hat{V}_{\mathrm{BIPS}}(\pi_{\boldsymbol{\vartheta}})$ can be derived as follows:*

$$\mathrm{Bias}\left(\hat{\mathrm{V}}_{\mathrm{BIPS}}(\pi_{\boldsymbol{\vartheta}})\right) = \mathsf{E}_D\left[\hat{V}_{\mathrm{BIPS}}(\pi_{\boldsymbol{\vartheta}}) \quad V(\pi_{\boldsymbol{\vartheta}})\right] = \mathsf{E}_{\pi_{\boldsymbol{\vartheta}}}\left[r_{\boldsymbol{x},a}\left(\frac{\beta\ (a|\boldsymbol{x})}{\hat{\beta}(a|\boldsymbol{x})} \quad 1\right)\right]$$

$$N\ \ \mathrm{Var}_D\left(\hat{V}_{\mathrm{BIPS}}(\pi_{\boldsymbol{\vartheta}})\right) = \mathrm{Var}_{\pi_{\boldsymbol{\vartheta}}}\left(\frac{\beta\ (a|\boldsymbol{x})}{\hat{\beta}(a|\boldsymbol{x})} r_{\boldsymbol{x},a}\right) + \mathsf{E}_{\pi_{\boldsymbol{\vartheta}}}\left[\left(\frac{\pi_{\boldsymbol{\vartheta}}(a|\boldsymbol{x})}{\beta\ (a|\boldsymbol{x})} \quad 1\right)\frac{\beta\ (a|\boldsymbol{x})^2}{\hat{\beta}(a|\boldsymbol{x})^2} r^2_{\boldsymbol{x},a}\right]$$

However, smaller $\hat{\beta}(a|\boldsymbol{x})$ usually implies less number of related training samples in the logged data, and thus $\hat{\beta}(a|\boldsymbol{x})$ can be inaccurate with a higher probability. To make it more explicit, let us revisit the empirical results shown in Figure 1. We followed the method introduced in [8] to estimate the logging policy on KuaiRec dataset [12] and plotted the estimated $\hat{\beta}(a|\boldsymbol{x})$ and its corresponding uncertainties on items of different observation frequencies in the logged dataset. We adopted the method in [45] to measure the confidence interval of $\hat{\beta}(a|\boldsymbol{x})$ on each instance. A wider confidence interval, i.e., higher uncertainty in estimation, implies that with a high probability the true value may be further away from the empirical mean estimate. We can observe in Figure 1 that as item frequency decreases, the estimated logging probability also decreases, but the estimation uncertainty increases. This implies that a smaller $\hat{\beta}(a|\boldsymbol{x})$ is usually 1) more inaccurate and 2) associated with a higher uncertainty.

As a result, with high bias and variance caused by inaccurate $\hat{\beta}(a|\boldsymbol{x})$, it is erroneous to learn $\pi_{\boldsymbol{\vartheta}}(a|\boldsymbol{x})$ by simply optimizing $\hat{V}_{\mathrm{BIPS}}(\pi_{\boldsymbol{\vartheta}})$. Furthermore, this approach may also hinder the convergence of off-policy learning, as discussed later in Section 3.2.

3 Uncertainty-aware off-policy learning

Our idea is to consider the uncertainty of the estimated logging policy by incorporating per-sample weight $\phi_{\boldsymbol{x},a}$, and perform policy learning by optimizing the following empirical estimator:

$$\hat{V}_{\mathrm{UIPS}}(\pi_{\boldsymbol{\vartheta}}) = \frac{1}{N}\sum_{n=1}^{N} \frac{\pi_{\boldsymbol{\vartheta}}(a_n|\boldsymbol{x}_n)}{\hat{\beta}(a_n|\boldsymbol{x}_n)}\ \ \phi_{\boldsymbol{x}_n,a_n}\ \ r_{\boldsymbol{x}_n,a_n}. \tag{4}$$

Intuitively, one should assign lower weights to samples whose $\hat{\beta}(a|\boldsymbol{x})$ is small and far away from the ground-truth $\beta\ (a|\boldsymbol{x})$. We then divide off-policy optimization into two iterative steps:

- **Deriving the optimal instance weight:** Find the optimal $\phi_{\boldsymbol{x},a}$ to make $\hat{V}_{\mathrm{UIPS}}(\pi_{\boldsymbol{\vartheta}})$ approach its ground-truth $V(\pi)$ as closely as possible, so as to facilitate policy learning. The derived optimal weight is denoted as $\phi^{*}_{\boldsymbol{x},a}$ (see Theorem 3.2).
- **Policy improvement**: Update the policy $\pi_{\boldsymbol{\vartheta}}(a|\boldsymbol{x})$ using the following gradient:

$$\nabla_{\vartheta}\hat{V}_{\mathrm{UIPS}}(\pi_{\boldsymbol{\vartheta}}) = \frac{1}{N}\sum_{n=1}^{N}\frac{\pi_{\boldsymbol{\vartheta}}(a_n|\boldsymbol{x}_n)}{\hat{\beta}(a_n|\boldsymbol{x}_n)}\ \ \phi_{\boldsymbol{x}_n,a_n}\ \ r_{\boldsymbol{x}_n,a_n}\nabla_{\boldsymbol{\vartheta}}\log(\pi_{\boldsymbol{\vartheta}}(a_n|\boldsymbol{x}_n)) \tag{5}$$

The whole algorithm framework and its computational cost, as well as important notations are summarized in Appendix 7.1.

3.1 Derive the optimal uncertainty-aware instance weight

We expect to find the optimal weight $\phi_{x,a}$ to make the empirical estimator $\hat{V}_{\mathrm{UIPS}}(\pi_{\boldsymbol{\vartheta}})$ as accurate as possible, taking into account the uncertainty in estimated logging probabilities. Intuitively, a high accuracy of the estimator is crucial for determining the correct direction of policy learning. We follow previous work [36, 29] and measure the mean squared error (MSE) of $\hat{V}_{\mathrm{UIPS}}(\pi_{\boldsymbol{\vartheta}})$ to the ground-truth policy value $V(\pi_{\boldsymbol{\vartheta}})$, which captures both the bias and variance of an estimator. A lower MSE indicates a more accurate estimator.

In UIPS, instead of directly minimizing the MSE, which is intractable, we find $\phi_{\boldsymbol{x},a}$ to minimize the upper bound of MSE. As we show later, the optimal $\phi^{*}_{\boldsymbol{x},a}$ has a closed-form solution which relates to both the value of $\pi_{\boldsymbol{\vartheta}}(a|\boldsymbol{x})/\hat{\beta}(a|\boldsymbol{x})$ and the estimation uncertainty of $\hat{\beta}(a|\boldsymbol{x})$.

Theorem 3.1. *The mean squared error (*MSE*) between $\hat{V}_{\mathrm{UIPS}}(\pi_{\boldsymbol{\vartheta}})$ and ground-truth estimator $V(\pi_{\boldsymbol{\vartheta}})$ is upper bounded as follows:*

$$\mathrm{MSE}\left(\hat{V}_{\mathrm{UIPS}}(\pi_{\boldsymbol{\vartheta}})\right) = \mathbb{E}_{D}\left[\left(\hat{V}_{\mathrm{UIPS}}(\pi_{\boldsymbol{\vartheta}}) \quad V(\pi_{\boldsymbol{\vartheta}})\right)^{2}\right] = \mathrm{Bias}\left(\hat{V}_{\mathrm{UIPS}}(\pi_{\boldsymbol{\vartheta}})\right)^{2} + \mathrm{Var}\left(\hat{V}_{\mathrm{UIPS}}(\pi_{\boldsymbol{\vartheta}})\right)$$

$$\mathbb{E}_{\pi_{\boldsymbol{\vartheta}}}\left[r^{2}_{\boldsymbol{x},a}\frac{\pi_{\boldsymbol{\vartheta}}(a|\boldsymbol{x})}{\beta\ (a|\boldsymbol{x})}\right]\ \ \mathbb{E}_{\beta}\ \ \left[\left(\frac{\beta\ (a|\boldsymbol{x})}{\hat{\beta}(a|\boldsymbol{x})}\phi_{\boldsymbol{x},a} \quad 1\right)^{2}\right] + \mathbb{E}_{\beta}\ \ \left[\frac{\pi_{\boldsymbol{\vartheta}}(a|\boldsymbol{x})^{2}}{\hat{\beta}(a|\boldsymbol{x})^{2}}\phi^{2}_{\boldsymbol{x},a}\right].$$

As the first expectation term $\mathbb{E}_{\pi_{\boldsymbol{\vartheta}}}\left[r^{2}_{\boldsymbol{x},a}\frac{\pi_{\boldsymbol{\vartheta}}(a|\boldsymbol{x})}{\beta\ (a|\boldsymbol{x})}\right]$ is a non-negative constant, we denote it as $\lambda \geq 0$ when searching for $\phi_{\boldsymbol{x},a}$. To minimize this upper bound of MSE, the optimal $\phi_{\boldsymbol{x},a}$ for each sample $(\boldsymbol{x}, a)$ should minimize the following,

$$\lambda\left(\frac{\beta\ (a|\boldsymbol{x})}{\hat{\beta}(a|\boldsymbol{x})}\phi_{\boldsymbol{x},a} \quad 1\right)^{2} + \frac{\pi_{\boldsymbol{\vartheta}}(a|\boldsymbol{x})^{2}}{\hat{\beta}(a|\boldsymbol{x})^{2}}\phi^{2}_{\boldsymbol{x},a}. \tag{6}$$

An interesting observation is that setting $\phi_{\boldsymbol{x},a} = \frac{\hat{\beta}(a|\boldsymbol{x})}{\beta\ (a|\boldsymbol{x})}$, i.e., turning $\frac{\pi(a|\boldsymbol{x})}{\hat{\beta}(a|\boldsymbol{x})}\phi_{\boldsymbol{x},a}$ into $\frac{\pi(a|\boldsymbol{x})}{\beta\ (a|\boldsymbol{x})}$ does not result in the optimal solution of Eq.(6). This is because such a setting only reduces bias (i.e., the first term of Eq.(6)), but fails to control the second term, which is related to the variance. Moreover, we cannot directly minimize Eq.(6) due to the unknown $\beta\ (a|\boldsymbol{x})$. But it is possible to obtain a confidence interval which contains $\beta\ (a|\boldsymbol{x})$ with a high probability, when $\hat{\beta}(a|\boldsymbol{x})$ is obtained via a specific estimator, e.g., (generalized) linear model or kernel methods.

Following previous work [23, 16, 22], we adopt the realizable assumption that $\beta\ (a|\boldsymbol{x})$ can be represented by a softmax function applied over a parametric function $f_{\boldsymbol{\theta}^{*}}(\boldsymbol{x}, a)$. Moreover, the universal approximation theorem [18] states that a parametric function with sufficient capacity, when combined with a softmax function, can approximate any distribution. Then we have:

$$\beta\ (a|\boldsymbol{x})\ /\ \exp(f_{\boldsymbol{\theta}^{*}}(\boldsymbol{x}, a)), \hat{\beta}(a|\boldsymbol{x})\ /\ \exp(f_{\boldsymbol{\theta}}(\boldsymbol{x}, a)), \tag{7}$$

where $f_{\boldsymbol{\theta}}(\boldsymbol{x}, a)$ is an estimate of $f_{\boldsymbol{\theta}^{*}}(\boldsymbol{x}, a)$. Following the conventional definition of confidence interval [20], we define γ and $U_{\boldsymbol{x},a}$ such that $|f_{\boldsymbol{\theta}^{*}}(\boldsymbol{x}, a) \quad f_{\boldsymbol{\theta}}(\boldsymbol{x}, a)| \quad \gamma U_{\boldsymbol{x},a}$ holds with probability

at least 1-δ, where γ is a function of δ (typically the smaller δ is, the larger γ is). Then $\gamma U_{\boldsymbol{x},a}$ measures the width of confidence interval of $f_{\boldsymbol{\theta}}(\boldsymbol{x}, a)$ against its ground-truth $f_{\boldsymbol{\theta}^*}(\boldsymbol{x}, a)$. As derived in Appendix 7.2, with probability at least 1-δ, we have $\beta\ (a|\boldsymbol{x}) \in \boldsymbol{B}_{\boldsymbol{x},a}$ and

$$\boldsymbol{B}_{\boldsymbol{x},a} = \left[\frac{\hat{Z}\exp(-\gamma U_{\boldsymbol{x},a})}{Z}\hat{\beta}(a|\boldsymbol{x}), \frac{\hat{Z}\exp(\gamma U_{\boldsymbol{x},a})}{Z}\hat{\beta}(a|\boldsymbol{x})\right],$$

where $Z = \sum_{a'} \exp(f_\theta\ (a'|\boldsymbol{x}))$ and $\hat{Z} = \sum_{a'} \exp(f_\theta(a'|\boldsymbol{x}))$.

As $\beta\ (a|\boldsymbol{x})$ can be any value in $\boldsymbol{B}_{\boldsymbol{x},a}$ with high probability, we aim to find the optimal $\phi_{\boldsymbol{x},a}$ that minimizes the worst case of Eq.(6), thereby ensuring that $\hat{V}_{\mathrm{UIPS}}(\pi_{\boldsymbol{\vartheta}})$ approaches its ground-truth $V(\pi_{\boldsymbol{\vartheta}})$ under the sense of MSE, even in the worst possible scenarios. This ensures the subsequent policy improvement direction will not be much worse with high probability. Thus, we formulate the following optimization problem:

$$\min_{\phi_{\boldsymbol{x},a}} \max_{\beta_{\boldsymbol{x},a} \in \boldsymbol{B}_{\boldsymbol{x},a}} \lambda \left(\frac{\beta_{\boldsymbol{x},a}}{\hat{\beta}(a|\boldsymbol{x})}\phi_{\boldsymbol{x},a} - 1\right)^2 + \frac{\pi_{\boldsymbol{\vartheta}}(a|\boldsymbol{x})^2}{\hat{\beta}(a|\boldsymbol{x})^2}\phi_{\boldsymbol{x},a}^2. \tag{8}$$

The following theorem derives a closed-form formula for the optimal solution of Eq.(8).

Theorem 3.2. *Let $\eta \in [\exp(-\gamma U_{\boldsymbol{x}}^{\max}), \exp(\gamma U_{\boldsymbol{x}}^{\max})]$, where $U_{\boldsymbol{x}}^{\max} = \max_a U_{\boldsymbol{x},a}$. The optimization problem in Eq.(8) has a closed-form solution:*

$$\phi_{\boldsymbol{x},a} = \min\left(\lambda / \left[\frac{\lambda}{\eta}\exp(-\gamma U_{\boldsymbol{x},a}) + \frac{\eta\pi_{\boldsymbol{\vartheta}}(a|\boldsymbol{x})^2}{\hat{\beta}(a|\boldsymbol{x})^2\exp(-\gamma U_{\boldsymbol{x},a})}\right], 2\eta / \left[\exp\big(\gamma U_{\boldsymbol{x},a}\big) + \exp(-\gamma U_{\boldsymbol{x},a})\right]\right).$$

The following corollary demonstrates the advantage of UIPS. The detailed proof of Theorem 3.2 and Corollary 3.3 can be found in Appendix 7.8.

Corollary 3.3. *With $\phi_{\boldsymbol{x},a}$ derived in Theorem 3.2, $\hat{V}_{\mathrm{UIPS}}(\pi_{\boldsymbol{\vartheta}})$ in Eq.(4) achieves a smaller upper bound of MSE than $\hat{V}_{\mathrm{BIPS}}(\pi_{\boldsymbol{\vartheta}})$ in Eq. (3).*

Insights about $\phi_{\boldsymbol{x},a}$. The detailed analysis of the effect of $\phi_{\boldsymbol{x},a}$ can be found in Lemma 7.1 in Appendix 7.8. In summary, we have the following key findings,

- For samples whose largest possible propensity score is under control: i.e., $\frac{\pi_{\boldsymbol{\vartheta}}(a|\boldsymbol{x})}{\min \boldsymbol{B}_{\boldsymbol{x},a}} < \overline{\lambda}$, higher uncertainty implies smaller values of $\pi/\hat{\beta}$. This suggests samples of this type with positive rewards are underestimated, and the extend of underestimation increases with the estimation uncertainty. UIPS thus chooses to increase $\phi_{\boldsymbol{x},a}$ with uncertainty, to emphasize these long-tail positive samples.
- Conversely, for samples with large propensity scores, UIPS decreases $\phi_{\boldsymbol{x},a}$ as the uncertainty increases, so as to prevent their distortion in policy learning.

Uncertainty estimation. Now we describe how to calculate $U_{\boldsymbol{x},a}$, i.e., the uncertainty of the estimated $\hat{\beta}(a|\boldsymbol{x})$. In this work, we choose to estimate $\beta\ (a|\boldsymbol{x})$ using a neural network, because 1) its representation learning capacity has been proved in numerous studies, and 2) various ways [11, 45] can be leveraged to perform the uncertainty estimation in a neural network. We adopt [45] due to its computational efficiency and theoretical soundness. Following the proof of Theorem 4.4 in [45], given the logged dataset D, we can get with a high probability that there exists γ such that:

$$|f_{\boldsymbol{\theta}}(\boldsymbol{x}_n, a_n) - f_{\boldsymbol{\theta}^*}(\boldsymbol{x}_n, a_n))| \le \gamma\sqrt{\boldsymbol{g}(\boldsymbol{x}_n, a_n)^\top \boldsymbol{M}_D^{-1}\boldsymbol{g}(\boldsymbol{x}_n, a_n)}$$

where $\boldsymbol{g}(\boldsymbol{x}_n, a_n)$ is the gradient of $f_{\boldsymbol{\theta}}(\boldsymbol{x}_n, a_n)$ with respect to the neural network's last layer's parameter $\boldsymbol{\theta}_w \in \boldsymbol{\theta}$, i.e., $\boldsymbol{g}(\boldsymbol{x}_n, a_n) = \nabla_{\boldsymbol{\theta}_w} f_{\boldsymbol{\theta}}(\boldsymbol{x}_n, a_n)$. And $M_D = \sum_{n=1}^N \boldsymbol{g}(\boldsymbol{x}_n, a_n)\boldsymbol{g}(\boldsymbol{x}_n, a_n)^\top$, implying $U_{\boldsymbol{x}_n,a_n} = \sqrt{\boldsymbol{g}(\boldsymbol{x}_n, a_n)^\top \boldsymbol{M}_D^{-1}\boldsymbol{g}(\boldsymbol{x}_n, a_n)}$.

3.2 Convergence of policy learning under UIPS

The following theorem provides the convergence result for UIPS, which converges to a stationary point of the expected reward function. The proof is provided in Appendix 7.9.

Theorem 3.4. *Denote $G_{\max}$ and Φ as the maximum value of $\|\frac{\partial \pi_\vartheta(a|\boldsymbol{x})}{\partial \vartheta}\|$ and $\mathbb{E}_\beta\left[\frac{\pi_\vartheta^2(a|\boldsymbol{x})}{\hat{\beta}^2(a|\boldsymbol{x})}(\phi_{\boldsymbol{x},a})^2\right]$ respectively, i.e., $\|\frac{\partial \pi_\vartheta(a|\boldsymbol{x})}{\partial \vartheta}\| \quad G_{\max}$ and $\mathbb{E}_\beta\left[\frac{\pi_\vartheta^2(a|\boldsymbol{x})}{\hat{\beta}^2(a|\boldsymbol{x})}(\phi_{\boldsymbol{x},a})^2\right] \quad \Phi$. And denote $V_{\max}$ as the finite maximum expected reward that can be achieved, and $\varphi_{\max} = \max_{\boldsymbol{x},a}\left\{\left|\frac{\beta(a|\boldsymbol{x})}{\hat{\beta}(a|\boldsymbol{x})}\phi_{\boldsymbol{x},a} \quad 1\right|\right\}$. Assume that the expected reward of π_ϑ, i.e., $V(\pi_\vartheta)$, is a differentiable and L-smooth function w.r.t ϑ. Denote the policy parameters obtained by Eq.(5) at iteration $k \in [K]$ as ϑ_k, then $\varphi_{\max} \in (0,1)$ and*

$$\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}[\|\nabla V(\pi_{\vartheta_k})\|]^2 \quad \frac{2LV_{\max}}{K(1 \quad \varphi_{\max})} + \left(L + \frac{2V_{\max}}{(1 \quad \varphi_{\max})}\right)\frac{G_{\max}\sqrt{\Phi}}{\sqrt{K}},$$

where $\nabla V(\pi_\vartheta)$ is the true policy gradient under ground-truth logging probability, i.e., $\nabla V(\pi_\vartheta) = E_\beta[\frac{\pi_\vartheta(a|\boldsymbol{x})}{\beta(a|\boldsymbol{x})}r_{\boldsymbol{x},a}\nabla_\vartheta \log(\pi_\vartheta(a|\boldsymbol{x}))]$.

Theorem 3.4 shows that, as $K \to \infty$ and with $1/(1 \quad \varphi_{\max})$ and Φ being controlled, UIPS leads policy update to converge to a stationary point where the true policy gradient $\nabla V(\pi_{\vartheta_k})$ is zero. And fortunately, UIPS is effective in controlling both $1/(1 \quad \varphi_{\max})$ and Φ. Specifically, we denote $\varphi_{\boldsymbol{x},a} = \left|\frac{\beta(a|\boldsymbol{x})}{\hat{\beta}(a|\boldsymbol{x})}\phi_{\boldsymbol{x},a} \quad 1\right|$ and $\Phi_{x,a} = \frac{\pi_\vartheta^2(a|\boldsymbol{x})}{\hat{\beta}^2(a|\boldsymbol{x})}(\phi_{\boldsymbol{x},a})^2$. It is clear to note that $\lambda\varphi_{x,a}^2 + \Phi_{x,a}$ corresponds to the objective in Eq.(6) for deriving $\phi_{\boldsymbol{x},a}$ for each sample $(\boldsymbol{x}, a)$. In other words, UIPS selects $\{\phi_{\boldsymbol{x},a}\}$ to minimize $\varphi_{\max} = \max\{\varphi_{x,a}\}$ and $\Phi = \mathbb{E}_\beta[\Phi_{\boldsymbol{x},a}]$, which directly accelerate the policy converge to a stationary point with the true policy gradient being zero.

In the case of BIPS in Eq.(3), we have $\phi_{\boldsymbol{x},a} \quad 1$. Although Φ may be large due to small logging probabilities, the more concerning issue is that the requirement $\varphi_{\max} \in (0,1)$ is no longer satisfied when $\beta(a|\boldsymbol{x}) \quad 2\hat{\beta}(a|\boldsymbol{x})$, which may happen with a non-negligible probability. Hence, the convergence of policy learning under $\hat{V}_{\mathrm{BIPS}}$ is no better than that under UIPS.

4 Empirical evaluations

We evaluate UIPS on both synthetic data and three real-world datasets with unbiased collection. We compare UIPS with the following baselines, which can be grouped into five categories:

- **Cross-Entropy (CE)**: A supervised learning method with the cross-entropy loss over its softmax output. No off-policy correction is performed in this method.
- **BIPS-Cap** [8]: The off-policy learning solution under the BIPS estimator in Eq.(3). The estimated propensity scores are further suppressed to control variance, i.e., taking $\min\left(c, \frac{\pi_\vartheta(a|\boldsymbol{x})}{\hat{\beta}(a|\boldsymbol{x})}\right)$ as the propensity score. Setting c to a small value can reduce variance, but introduces bias.
- **MinVar** & **stableVar** [46], **Shrinkage** [36]: This line of work improves off-policy evaluation by reweighing each sample. For example, MinVar and stableVar reweigh each sample by $\frac{h_{\boldsymbol{x},a}}{\sum_{a'} h_{\boldsymbol{x},a'}}$ with $h_{\boldsymbol{x},a} = \frac{\hat{\beta}(a|\boldsymbol{x})}{\pi_\vartheta(a|\boldsymbol{x})^2}$ and $h_{\boldsymbol{x},a} = \frac{\sqrt{\Phi}}{\hat{\beta}(a|\boldsymbol{x})\pi_\vartheta(a|\boldsymbol{x})}$ respectively, since they find that $\pi_\vartheta(a|\boldsymbol{x})^2/\hat{\beta}(a|\boldsymbol{x})$ is directly related to policy evaluation variance. Su et al. [36] propose to shrink the propensity score by $\lambda/(\lambda + \frac{\pi_\vartheta(a|\boldsymbol{x})^2}{\hat{\beta}(a|\boldsymbol{x})^2})$, which is a special case of our UIPS with $U_{\boldsymbol{x},a} = 0$ and $\eta = 1$. All these methods simply treat $\hat{\beta}(a|\boldsymbol{x})$ as $\beta(a|\boldsymbol{x})$, and none of them consider the uncertainty of $\hat{\beta}(a|\boldsymbol{x})$.
- **SNIPS** [39], **BanditNet** [16], **POEM** [38], **POXM** [23], **Adaptive** [22]: This line of work aims for more stable and accurate policy learning. For example, SNIPS normalizes the estimator by the sum of propensity scores in each batch. BanditNet extends SNIPS and leverages an additional Lagrangian term to normalize the estimator by an approximated sum of propensity scores of all samples. POEM jointly optimizes the estimator and its variance. POXM controls estimation variance by pruning samples with small logging probabilities. Adaptive proposes a new formulation to utilize negative samples.
- **ApproxKNN** [5] and **IPS-C-TS**: The line of work improves off-policy learning by applying calibration to estimated logging probabilities. ApproxKNN utilizes the K-Nearest Neighbor algorithm for calibration, which exhibits the lowest calibration error in [5]. IPS-C-TS, on the other hand, employs temperature scaling, a widely recognized and effective calibration method for probability distribution [13].

Table 1: Experiment results on synthetic datasets. The best and second best results are highlighted with **bold** and underline respectively. The p-value under the t-test between UIPS and the best baseline on each dataset is also provided.

	$\tau = 0.5$			$\tau = 1$			$\tau = 2$		
Algorithm	P@5	R@5	NDCG@5	P@5	R@5	NDCG@5	P@5	R@5	NDCG@5
IPS-GT	0.5589 $_{1e^{-3}}$	0.1582 $_{6e^{-4}}$	0.6093 $_{1e^{-3}}$	0.5526 $_{2e^{-3}}$	0.1565 $_{6e^{-4}}$	0.6007 $_{1e^{-3}}$	0.5531 $_{2e^{-3}}$	0.1557 $_{7e^{-4}}$	0.6037 $_{1e^{-3}}$
CE	0.5553 $_{6e^{-4}}$	0.1573 $_{2e^{-4}}$	0.6037 $_{5e^{-4}}$	0.5510 $_{6e^{-4}}$	0.1561 $_{2e^{-4}}$	0.5995 $_{4e^{-4}}$	0.5386 $_{2e^{-3}}$	0.1524 $_{7e^{-4}}$	0.5874 $_{2e^{-3}}$
BIPS-Cap	0.5515 $_{2e^{-3}}$	0.1553 $_{8e^{-4}}$	0.6031 $_{2e^{-3}}$	0.5526 $_{2e^{-3}}$	0.1561 $_{6e^{-4}}$	0.6016 $_{1e^{-3}}$	0.5409 $_{3e^{-3}}$	0.1529 $_{9e^{-4}}$	0.5901 $_{2e^{-3}}$
MinVar	0.5340 $_{2e^{-3}}$	0.1509 $_{6e^{-4}}$	0.5857 $_{2e^{-3}}$	0.5282 $_{2e^{-3}}$	0.1491 $_{7e^{-4}}$	0.5791 $_{2e^{-3}}$	0.5036 $_{4e^{-3}}$	0.1415 $_{1e^{-3}}$	0.5543 $_{3e^{-3}}$
stableVar	0.4577 $_{5e^{-3}}$	0.1310 $_{1e^{-3}}$	0.5111 $_{2e^{-3}}$	0.5373 $_{3e^{-3}}$	0.1523 $_{9e^{-4}}$	0.5866 $_{3e^{-3}}$	0.5279 $_{3e^{-3}}$	0.1492 $_{8e^{-4}}$	0.5781 $_{3e^{-3}}$
Shrinkage	0.5526 $_{2e^{-3}}$	0.1562 $_{7e^{-4}}$	0.6024 $_{1e^{-3}}$	0.5499 $_{4e^{-3}}$	0.1545 $_{1e^{-3}}$	0.6040 $_{3e^{-3}}$	0.5347 $_{2e^{-3}}$	0.1513 $_{6e^{-4}}$	0.5824 $_{2e^{-3}}$
SNIPS	0.2616 $_{6e^{-2}}$	0.0749 $_{2e^{-2}}$	0.3150 $_{7e^{-2}}$	0.3538 $_{5e^{-2}}$	0.0987 $_{1e^{-2}}$	0.4144 $_{6e^{-2}}$	0.4379 $_{3e^{-2}}$	0.1226 $_{9e^{-3}}$	0.5177 $_{3e^{-2}}$
BanditNet	0.4011 $_{3e^{-2}}$	0.1131 $_{8e^{-3}}$	0.4830 $_{2e^{-2}}$	0.3894 $_{4e^{-2}}$	0.1095 $_{1e^{-2}}$	0.4741 $_{3e^{-2}}$	0.4122 $_{3e^{-2}}$	0.1153 $_{8e^{-3}}$	0.4934 $_{3e^{-2}}$
POEM	0.5480 $_{2e^{-3}}$	0.1539 $_{8e^{-4}}$	0.6008 $_{2e^{-3}}$	0.5502 $_{2e^{-3}}$	0.1551 $_{6e^{-4}}$	0.6000 $_{2e^{-3}}$	0.5399 $_{2e^{-3}}$	0.1526 $_{8e^{-4}}$	0.5893 $_{2e^{-3}}$
POXM	0.4006 $_{3e^{-2}}$	0.1130 $_{8e^{-3}}$	0.4828 $_{2e^{-2}}$	0.3616 $_{4e^{-2}}$	0.1019 $_{1e^{-2}}$	0.4522 $_{4e^{-2}}$	0.3816 $_{4e^{-2}}$	0.1069 $_{1e^{-2}}$	0.4680 $_{4e^{-2}}$
Adaptive	0.3831 $_{2e^{-2}}$	0.1050 $_{4e^{-3}}$	0.4382 $_{2e^{-2}}$	0.4734 $_{4e^{-3}}$	0.1325 $_{1e^{-3}}$	0.5326 $_{3e^{-3}}$	0.3936 $_{1e^{-2}}$	0.1097 $_{4e^{-3}}$	0.4368 $_{2e^{-2}}$
ApproxKNN	0.5576 $_{1e^{-3}}$	0.1580 $_{4e^{-4}}$	0.6059 $_{2e^{-3}}$	0.5527 $_{9e^{-4}}$	0.1567 $_{1e^{-4}}$	0.6010 $_{1e^{-3}}$	0.5409 $_{2e^{-3}}$	0.1532 $_{6e^{-4}}$	0.5890 $_{1e^{-3}}$
IPS-C-TS	0.5565 $_{1e^{-3}}$	0.1577 $_{3e^{-4}}$	0.6048 $_{7e^{-4}}$	0.5517 $_{6e^{-4}}$	0.1563 $_{2e^{-4}}$	0.6002 $_{6e^{-4}}$	0.5393 $_{1e^{-3}}$	0.1526 $_{5e^{-4}}$	0.5879 $_{1e^{-3}}$
UIPS-P	0.4019 $_{3e^{-2}}$	0.1131 $_{1e^{-2}}$	0.4831 $_{3e^{-2}}$	0.3904 $_{4e^{-2}}$	0.1096 $_{1e^{-2}}$	0.4749 $_{3e^{-2}}$	0.4109 $_{3e^{-2}}$	0.1149 $_{1e^{-2}}$	0.4922 $_{3e^{-2}}$
UIPS-O	0.4135 $_{4e^{-2}}$	0.1167 $_{1e^{-2}}$	0.4954 $_{4e^{-2}}$	0.3896 $_{4e^{-2}}$	0.1096 $_{1e^{-2}}$	0.4739 $_{3e^{-2}}$	0.4519 $_{3e^{-2}}$	0.1268 $_{8e^{-3}}$	0.5296 $_{2e^{-2}}$
UIPS	**0.5608** $_{2e^{-3}}$	**0.1589** $_{8e^{-4}}$	**0.6113** $_{3e^{-3}}$	**0.5572** $_{2e^{-3}}$	**0.1571** $_{8e^{-4}}$	**0.6074** $_{2e^{-3}}$	**0.5432** $_{3e^{-3}}$	**0.1534** $_{8e^{-4}}$	**0.5946** $_{2e^{-3}}$
p-value	$3e^{-3}$	$1e^{-2}$	$4e^{-5}$	$2e^{-5}$	$2e^{-1}$	$4e^{-10}$	$1e^{-1}$	$5e^{-1}$	$4e^{-2}$

Table 2: Performance under different uncertainties.

	Actions on Samples with High Uncertainty			**Actions on Samples with Low Uncertainty**		
Algorithm	P@5(RI)	R@5(RI)	NDCG@5(RI)	P@5(RI)	R@5(RI)	NDCG@5(RI)
CE	0.5190	0.1231	0.5526	0.5913	0.1915	0.6549
BIPS-Cap	0.5117 (-1.41%)	0.1202 (-2.33%)	0.5488 (-0.68%)	0.5913 (+0.00%)	0.1903 (-0.64%)	0.6574 (+0.39%)
Shrinkage	0.5158 (-0.62%)	0.1217 (-1.11%)	0.5505 (-0.37%)	0.5892 (-0.35%)	0.1905 (-0.55%)	0.6546 (-0.05%)
UIPS	0.5222 (+0.61%)	0.1237 (+0.50%)	0.5568 (+0.77%)	0.5994 (+1.38%)	0.1940 (+1.28%)	0.6658 (+1.66%)

- **UIPS-P** and **UIPS-O**: These are two variants of our UIPS with different ways of leveraging uncertainties. UIPS-P directly penalizes samples whose estimated logging probabilities are of high uncertainty, i.e., taking $\phi_{\boldsymbol{x},a} = 1.0/\exp(\gamma U_{\boldsymbol{x},a})$, which follows previous work on offline reinforcement learning [43, 4]. UIPS-O adversarially uses the worst propensity score for policy learning, i.e., $\phi_{\boldsymbol{x},a} = 1.0/\exp(\ \ \gamma U_{\boldsymbol{x},a})$.

4.1 Synthetic data

Data generation. Following previous work [24, 23], we generate a synthetic dataset by a supervise-to-bandit conversion on Wiki10-31K dataset [7], which is an extreme multi-label classification dataset. The Wiki10-31K dataset contains approximately 20K samples. Each sample is associated with a feature vector $\tilde{\boldsymbol{x}}$ of 101,938 dimensions and a label vector $\boldsymbol{y}_{\tilde{\boldsymbol{x}}}$ of 31K classes with more than one positive class. Let $\boldsymbol{y}_{\tilde{\boldsymbol{x}},a}$ denote the label of class a under $\tilde{\boldsymbol{x}}$, and we take each class as an action. The huge action space creates great challenges in off-policy learning, e.g., sparse observations, and therefore better evaluates different methods.

We split the dataset into train, validation and test sets with size 11K:3K:6K. The test set is from the official split. Since the original feature vector $\tilde{\boldsymbol{x}}$ is too sparse, for ease of learning, we first embedded it to dimension d by $\boldsymbol{x} = \boldsymbol{W}\tilde{\boldsymbol{x}}$, and synthesized the ground-truth logging policy $\beta\ (a|\boldsymbol{x})$ by:

$$\beta\ (a|\boldsymbol{x})\ \propto\ \exp\left(\boldsymbol{x}^\top \boldsymbol{\theta}_a/\tau\right), \tag{9}$$

where $\boldsymbol{W}$ and $\{\boldsymbol{\theta}_a\}$ are pre-trained parameters by applying a logistic regression model on the train set, τ is a positive hyper-parameter that controls the skewness of logging distribution. A small value of τ leads to a near-deterministic logging policy, while a larger τ makes it flatter. More implementation details can be found in Appendix 7.3.

Evaluation metrics. To evaluate the learned policy $\pi_{\boldsymbol{\vartheta}}(a|\boldsymbol{x})$, we calculate Precision@K (P@K), Recall@K (R@K) and NDCG@K following previous work [23, 24]. Higher P@K, R@K and NDCG@K imply a better policy.

Effectiveness of policy learning. Table 1 shows the average performance and standard deviations of all algorithms under 10 random seeds on three synthetic datasets generated under different τ. As the ground-truth logging policy is accessible on the synthetic datasets, we included a new baseline IPS-GT, which uses the IPS estimator with the ground-truth logging probabilities. We calculated p-value under t-test between UIPS and the best baseline to investigate the significance of improvement.

First, we can observe that UIPS achieved similar and even better performance than IPS-GT when $\tau = 0.5$ and $\tau = 1$, but performed worse than IPS-GT when $\tau = 2$. Despite using ground-truth

logging probabilities, IPS-GT still suffered from high variance caused by samples with small logging probabilities, which is the main cause of its worse performance when $\tau = 0.5$ and $\tau = 1$. In contrast, UIPS effectively controlled the negative impact of these high-variance samples, resulting in a better bias-variance trade-off.

With an increasing τ, suggesting a decrease in the probability of selecting positive actions, most algorithms experienced a drop in performance. However, UIPS consistently outperformed all other algorithms across all three datasets and metrics. Interestingly, as τ decreases, the performance improvement of UIPS became even more pronounced, despite SNIPS, BanditNet, and POXM being designed to handle small logging probabilities of positive actions.

ApproxKNN and IPS-C-TS generally achieved better performance than BIPS-Cap, implying the effectiveness of calibration of estimated logging probabilities. However, UIPS still consistently outperformed both ApproxKNN and IPS-C-TS. The main reason is that calibration primarily focuses on adjusting the estimated probabilities to ensure on *average* the model's predictions are reliable and accurate. In contrast, UIPS specifically handles the impact from each *individual* sample in policy learning.

UIPS also consistently outperformed Shrinkage (a special case of UIPS with uncertainties always being zero) on all three datasets, demonstrating the benefits of considering the estimation uncertainty. Finally, blindly reweighing through uncertainties, regardless of their impact on the accuracy of the resulting estimator and the learned policy, ultimately resulted in poor performance, as demonstrated by UIPS-P and UIPS-O.

Performance under different uncertainty levels. As shown in Figure 1, low-frequency samples in the logged dataset suffer higher uncertainties in their propensity estimation. Thus, we divided the test set into two subsets according to the average frequency of associated actions, where the uncertainty in the subset associated with low-frequency actions is on average 8% higher than that in high-frequency actions. Table 2 shows the results on these two subsets when $\tau = 0.5$. In addition, we include the results of the top three baselines that directly utilize the estimated logging policy. Table 2 clearly demonstrates that only UIPS performed better than CE on the test set with low-frequency actions, implying the distortion of inaccurately estimated logging probabilities and the effectiveness of UIPS in efficiently handling them.

Off-policy Evaluation. We further inspected whether $\hat{V}_{\mathrm{UIPS}}$ in Eq.(4) leads to more accurate off-policy evaluation. Following previous work [29, 46, 36], we evaluated the following ϵ-greedy policy: $\pi(a|\boldsymbol{x}) = \frac{1}{|M_x|}$ If $a \in M_x$ else $\epsilon/|\mathcal{A}|$, where M_x contains all positive actions associated with instance $\boldsymbol{x}$. For each $\boldsymbol{x}$ in the test set, we randomly sample 100 actions following the logging policy in Eq.(9) to generate the logged dataset. Table 3 shows the MSE of the estimators to the ground-truth policy value under 20 different random seeds. From Table 3, one can observe that: 1) IPS-GT with a skewer logging policy (i.e., smaller τ) leads to higher MSE, consistent with previous findings [29, 46, 36]; 2) inaccurate logging probabilities result in high bias and variance, leading to much larger MSE of BIPS compared to IPS-GT. Furthermore, this distortion is particularly pronounced when the ground-truth logging policy is skewed ($\tau = 0.5$); and 3) although all using the estimated logging policy, $\hat{V}_{\mathrm{UIPS}}$ yields the smallest MSE, comparing to other baselines that are designed to improve over BIPS.

Hyper-parameter Tuning. Discussions about hyper-parameter tuning and performance of UIPS under different hyper-parameters can also be found in Appendix 7.3.1.

4.2 Real-world data

To demonstrate the effectiveness of UIPS in real-world scenarios, we evaluate it on three recommendation datasets: (1) Yahoo! R3[1]; (2) Coat[2]; (3) KuaiRec [12], for music, fashion and short-video recommendations respectively. All these datasets contain an unbiased test set collected from a randomized controlled trial where items are randomly selected. The statistics of the three datasets and implementation details, e.g., model architectures and dataset splits, can be found in Appendix 7.3.2.

Following [10], we take $K = 5$ on Yahoo! R3 and Coat datasets, and $K = 50$ on KuaiRec dataset. The p-value under the t-test between UIPS and the best baseline on each dataset is also reported to investigate the significance of improvement.

[1] `https://webscope.sandbox.yahoo.com/`

[2] `https://www.cs.cornell.edu/~schnabts/mnar/`

Table 3: MSE of different off-policy estimators. A lower MSE indicates a more accurate estimator.

Algorithm	IPS-GT	BIPS	minVar	stableVar	Shrinage	UIPS
$\tau = 0.5$	$0.0875 \pm 4e^{-4}$	15.786 ± 1.51	$0.9021 \pm 7e^{-13}$	$0.8612 \pm 5e^{-8}$	$0.0718 \pm 5e^{-6}$	$\mathbf{0.0210} \pm 2e^{-6}$
$\tau = 1.0$	$0.0209 \pm 8e^{-5}$	0.5510 ± 0.388	$0.9019 \pm 8e^{-12}$	$0.8578 \pm 2e^{-7}$	$0.1978 \pm 2e^{-5}$	$\mathbf{0.0093} \pm 1e^{-6}$
$\tau = 2.0$	$0.0020 \pm 6e^{-6}$	0.5669 ± 0.013	$0.9015 \pm 5e^{-15}$	$0.8342 \pm 5e^{-7}$	$0.2952 \pm 3e^{-5}$	$\mathbf{0.0043} \pm 4e^{-7}$

Table 4: Experimental results on real-world datasets. The best and second best results are highlighted with **bold** and underline respectively. The p-value under the t-test between UIPS and the best baseline on each dataset is also provided.

	Yahoo			**Coat**			**KuaiRec**		
Algorithm	P@5	R@5	NDCG@5	P@5	R@5	NDCG@5	P@50	R@50	NDCG@50
CE	$0.2819 \pm 2e^{-3}$	$0.7594 \pm 6e^{-3}$	$0.6073 \pm 7e^{-3}$	$0.2799 \pm 5e^{-3}$	$0.4618 \pm 1e^{-2}$	$\underline{0.4529} \pm 7e^{-3}$	$0.8802 \pm 2e^{-3}$	$0.0240 \pm 8e^{-5}$	$0.8810 \pm 6e^{-3}$
BIPS-Cap	$0.2808 \pm 2e^{-3}$	$0.7576 \pm 5e^{-3}$	$0.6099 \pm 8e^{-3}$	$0.2758 \pm 6e^{-3}$	$0.4582 \pm 7e^{-3}$	$0.4399 \pm 9e^{-3}$	$0.8750 \pm 3e^{-3}$	$0.0238 \pm 7e^{-5}$	$0.8788 \pm 5e^{-3}$
MinVar	$0.2843 \pm 4e^{-3}$	$\underline{0.7685} \pm 1e^{-2}$	$0.6168 \pm 1e^{-2}$	$0.2813 \pm 3e^{-3}$	$\underline{0.4668} \pm 9e^{-3}$	$0.4414 \pm 8e^{-3}$	$0.8827 \pm 1e^{-3}$	$0.0240 \pm 5e^{-5}$	$0.8886 \pm 2e^{-3}$
stableVar	$0.2787 \pm 2e^{-3}$	$0.7499 \pm 7e^{-3}$	$0.5919 \pm 7e^{-3}$	$\underline{0.2840} \pm 3e^{-3}$	$0.4662 \pm 5e^{-3}$	$0.4393 \pm 7e^{-3}$	$0.8524 \pm 7e^{-3}$	$0.0231 \pm 2e^{-4}$	$0.8570 \pm 4e^{-3}$
Shrinkage	$\underline{0.2843} \pm 3e^{-3}$	$0.7654 \pm 8e^{-3}$	$\underline{0.6204} \pm 7e^{-3}$	$0.2790 \pm 5e^{-3}$	$0.4636 \pm 4e^{-3}$	$0.4464 \pm 1e^{-2}$	$0.8744 \pm 3e^{-3}$	$0.0238 \pm 9e^{-5}$	$0.8771 \pm 6e^{-3}$
SNIPS	$0.2222 \pm 4e^{-3}$	$0.5828 \pm 1e^{-2}$	$0.4357 \pm 1e^{-2}$	$0.2643 \pm 7e^{-3}$	$0.4287 \pm 1e^{-2}$	$0.4009 \pm 9e^{-3}$	$0.8411 \pm 6e^{-3}$	$0.0228 \pm 2e^{-4}$	$0.8431 \pm 6e^{-3}$
BanditNet	$0.2413 \pm 8e^{-3}$	$0.6442 \pm 2e^{-2}$	$0.4988 \pm 2e^{-2}$	$0.2781 \pm 8e^{-3}$	$0.4527 \pm 1e^{-2}$	$0.4251 \pm 1e^{-2}$	$0.8758 \pm 5e^{-3}$	$0.0239 \pm 2e^{-4}$	$0.8810 \pm 4e^{-3}$
POEM	$0.2732 \pm 3e^{-3}$	$0.7357 \pm 1e^{-2}$	$0.5880 \pm 1e^{-2}$	$0.2791 \pm 4e^{-3}$	$0.4566 \pm 6e^{-3}$	$0.4375 \pm 6e^{-3}$	$0.7785 \pm 1e^{-2}$	$0.0210 \pm 2e^{-4}$	$0.7779 \pm 6e^{-3}$
POXM	$0.2250 \pm 5e^{-3}$	$0.5940 \pm 1e^{-2}$	$0.4542 \pm 2e^{-2}$	$0.2663 \pm 6e^{-3}$	$0.4308 \pm 9e^{-3}$	$0.4006 \pm 1e^{-2}$	$\underline{0.8962} \pm 1e^{-2}$	$\underline{0.0245} \pm 4e^{-4}$	$\underline{0.9041} \pm 1e^{-2}$
Adaptive	$0.2762 \pm 3e^{-3}$	$0.7451 \pm 9e^{-3}$	$0.5919 \pm 8e^{-3}$	$0.2830 \pm 3e^{-3}$	$0.4634 \pm 5e^{-3}$	$0.4217 \pm 5e^{-3}$	$0.8375 \pm 1e^{-2}$	$0.0227 \pm 4e^{-4}$	$0.8460 \pm 1e^{-2}$
ApproxKNN	$0.2697 \pm 2e^{-3}$	$0.7225 \pm 5e^{-3}$	$0.5760 \pm 6e^{-3}$	$0.2755 \pm 2e^{-3}$	$0.4594 \pm 5e^{-3}$	$0.4490 \pm 4e^{-3}$	$0.8839 \pm 2e^{-6}$	$0.0240 \pm 5e^{-5}$	$0.8895 \pm 2e^{-3}$
IPS-C-TS	$0.2816 \pm 2e^{-3}$	$0.7582 \pm 5e^{-3}$	$0.6114 \pm 5e^{-3}$	$0.2799 \pm 3e^{-3}$	$0.4625 \pm 7e^{-3}$	$0.4462 \pm 6e^{-3}$	$0.8781 \pm 3e^{-3}$	$0.0239 \pm 1e^{-4}$	$0.8749 \pm 3e^{-3}$
UIPS-P	$0.1829 \pm 8e^{-3}$	$0.4560 \pm 3e^{-2}$	$0.3300 \pm 1e^{-2}$	$0.2685 \pm 7e^{-3}$	$0.4364 \pm 9e^{-3}$	$0.4087 \pm 7e^{-3}$	$0.8638 \pm 8e^{-3}$	$0.0235 \pm 3e^{-4}$	$0.8685 \pm 7e^{-3}$
UIPS-O	$0.1947 \pm 3e^{-3}$	$0.4959 \pm 1e^{-2}$	$0.3600 \pm 8e^{-3}$	$0.2657 \pm 5e^{-3}$	$0.4306 \pm 9e^{-3}$	$0.4146 \pm 9e^{-3}$	$0.8651 \pm 8e^{-3}$	$0.0235 \pm 2e^{-4}$	$0.8697 \pm 7e^{-3}$
UIPS	$\mathbf{0.2868} \pm 2e^{-3}$	$\mathbf{0.7742} \pm 5e^{-3}$	$\mathbf{0.6274} \pm 5e^{-3}$	$\mathbf{0.2877} \pm 3e^{-3}$	$\mathbf{0.4757} \pm 5e^{-3}$	$\mathbf{0.4576} \pm 8e^{-3}$	$\mathbf{0.9120} \pm 1e^{-3}$	$\mathbf{0.0250} \pm 5e^{-5}$	$\mathbf{0.9174} \pm 7e^{-4}$
p-value	$4e^{-2}$	$1e^{-2}$	$3e^{-2}$	$2e^{-2}$	$6e^{-4}$	$5e^{-5}$	$6e^{-4}$	$6e^{-4}$	$1e^{-3}$

We can first observe from Table 4 that on all three datasets, the proposed UIPS achieved the highest precision, recall and NDCG. Apparently, accurate estimation of logging probabilities in real-world scenarios with large action spaces and sparse interactions is challenging to achieve, causing BIPS-Cap to underperform CE. Additionally, calibration poses difficulties in such scenarios, leading to poor performance of ApproxKNN and IPS-C-TS across all three real-world datasets. BanditNet, POEM and POXM performed better in problems with a larger action space, while MinVar, stableVar and Shrinkage as well as Adaptive suited better for scenarios with a smaller action size. Again, UIPS outperformed Shrinkage, highlighting the importance of handling uncertainty in the estimated logging policy. But simply reweighing based on uncertainties, without considering their impact on the accuracy of the resulting estimator and the learned policy, led to poor performance, as shown by UIPS-P and UIPS-O.

4.3 Comparisons against other lines of off-policy learning

We discuss about the difference between UIPS and recent work with direct propensity estimation [34, 21], the DICE line of work [26, 47, 9] as well as the work on the distributionally robust off-policy learning [32, 17, 44] in Appendix 7.6, Appendix 7.5 and Appendix 7.7 respectively. Our empirical evaluations on recent work [26, 44, 34] from all three lines suggest that these lines of work cannot properly handle inaccuracies in the estimated logging probabilities that hinder policy improvement. Moreover, we also integrated the proposed UIPS with the doubly-robust (DR) estimator and our results suggest the accuracy of the imputation model greatly affected policy learning under DR, but UIPS still provides benefits. More details can be found in Appendix 7.4.

5 Related work

Our work is the first of its kind to account for the uncertainty of logging policy estimation in off-policy learning. The following two lines of work are most related to this work.

Off-policy learning. In many real-world applications, such as search engines and recommender systems, interactive online model update is expensive and risky [15]. Off-policy learning has therefore attracted great interest, since it can leverage the already logged feedback data [2, 8, 22]. The main challenge in off-policy learning is to address the mismatch between the logging policy and the learning policy. One common and widely-applied approach is to leverage the Inverse Propensity Scoring (IPS) method to correct the discrepancy between the two policies. And various methods are proposed to enhance IPS for more stabilizing learning [39, 37, 38] and improved variance control [23, 22], as well as extensions to more complex problems [40, 24].

However, all these solutions directly use the estimated logging policy for off-policy correction, leading to sub-optimal performance as shown in our experiments. A recent study on causal recommendation [10] also argues that propensity scores may not be correct due to unobserved confounders. They assume the effect of unobserved confounder for any sample can be bounded by a pre-defined hyper-parameter, and adversarially search for the worst-case propensity for learning. Mapping to off-policy learning, their solution is a special case of our UIPS-O variant with uncertainty as a pre-defined constant.

There were existing studies [34, 21, 26, 47, 9] also explore direct estimation of the propensity ratio to bypass estimating the logging policy. However, as discussed in Appendix 7.6 and Appendix 7.5, they demonstrate inferior performance compared to UIPS. This is primarily due to either the lack of consideration for the accuracy of the estimated propensity ratio, similar to the limitations of existing IPS-type algorithms in handling inaccurately estimated logging probabilities, or the degeneration to a specific IPS estimator that suffers high variance.

Recent work on distributionally robust off-policy evaluation and learning [32, 17, 44] also addresses uncertainty in off-policy learning. However, their approach to handling uncertainty and the underlying motivation differ significantly from ours, resulting in distinct techniques employed. Further details can be found in Appendix 7.7. Additionally, experiments conducted in Appendix 7.7 demonstrate that directly adapting methods from distributionally robust off-policy learning to handle inaccurately estimated logging probabilities leads to poor performance.

Off-policy learning can also be directly built on off-policy evaluation. Several work [36, 46] also propose to control the variance of the estimator caused by small logging probabilities through instance reweighing. Again, they directly use the estimated logging policy for correction, and thus performed worse than UIPS as observed in our experiments. A recent study [29] assumed additional structure in the action space and proposed the marginalized IPS. Instead, our work considers the uncertainty in the estimated logging policy and thus does not add any new assumptions about the problem space.

Uncertainty-aware learning. Estimation uncertainty has been extensively studied [45, 50, 1]. In the context of on-policy reinforcement leanring and bandits [1, 48, 49], the use of uncertainty aims to strike a balance between exploration and exploitation by adopting an optimistic approach (i.e., UCB in bandits). One the other hand, most reseach on offline reinforcement learning/bandits [43, 4, 6] tends to be more conservative, employing techniques such as Lower Confidence Bounds (LCB) or penalizing out-of-distribution states and actions based on uncertainty to address extrapolation errors. However, these principles differ fundamentally from UIPS, which directly minimizes the mean square error of off-policy evaluation. The closed-form solution of the resulting per-instance weight in UIPS reflects how uncertainty contributes to the policy evaluation error. Moreover, Our UIPS-O and UIPS-P baselines leverage uncertainties using the two aforementioned general principles respectively. However, empirical findings indicate that blindly penalizing or boosting samples based on uncertainty is problematic. Proper correction depends on both uncertainty in logging policy estimation and the actual value of estimated logging probabilities.

6 Conclusion

In this paper, we propose a Uncertainty-aware Inverse Propensity Score estimator (UIPS) to explicitly model the uncertainty of the estimated logging policy for improved off-policy learning. UIPS weighs each logged instance to reduce its policy evaluation error, where the optimal weights have a closed-form solution derived by minimizing the upper bound of the resulting estimator's mean squared error (MSE) to its ground-truth value. An improved policy is then obtained by optimizing the resulting estimation. Extensive experiments on synthetic and three real-world datasets as well as the theoretical convergence guarantee demonstrate the efficiency of UIPS.

As demonstrated in this work, explicitly modeling the uncertainty of the estimated logging policy is crucial for effective off-policy learning; but the best use of this uncertainty is not to simply down-weigh or drop instances with uncertain estimations, but to balance it with the actually estimated logging probabilities in a per-instance basis. As our future work, it is promising to investigate how UIPS can be extended to value-based learning methods, e.g., actor-critics. And on the other hand, it is also important to analyze how tight our upper bound analysis of policy evaluation error is; and if possible, find new ways to tighten it for improvements.

References

[1] Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems*, pages 2312–2320, 2011.

[2] Aman Agarwal, Ivan Zaitsev, Xuanhui Wang, Cheng Li, Marc Najork, and Thorsten Joachims. Estimating position bias without intrusive interventions. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pages 474–482, 2019.

[3] Ahmad Ajalloeian and Sebastian U Stich. On the convergence of sgd with biased gradients. *arXiv preprint arXiv:2008.00051*, 2020.

[4] Gaon An, Seungyong Moon, Jang-Hyun Kim, and Hyun Oh Song. Uncertainty-based offline reinforcement learning with diversified q-ensemble. *Advances in neural information processing systems*, 34:7436–7447, 2021.

[5] Raghu Aniruddh, Gottesman Omer, Liu Yao, Komorowski Matthieu, Faisal Aldo, Doshi-Velez Finale, and Brunskill Emma. Behaviour policy estimation in off-policy policy evaluation: Calibration matters. *arXiv preprint arXiv:1807.01066*, 2018.

[6] Chenjia Bai, Lingxiao Wang, Zhuoran Yang, Zhihong Deng, Animesh Garg, Peng Liu, and Zhaoran Wang. Pessimistic bootstrapping for uncertainty-driven offline reinforcement learning. *arXiv preprint arXiv:2202.11566*, 2022.

[7] K. Bhatia, K. Dahiya, H. Jain, P. Kar, A. Mittal, Y. Prabhu, and M. Varma. The extreme classification repository: Multi-label datasets and code, 2016.

[8] Minmin Chen, Alex Beutel, Paul Covington, Sagar Jain, Francois Belletti, and Ed H Chi. Top-k off-policy correction for a reinforce recommender system. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pages 456–464, 2019.

[9] Bo Dai, Ofir Nachum, Yinlam Chow, Lihong Li, Csaba Szepesvári, and Dale Schuurmans. Coindice: Off-policy confidence interval estimation. *Advances in neural information processing systems*, 33:9398–9411, 2020.

[10] Sihao Ding, Peng Wu, Fuli Feng, Yitong Wang, Xiangnan He, Yong Liao, and Yongdong Zhang. Addressing unmeasured confounder for recommendation with sensitivity analysis. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 305–315, 2022.

[11] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR, 2016.

[12] Chongming Gao, Shijun Li, Wenqiang Lei, Jiawei Chen, Biao Li, Peng Jiang, Xiangnan He, Jiaxin Mao, and Tat-Seng Chua. Kuairec: A fully-observed dataset and insights for evaluating recommender systems. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management*, CIKM '22, 2022.

[13] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR, 2017.

[14] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. Deepfm: a factorization-machine based neural network for ctr prediction. *arXiv preprint arXiv:1703.04247*, 2017.

[15] Nan Jiang and Lihong Li. Doubly robust off-policy value evaluation for reinforcement learning. In *International Conference on Machine Learning*, pages 652–661. PMLR, 2016.

[16] Thorsten Joachims, Adith Swaminathan, and Maarten De Rijke. Deep learning with logged bandit feedback. In *International Conference on Learning Representations*, 2018.

[17] Nathan Kallus, Xiaojie Mao, Kaiwen Wang, and Zhengyuan Zhou. Doubly robust distributionally robust off-policy evaluation and learning. In *International Conference on Machine Learning*, pages 10598–10632. PMLR, 2022.

[18] Patrick Kidger and Terry Lyons. Universal approximation with deep narrow networks. In *Conference on learning theory*, pages 2306–2327. PMLR, 2020.

[19] Sergey Levine and Vladlen Koltun. Guided policy search. In *International conference on machine learning*, pages 1–9. PMLR, 2013.

[20] Lihong Li, Yu Lu, and Dengyong Zhou. Provably optimal algorithms for generalized linear contextual bandits. In *International Conference on Machine Learning*, pages 2071–2080. PMLR, 2017.

[21] Xu-Hui Liu, Zhenghai Xue, Jingcheng Pang, Shengyi Jiang, Feng Xu, and Yang Yu. Regret minimization experience replay in off-policy reinforcement learning. *Advances in Neural Information Processing Systems*, 34:17604–17615, 2021.

[22] Yaxu Liu, Jui-Nan Yen, Bowen Yuan, Rundong Shi, Peng Yan, and Chih-Jen Lin. Practical counterfactual policy learning for top-k recommendations. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1141–1151, 2022.

[23] Romain Lopez, Inderjit S Dhillon, and Michael I Jordan. Learning from extreme bandit feedback. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 8732–8740, 2021.

[24] Jiaqi Ma, Zhe Zhao, Xinyang Yi, Ji Yang, Minmin Chen, Jiaxi Tang, Lichan Hong, and Ed H Chi. Off-policy learning in two-stage recommender systems. In *Proceedings of The Web Conference 2020*, pages 463–473, 2020.

[25] Rémi Munos, Tom Stepleton, Anna Harutyunyan, and Marc Bellemare. Safe and efficient off-policy reinforcement learning. *Advances in neural information processing systems*, 29, 2016.

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

[27] Doina Precup. Eligibility traces for off-policy policy evaluation. *Computer Science Department Faculty Publication Series*, page 80, 2000.

[28] Aniruddh Raghu, Omer Gottesman, Yao Liu, Matthieu Komorowski, Aldo Faisal, Finale Doshi-Velez, and Emma Brunskill. Behaviour policy estimation in off-policy policy evaluation: Calibration matters. *arXiv preprint arXiv:1807.01066*, 2018.

[29] Yuta Saito and Thorsten Joachims. Off-policy evaluation for large action spaces via embeddings. *arXiv preprint arXiv:2202.06317*, 2022.

[30] Tobias Schnabel, Adith Swaminathan, Ashudeep Singh, Navin Chandak, and Thorsten Joachims. Recommendations as treatments: Debiasing learning and evaluation. In *international conference on machine learning*, pages 1670–1679. PMLR, 2016.

[31] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897. PMLR, 2015.

[32] Nian Si, Fan Zhang, Zhengyuan Zhou, and Jose Blanchet. Distributionally robust policy evaluation and learning in offline contextual bandits. In *International Conference on Machine Learning*, pages 8884–8894. PMLR, 2020.

[33] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

[34] Samarth Sinha, Jiaming Song, Animesh Garg, and Stefano Ermon. Experience replay with likelihood-free importance weights. In *Learning for Dynamics and Control Conference*, pages 110–123. PMLR, 2022.

[35] Alex Strehl, John Langford, Lihong Li, and Sham M Kakade. Learning from logged implicit exploration data. *Advances in neural information processing systems*, 23, 2010.

[36] Yi Su, Maria Dimakopoulou, Akshay Krishnamurthy, and Miroslav Dudík. Doubly robust off-policy evaluation with shrinkage. In *International Conference on Machine Learning*, pages 9167–9176. PMLR, 2020.

[37] Adith Swaminathan and Thorsten Joachims. Batch learning from logged bandit feedback through counterfactual risk minimization. *The Journal of Machine Learning Research*, 16(1):1731–1755, 2015.

[38] Adith Swaminathan and Thorsten Joachims. Counterfactual risk minimization: Learning from logged bandit feedback. In *International Conference on Machine Learning*, pages 814–823. PMLR, 2015.

[39] Adith Swaminathan and Thorsten Joachims. The self-normalized estimator for counterfactual learning. *advances in neural information processing systems*, 28, 2015.

[40] Adith Swaminathan, Akshay Krishnamurthy, Alekh Agarwal, Miro Dudik, John Langford, Damien Jose, and Imed Zitouni. Off-policy evaluation for slate recommendation. *Advances in Neural Information Processing Systems*, 30, 2017.

[41] Sebastian Thrun and Michael L Littman. Reinforcement learning: an introduction. *AI Magazine*, 21(1):103–103, 2000.

[42] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256, 1992.

[43] Yue Wu, Shuangfei Zhai, Nitish Srivastava, Joshua Susskind, Jian Zhang, Ruslan Salakhutdinov, and Hanlin Goh. Uncertainty weighted actor-critic for offline reinforcement learning. *arXiv preprint arXiv:2105.08140*, 2021.

[44] Da Xu, Yuting Ye, Chuanwei Ruan, and Bo Yang. Towards robust off-policy learning for runtime uncertainty. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 10101–10109, 2022.

[45] Pan Xu, Zheng Wen, Handong Zhao, and Quanquan Gu. Neural contextual bandits with deep representation and shallow exploration. In *International Conference on Learning Representations*, 2021.

[46] Ruohan Zhan, Vitor Hadad, David A Hirshberg, and Susan Athey. Off-policy evaluation via adaptive weighting with data from contextual bandits. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 2125–2135, 2021.

[47] Ruiyi Zhang, Bo Dai, Lihong Li, and Dale Schuurmans. Gendice: Generalized offline estimation of stationary values. *arXiv preprint arXiv:2002.09072*, 2020.

[48] Xiaoying Zhang, Hong Xie, Hang Li, and John CS Lui. Conversational contextual bandit: Algorithm and application. In *Proceedings of the web conference 2020*, pages 662–672, 2020.

[49] Xiaoying Zhang, Hong Xie, and John CS Lui. Heterogeneous information assisted bandit learning: Theory and application. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, pages 2135–2140. IEEE, 2021.

[50] Dongruo Zhou, Lihong Li, and Quanquan Gu. Neural contextual bandits with ucb-based exploration. In *International Conference on Machine Learning*, pages 11492–11502. PMLR, 2020.

[51] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. Deep interest network for click-through rate prediction. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1059–1068, 2018.

Algorithm 1 UIPS

1: **Input:** The logged dataset $D := \{(\boldsymbol{x}_n, a_n, r_{\boldsymbol{x}_n,a_n})\}_{n=1}^{N}$, the estimated logging policy model $\hat{\beta}(a|\boldsymbol{x}) = \frac{\exp(f_{\boldsymbol{\theta}}(\boldsymbol{x},a))}{\sum_{a'}\exp(f_{\boldsymbol{\theta}}(\boldsymbol{x},a'))}$, dimension d.
2: Initialize $\boldsymbol{M}_D = \boldsymbol{I}_{d\times d}$.
3: **for** $n = 1$ **to** N **do**
4: $\quad \boldsymbol{M}_D = \boldsymbol{M}_D + \nabla_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}}(\boldsymbol{x}_n, a_n) \nabla_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}}(\boldsymbol{x}_n, a_n)^\top$ ▷ $\boldsymbol{M}_D$ for uncertainty calculation.
5: **end for**
6: **for** $n = 1$ **to** N **do**
7: $\quad U_{\boldsymbol{x}_n,a_n} = \sqrt{\nabla_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}}(\boldsymbol{x}_n, a_n)^\top \boldsymbol{M}_D^{-1} \nabla_{\boldsymbol{\theta}} f_{\boldsymbol{\theta}}(\boldsymbol{x}_n, a_n)}$
8: **end for**
9: **while** not converge **do**
10: $\quad$ **for** $n = 1$ **to** N **do**
11: $\quad\quad$ Calculate $\phi_{\boldsymbol{x}_n,a_n}$ as in Theorem 3.2
12: $\quad\quad$ Calculate gradients as in Eq.(5) and updating $\pi_{\boldsymbol{\vartheta}}(a|\boldsymbol{x})$.
13: $\quad$ **end for**
14: **end while**
15: **Output:**The learnt policy $\pi_{\boldsymbol{\vartheta}}(a|\boldsymbol{x})$.

7 Appendix

7.1 Notations and Algorithm Framework.

For ease of reading, we list important notations in Table 5 and summarize the main framework of the proposed UIPS in Algorithm 1.

Notation	Description
$\mathcal{X}$	context space
$\mathcal{A}$	action set
$\boldsymbol{x} \in \mathbb{R}^d$	context vector
a	action
$r_{\boldsymbol{x},a}$	reward
$\pi(a\|\boldsymbol{x})$	targeted policy to evaluate
$\beta(a\|\boldsymbol{x})$	the unknown ground-truth logging policy
$\hat{\beta}(a\|\boldsymbol{x})$	the estimated logging policy
$V(\pi)$	value function
$D := \{(\boldsymbol{x}_n, a_n, r_{\boldsymbol{x}_n,a_n})\}_{n=1}^{N}$	logged dataset containing N samples
$\phi_{\boldsymbol{x},a}$	the optimal uncertainty-aware weight
$f_{\boldsymbol{\theta}^*}(\boldsymbol{x}, a)$	the unknown ground-truth function that generates $\beta(a\|\boldsymbol{x}) = \frac{\exp(f_{\boldsymbol{\theta}^*}(\boldsymbol{x},a))}{\sum_{a'}\exp(f_{\boldsymbol{\theta}^*}(\boldsymbol{x},a'))}$
$f_{\boldsymbol{\theta}}(\boldsymbol{x}, a)$	the estimate of $f_{\boldsymbol{\theta}^*}(\boldsymbol{x}, a)$ that generates $\hat{\beta}(a\|\boldsymbol{x})$
$\boldsymbol{B}_{\boldsymbol{x},a}$	confidence interval of $\hat{\beta}(a\|\boldsymbol{x})$
$U_{\boldsymbol{x},a}$	uncertainty defined as $\|f_{\boldsymbol{\theta}^*}(\boldsymbol{x}, a) - f_{\boldsymbol{\theta}}(\boldsymbol{x}, a)\| \leq \gamma U_{\boldsymbol{x},a}$
$\boldsymbol{g}(\boldsymbol{x}_n, a_n)$	gradient of $f_{\boldsymbol{\theta}}(\boldsymbol{x}, a)$ regarding to the last layer.

Table 5: Notations

Computation Cost. The additional computation cost of UIPS over IPS comes from two parts:

- Pre-calculate uncertainties (line 1-5 in Algorithm 1) : This part calculates uncertainty of the logging probability for each (s, a) pair, and it *only needs to be executed once*. The computational cost of this step is $O(Nd^2 + d^3)$, where $O(Nd^2)$ is for calculating uncertainties in each (s, a) pair and $O(d^3)$ is for matrix inverse.
- Calculate $\phi_{\boldsymbol{x},a}$ during training (line 8 in Algorithm 1): It only takes O(1) time, the same computational cost as calculating IPS score.

Note that calculating the logging probability for each sample, which is essential for both UIPS and IPS, takes $O(Nd|\mathcal{A}|)$ time. Since the dimension d is usually much less than action size $|\mathcal{A}|$ and sample size N, UIPS does not introduce significant computational overhead compared to the original IPS solution.

7.2 Derivation of confidence interval of logging probability.

Given that $|f_{\theta^*}(\boldsymbol{x}, a) \quad f_{\boldsymbol{\theta}}(\boldsymbol{x}, a)| \quad \gamma U_{\boldsymbol{x},a}$ holds with probability at least $1 \quad \sigma$ and

$$\beta \ (a|\boldsymbol{x}) = \frac{\exp(f_{\boldsymbol{\theta}^*}(\boldsymbol{x}, a))}{Z}, \hat{\beta}(a|\boldsymbol{x}) = \frac{\exp(f_{\boldsymbol{\theta}}(\boldsymbol{x}, a))}{\hat{Z}},$$

where $Z \ = \sum_{a'} \exp(f_\theta \ (a'|\boldsymbol{x}))$ and $\hat{Z} = \sum_{a'} \exp(f_\theta(a'|\boldsymbol{x}))$, we can get that with probability at least $1 \quad \sigma$:

$$\begin{aligned}
& |f_{\boldsymbol{\theta}^*}(\boldsymbol{x}, a) \quad f_{\boldsymbol{\theta}}(\boldsymbol{x}, a)| \quad \gamma U_{\boldsymbol{x},a} \\
\Leftrightarrow\ & f_{\boldsymbol{\theta}}(\boldsymbol{x}, a) \quad \gamma U_{\boldsymbol{x},a} \quad f_{\boldsymbol{\theta}^*}(\boldsymbol{x}, a) \quad f_{\boldsymbol{\theta}}(\boldsymbol{x}, a) + \gamma U_{\boldsymbol{x},a} \\
\Leftrightarrow\ & \exp(f_{\boldsymbol{\theta}}(\boldsymbol{x}, a)) \exp(\ \gamma U_{\boldsymbol{x},a}) \quad \exp(f_{\boldsymbol{\theta}^*}(\boldsymbol{x}, a)) \quad \exp(f_{\boldsymbol{\theta}}(\boldsymbol{x}, a)) \exp(\gamma U_{\boldsymbol{x},a}) \\
\overset{(1)}{\Leftrightarrow}\ & \hat{Z}\hat{\beta}(a|\boldsymbol{x}) \exp(\ \gamma U_{\boldsymbol{x},a}) \quad \exp(f_{\boldsymbol{\theta}^*}(\boldsymbol{x}, a)) \quad \hat{Z}\hat{\beta}(a|\boldsymbol{x}) \exp(\gamma U_{\boldsymbol{x},a}) \\
\overset{(2)}{\Leftrightarrow}\ & \frac{\hat{Z} \exp(\ \gamma U_{\boldsymbol{x},a})}{Z} \hat{\beta}(a|\boldsymbol{x}) \quad \beta \ (a|\boldsymbol{x}) \quad \frac{\hat{Z} \exp(\gamma U_{\boldsymbol{x},a})}{Z} \hat{\beta}(a|\boldsymbol{x})
\end{aligned}$$

The step labeled as (1) is due to the modelling of $\hat{\beta}(a|\boldsymbol{x})$. And the step labeled as (2) is because Z is a positive constant independent of $\hat{\beta}(a|\boldsymbol{x})$. Thus with probability at least 1-δ, we have $\beta \ (a|\boldsymbol{x}) \in \boldsymbol{B}_{\boldsymbol{x},a}$ and

$$\boldsymbol{B}_{\boldsymbol{x},a} = \left[\frac{\hat{Z} \exp(\ \gamma U_{\boldsymbol{x},a})}{Z} \hat{\beta}(a|\boldsymbol{x}), \frac{\hat{Z} \exp(\gamma U_{\boldsymbol{x},a})}{Z} \hat{\beta}(a|\boldsymbol{x}) \right].$$

7.3 Experiment details

7.3.1 Synthetic Data

Data generation. Given the ground-truth logging policy $\beta \ (a|\boldsymbol{x})$, we generate the logged dataset as follows. For each sample in train set, we first get the embedded context vector $\boldsymbol{x}$ from its original feature vector $\tilde{\boldsymbol{x}}$. We then sample an action a according to $\beta \ (a|\boldsymbol{x})$, and obtain the reward $r_{\boldsymbol{x},a} = \boldsymbol{y}_{\tilde{\boldsymbol{x}},a}$, resulting bandit feedback $(\boldsymbol{x}, a, r_{\boldsymbol{x},a})$, where $\boldsymbol{y}_{\tilde{\boldsymbol{x}},a}$ is the label of class a under the original feature vector $\tilde{\boldsymbol{x}}$. We repeat above process N times to collect the logged dataset. In our experiments, we take $d = 64, N = 100$.

We model the logging policy as in Eq.(9), where $\{\boldsymbol{\theta}_a\}$ are the parameters to be estimated. To train the logging policy, we take all samples in the logged dataset D as positive instances, and randomly sample non-selected actions as negative instances as in [8].

7.3.2 Real-world Data

Statistics of datasets. The statistics of three real-world recommendation datasets with unbiased data can be found in Table 6.

Table 6: The statistics of three real-world datasets.

Dataset	#User	#Item	#Biased Data	#Unbiased Data
Yahoo R3	15,400	1,000	311,704	54,000
Coat	290	300	6,960	4,640
KuaiRec	7,176	10,729	12,530,806	4,676,570

All these datasets contain a set of biased data collected from users' interactions on the platform, and a set of unbiased data collected from a randomized controlled trial where items are randomly selected. As in [10], on each dataset, the biased data is used for training, and the unbiased data is for testing, with a small part of unbiased data split for validation purpose (5% on Yahoo R3 and Coat,

and 15% on KuaiRec). We take the reward as 1 if : (1) the rating is larger than 3 in Yahoo! R3 and Coat datasets; (2) the user watched more than 70% of the video in KuaiRec. Otherwise, the reward is labeled as 0.

We adopted a two-tower neural network architecture to implement both the logging and learning policy, as shown in Figure 2. For the learning policy, the user representation and item representation are first modelled through two separate neural networks (i.e., the user tower and the item tower), and then their element-by-element product vector is projected to predict the user's preference for the item. We then re-use the user state generated from the user tower of the learning policy, and model the logging policy with another separate item tower, following [8]. We also block gradients to prevent the logging policy learning interfering the user state of the learning policy. In each learning epoch, we will first estimate the logging policy, and then take the estimated logging probabilities as well as their uncertainties to optimize the learning policy.

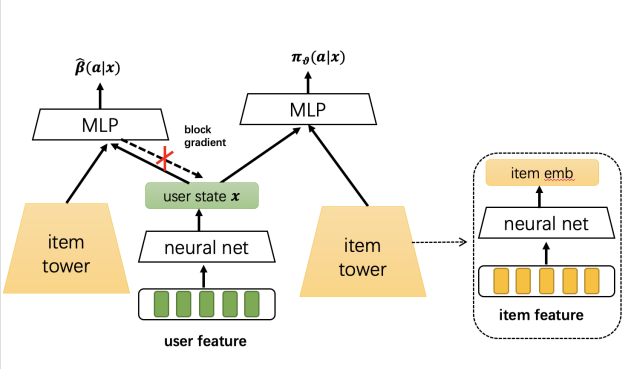


Figure 2: Model architecture of the logging and the learning policy in real-world datasets

7.3.3 Implementation details.

To facilitate hyper-parameter tuning, we disentangled two ηs in two the terms of $\phi_{\boldsymbol{x},a}$ in Theorem 3.2, and introduce η_1 and η_2 to represent η in the first and second term respectively in our implementation. This is due to the scale of η in the first term is closely related to the scale of λ, with $\lambda/\overline{p}\,\overline{\eta}$ as the truly effective hyper-parameter. But the scale of η in the second term is independent from λ.

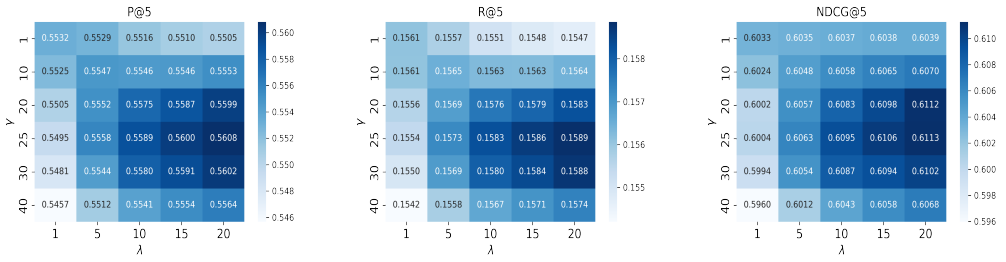
Moreover, while $\lambda = \mathsf{E}_{\pi_\vartheta}\left[r^2_{\boldsymbol{x},a}\frac{\pi_\vartheta(aj\boldsymbol{x})}{\beta\ (aj\boldsymbol{x})}\right]$ depends on π_θ as discussed in Eq.(6), we cannot adaptively set the value of λ since the ground-truth logging policy $\beta\ (aj\boldsymbol{x})$ is unknown. However, we have:

$$\lambda := E_\pi\left[r^2_{\boldsymbol{x},a}\frac{\pi_\theta(aj\boldsymbol{x})}{\beta\ (aj\boldsymbol{x})}\right] \quad \left(\sum_a \pi_\theta(aj\boldsymbol{x})^4\right)\left(\sum_a \frac{r^4_{\boldsymbol{x},a}}{\beta\ (aj\boldsymbol{x})^2}\right) \quad \left(\sum_a \frac{r^4_{\boldsymbol{x},a}}{\beta\ (aj\boldsymbol{x})^2}\right),$$

where the first inequality is due to the Cauchy–Schwarz inequality and the second inequality is because $\sum_a \pi_\theta(aj\boldsymbol{x})^4 \quad \sum_a \pi_\theta(aj\boldsymbol{x}) = 1$ with $\pi_\theta(aj\boldsymbol{x})$ 2 $[0,1]$. We denote $\tilde{\lambda} = \left(\sum_a \frac{r^4_{\boldsymbol{x},a}}{\beta\ (aj\boldsymbol{x})^2}\right)$, which is dataset-specific constant and independent from π_θ. By replacing λ in Eq.(6) with $\tilde{\lambda}$, we can still minimize an upper bound of $\mathrm{MSE}(\hat{V}_{\mathrm{UIPS}}(\pi_\vartheta))$, which ensures that the result of our analysis still holds. Thus, considering ease of computation and efficiency, we take a fixed λ during our policy learning.

We then use grid search to select hyperparameters based on the model's performance on the validation dataset: the learning rate was searched in $\{1e^{-5}, 1e^{-4}, 1e^{-3}, 1e^{-2}\}$; λ, γ, η_1 were searched in {0.5, 0.1, 1, 2,5, 10, 15, 20, 25, 30, 40, 50}. And η_2 was searched in {1, 10, 100, 1000}. For baseline algorithms, we also performed a similar grid search as mentioned above, and the search range follows the original papers.

Ablation study: hyper-parameter tuning. Although UIPS has four hyperparameters (λ, γ, η_1, and η_2), one only needs to carefully finetune two of them, i.e., γ and η_1^2/λ, to obtain good performance of UIPS. This is because:



(a) Precision@5 (b) Recall@5 (c) NDCG@5

Figure 3: Effect of λ and γ on synthetic dataset with $\tau = 0.5$

- η_2 acts as a capping threshold to ensure $\phi_{\boldsymbol{x},a} \quad 2\eta_2$ holds even when the corresponding propensity scores are very low. Hence, it should be set to a reasonably large value (e.g., 100).
- The key component (i.e., the first term) of $\phi_{\boldsymbol{x},a}$ can be rewritten in the following way. While all $(\boldsymbol{x}, a)$ pairs will be multiplied by $\phi_{\boldsymbol{x},a}$, η_1 in the numerator will not affect final performance too much, and the key is to find a good value of η_1^2/λ to balance the two terms in the denominator:

$$\eta_1 / \left[\exp\left(\quad \gamma U_{\boldsymbol{x},a} \right) + \frac{\eta_1^2/\lambda \quad \pi_{\boldsymbol{\vartheta}}(a|\boldsymbol{x})^2}{\hat{\beta}(a|\boldsymbol{x})^2 \exp\left(\quad \gamma U_{\boldsymbol{x},a} \right)} \right].$$

Thus with η_1 and η_2 fixed, the effect of hyper-parameter γ and λ on precision, recall as well as NDCG can be found in Figure 3. We can observe that to make UIPS excel, $\boldsymbol{B}_{\boldsymbol{x},a}$ needs to be of high confidence, e.g., $\gamma = 25$ performed the best on the dataset with $\tau = 0.5$. Moreover, the threshold $\overline{\lambda}/\eta_1$ cannot be too small or too large.

7.4 Experiments on the doubly robust estimators.

The doubly robust (DR) estimator [15], which is a hybrid of *direct method* (DM) estimator and *inverse propensity score* (IPS) estimator, is also widely used for off-policy evaluation. More specifically, let $\hat{\eta} : \mathcal{X} \quad \mathcal{A} \quad ! \quad \mathbb{R}$ be the imputation model in DM that estimates the reward of action a under context vector $\boldsymbol{x}$, and $\hat{\beta}(a|\boldsymbol{x})$ be the estimated logging policy in the IPS estimator. The DR estimator evaluates policy π based on the logged dataset $D := \{(\boldsymbol{x}_n, a_n, r_{\boldsymbol{x}_n,a_n})\}_{n \in [N]}$, by:

$$\hat{V}_{\mathrm{DR}}(\pi) = \hat{V}_{\mathrm{DM}}(\pi) + \frac{1}{N} \sum_{n=1}^{N} \frac{\pi(a_n|\boldsymbol{x}_n)}{\hat{\beta}(a_n|\boldsymbol{x}_n)} \left(r_{\boldsymbol{x}_n,a_n} \quad \hat{\eta}(\boldsymbol{x}_n, a_n) \right) \tag{10}$$

where $\hat{V}_{\mathrm{DM}}(\pi)$ is the DM estimator:

$$\hat{V}_{\mathrm{DM}}(\pi) = \frac{1}{N} \sum_{n=1}^{N} \sum_{a \in \mathcal{A}} \pi(a|\boldsymbol{x}_n) \hat{\eta}(\boldsymbol{x}_n, a). \tag{11}$$

Again assume the policy $\pi(a|\boldsymbol{x})$ is parameterized by $\boldsymbol{\vartheta}$, the REINFORCE gradient of $\hat{V}_{\mathrm{DR}}(\pi_{\boldsymbol{\vartheta}})$ with respect to $\boldsymbol{\vartheta}$ can be readily derived as follows:

$$\begin{aligned} \nabla_{\boldsymbol{\vartheta}} \hat{V}_{\mathrm{DR}}(\pi_{\boldsymbol{\vartheta}}) = & \frac{1}{N} \sum_{n=1}^{N} \left(\sum_{a \in \mathcal{A}} \pi_{\boldsymbol{\vartheta}}(a|\boldsymbol{x}_n) \hat{\eta}(\boldsymbol{x}_n, a) \nabla_{\boldsymbol{\vartheta}} \log(\pi_{\boldsymbol{\vartheta}}(a|\boldsymbol{x}_n)) \right) \\ & + \frac{1}{N} \sum_{n=1}^{N} \left(\frac{\pi(a_n|\boldsymbol{x}_n)}{\hat{\beta}(a_n|\boldsymbol{x}_n)} \left(r_{\boldsymbol{x}_n,a_n} \quad \hat{\eta}(\boldsymbol{x}_n, a_n) \right) \nabla_{\boldsymbol{\vartheta}} \log(\pi_{\boldsymbol{\vartheta}}(a_n|\boldsymbol{x}_n)) \right). \end{aligned} \tag{12}$$

The imputation model $\hat{\eta}(\boldsymbol{x}, a)$ is pre-trained following previous work [22] with the same neural network architecture as the logging policy model. Besides the standard DR estimator, we also adapt

Table 7: Experiment results on synthetic datasets. The best and second best results are highlighted with **bold** and underline respectively. Two p-values are calculated: (1) p-value (UIPSDR): The p-value under the t-test between UIPSDR and the best DR baseline on each dataset; (2) p-value (UIPS): The p-value under the t-test between UIPS and the best DR baseline on each dataset.

	$\tau = 0.5$			$\tau = 1$			$\tau = 2$		
Algorithm	P@5	R@5	NDCG@5	P@5	R@5	NDCG@5	P@5	R@5	NDCG@5
BIPS-Cap	0.5515 $_{2e^{-3}}$	0.1553 $_{8e^{-4}}$	0.6031 $_{2e^{-3}}$	0.5526 $_{2e^{-3}}$	0.1561 $_{6e^{-4}}$	0.6016 $_{1e^{-3}}$	0.5409 $_{3e^{-3}}$	0.1529 $_{9e^{-4}}$	0.5901 $_{2e^{-3}}$
UIPS	0.5589 $_{3e^{-3}}$	0.1583 $_{9e^{-4}}$	0.6095 $_{3e^{-3}}$	0.5572 $_{2e^{-3}}$	0.1571 $_{8e^{-4}}$	0.6074 $_{2e^{-3}}$	0.5432 $_{3e^{-3}}$	0.1534 $_{8e^{-4}}$	0.5946 $_{2e^{-3}}$
DR	0.3846 $_{3e^{-2}}$	0.1082 $_{8e^{-3}}$	0.4684 $_{3e^{-2}}$	0.3631 $_{3e^{-2}}$	0.1017 $_{9e^{-3}}$	0.4494 $_{3e^{-2}}$	0.3560 $_{3e^{-2}}$	0.0995 $_{7e^{-3}}$	0.4470 $_{2e^{-3}}$
MinVarDR	0.3212 $_{3e^{-2}}$	0.0908 $_{8e^{-3}}$	0.4062 $_{3e^{-2}}$	0.3240 $_{5e^{-2}}$	0.0903 $_{1e^{-2}}$	0.3905 $_{5e^{-2}}$	0.3234 $_{5e^{-2}}$	0.0910 $_{1e^{-2}}$	0.4059 $_{4e^{-2}}$
ShrinkageDR	0.4139 $_{2e^{-2}}$	0.1161 $_{7e^{-3}}$	0.4969 $_{3e^{-2}}$	0.3944 $_{3e^{-2}}$	0.1101 $_{8e^{-3}}$	0.4797 $_{2e^{-2}}$	0.4080 $_{3e^{-2}}$	0.1135 $_{7e^{-3}}$	0.4901 $_{2e^{-2}}$
UIPSDR	**0.4278** $_{2e^{-2}}$	**0.1200** $_{6e^{-3}}$	**0.5069** $_{2e^{-2}}$	**0.4008** $_{2e^{-2}}$	**0.1126** $_{7e^{-3}}$	**0.4847** $_{2e^{-2}}$	**0.4144** $_{2e^{-2}}$	**0.1162** $_{8e^{-3}}$	**0.4972** $_{2e^{-2}}$
p-value (UIPSDR)	$2e^{-1}$	$2e^{-1}$	$3e^{-1}$	$6e^{-1}$	$4e^{-1}$	$6e^{-1}$	$6e^{-1}$	$4e^{-1}$	$5e^{-1}$
p-value (UIPS)	$6e^{-13}$	$4e^{-13}$	$4e^{-12}$	$2e^{-12}$	$1e^{-12}$	$5e^{-12}$	$8e^{-12}$	$8e^{-12}$	$2e^{-11}$

Table 8: Experiment results on real-world unbiased datasets. The best and second best results are highlighted with **bold** and underline respectively. Two p-values are calculated: (1) p-value (UIPSDR): The p-value under the t-test between UIPSDR and the best DR baseline on each dataset; (2) p-value (UIPS): The p-value under the t-test between UIPS and the best DR baseline on each dataset.

	Yahoo			**Coat**			**KuaiRec**		
Algorithm	P@5	R@5	NDCG@5	P@5	R@5	NDCG@5	P@50	R@50	NDCG@50
BIPS-Cap	0.2808 $_{2e^{-3}}$	0.7576 $_{5e^{-3}}$	0.6099 $_{8e^{-3}}$	0.2758 $_{6e^{-3}}$	0.4582 $_{7e^{-3}}$	0.4399 $_{9e^{-3}}$	0.8750 $_{3e^{-3}}$	0.0238 $_{7e^{-5}}$	0.8788 $_{5e^{-3}}$
UIPS	0.2868 $_{2e^{-3}}$	0.7742 $_{5e^{-3}}$	0.6274 $_{5e^{-3}}$	0.2877 $_{3e^{-3}}$	0.4757 $_{5e^{-3}}$	0.4576 $_{8e^{-3}}$	0.9120 $_{1e^{-3}}$	0.0250 $_{5e^{-5}}$	0.9174 $_{7e^{-4}}$
DR	0.2670 $_{2e^{-3}}$	0.7174 $_{6e^{-3}}$	0.5636 $_{6e^{-3}}$	0.2884 $_{3e^{-3}}$	0.4760 $_{5e^{-3}}$	0.4541 $_{5e^{-3}}$	0.8794 $_{1e^{-2}}$	0.0240 $_{5e^{-4}}$	0.8824 $_{2e^{-2}}$
MinVarDR	0.2272 $_{5e^{-3}}$	0.5989 $_{1e^{-2}}$	0.4525 $_{1e^{-2}}$	0.2704 $_{4e^{-3}}$	0.4434 $_{9e^{-3}}$	0.4137 $_{6e^{-3}}$	0.8640 $_{7e^{-3}}$	0.0235 $_{2e^{-4}}$	0.8657 $_{7e^{-3}}$
ShrinkageDR	0.2697 $_{2e^{-3}}$	0.7226 $_{6e^{-3}}$	0.5713 $_{5e^{-3}}$	0.2895 $_{4e^{-3}}$	0.4749 $_{6e^{-3}}$	0.4526 $_{6e^{-3}}$	0.8778 $_{2e^{-2}}$	0.0239 $_{5e^{-4}}$	0.8800 $_{2e^{-2}}$
UIPSDR	**0.2721** $_{1e^{-3}}$	**0.7294** $_{6e^{-3}}$	**0.5750** $_{5e^{-3}}$	**0.2946** $_{4e^{-3}}$	**0.4854** $_{8e^{-3}}$	**0.4647** $_{8e^{-3}}$	**0.8849** $_{1e^{-2}}$	**0.0242** $_{4e^{-4}}$	**0.8896** $_{1e^{-2}}$
p-value (UIPSDR)	$1e^{-2}$	$2e^{-2}$	$1e^{-1}$	$7e^{-3}$	$5e^{-3}$	$2e^{-3}$	$4e^{-1}$	$4e^{-1}$	$3e^{-1}$
p-value (UIPS)	$1e^{-12}$	$6e^{-14}$	$6e^{-15}$	$3e^{-1}$	$8e^{-1}$	$1e^{-1}$	$2e^{-6}$	$2e^{-6}$	$1e^{-3}$

UIPS and the best two baselines on off-policy evaluation estimator (i.e., MinVar and Shrinkage based on the results in Table 1 and Table 4) to doubly robust setting using the same imputation model.

Table 7 and Table 8 report the empirical performance of the learned policy on the synthetic datasets and three real-world datasets respectively. For ease of comparison, we also include the experiment results of BIPS-Cap and UIPS on each dataset in these two tables. Two p-values are also provided: (1) p-value (UIPSDR): The p-value under the t-test between UIPSDR and the best DR baseline on each dataset; (2) p-value (UIPS): The p-value under the t-test between UIPS and the best DR baseline on each dataset. From Table 7 and Table 8, we can first observe that DR cannot consistently outperform BIPS-Cap: It outperformed BIPS-Cap on the Coat and KuaiRec dataset, while achieving much worse performance on the synthetic datasets and Yahoo dataset. This is because the imputation model also plays a very important role in gradient calculation as shown in Eq.(12). Its accuracy greatly affects policy learning. When the imputation model is sufficiently accurate, for example, on the Coat dataset with only 300 actions, incorporating the DM estimator not only led to better performance of DR over IPS, but also improved performance of UIPSDR over UIPS. And in particular, in this situation UIPSDR performed better than DR with the gain being statistically significant. When the imputation model is not accurate enough, for example, on the KuaiRec dataset with a large action space but sparse reward feedback, DR is still worse than UIPS, and UIPSDR also performs worse than UIPS due to the distortion of the imputation model.

7.5 Difference against DICE-type algorithms.

The DICE line of work [26, 47, 9] is proposed for off-policy correction in the multi-step RL setting with the environment following the Markov Decision Process (MDP) assumption. Although the DICE line of work does not require the knowledge of the logging policy, it is fundamentally different from our work. Given a logged dataset $D := \mathsf{f}(s_t, a_t, r_{s_t,a_t}, s_{t+1})\mathsf{g}$ collected from an unknown logging policy $\beta\ (a_t \mathit{j} s_t)$, DICE-type algorithms propose to directly estimate the discounted stationary distribution correction $w_{\pi/D}(s_t, a_t) = \frac{d^\pi(s_t,a_t)}{d^D(s_t,a_t)}$ to replace the product-based off-policy correction weight $\prod_t \frac{\pi(a_t j s_t)}{\beta\ (a_t j s_t)}$, which suffers high variance due to the series of products. While the estimation of $w_{\pi/D}(s_t, a_t)$ is agnostic to the logging policy, it highly depends on two assumptions: (1) Environment follows an MDP, i.e., $s_{t+1} \quad T(s_t, a_t)$ with $T(\ ,\ )$ denoting the state transition function; (2) Each logged sample should be of a state-action-next-state tuple $(s_t, a_t, r_{s_t,a_t}, s_{t+1})$ that contains state transition information.

Table 9: Empirical performance of DICE-S on synthetic datasets.

	$\tau = 0.5$			$\tau = 1$			$\tau = 2$		
Algorithm	P@5	R@5	NDCG@5	P@5	R@5	NDCG@5	P@5	R@5	NDCG@5
BIPS-Cap	0.5515 $_{2e^{-3}}$	0.1553 $_{8e^{-4}}$	0.6031 $_{2e^{-3}}$	0.5526 $_{2e^{-3}}$	0.1561 $_{6e^{-4}}$	0.6016 $_{1e^{-3}}$	0.5409 $_{3e^{-3}}$	0.1529 $_{9e^{-4}}$	0.5901 $_{2e^{-3}}$
DICE-S	0.5416 $_{4e^{-3}}$	0.1520 $_{1e^{-3}}$	0.5968 $_{4e^{-3}}$	0.5508 $_{2e^{-3}}$	0.1553 $_{7e^{-4}}$	0.6010 $_{2e^{-3}}$	0.5403 $_{2e^{-3}}$	0.1526 $_{7e^{-4}}$	0.5903 $_{1e^{-3}}$
UIPS	0.5589 $_{3e^{-3}}$	0.1583 $_{9e^{-4}}$	0.6095 $_{3e^{-3}}$	0.5572 $_{2e^{-3}}$	0.1571 $_{8e^{-4}}$	0.6074 $_{2e^{-3}}$	0.5432 $_{3e^{-3}}$	0.1534 $_{8e^{-4}}$	0.5946 $_{2e^{-3}}$

Table 10: Empirical performance of DICE-S on real-world datasets.

	Yahoo			**Coat**			**KuaiRec**		
Algorithm	P@5	R@5	NDCG@5	P@5	R@5	NDCG@5	P@50	R@50	NDCG@50
BIPS-Cap	0.2808 $_{2e^{-3}}$	0.7576 $_{5e^{-3}}$	0.6099 $_{8e^{-3}}$	0.2758 $_{6e^{-3}}$	0.4582 $_{7e^{-3}}$	0.4399 $_{9e^{-3}}$	0.8750 $_{3e^{-3}}$	0.0238 $_{7e^{-5}}$	0.8788 $_{5e^{-3}}$
DICE-S	0.2618 $_{4e^{-3}}$	0.7010 $_{1e^{-2}}$	0.5627 $_{9e^{-3}}$	0.2686 $_{5e^{-3}}$	0.4422 $_{6e^{-3}}$	0.4265 $_{5e^{-3}}$	0.8842 $_{2e^{-3}}$	0.0241 $_{6e^{-5}}$	0.8908 $_{2e^{-3}}$
UIPS	0.2868 $_{2e^{-3}}$	0.7742 $_{5e^{-3}}$	0.6274 $_{5e^{-3}}$	0.2877 $_{3e^{-3}}$	0.4757 $_{5e^{-3}}$	0.4576 $_{8e^{-3}}$	0.9120 $_{1e^{-3}}$	0.0250 $_{5e^{-5}}$	0.9174 $_{7e^{-4}}$

We further inspected how DICE-type algorithms would work in the contextual bandit setting, which can be taken as a one-state MDP. We take DualDICE [1] as an example for illustration, and similar conclusions can be drawn for other DICE-type algorithms. We can easily derive that in the contextual bandit setting, DualDICE degenerates to the IPS estimator that approximates the unknown ground-truth logging policy $\beta(a|s)$ with its empirical estimate from the given logged dataset $D := \{(s_i, a_i, r_{s_i,a_i})|i \in [N]\}$, which inherits all limitations of IPS estimator, such as high variance on instances with low support in D.

More specifically, recall that DualDICE estimates the discounted stationary distribution correction by optimizing the following objective function (Eq.(8) in [26]):

$$\min_{x:S \times A \to \mathbb{R}} \quad \frac{1}{2} E_{(s,a) \sim d^D}[x(s,a)^2] \quad - E_{(s,a) \sim d^\pi}[x(s,a)],$$

with the optimizer $x(s,a) = w_{\pi/D}(s,a)$. In a multi-step RL setting, one cannot directly calculate the second expectation as $d^\pi(s,a)$ is inaccessible, thus DualDICE takes the change-of-variable trick to address the above optimization problem. However, in the contextual bandit setting, let $p(s)$ denote the state distribution, $d^\pi(s,a) = p(s)\pi(a|s)$ and $d^D(s,a) = p(s)\beta(a|s)$. The optimization problem can be rewritten as :

$$\min_{x:S \times A \to \mathbb{R}} \quad E_{s \sim p(s)} \left[\frac{1}{2} E_{a \sim \beta(a|s)}[x(s,a)^2] \quad - E_{a \sim \pi(a|s)}[x(s,a)] \right].$$

With the logged dataset D, the empirical estimate of above optimization problem is :

$$\min_{x:S \times A \to \mathbb{R}} \quad \sum_s \frac{N_s}{N} \left(\sum_a \frac{N_{s,a}}{2N_s} x(s,a)^2 \quad - \sum_a \pi(a|s) x(s,a) \right),$$

yielding $x(s,a) = \pi(a|s)N_s/N_{s,a}$, where N_s and $N_{s,a}$ denote the number of logged samples with $s_i = s$ and $(s_i = s, a_i = a)$ respectively. This is actually equivalent to the IPS estimator with $N_{s,a}/N_s$ as the estimate of $\beta(a|s)$. We refer to the above estimator as DICE-S, and Table 9 and Table 10 show the empirical performance of the policy learned through DICE-S on the synthetic and real-world datasets respectively. Similar as BIPS-Cap described in Section 4, we also clip propensity scores to control variance. One can observe from Table 9 and Table 10 that DICE-S performs worse than BIPS-Cap in all datasets except the KuaiRec dataset. Additionally, DICE-S underperforms compared to UIPS in all datasets. This is due to the fact that although DICE-S is unbiased, it only becomes accurate with numerous logged samples, and suffers high variance when logged samples are limited, resulting in its poor performance in our experiments.

7.6 Comparison against work on direct propensity estimation.

In addition to DICE, several other works [34, 21] propose methods to directly estimate the propensity ratio without requiring a behavior policy. These methods then use the propensity estimates to prioritize instances in the replay buffer for better TD learning. To compare its effectiveness, we also included a new baseline called IPS-LFIW, which implements the approach proposed in [1] to directly estimate the propensity ratio for off-policy learning.

The average performance and standard deviations of IPS-LFIW and UIPS on three synthetic datasets with different τ are reported in Table 11. Recall that smaller τ indicates a more skewed ground-truth logging policy. The p-value under the t-test between UIPS and IPS-LFIW is also provided to

Table 11: Empirical performance of IPS-LFIW and UIPS on synthetic datasets

	$\tau = 0.5$			$\tau = 1$			$\tau = 2$		
Algorithm	P@5	R@5	NDCG@5	P@5	R@5	NDCG@5	P@5	R@5	NDCG@5
IPS-LFIW	0.5542 $1e^{-3}$	0.1568 $4e^{-4}$	0.6033 $1e^{-3}$	0.5472 $1e^{-3}$	0.1549 $5e^{-4}$	0.5975 $1e^{-3}$	0.5255 $6e^{-3}$	0.1485 $1e^{-3}$	0.5769 $5e^{-3}$
UIPS	**0.5608** $2e^{-3}$	**0.1589** $8e^{-4}$	**0.6113** $3e^{-3}$	**0.5572** $2e^{-3}$	**0.1571** $8e^{-4}$	**0.6074** $2e^{-3}$	**0.5432** $3e^{-3}$	**0.1534** $8e^{-4}$	**0.5946** $2e^{-3}$
p-value	$2e^{-6}$	$5e^{-6}$	$6e^{-9}$	$2e^{-8}$	$2e^{-6}$	$4e^{-10}$	$4e^{-7}$	$8e^{-7}$	$1e^{-7}$

investigate the significance of improvements. Notably, UIPS consistently outperformed IPS-LFIW with statistically significant improvements. One major reason for the worse performance of IPS-LFIW is that it does not consider the accuracy of the estimated propensity ratio, in a direct analogy to failing to handle uncertainty in the estimated logging probabilities in existing IPS-type algorithms.

7.7 Comparison against distributionally robust off-policy evaluation and learning.

Our work is fundamentally different from the line of work on distributionally robust off-policy evaluation and learning [32, 17, 44]. This results in different objectives that guide the use of min-max optimization.

Specifically, the work in [32, 17, 44] assumes unknown changes exist between their training and deployment environments, such as user preference drift or unforeseen events during policy execution. Thus they choose to maximize the policy value (e.g., $\hat{V}_{\mathrm{IPS}}(\pi)$) in the worst environment within an uncertainty set around the training environment,

$$\max_{\pi} \min_{U} \quad \hat{V}_{\mathrm{IPS}}(\pi).$$

As a result, their uncertainty set is created by introducing a small perturbation to the training environment. For example, the work in [32, 17] searches the worst environment P_1 in the σ-close perturbed environments around the training environment P_0 (Eq.(1) in [17]):

$$U(\sigma) = \{P_1 : P_1 << P_0 \quad \text{and} \quad D_{KL}(P_1 || P_0) \quad \sigma\}.$$

And the work in [44] adversarially perturbs the known ground-truth logging policy $\pi_0(\cdot)$ in searching for the worst case (Eq.(4) in [44]):

$$U(\alpha) = \{\pi_u : \max_{a,x} \quad \max\{\frac{\pi_u(a|x)}{\pi_0(a|x)}, \frac{\pi_0(a|x)}{\pi_u(a|x)}\} \quad e^{\alpha}\}.$$

In contrast, UIPS assumes the training and deployment environments stay the same, but the ground-truth logging policy is unknown. To control the high bias and high variance caused by inaccurately and small estimated logging probabilities, UIPS explicitly models the uncertainty in the estimated logging policy by incorporating a per-sample weight $\phi_{\boldsymbol{x},a}$ as discussed in Eq.(4). In order to make $\hat{V}_{\mathrm{UIPS}}(\pi_{\boldsymbol{\vartheta}})$ as accurate as possible despite the unknown ground-truth logging policy $\beta(\cdot)$, UIPS solves a min-max optimization problem in Eq.(8). This optimization problem seeks to find the optimal $\phi_{\boldsymbol{x},a}$ that minimizes the upper bound of the mean squared error (MSE) of $\hat{V}_{\mathrm{UIPS}}$ to its ground-truth value, within an uncertainty set of the unknown ground-truth logging policy $\beta(\cdot)$. The closed-form solution for the min-max optimization is also derived as in Theorem 3.2.

Furthermore, observing that work in [44] also performs optimization over an uncertainty set of the logging policy, we further adapted their method to handle the inaccuracy of the estimated logging policy, by taking $\pi_0(\cdot)$ as the estimated logging policy, i.e., $\pi_0(\cdot) = \hat{\beta}(\cdot)$. We name the adapted methods as IPS-UN. Table 12 demonstrates the performance of the learned policy under IPS-UN on three synthetic datasets. The results suggest that directly applying IPS-UN to handle inaccurately estimated logging probabilities is not be a feasible solution to our problem. One important reason for the worse performance of IPS-UN is that it strives to optimize for the worst potential environment, which might not be the case in our experiment datatsets. On the other hand, UIPS assumes the training and deployment environments stay same and strives to identify the optimal policy with an unknown ground-truth logging policy.

7.8 Theoretical Proofs.

Proof of Proposition 2.1:

Table 12: Empirical performance of IPS-UN on synthetic datasets.

	$\tau = 0.5$			$\tau = 1$			$\tau = 2$		
Algorithm	P@5	R@5	NDCG@5	P@5	R@5	NDCG@5	P@5	R@5	NDCG@5
BIPS-Cap	0.5515 $_{2e^{-3}}$	0.1553 $_{8e^{-4}}$	0.6031 $_{2e^{-3}}$	0.5526 $_{2e^{-3}}$	0.1561 $_{6e^{-4}}$	0.6016 $_{1e^{-3}}$	0.5409 $_{3e^{-3}}$	0.1529 $_{9e^{-4}}$	0.5901 $_{2e^{-3}}$
IPS-UN	0.4089 $_{3e^{-2}}$	0.1152 $_{8e^{-3}}$	0.4916 $_{2e^{-2}}$	0.3911 $_{3e^{-3}}$	0.1100 $_{9e^{-3}}$	0.4754 $_{3e^{-2}}$	0.3599 $_{4e^{-2}}$	0.1009 $_{1e^{-2}}$	0.4353 $_{4e^{-2}}$
UIPS	0.5589 $_{3e^{-3}}$	0.1583 $_{9e^{-4}}$	0.6095 $_{3e^{-3}}$	0.5572 $_{2e^{-3}}$	0.1571 $_{8e^{-4}}$	0.6074 $_{2e^{-3}}$	0.5432 $_{3e^{-3}}$	0.1534 $_{8e^{-4}}$	0.5946 $_{2e^{-3}}$

Proof. Because of the linearity of expectation, we have $\mathsf{E}_D\left[\hat{V}_{\mathrm{BIPS}}(\pi_{\boldsymbol{\vartheta}})\right] = \mathsf{E}_\beta\left[\frac{\pi_{\boldsymbol{\vartheta}}(aj\boldsymbol{x})}{\hat{\beta}(aj\boldsymbol{x})} r_{\boldsymbol{x},a}\right]$, and thus:

$$\begin{aligned}
\mathrm{Bias}\left(\hat{\mathrm{V}}_{\mathrm{BIPS}}(\pi_{\boldsymbol{\vartheta}})\right) &= \mathsf{E}_D\left[\hat{V}_{\mathrm{BIPS}}(\pi_{\boldsymbol{\vartheta}}) \quad V(\pi_{\boldsymbol{\vartheta}})\right] \\
&= \mathsf{E}_\beta\left[\frac{\pi_{\boldsymbol{\vartheta}}(aj\boldsymbol{x})}{\hat{\beta}(aj\boldsymbol{x})} r_{\boldsymbol{x},a}\right] \quad \mathsf{E}_\beta\left[\frac{\pi_{\boldsymbol{\vartheta}}(aj\boldsymbol{x})}{\beta\ (aj\boldsymbol{x})} r_{\boldsymbol{x},a}\right] \\
&= \mathsf{E}_\beta\left[\frac{\pi_{\boldsymbol{\vartheta}}(aj\boldsymbol{x})}{\beta\ (aj\boldsymbol{x})} r_{\boldsymbol{x},a}\left(\frac{\beta\ (aj\boldsymbol{x})}{\hat{\beta}(aj\boldsymbol{x})} \quad 1\right)\right] \\
&= \mathsf{E}_{\pi_{\boldsymbol{\vartheta}}}\left[r_{\boldsymbol{x},a}\left(\frac{\beta\ (aj\boldsymbol{x})}{\hat{\beta}(aj\boldsymbol{x})} \quad 1\right)\right]. \qquad (13)
\end{aligned}$$

Since samples are independently sampled from logging policy, the variance can be computed as:

$$\mathrm{Var}_D\left(\hat{V}_{\mathrm{BIPS}}(\pi_{\boldsymbol{\vartheta}})\right) = \frac{1}{N}\mathrm{Var}_\beta\left(\frac{\pi_{\boldsymbol{\vartheta}}(aj\boldsymbol{x})}{\hat{\beta}(aj\boldsymbol{x})} r_{\boldsymbol{x},a}\right).$$

By re-scaling, we get:

$$\begin{aligned}
N \quad \mathrm{Var}_D\left(\hat{V}_{\mathrm{BIPS}}(\pi_{\boldsymbol{\vartheta}})\right) &= \mathrm{Var}_\beta\left(\frac{\pi_{\boldsymbol{\vartheta}}(aj\boldsymbol{x})}{\hat{\beta}(aj\boldsymbol{x})} r_{\boldsymbol{x},a}\right) \qquad (14) \\
&= \mathsf{E}_\beta\left[\frac{\pi_{\boldsymbol{\vartheta}}(aj\boldsymbol{x})^2}{\hat{\beta}(aj\boldsymbol{x})^2} r_{\boldsymbol{x},a}^2\right] \quad \left(\mathsf{E}_\beta\left[\frac{\pi_{\boldsymbol{\vartheta}}(aj\boldsymbol{x})}{\hat{\beta}(aj\boldsymbol{x})} r_{\boldsymbol{x},a}\right]\right)^2 \\
&= \mathsf{E}_{\pi_{\boldsymbol{\vartheta}}}\left[\frac{\pi_{\boldsymbol{\vartheta}}(aj\boldsymbol{x})}{\beta\ (aj\boldsymbol{x})} \frac{\beta\ (aj\boldsymbol{x})^2}{\hat{\beta}(aj\boldsymbol{x})^2} r_{\boldsymbol{x},a}^2\right] \quad \left(\mathsf{E}_{\pi_{\boldsymbol{\vartheta}}}\left[\frac{\beta\ (aj\boldsymbol{x})}{\hat{\beta}(aj\boldsymbol{x})} r_{\boldsymbol{x},a}\right]\right)^2
\end{aligned}$$

This completes the proof. □

Proof of Theorem 3.1:

Proof. We can get:

$$\begin{aligned}
\mathrm{MSE}\left(\hat{V}_{\mathrm{UIPS}}(\pi_{\boldsymbol{\vartheta}})\right) &= \mathsf{E}_D\left[\left(\hat{V}_{\mathrm{UIPS}}(\pi_{\boldsymbol{\vartheta}}) \quad V(\pi_{\boldsymbol{\vartheta}})\right)^2\right] \\
&= \left(\mathsf{E}_D\left[\hat{V}_{\mathrm{UIPS}}(\pi_{\boldsymbol{\vartheta}}) \quad V(\pi_{\boldsymbol{\vartheta}})\right]\right)^2 + \mathrm{Var}_D\left(\hat{V}_{\mathrm{UIPS}}(\pi_{\boldsymbol{\vartheta}}) \quad V(\pi_{\boldsymbol{\vartheta}})\right) \\
&= \left(\mathsf{E}_D\left[\hat{V}_{\mathrm{UIPS}}(\pi_{\boldsymbol{\vartheta}}) \quad V(\pi_{\boldsymbol{\vartheta}})\right]\right)^2 + \mathrm{Var}_D\left(\hat{V}_{\mathrm{UIPS}}(\pi_{\boldsymbol{\vartheta}})\right) \\
&= \mathrm{Bias}(\hat{V}_{\mathrm{UIPS}}(\pi_{\boldsymbol{\vartheta}}))^2 + \mathrm{Var}(\hat{V}_{\mathrm{UIPS}}(\pi_{\boldsymbol{\vartheta}})).
\end{aligned}$$

We first bound the bias term:

$$\begin{aligned}
\mathrm{Bias}(\hat{V}_{\mathrm{UIPS}}(\pi_{\boldsymbol{\vartheta}})) &= \mathsf{E}_D\left[\hat{V}_{\mathrm{UIPS}}(\pi_{\boldsymbol{\vartheta}}) \quad V(\pi_{\boldsymbol{\vartheta}})\right] \\
&\overset{(1)}{=} \mathsf{E}_\beta\left[\frac{\pi_{\boldsymbol{\vartheta}}(aj\boldsymbol{x})}{\hat{\beta}(aj\boldsymbol{x})}\phi_{\boldsymbol{x},a} r_{\boldsymbol{x},a}\right] \quad V(\pi_{\boldsymbol{\vartheta}}) \\
&= \mathsf{E}_\beta\left[\frac{\pi_{\boldsymbol{\vartheta}}(aj\boldsymbol{x})}{\hat{\beta}(aj\boldsymbol{x})}\phi_{\boldsymbol{x},a} r_{\boldsymbol{x},a} \quad \frac{\pi_{\boldsymbol{\vartheta}}(aj\boldsymbol{x})}{\beta\ (aj\boldsymbol{x})} r_{\boldsymbol{x},a}\right] \\
&= \mathsf{E}_\beta\left[r_{\boldsymbol{x},a}\frac{\pi_{\boldsymbol{\vartheta}}(aj\boldsymbol{x})}{\beta\ (aj\boldsymbol{x})} \quad \left(\frac{\beta\ (aj\boldsymbol{x})}{\hat{\beta}(aj\boldsymbol{x})}\phi_{\boldsymbol{x},a} \quad 1\right)\right] \\
&\overset{(2)}{\sqrt{\mathsf{E}_{\pi_{\boldsymbol{\vartheta}}}\left[r^2_{\boldsymbol{x},a}\frac{\pi_{\boldsymbol{\vartheta}}(aj\boldsymbol{x})}{\beta\ (aj\boldsymbol{x})}\right]}} \sqrt{\mathsf{E}_\beta\left[\left(\frac{\beta\ (aj\boldsymbol{x})}{\hat{\beta}(aj\boldsymbol{x})}\phi_{\boldsymbol{x},a} \quad 1\right)^2\right]}
\end{aligned}$$

Step (1) follows the linearity of expectation and step (2) is due to the Cauchy-Schwarz inequality. We then bound the variance term:

$$\begin{aligned}
\mathrm{Var}(\hat{V}_{\mathrm{UIPS}}(\pi_{\boldsymbol{\vartheta}})) &= \frac{1}{N}\mathrm{Var}_\beta\left(\frac{\pi_{\boldsymbol{\vartheta}}(aj\boldsymbol{x})}{\hat{\beta}(aj\boldsymbol{x})}\phi_{\boldsymbol{x},a} r_{\boldsymbol{x},a}\right) \\
&= \frac{1}{N}\left(\mathsf{E}_\beta\left[\frac{\pi_{\boldsymbol{\vartheta}}(aj\boldsymbol{x})^2}{\hat{\beta}(aj\boldsymbol{x})^2}\phi^2_{\boldsymbol{x},a} r^2_{\boldsymbol{x},a}\right] \quad \left(\mathsf{E}_\beta\left[\frac{\pi_{\boldsymbol{\vartheta}}(aj\boldsymbol{x})}{\hat{\beta}(aj\boldsymbol{x})}\phi_{\boldsymbol{x},a} r_{\boldsymbol{x},a}\right]\right)^2\right) \\
&\quad \frac{1}{N}\mathsf{E}_\beta\left[\frac{\pi_{\boldsymbol{\vartheta}}(aj\boldsymbol{x})^2}{\hat{\beta}(aj\boldsymbol{x})^2}\phi^2_{\boldsymbol{x},a} r^2_{\boldsymbol{x},a}\right] \quad \mathsf{E}_\beta\left[\frac{\pi_{\boldsymbol{\vartheta}}(aj\boldsymbol{x})^2}{\hat{\beta}(aj\boldsymbol{x})^2}\phi^2_{\boldsymbol{x},a}\right]
\end{aligned}$$

Combining the bound of bias and variance completes the proof. □

Proof of Theorem 3.2:

Proof. We first define several notations:

- $T(\phi_{\boldsymbol{x},a}, \beta_{\boldsymbol{x},a}) = \lambda \mathsf{E}_\beta\left[\left(\frac{\beta_{\boldsymbol{x},a}}{\hat{\beta}(aj\boldsymbol{x})}\phi_{\boldsymbol{x},a} \quad 1\right)^2\right] + \mathsf{E}_\beta\left[\frac{\pi_{\boldsymbol{\vartheta}}(aj\boldsymbol{x})^2}{\hat{\beta}(aj\boldsymbol{x})^2}\phi^2_{\boldsymbol{x},a}\right]$.
- $\tilde{T}(\phi_{\boldsymbol{x},a}) = \max_{\beta_{\boldsymbol{x},a}\, 2\, \boldsymbol{B}_{\boldsymbol{x},a}} T(\phi_{\boldsymbol{x},a}, \beta_{\boldsymbol{x},a})$ denotes the maximum value of inner optimization problem.
- $T \ = \min_{\phi_{\boldsymbol{x},a}} \tilde{T}(\phi_{\boldsymbol{x},a}) = \min_{\phi_{\boldsymbol{x},a}} \max_{\beta_{\boldsymbol{x},a}\, 2\, \boldsymbol{B}_{\boldsymbol{x},a}} T(\phi_{\boldsymbol{x},a}, \beta_{\boldsymbol{x},a})$ denote the optimal min-max value. And $\phi_{\boldsymbol{x},a} = \arg\min_{\phi_{\boldsymbol{x},a}} \tilde{T}(\phi_{\boldsymbol{x},a})$.
- $\boldsymbol{B}_{\boldsymbol{x},a} := \frac{\hat{Z}\exp(\ \gamma U_{\boldsymbol{x},a})}{Z}\hat{\beta}(aj\boldsymbol{x})$, and $\boldsymbol{B}^+_{\boldsymbol{x},a} := \frac{\hat{Z}\exp(\gamma U_{\boldsymbol{x},a})}{Z}\hat{\beta}(aj\boldsymbol{x})$.

We first find the maximum value of the inner optimization problem, i.e., $\tilde{T}(\phi_{\boldsymbol{x},a})$ for any fixed $\phi_{\boldsymbol{x},a}$. And there are three cases shown in Figure 4:

Case I: When $\frac{\hat{\beta}(aj\boldsymbol{x})}{\phi_{\boldsymbol{x},a}} \quad \boldsymbol{B}^+_{\boldsymbol{x},a}$, , $\tilde{T}(\phi_{\boldsymbol{x},a})$ achieves the maximum value at $\beta_{\boldsymbol{x},a} = \boldsymbol{B}_{\boldsymbol{x},a}$. In other words, $\tilde{T}(\phi_{\boldsymbol{x},a}) = T(\phi_{\boldsymbol{x},a}, \boldsymbol{B}_{\boldsymbol{x},a})$ when $\phi_{\boldsymbol{x},a} \quad \frac{\hat{\beta}(aj\boldsymbol{x})}{\boldsymbol{B}^+_{\boldsymbol{x},a}}$.

Case II: When $\boldsymbol{B}_{\boldsymbol{x},a} \quad \frac{\hat{\beta}(aj\boldsymbol{x})}{\phi_{\boldsymbol{x},a}} \quad \boldsymbol{B}^+_{\boldsymbol{x},a}$, i.e., $\frac{Z\ \exp(\ \gamma U_{\boldsymbol{x},a})}{\hat{Z}} \quad \phi_{\boldsymbol{x},a} \quad \frac{Z\ \exp(\gamma U_{\boldsymbol{x},a})}{\hat{Z}}$, then $\tilde{T}(\phi_{\boldsymbol{x},a})$ will be the maximum between $T(\phi_{\boldsymbol{x},a}, \boldsymbol{B}_{\boldsymbol{x},a})$ and $T(\phi_{\boldsymbol{x},a}, \boldsymbol{B}^+_{\boldsymbol{x},a})$.

More specifically, when $\frac{\hat{\beta}(aj\boldsymbol{x})}{\phi_{\boldsymbol{x},a}} \quad \frac{\boldsymbol{B}^+_{\boldsymbol{x},a}+\boldsymbol{B}_{\boldsymbol{x},a}}{2}$, i.e., $\phi_{\boldsymbol{x},a} \quad \frac{2\hat{\beta}(aj\boldsymbol{x})}{\boldsymbol{B}^+_{\boldsymbol{x},a}+\boldsymbol{B}_{\boldsymbol{x},a}}$, $\tilde{T}(\phi_{\boldsymbol{x},a}) = T(\phi_{\boldsymbol{x},a}, \boldsymbol{B}^+_{\boldsymbol{x},a})$. Otherwise when $\phi_{\boldsymbol{x},a} < \frac{2\hat{\beta}(aj\boldsymbol{x})}{\boldsymbol{B}^+_{\boldsymbol{x},a}+\boldsymbol{B}_{\boldsymbol{x},a}}$, $\tilde{T}(\phi_{\boldsymbol{x},a}) = T(\phi_{\boldsymbol{x},a}, \boldsymbol{B}_{\boldsymbol{x},a})$.

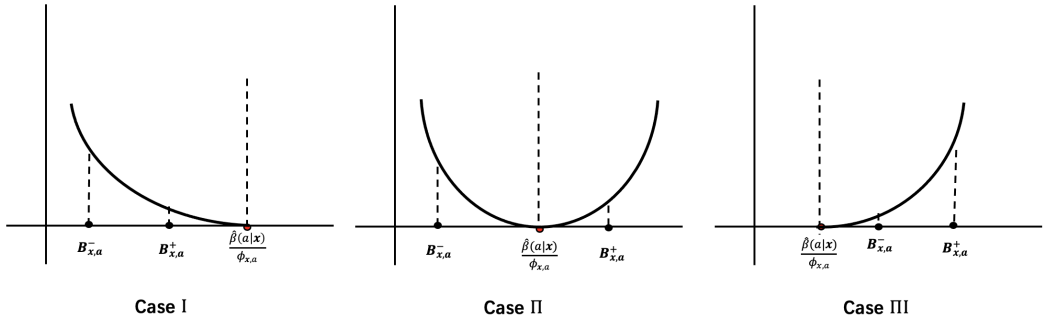


Figure 4: Three cases for maximizing the inner optimization problem.

Case III: When $\phi_{\boldsymbol{x},a}$ $\frac{\hat{\beta}(aj\boldsymbol{x})}{\boldsymbol{B}_{\boldsymbol{x},a}}$. implying $\frac{\hat{\beta}(aj\boldsymbol{x})}{\phi_{\boldsymbol{x},a}}$ $\boldsymbol{B}_{\boldsymbol{x},a}, \tilde{T}(\phi_{\boldsymbol{x},a}) = T(\phi_{\boldsymbol{x},a}, \boldsymbol{B}^{+}_{\boldsymbol{x},a})$.

Overall, we get that:

$$\tilde{T}(\phi_{\boldsymbol{x},a}) = \begin{cases} T(\phi_{\boldsymbol{x},a}, \boldsymbol{B}_{\boldsymbol{x},a}), & \phi_{\boldsymbol{x},a} \in (\quad 1\ , \frac{2\hat{\beta}(aj\boldsymbol{x})}{\boldsymbol{B}^{+}_{\boldsymbol{x},a}+\boldsymbol{B}_{\boldsymbol{x},a}}] \\ T(\phi_{\boldsymbol{x},a}, \boldsymbol{B}^{+}_{\boldsymbol{x},a}) & \phi_{\boldsymbol{x},a} \in [\frac{2\hat{\beta}(aj\boldsymbol{x})}{\boldsymbol{B}^{+}_{\boldsymbol{x},a}+\boldsymbol{B}_{\boldsymbol{x},a}},\ 1\ ) \end{cases} \tag{15}$$

Next we try to find the minimum value of $\tilde{T}(\phi_{\boldsymbol{x},a})$. We first observe that without considering constraint on $\phi_{\boldsymbol{x},a}$, when

$$\phi^{+}_{\boldsymbol{x},a} = \frac{\lambda}{\lambda\frac{\boldsymbol{B}^{+}_{\boldsymbol{x},a}}{\hat{\beta}(aj\boldsymbol{x})} + \frac{\pi_{\vartheta}(aj\boldsymbol{x})^2}{\hat{\beta}(aj\boldsymbol{x})\boldsymbol{B}^{+}_{\boldsymbol{x},a}}}$$

$T(\phi_{\boldsymbol{x},a}, \boldsymbol{B}^{+}_{\boldsymbol{x},a})$ achieves the global minimum value. However, $\phi^{+}_{\boldsymbol{x},a}$ $\frac{\hat{\beta}(aj\boldsymbol{x})}{\boldsymbol{B}^{+}_{\boldsymbol{x},a}}$ $\frac{2\hat{\beta}(aj\boldsymbol{x})}{\boldsymbol{B}^{+}_{\boldsymbol{x},a}+\boldsymbol{B}_{\boldsymbol{x},a}}$, which implies when $\phi_{\boldsymbol{x},a} \in \left[\frac{2\hat{\beta}(aj\boldsymbol{x})}{\boldsymbol{B}^{+}_{\boldsymbol{x},a}+\boldsymbol{B}_{\boldsymbol{x},a}},\ 1\right)$, the minimum value of $T(\phi_{\boldsymbol{x},a}, \boldsymbol{B}^{+}_{\boldsymbol{x},a})$ is achieved at $\frac{2\hat{\beta}(aj\boldsymbol{x})}{\boldsymbol{B}^{+}_{\boldsymbol{x},a}+\boldsymbol{B}_{\boldsymbol{x},a}}$.

On the other hand, without considering any constraint on $\phi_{\boldsymbol{x},a}$, the global minimum value of $T(\phi_{\boldsymbol{x},a}, \boldsymbol{B}_{\boldsymbol{x},a})$ is achieved at:

$$\phi_{\boldsymbol{x},a} = \frac{\lambda}{\lambda\frac{\boldsymbol{B}_{\boldsymbol{x},a}}{\hat{\beta}(aj\boldsymbol{x})} + \frac{\pi_{\vartheta}(aj\boldsymbol{x})^2}{\hat{\beta}(aj\boldsymbol{x})\boldsymbol{B}_{\boldsymbol{x},a}}}. \tag{16}$$

Thus if $\phi_{\boldsymbol{x},a}$ $\frac{2\hat{\beta}(aj\boldsymbol{x})}{\boldsymbol{B}^{+}_{\boldsymbol{x},a}+\boldsymbol{B}_{\boldsymbol{x},a}}$, $\phi_{\boldsymbol{x},a} = \phi_{\boldsymbol{x},a}$, since $T(\phi_{\boldsymbol{x},a}, \boldsymbol{B}_{\boldsymbol{x},a})$ $T(\frac{2\hat{\beta}(aj\boldsymbol{x})}{\boldsymbol{B}^{+}_{\boldsymbol{x},a}+\boldsymbol{B}_{\boldsymbol{x},a}}, \boldsymbol{B}_{\boldsymbol{x},a}) = T(\frac{2\hat{\beta}(aj\boldsymbol{x})}{\boldsymbol{B}^{+}_{\boldsymbol{x},a}+\boldsymbol{B}_{\boldsymbol{x},a}}, \boldsymbol{B}^{+}_{\boldsymbol{x},a})$. Otherwise, when $\phi_{\boldsymbol{x},a} > \frac{2\hat{\beta}(aj\boldsymbol{x})}{\boldsymbol{B}^{+}_{\boldsymbol{x},a}+\boldsymbol{B}_{\boldsymbol{x},a}}$, the minimum value of $T(\phi_{\boldsymbol{x},a}, \boldsymbol{B}_{\boldsymbol{x},a})$ is also achieved at $\frac{2\hat{\beta}(aj\boldsymbol{x})}{\boldsymbol{B}^{+}_{\boldsymbol{x},a}+\boldsymbol{B}_{\boldsymbol{x},a}}$, implying $\phi_{\boldsymbol{x},a} = \frac{2\hat{\beta}(aj\boldsymbol{x})}{\boldsymbol{B}^{+}_{\boldsymbol{x},a}+\boldsymbol{B}_{\boldsymbol{x},a}}$.

Overall,

$$\phi_{\boldsymbol{x},a} = \min\left(\frac{\lambda}{\lambda\frac{\boldsymbol{B}_{\boldsymbol{x},a}}{\hat{\beta}(aj\boldsymbol{x})} + \frac{\pi_{\vartheta}(aj\boldsymbol{x})^2}{\hat{\beta}(aj\boldsymbol{x})\boldsymbol{B}_{\boldsymbol{x},a}}}, \frac{2\hat{\beta}(aj\boldsymbol{x})}{\boldsymbol{B}^{+}_{\boldsymbol{x},a}+\boldsymbol{B}_{\boldsymbol{x},a}}\right) \tag{17}$$

Let $\eta = \frac{Z}{\hat{Z}}$, we can get

$$\phi_{\boldsymbol{x},a} = \min\left(\frac{\lambda}{\frac{\lambda}{\eta}\exp(\quad\gamma U_{\boldsymbol{x},a}) + \frac{\eta\pi_{\vartheta}(aj\boldsymbol{x})^2}{\hat{\beta}(aj\boldsymbol{x})^2\exp(\quad\gamma U_{\boldsymbol{x},a})}}, \frac{2\eta}{\exp(\gamma U_{\boldsymbol{x},a}) + \exp(\quad\gamma U_{\boldsymbol{x},a})}\right). \tag{18}$$

Note that η 2 $[\exp(\ \gamma U_s^{\max}), \exp(\gamma U_s^{\max})]$, since $\hat{Z}\exp(\ \gamma U_s^{\max})$ Z $=$ $\sum_{a^0}\exp(f_\theta\ (a^0 j\boldsymbol{x}))$ $\hat{Z}\exp(U_s^{\max})$.

This completes the proof . □

Proof of Corollary 3.3:

Proof. Following the similar procedure as in Theorem 3.1, we can derive the mean squared error (MSE) between $\hat{V}_{\mathrm{BIPS}}(\pi_{\boldsymbol{\vartheta}})$ and ground-truth estimator $V(\pi_{\boldsymbol{\vartheta}})$ is upper bounded as follows:

$$\mathrm{MSE}\left(\hat{V}_{\mathrm{BIPS}}(\pi_{\boldsymbol{\vartheta}})\right) = \mathsf{E}_D\left[\left(\hat{V}_{\mathrm{BIPS}}(\pi_{\boldsymbol{\vartheta}}) \quad V(\pi_{\boldsymbol{\vartheta}})\right)^2\right]$$
$$\mathsf{E}_{\pi_{\boldsymbol{\vartheta}}}\left[r_{\boldsymbol{x},a}^2 \frac{\pi_{\boldsymbol{\vartheta}}(aj\boldsymbol{x})}{\beta\ (aj\boldsymbol{x})}\right]\ \mathsf{E}_\beta\ \left[\left(\frac{\beta\ (aj\boldsymbol{x})}{\hat{\beta}(aj\boldsymbol{x})} \quad 1\right)^2\right] + \mathsf{E}_\beta\ \left[\frac{\pi_{\boldsymbol{\vartheta}}(aj\boldsymbol{x})^2}{\hat{\beta}(aj\boldsymbol{x})^2}\right].$$

When $\beta\ (aj\boldsymbol{x})$ 2 $[\boldsymbol{B}_{\boldsymbol{x},a}, \boldsymbol{B}^+_{\boldsymbol{x},a}]$, with $\boldsymbol{B}_{\boldsymbol{x},a}$ $:=$ $\frac{\hat{Z}\exp(\ \gamma U_{\boldsymbol{x},a})}{Z}\hat{\beta}(aj\boldsymbol{x})$ and $\boldsymbol{B}^+_{\boldsymbol{x},a}$ $:=$ $\frac{\hat{Z}\exp(\gamma U_{\boldsymbol{x},a})}{Z}\hat{\beta}(aj\boldsymbol{x})$, $\mathrm{MSE}\left(\hat{V}_{\mathrm{BIPS}}(\pi_{\boldsymbol{\vartheta}})\right)$ can be further upper bounded as follows.

For ease of illustration, we set $\lambda = \mathsf{E}_{\pi_{\boldsymbol{\vartheta}}}\left[r_{\boldsymbol{x},a}^2 \frac{\pi_{\boldsymbol{\vartheta}}(aj\boldsymbol{x})}{\beta\ (aj\boldsymbol{x})}\right]$ and

$$T(\phi_{\boldsymbol{x},a}, \beta_{\boldsymbol{x},a}) = \lambda\mathsf{E}_\beta\ \left[\left(\frac{\beta_{\boldsymbol{x},a}}{\hat{\beta}(aj\boldsymbol{x})}\phi_{\boldsymbol{x},a} \quad 1\right)^2\right] + \mathsf{E}_\beta\ \left[\frac{\pi_{\boldsymbol{\vartheta}}(aj\boldsymbol{x})^2}{\hat{\beta}(aj\boldsymbol{x})^2}\phi_{\boldsymbol{x},a}^2\right].$$

Let T_{BIPS} denote the upper bound of $\mathrm{MSE}\left(\hat{V}_{\mathrm{BIPS}}(\pi_{\boldsymbol{\vartheta}})\right)$, then we can derive that

$$T_{\mathrm{BIPS}} = \begin{cases} T(1, \boldsymbol{B}^+_{\boldsymbol{x},a}), & \hat{\beta}(aj\boldsymbol{x}) \quad \frac{\boldsymbol{B}^+_{\boldsymbol{x},a}+\boldsymbol{B}_{\boldsymbol{x},a}}{2} \\ T(1, \boldsymbol{B}_{\boldsymbol{x},a}), & \hat{\beta}(aj\boldsymbol{x}) > \frac{\boldsymbol{B}^+_{\boldsymbol{x},a}+\boldsymbol{B}_{\boldsymbol{x},a}}{2}. \end{cases} \tag{19}$$

Recall that in Theorem 3.2, we show that the upper bound of $\mathrm{MSE}\left(\hat{V}_{\mathrm{UIPS}}(\pi_{\boldsymbol{\vartheta}})\right)$ is as follows:

$$T_{\mathrm{UIPS}} = \begin{cases} T(\phi_{\boldsymbol{x},a}, \boldsymbol{B}_{\boldsymbol{x},a}), & \phi_{\boldsymbol{x},a} < \frac{2\hat{\beta}(aj\boldsymbol{x})}{\boldsymbol{B}^+_{\boldsymbol{x},a}+\boldsymbol{B}_{\boldsymbol{x},a}} \\ T(\frac{2\hat{\beta}(aj\boldsymbol{x})}{\boldsymbol{B}^+_{\boldsymbol{x},a}+\boldsymbol{B}_{\boldsymbol{x},a}}, \boldsymbol{B}^+_{\boldsymbol{x},a}), & \phi_{\boldsymbol{x},a} \quad \frac{2\hat{\beta}(aj\boldsymbol{x})}{\boldsymbol{B}^+_{\boldsymbol{x},a}+\boldsymbol{B}_{\boldsymbol{x},a}}. \end{cases} \tag{20}$$

where $\phi_{\boldsymbol{x},a}$ is defined in Eq.(16). Next we show that T_{UIPS} T_{BIPS}.

- When $\hat{\beta}(aj\boldsymbol{x})$ $\frac{\boldsymbol{B}^+_{\boldsymbol{x},a}+\boldsymbol{B}_{\boldsymbol{x},a}}{2}$, we can get $T_{\mathrm{BIPS}} = T(1, \boldsymbol{B}^+_{\boldsymbol{x},a})$ and $\frac{2\hat{\beta}(aj\boldsymbol{x})}{\boldsymbol{B}^+_{\boldsymbol{x},a}+\boldsymbol{B}_{\boldsymbol{x},a}}$ 1. If $\phi_{\boldsymbol{x},a} < \frac{2\hat{\beta}(aj\boldsymbol{x})}{\boldsymbol{B}^+_{\boldsymbol{x},a}+\boldsymbol{B}_{\boldsymbol{x},a}}$, then $T_{\mathrm{UIPS}} = T(\phi_{\boldsymbol{x},a}, \boldsymbol{B}_{\boldsymbol{x},a}) < T(\frac{2\hat{\beta}(aj\boldsymbol{x})}{\boldsymbol{B}^+_{\boldsymbol{x},a}+\boldsymbol{B}_{\boldsymbol{x},a}}, \boldsymbol{B}_{\boldsymbol{x},a}) = T(\frac{2\hat{\beta}(aj\boldsymbol{x})}{\boldsymbol{B}^+_{\boldsymbol{x},a}+\boldsymbol{B}_{\boldsymbol{x},a}}, \boldsymbol{B}^+_{\boldsymbol{x},a})$ $T(1, \boldsymbol{B}^+_{\boldsymbol{x},a})$. And if $\phi_{\boldsymbol{x},a}$ $\frac{2\hat{\beta}(aj\boldsymbol{x})}{\boldsymbol{B}^+_{\boldsymbol{x},a}+\boldsymbol{B}_{\boldsymbol{x},a}}$, $T_{\mathrm{UIPS}} = T(\frac{2\hat{\beta}(aj\boldsymbol{x})}{\boldsymbol{B}^+_{\boldsymbol{x},a}+\boldsymbol{B}_{\boldsymbol{x},a}}, \boldsymbol{B}^+_{\boldsymbol{x},a})$ $T(1, \boldsymbol{B}^+_{\boldsymbol{x},a})$;
- When $\hat{\beta}(aj\boldsymbol{x}) > \frac{\boldsymbol{B}^+_{\boldsymbol{x},a}+\boldsymbol{B}_{\boldsymbol{x},a}}{2}$, we can get $T_{\mathrm{BIPS}} = T(1, \boldsymbol{B}_{\boldsymbol{x},a})$ and $\frac{2\hat{\beta}(aj\boldsymbol{x})}{\boldsymbol{B}^+_{\boldsymbol{x},a}+\boldsymbol{B}_{\boldsymbol{x},a}} > 1$. If $\phi_{\boldsymbol{x},a} < \frac{2\hat{\beta}(aj\boldsymbol{x})}{\boldsymbol{B}^+_{\boldsymbol{x},a}+\boldsymbol{B}_{\boldsymbol{x},a}}$, then $T_{\mathrm{UIPS}} = T(\phi_{\boldsymbol{x},a}, \boldsymbol{B}_{\boldsymbol{x},a})$ $T(1, \boldsymbol{B}_{\boldsymbol{x},a})$, since $\phi_{\boldsymbol{x},a}$ is a global minimum. Otherwise when $\phi_{\boldsymbol{x},a}$ $\frac{2\hat{\beta}(aj\boldsymbol{x})}{\boldsymbol{B}^+_{\boldsymbol{x},a}+\boldsymbol{B}_{\boldsymbol{x},a}} > 1$, $T_{\mathrm{UIPS}} = T(\frac{2\hat{\beta}(aj\boldsymbol{x})}{\boldsymbol{B}^+_{\boldsymbol{x},a}+\boldsymbol{B}_{\boldsymbol{x},a}}, \boldsymbol{B}^+_{\boldsymbol{x},a}) = T(\frac{2\hat{\beta}(aj\boldsymbol{x})}{\boldsymbol{B}^+_{\boldsymbol{x},a}+\boldsymbol{B}_{\boldsymbol{x},a}}, \boldsymbol{B}_{\boldsymbol{x},a}) < T(1, \boldsymbol{B}_{\boldsymbol{x},a})$.

In both cases, we have T_{UIPS} T_{BIPS}, thus completing the proof.

□

Lemma 7.1. *Under fixed $\pi_{\vartheta}(a|\boldsymbol{x})$ and $\hat{\beta}(a|\boldsymbol{x})$, and $\alpha_{\boldsymbol{x},a} = \sqrt{\frac{\lambda}{2\eta^2} \quad \frac{\lambda(1 \quad \eta)}{\eta^2} \exp(\quad 2\gamma U_{\boldsymbol{x},a})}$, we have the following observations:*

- *If $\frac{\pi_{\vartheta}(a|\boldsymbol{x})}{\hat{\beta}(a|\boldsymbol{x})} \qquad \alpha_{\boldsymbol{x},a}$, $\phi_{\boldsymbol{x},a} \quad = \quad 2\eta / [\exp(\gamma U_{\boldsymbol{x},a}) + \exp(\quad \gamma U_{\boldsymbol{x},a})]$. Otherwise $\phi_{\boldsymbol{x},a} \quad = \quad \lambda / \left[\frac{\lambda}{\eta} \exp(\quad \gamma U_{\boldsymbol{x},a}) + \frac{\eta \pi_{\vartheta}(a|\boldsymbol{x})^2}{\hat{\beta}^2(a|\boldsymbol{x}) \exp(\quad \gamma U_{\boldsymbol{x},a})}\right]$. In other words, $\phi_{\boldsymbol{x},a} \quad 2\eta$ always holds.*
- *If $\alpha_{\boldsymbol{x},a} \quad \frac{\pi_{\vartheta}(a|\boldsymbol{x})}{\hat{\beta}(a|\boldsymbol{x})}$ and $\frac{\pi_{\vartheta}(a|\boldsymbol{x})}{\boldsymbol{B}_{\boldsymbol{x},a}} < \overline{\lambda}$, larger $U_{\boldsymbol{x},a}$ brings larger $\phi_{\boldsymbol{x},a}$.*
- *Otherwise $\phi_{\boldsymbol{x},a}$ decreases as $U_{\boldsymbol{x},a}$ increases.*

Proof. The following inequality validates the first observation:

$$\frac{\lambda}{\frac{\lambda}{\eta} \exp(\quad \gamma U_{\boldsymbol{x},a}) + \frac{\eta \pi_{\vartheta}(a|\boldsymbol{x})^2}{\hat{\beta}^2(a|\boldsymbol{x}) \exp(\quad \gamma U_{\boldsymbol{x},a})}} \qquad \frac{2\eta}{\exp(\gamma U_{\boldsymbol{x},a}) + \exp(\quad \gamma U_{\boldsymbol{x},a})}$$

For the second and third observations, $\alpha_{\boldsymbol{x},a} \qquad \overline{\lambda} \qquad \frac{\overline{\lambda}}{2\eta} \qquad \frac{\overline{\lambda}}{\eta}$. Let $L(u) \quad = \quad \frac{\lambda}{\eta} \exp(\quad \gamma u) + \frac{\eta \pi_{\vartheta}(a|\boldsymbol{x})^2}{\hat{\beta}(a|\boldsymbol{x})^2 \exp(\quad \gamma u)}$, we can have:

$$\nabla_u L(u) = \quad \gamma \frac{\lambda}{\eta} \exp(\quad \gamma u) + \gamma \frac{\eta \pi_{\vartheta}(a|\boldsymbol{x})^2}{\hat{\beta}(a|\boldsymbol{x})^2} \exp(\gamma u)$$

To make $\nabla_u L(u) \qquad 0$, we need $u \qquad \frac{1}{\gamma} \log\left(\frac{\overline{\lambda} \hat{\beta}(a|\boldsymbol{x})}{\pi_{\vartheta}(a|\boldsymbol{x})}\right)$. This implies when $U_{\boldsymbol{x},a}$ $\frac{1}{\gamma} \log\left(\frac{\overline{\lambda} \hat{\beta}(a|\boldsymbol{x})}{\eta \pi_{\vartheta}(a|\boldsymbol{x})}\right)$, $\phi_{\boldsymbol{x},a}$ will decrease as $U_{\boldsymbol{x},a}$ increases; otherwise as $U_{\boldsymbol{x},a}$ increases, $\phi_{\boldsymbol{x},a}$ also increases.

More specifically, when $\alpha_{\boldsymbol{x},a} \qquad \frac{\pi_{\vartheta}(a|\boldsymbol{x})}{\hat{\beta}(a|\boldsymbol{x})}$, we have:

$$U_{\boldsymbol{x},a} \quad \frac{1}{\gamma} \log\left(\frac{\overline{\lambda} \hat{\beta}(a|\boldsymbol{x})}{\eta \pi_{\vartheta}(a|\boldsymbol{x})}\right) \ , \ \frac{\pi_{\vartheta}(a|\boldsymbol{x})}{\boldsymbol{B}_{\boldsymbol{x},a}} < \overline{\lambda} \quad , \quad \frac{\pi_{\vartheta}(a|\boldsymbol{x})}{\hat{\beta}(a|\boldsymbol{x})} < \frac{\overline{\lambda}}{\eta} \exp(\quad \gamma U_{\boldsymbol{x},a}).$$

In other words, for these samples, higher uncertainty implies smaller value of $\pi/\hat{\beta}$, and UIPS tends to boost such safe sample with higher $\phi_{\boldsymbol{x},a}$.

In other cases, $\phi_{\boldsymbol{x},a}$ decreases as $U_{\boldsymbol{x},a}$ increases. This completes the proof. □

7.9 Convergence Analysis

Next we provide the convergence analysis of policy improvement under UIPS.

Definition 7.2. A function $f : \mathbb{R}^d \ ! \ \mathbb{R}$ is L-smooth when $\|\nabla f(x) \quad \nabla f(y)\|_2 \quad L\|x \quad y\|_2$, for all $x, y \ 2 \ \mathbb{R}^d$.

We first state and prove a general result, which serves as the basis to complete our convergence analysis of policy improvement under UIPS. The proof is a special case of convergence proof of stochastic gradient descent with biased gradients in [3]. Suppose we have a differentiable function $f : \mathbb{R}^d \ ! \ \mathbb{R}$, which is L-smooth, and attains a finite minimum value $f \ := \min_{x \in \mathbb{R}^d} f(x)$.

Suppose we cannot directly assess the gradient $\nabla f(x)$. Instead we can only assess a noisy but unbiased gradient $\zeta(x) \ 2 \ \mathbb{R}^d$ at the given x of the function $\tilde{f}(x)$.

Let $b(x) = \nabla \tilde{f}(x) \quad \nabla f(x)$ denote the difference between $\nabla \tilde{f}(x)$ and $\nabla f(x)$, and $\delta(x) = \zeta(x)$ $\nabla \tilde{f}(x)$ denote the noise in gradients. We assume that:

$$\|b(x)\| \quad \varphi \|\nabla f(x)\| \quad \text{and} \quad \mathbb{E}[\delta(x)] = 0 \quad \text{and} \quad \mathbb{E}[\|\delta(x)\|^2 | x] \quad M\mathbb{E}[\|\nabla \tilde{f}(x)\|^2] + \sigma^2 \qquad (21)$$

where the constants φ and M satisfy $0 < \varphi < 1$ and $M \quad 0$ respectively. When running stochastic gradient descent algorithms, i.e. $x_{k+1} = x_k \quad \eta_k \zeta(x_k)$, we have the following guarantee on the convergence of x_k to an approximate stationary point of f.

Theorem 7.3. *Suppose $f(\ )$ is differentiable and L-smooth, and the assessed approximate gradient meets the conditions in Eq.* (21) *with parameters (σ_k, φ_k) at iteration k. Denote $\sigma_{\max} = \max_k \sigma_k$ and $\varphi_{\max} = \max_k \varphi_k$. Set the stepsizes $\{\eta_k\}$ to $\eta_k = \min\{\frac{1}{(M+1)L}, 1/(\sigma_{\max}\sqrt{K})\}$, after K iterations, the stochastic gradient descent satisfies :*

$$\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\left[\|\nabla f(x_k)\|^2\right] \quad \frac{2L(f(x_1) \quad f^{\ })}{K(1 \quad \varphi_{\max})} + \left(L + \frac{2(f(x_1) \quad f^{\ })}{(1 \quad \varphi_{\max})}\right)\frac{\sigma_{\max}}{\sqrt{K}}$$

Proof. With $\eta_k \quad \frac{1}{(M+1)L}$, we have:

$$\begin{aligned} f(x_{k+1}) \quad & f(x_k) + \langle \nabla f(x_k), x_{k+1} \quad x_k\rangle + \frac{L}{2}\|x_{k+1} \quad x_k\|^2 \\ &= f(x_k) \quad \eta_k\langle \nabla f(x_k), \zeta(x_k)\rangle + \frac{L\eta_k^2}{2}\|\zeta(x_k)\|^2 \\ &= f(x_k) \quad \eta_k\langle \nabla f(x_k), \delta(x_k) + b(x_k) + \nabla f(x_k)\rangle + \frac{L\eta_k^2}{2}\|\delta(x_k) + b(x_k) + \nabla f(x_k)\|^2 \end{aligned} \tag{22}$$

By taking expectations on both side, we have:

$$\begin{aligned} \mathbb{E}[f(x_{k+1})] \quad & \mathbb{E}[f(x_k)] \quad \eta_k\mathbb{E}[\langle \nabla f(x_k), \delta(x_k)\rangle] \quad \eta_k\mathbb{E}[\langle \nabla f(x_k), b(x) + \nabla f(x_k)\rangle] + \frac{L\eta_k^2}{2}\mathbb{E}[\|\delta(x_k) + b(x_k) + \nabla f(x_k)\|^2] \\ &\overset{(1)}{=} \mathbb{E}[f(x_k)] \quad \eta_k\mathbb{E}[\langle \nabla f(x_k), b(x) + \nabla f(x_k)\rangle] + \frac{L\eta_k^2}{2}\left(\mathbb{E}[\|\delta(x_k)\|^2] + \mathbb{E}[\|b(x) + \nabla f(x_k)\|^2]\right) \\ &\quad \mathbb{E}[f(x_k)] \quad \eta_k\mathbb{E}[\langle \nabla f(x_k), b(x) + \nabla f(x_k)\rangle] + \frac{L\eta_k^2}{2}\left((M+1)\mathbb{E}[\|b(x) + \nabla f(x_k)\|^2] + \sigma_k^2\right) \\ &\overset{(2)}{\le} \mathbb{E}[f(x_k)] + \frac{\eta_k}{2}\mathbb{E}\left[\left(\quad 2\langle \nabla f(x_k), b(x) + \nabla f(x_k)\rangle + \|b(x) + \nabla f(x_k)\|^2\right)\right] + \frac{L\eta_k^2}{2}\sigma_k^2 \\ &\quad \mathbb{E}[f(x_k)] + \frac{\eta_k}{2}\mathbb{E}[\left(\quad \|\nabla f(x_k)\|^2 + \|b(x)\|^2\right)] + \frac{L\eta_k^2}{2}\sigma_k^2 \\ &\overset{(3)}{\le} \mathbb{E}[f(x_k)] + \frac{\eta_k}{2}(\varphi_k \quad 1)\mathbb{E}[\|\nabla f(x_k)\|^2] + \frac{L\eta_k^2}{2}\sigma_k^2 \end{aligned}$$

where the inequality labeled as (1) is due to $\mathbb{E}[\delta(x)] = 0$, inequality labeled as (2) is due to $\eta_k \quad \frac{1}{(M+1)L}$, and inequality labeled as (3) is due to $\|b(x_k)\| \quad \varphi_k\|\nabla f(x_k)\|$.

By summing over iterations $k = 1, 2, \ldots, K$ and re-arranging the terms, we obtain:

$$\begin{aligned} \frac{1}{2}\sum_{k=1}^{K}(1 \quad \varphi_k)\eta_k\mathbb{E}[\|\nabla f(x_k)\|^2] \quad & f(x_1) \quad \mathbb{E}[f(x_{K+1})] + \frac{L}{2}\sum_{k=1}^{K}\eta_k^2\sigma_k^2 \\ & f(x_1) \quad f^{\ } + \frac{L}{2}\sum_{k=1}^{K}\eta_k^2\sigma_k^2 \end{aligned}$$

where the last inequality follows from $f(x_{K+1}) \quad f^{\ }$. Since $\eta_k = \min\{\frac{1}{(M+1)L}, \frac{1}{\sigma_{\max}\sqrt{K}}\}, \forall k = 1, \ldots, K$, we can obtain:

$$(1 \quad \varphi_{\max})\eta_1\sum_{k=1}^{K}\mathbb{E}[\|\nabla f(x_k)\|^2] \quad 2(f(x_1) \quad f^{\ }) + LK\eta_1^2\sigma_{\max}^2$$

Dividing both sides of the above inequality by $K\eta_1(1 \quad \varphi_{\max})$, we obtain the following,

$$\begin{aligned} \frac{1}{K}\sum_{k=1}^{K}\mathbb{E}[\|\nabla f(x_k)\|^2] \quad & \frac{2(f(x_1) \quad f^{\ }) + LK\eta_1^2\sigma_{\max}^2}{K\eta_1(1 \quad \varphi_{\max})} \\ & \frac{2(f(x_1) \quad f^{\ })}{K(1 \quad \varphi_{\max})}\max\{L, \sigma_{\max}\sqrt{K}\} + L\sigma_{\max}^2\frac{1}{\sigma_{\max}\sqrt{K}} \\ & \frac{2L(f(x_1) \quad f^{\ })}{K(1 \quad \varphi_{\max})} + \left(L + \frac{2(f(x_1) \quad f^{\ })}{(1 \quad \varphi_{\max})}\right)\frac{\sigma_{\max}}{\sqrt{K}} \end{aligned}$$

□

Given this general result, we can now prove Theorem 3.4 by showing that the gradient in UIPS meets the requirements in Theorem 7.3.

Proof of Theorem 3.4:

Proof. UIPS aims to maximize the expected return $V(\pi_\vartheta)$. Therefore, we can utilize Theorem 7.3 by setting $f = V(\pi_\vartheta)$. Since $f = V_{\max}$ and the expected reward is always non-negative, it follows that $f(x_1) \quad f \quad V_{\max}$.

We first introduce some additional notations. Let $\rho_\vartheta(\boldsymbol{x}, a) = \frac{\pi_\vartheta(aj\boldsymbol{x})}{\beta\ (aj\boldsymbol{x})}$ denote the propensity score under ground-truth logging policy, and $\hat{\rho}_\vartheta(\boldsymbol{x}, a) = \frac{\pi_\vartheta(aj\boldsymbol{x})}{\hat{\beta}(aj\boldsymbol{x})}\phi_{\boldsymbol{x},a}$ represet the propensity score of UIPS. Recall that $\phi_{\boldsymbol{x},a}$ is derived through solving the optimization problem in Eq (8). Let $g_\vartheta(\boldsymbol{x}, a) = \frac{\partial \pi_\vartheta(aj\boldsymbol{x})}{\partial \vartheta}$. The true off-policy policy gradient is computed as follows:

$$\mathsf{r}\, V(\pi_\vartheta) = \mathsf{E}_\beta\ [\rho\ \mathsf{r} g_\vartheta],$$

For UIPS, the approximate policy gradient in each batch with batch size as B is:

$$\mathsf{r}\, \hat{V}_{\mathrm{UIPS}}(\pi_\vartheta) = \frac{1}{B}\sum_{i=1}^{B} \hat{\rho}_i r_i g_\vartheta^i,$$

which is an unbiased estimate of :

$$\mathsf{r}\, V_{\mathrm{UIPS}}(\pi_\vartheta) = \mathsf{E}_\beta\ [\hat{\rho}\mathsf{r} g_\vartheta].$$

To utilize Theorem 7.3, we set $\mathsf{r}\, f = \mathsf{r}\, V(\pi_\vartheta)$, $\mathsf{r}\, \tilde{f} = \mathsf{r}\, V_{\mathrm{UIPS}}(\pi_\vartheta)$, and $\zeta = \mathsf{r}\, \hat{V}_{\mathrm{UIPS}}(\pi_\vartheta)$. We will now demonstrate that the assumptions in Eq. (21) can be satisfied.

Let $\varphi_\vartheta = \max\left\{\left|\frac{\hat{\rho}_\vartheta(\boldsymbol{x},a)}{\rho_\vartheta(\boldsymbol{x},a)} \quad 1\right|\right\}$, We first have:

$$\begin{aligned}
\mathsf{k} b(\vartheta)\mathsf{k} &= \mathsf{k}\, \mathsf{r}\, V_{\mathrm{UIPS}}(\pi_\vartheta) \quad \mathsf{r}\, V(\pi_\vartheta)\mathsf{k} \\
&= \mathsf{k}\mathsf{E}_\beta\ [(\hat{\rho} \quad \rho\ )r g_\vartheta]\mathsf{k} = \mathsf{k}\mathsf{E}_\beta\ [\rho\ (\frac{\hat{\rho}}{\rho} \quad 1)r g_\vartheta]\mathsf{k} \\
&\quad \varphi_\vartheta \mathsf{k}\mathsf{E}_\beta\ [\rho\ \mathsf{r} g_\vartheta]\mathsf{k}
\end{aligned} \tag{23}$$

And next we show that $0 < \varphi_\vartheta < 1$.

$$\varphi_\vartheta = \max\left\{\left|\frac{\hat{\rho}_\vartheta(\boldsymbol{x},a)}{\rho_\vartheta(\boldsymbol{x},a)} \quad 1\right|\right\} = \max\left\{\left|\frac{\beta\ (aj\boldsymbol{x})}{\hat{\beta}(aj\boldsymbol{x})} \quad \phi_{\boldsymbol{x},a} \quad 1\right|\right\} \tag{24}$$

Recall from Eq.(17) in proof of Theorem 3.2, we have :

$$\phi_{\boldsymbol{x},a} = \min\left(\frac{\lambda}{\lambda\frac{\boldsymbol{B}_{\boldsymbol{x},a}}{\hat{\beta}(aj\boldsymbol{x})} + \frac{\pi_{\boldsymbol{\vartheta}}(aj\boldsymbol{x})^2}{\hat{\beta}(aj\boldsymbol{x})\boldsymbol{B}_{\boldsymbol{x},a}}}, \frac{2\hat{\beta}(aj\boldsymbol{x})}{\boldsymbol{B}^+_{\boldsymbol{x},a} + \boldsymbol{B}_{\boldsymbol{x},a}}\right)$$

where $\boldsymbol{B}_{\boldsymbol{x},a} := \frac{\hat{Z}\exp(\quad \gamma U_{\boldsymbol{x},a})}{Z}\hat{\beta}(aj\boldsymbol{x})$, and $\boldsymbol{B}^+_{\boldsymbol{x},a} := \frac{\hat{Z}\exp(\gamma U_{\boldsymbol{x},a})}{Z}\hat{\beta}(aj\boldsymbol{x})$. Thus we have:

$$\frac{\beta\ (aj\boldsymbol{x})}{\hat{\beta}(aj\boldsymbol{x})} \quad \phi_{\boldsymbol{x},a} = \min\left(\frac{\lambda}{\lambda\frac{\boldsymbol{B}_{\boldsymbol{x},a}}{\beta\ (aj\boldsymbol{x})} + \frac{\pi_{\boldsymbol{\vartheta}}(aj\boldsymbol{x})^2}{\beta\ (aj\boldsymbol{x})\boldsymbol{B}_{\boldsymbol{x},a}}}, \frac{2}{\frac{\boldsymbol{B}^+_{\boldsymbol{x},a}}{\beta\ (aj\boldsymbol{x})} + \frac{\boldsymbol{B}_{\boldsymbol{x},a}}{\beta\ (aj\boldsymbol{x})}}\right) \tag{25}$$

Since $\boldsymbol{B}^+_{\boldsymbol{x},a} \quad \beta\ (aj\boldsymbol{x})$, thus $0 \quad \frac{\beta\ (aj\boldsymbol{x})}{\hat{\beta}(aj\boldsymbol{x})} \quad \phi_{\boldsymbol{x},a} \quad 2$, implying $0 < \varphi_\vartheta < 1$.

Also, since $\mathsf{r}\, \hat{V}_{\mathrm{UIPS}}(\pi_\vartheta)$ is an unbiased estimate of $\mathsf{r}\, V_{\mathrm{UIPS}}(\pi_\vartheta)$, we have:

$$\mathsf{E}[\delta(\vartheta)] = \mathsf{E}[\,\mathsf{r}\, \hat{V}_{\mathrm{UIPS}}(\pi_\vartheta) \quad \mathsf{r}\, V_{\mathrm{UIPS}}(\pi_\vartheta)] = \mathbf{0} \tag{26}$$

Finally, we have the following,

$$\mathsf{E}[\|\delta(\vartheta)\|^2] = \mathsf{E}\left[\| r\hat{V}_{\mathrm{UIPS}}(\pi_\vartheta) \quad r V_{\mathrm{UIPS}}(\pi_\vartheta)\|^2\right] = \mathsf{E}\left[\left\| \frac{1}{B}\sum_{i=1}^{B}\left(\hat{\rho}_i r_i g_\vartheta^i \quad \mathsf{E}_\beta\ [\hat{\rho} r g_\vartheta]\right)\right\|^2\right]$$

$$\frac{1}{B^2}\mathsf{E}\left[\left(\sum_{i=1}^{B}\|\hat{\rho}_i r_i g_\vartheta^i \quad E_\beta\ [\hat{\rho} r g_\vartheta]\|\right)^2\right] \quad \frac{1}{B}\mathsf{E}\left[\sum_{i=1}^{B}\|\hat{\rho}_i r_i g_\vartheta^i \quad \mathsf{E}_\beta\ [\hat{\rho} r g_\vartheta]\|^2\right]$$

$$\mathsf{E}_\beta\ [\|\hat{\rho} r g_\vartheta \quad \mathsf{E}_\beta\ [\hat{\rho} r g_\vartheta]\|^2] \overset{(1)}{=} \mathsf{E}_\beta\ [\|\hat{\rho} r g_\vartheta\|^2] \quad \|\mathsf{E}_\beta\ [\hat{\rho} r g_\vartheta]\|^2$$

$$\mathsf{E}_\beta\ [\|\hat{\rho} r g_\vartheta\|^2] \quad G_{\max}^2\Phi \tag{27}$$

where the equality labeled as (1) is due to $E[\|Y \quad E[Y]\|^2] = E[\|Y\|^2] \quad \|E[Y]\|^2$.
Hence, by applying Theorem 7.3, we have:

$$\frac{1}{K}\sum_{k=1}^{K}\mathsf{E}[\|r V(\pi_{\vartheta_k})\|^2 \quad \frac{2LV_{\max}}{K(1 \quad \varphi_{\max})} + \left(L + \frac{2V_{\max}}{(1 \quad \varphi_{\max})}\right)\frac{G_{\max}\sqrt{p\Phi}}{K} \tag{28}$$

□