Local Metric Learning for Off-Policy Evaluation in Contextual Bandits with Continuous Actions

Haanvid Lee[1], **Jongmin Lee**[2], **Yunseon Choi**[1], **Wonseok Jeon**[†],
Byung-Jun Lee[3,4], **Yung-Kyun Noh**[5,6], **Kee-Eung Kim**[1]
[1]KAIST, [2]UC Berkeley, [3]Korea Univ., [4]Gauss Labs Inc., [5]Hanyang Univ., [6]KIAS
haanvid@kaist.ac.kr, jongmin.lee@berkeley.edu, cys9506@kaist.ac.kr
byungjunlee@korea.ac.kr, nohyung@hanyang.ac.kr, kekim@kaist.ac.kr

Abstract

We consider local kernel metric learning for off-policy evaluation (OPE) of deterministic policies in contextual bandits with continuous action spaces. Our work is motivated by practical scenarios where the target policy needs to be deterministic due to domain requirements, such as prescription of treatment dosage and duration in medicine. Although importance sampling (IS) provides a basic principle for OPE, it is ill-posed for the deterministic target policy with continuous actions. Our main idea is to relax the target policy and pose the problem as kernel-based estimation, where we learn the kernel metric in order to minimize the overall mean squared error (MSE). We present an analytic solution for the optimal metric, based on the analysis of bias and variance. Whereas prior work has been limited to scalar action spaces or kernel bandwidth selection, our work takes a step further being capable of vector action spaces and metric optimization. We show that our estimator is consistent, and significantly reduces the MSE compared to baseline OPE methods through experiments on various domains.

1 Introduction

In order to deploy a contextual bandit policy to a real-world environment, such as personalized pricing[1], treatments [2], recommendation [3], and advertisements [4], the performance of the policy should be evaluated prior to the deployment to decide whether the trained policy is suitable for deployment. This is because the interaction of the policy with the environment could be costly and/or dangerous. For example, using an erroneous medical treatment policy to prescribe drugs for patients could result in dire consequences. There then emerges a necessity for an algorithm that can evaluate a policy's performance without having it interact with the environment in an online manner. Such algorithms are called "off-policy evaluation" (OPE) algorithms. [5]. OPE algorithms for contextual bandits evaluate a target policy by estimating its expected reward from the data sampled by a behavior policy and without the target policy interacting with the environment.

Previous works on OPE for contextual bandits have mainly focused on environments with finite actions [4, 6–10]. The works can be largely divided into three approaches [6, 9, 11]. The first approach is the direct method (DM) which learns an environment model for policy evaluation. DM is known to have low variance [9]. However, since the environment model is learned with function approximation, the estimator is biased. The second approach is importance sampling (IS) which corrects the data distribution induced by a behavior policy to that of a target policy [12], and uses the corrected distribution to approximate the expected reward of the target policy. IS estimates are unbiased when the behavior and target policies are given. However, IS estimates can have large

[†]The research was done while in Mila/McGill University, but the author is currently employed by Qualcomm Technologies Inc.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

variance when there is a large mismatch between the behavior and target policy distributions [9]. The last approach is doubly robust (DR), which uses DM to reduce the variance of IS while keeping the unbiasedness of an IS estimate [6].

Although there are existing works on IS and DR that can be applied to continuous action spaces [13–16], most of them cannot be easily extended to evaluate *deterministic* contextual bandit policies with continuous actions. This is because IS weights used for both IS and DR estimators are almost surely zero for deterministic target policies in continuous action spaces [2]. However, in practice, such OPE algorithms are needed. For example, treatment prescription policies should not stochastically prescribe drugs to patients.

To meet the needs, there are works for evaluating deterministic contextual bandit policies with continuous actions [2, 5, 17]. These works focus on measuring the similarity between behavior and target actions for assigning IS ratios. However, these works either assume a single-dimensional action space [17] which cannot be straightforwardly extended to multiple action dimensions, or, use Euclidean distances for the similarity measures [2, 5]. In general, similarity measures should be learned locally at a state to weigh differences between behavior and target actions in each action dimension differently. For example, in the case of multi-drug prescription, the synergies and side effects of the prescribed drugs are often very complex. Moreover, the different kinds of drugs are likely to have different degrees of effect from person to person [18, 19].

To this end, we propose local kernel metric learning for IS (KMIS) OPE estimation of deterministic contextual bandit policies with multi-dimensional continuous action spaces. Our proposed method learns a Mahalanobis distance metric [20–23] locally at each state that lengthens or shortens the distance between behavior and target actions to reduce the MSE. The metric-applied kernels measure the similarities between actions according to the Mahalanobis distances induced by the metric. In our work, we first analytically show that the leading-order bias [2] of a kernel-based IS estimator becomes a dominant factor in the leading-order MSE [2] as the action dimension increases given the optimal bandwidth [2] that minimizes the leading-order MSE without a metric. Then we derive the kernel metric that minimizes the upper bound of the leading-order bias, which is bandwidth-agnostic. Our analysis shows that the convergence speed of a kernel-based IS OPE estimator can be improved with the application of the KMIS metrics. In the experiments, we demonstrate that MSEs of kernel-based IS estimators are significantly reduced when combined with the proposed kernel metric learning. We report empirical results in various synthetic domains as well as a real-world dataset.

2 Related Work

The works on OPE of deterministic contextual bandits with continuous actions can be largely divided into importance sampling (IS) [2, 5] and doubly robust estimators (DR) [17, 24]. Both methods eliminate the problem of almost surely having zero IS estimates when given a deterministic target policy and a stochastic behavior policy in a continuous action space in two ways. Most of the works relax the deterministic target policy, which can be seen as a Dirac delta function [2], to a kernel [2, 5, 24], and the other work discretizes the action space [17].

Among these, kernel-based IS methods use a kernel to measure the similarities between the target and behavior actions and focus on selecting the bandwidth of the kernel [2, 5, 24]. Su et al. [5] proposed the bandwidth selection algorithm that uses the Lepski's principle for bandwidth selection in the study of nonparametric statistics [25]. Kallus and Zhou [2] derived the leading-order MSE of a kernel-based IS OPE estimation and chose the optimal bandwidth that minimizes the leading-order MSE for OPE estimations. One of the limitations of the existing kernel-based IS methods [2, 5, 24] is that these methods use Euclidean distances for measuring the similarities between behavior and target actions. The Euclidean distance is inadequate for measuring the similarities as assigning a high similarity measure to the actions having similar rewards will induce less bias (bias definition in Section 3.2). The other limitation is that these methods use one global bandwidth for the whole action space [17]. Since the second-order derivative of the expected reward w.r.t. an action is related to the leading-order bias derived by Kallus and Zhou [2], the optimal bandwidth that balances the bias-variance trade-off may vary depending on actions.

As for discretization methods, Cai et al. [17] proposed deep jump learning (DJL) [17] that avoids the limitation of kernel-based methods due to using one global bandwidth by adaptively discretizing one-dimensional continuous action spaces. The action space is discretized to have similar expected

rewards for each discretized action interval given a state. By using the discretized intervals for DR estimation, DJL can estimate the policy value more accurately in comparison to kernel-based methods when the action space has second-order derivatives of the expected rewards which change significantly across the action space. In such cases, kernel-based methods use the same bandwidth for all actions even though the optimal bandwidth varies depending on actions. On the other hand, DJL can discretize the action space into smaller intervals for the parts of the action space where the second-order derivative is high, and larger intervals when it is low. However, their work focuses on domains with a single action dimension and cannot be easily extended to environments with multi-dimensional action spaces.

To tackle the limitation of kernel-based methods caused by using Euclidean distances, kernel metric learning can be applied to shrink or extend distances in the directions that reduce MSE. Metric learning has been used for nearest neighbor classification [22, 26–29], and kernel regression [22, 23]. Among them, our work was inspired by the work of Noh et al. that learns a metric for reducing the bias and MSE of kernel regression [23]. In their work, they made the assumption on the generative model of the data where the input and output are jointly Gaussian. Under this assumption, they derived the Mahalanobis metric that reduces the bias and MSE of Nadaraya-Watson kernel regression. We learn our metric in a similar fashion in the context of OPE in contextual bandits except that we do not have the generative assumption on the data as the assumption is unnatural in our problem setting.

3 Preliminaries

3.1 Problem Setting

In this work, we focus on OPE of a deterministic target policy in an environment with multi-dimensional continuous action space $\mathcal{A} \subset \mathbb{R}^{D_A}$, state space $\mathcal{S} \subset \mathbb{R}^{D_S}$, reward $r \in \mathbb{R}$ sampled from the conditional distribution of the reward $p(r|\,\mathbf{s}, \mathbf{a})$, state distribution $p(\mathbf{s})$. The *deterministic* target policy π can be regarded as having a Dirac delta distribution $\pi(\mathbf{a}\,|\,\mathbf{s}) = \delta(\pi(\mathbf{s}) - \mathbf{a})$, where the probability density function (PDF) value is zero everywhere except at the selected target action given a state $\pi(\mathbf{s})$. The offline dataset $D = \{(\mathbf{s}_i, \mathbf{a}_i, r_i)\}_{i=1}^N$ used for the evaluation of the deterministic target policy π is sampled with a known stochastic behavior policy $\pi_b : \mathcal{S} \to \Delta(\mathcal{A})$. We assume that the support of the behavior policy π_b contains the actions selected by the target policy $\pi(\mathbf{s})$. The goal of OPE is to evaluate the target policy value $\rho^\pi = \mathbb{E}_{\mathbf{s}\sim p(\mathbf{s}), \mathbf{a}\sim\pi(\mathbf{a}\,|\,\mathbf{s}), r\sim p(r|\,\mathbf{s},\mathbf{a})}[r]$ using D and without interaction with the environment.

3.2 Bandwidth Selection for the Isotropic Kernel-Based IS Estimator

One of the methods to evaluate the target policy value with the offline data D sampled from π_b is to perform IS estimation. The IS ratios correct the action sampling distribution for the expectation from π_b to π. However, since the density of a deterministic target policy π is a Dirac delta function, its PDF value at the behavior action sampled from π_b is almost surely 0. Existing works deal with the problem by relaxing the target policy π to an isotropic kernel K with a bandwidth h and computing the IS estimate of the policy value $\hat{\rho}^K$ as in Eq. (1) [2, 5, 24].

$$\begin{aligned}\rho^\pi &= \mathbb{E}_{\mathbf{s}\sim p(\mathbf{s}), \mathbf{a}\sim\pi_b(\mathbf{a}|\mathbf{s}), r\sim p(r|\mathbf{s},\mathbf{a})}\left[\frac{\pi(\mathbf{a}\mid\mathbf{s})}{\pi_b(\mathbf{a}\mid\mathbf{s})} r\right], \\ &\approx \frac{1}{Nh^{D_A}} \sum_{i=1}^N K\left(\frac{\mathbf{a}_i - \pi(\mathbf{s}_i)}{h}\right) \frac{r_i}{\pi_b(\mathbf{a}_i \mid \mathbf{s}_i)} \end{aligned} \tag{1}$$

As the Dirac delta function can be regarded as a kernel having its bandwidth h approaching zero, the relaxation of the Dirac delta function to a kernel can be seen as increasing its h. By the relaxation, the bias of the kernel-based IS estimation $\mathrm{Bias}\left[\hat{\rho}^K\right] := \mathbb{E}_{\mathbf{s}\sim p(\mathbf{s}), \mathbf{a}\sim\pi_b(\mathbf{a}|\mathbf{s}), r\sim p(r|\mathbf{s},\mathbf{a})}\left[\hat{\rho}^K - \rho^\pi\right]$ increases while its variance is reduced. As the bias and the variance compose the MSE of the estimate, the bandwidth that best balances between them and reduce the MSE should be selected. Kallus and Zhou [2] derived the leading-order MSE (LOMSE) in terms of bandwidth h, sample size N, and action dimension D_A for selecting a bandwidth (Eq. (2)) assuming that $h \to 0$ and $\frac{1}{Nh^{D_A}} \to 0$ as $N \to \infty$ (derivation in Appendix A.1). They also derived the optimal bandwidth h^* that minimizes the LOMSE (derivation in Appendix A.2).

(a) $K(\mathbf{a}-\mathbf{a}_t) = \frac{1}{2\pi}\exp\left(-\frac{(\boldsymbol{a}-\boldsymbol{a_t})^\top(\boldsymbol{a}-\boldsymbol{a_t})}{2}\right)$ (b) $K(\mathbf{z}-\mathbf{z}_t) = \frac{1}{2\pi}\exp\left(-\frac{(\boldsymbol{a}-\boldsymbol{a_t})^\top A(\boldsymbol{s})(\boldsymbol{a}-\boldsymbol{a_t})}{2}\right)$

Figure 1: Illustration of bias reduction in a kernel-based IS estimate by the metric $A(\mathbf{s})$ locally learned at a given state $\mathbf{s}$. The contour line is drawn for the reward over the action space given $\mathbf{s}$. Although behavior actions $\mathbf{a}'$ and $\mathbf{a}''$ are away from target action $\mathbf{a}_t$ $(=\pi(\mathbf{s}))$ by an equal Euclidean distance, their corresponding rewards are different. When an (a) isotropic Gaussian kernel is used, bias can come from $\mathbf{a}'$ since the similarity measures of $\mathbf{a}'$, and $\mathbf{a}''$ from $\mathbf{a}_t$ are the same even though their rewards are different. However, when the (b) metric is applied, bias is reduced as the similarity measure is higher on $\mathbf{a}''$ that has a similar reward to $\mathbf{a}_t$ compared to that of $\mathbf{a}'$.

$$\mathrm{LOMSE}(h, N, D_A) = \underbrace{h^4 C_b}_{\text{(leading-order bias)}^2} + \underbrace{\frac{C_v}{N h^{D_A}}}_{\text{(leading-order variance)}}, \tag{2}$$

$$h^* = \arg\min_h \mathrm{LOMSE}(h, N, D_A) = \left(\frac{D_A C_v}{4 N C_b}\right)^{\frac{1}{D_A+4}}, \tag{3}$$

$$C_b := \left(\frac{1}{2}\mathbb{E}_{\mathbf{s}\sim p(\mathbf{s})}\left[\nabla^2_{\mathbf{a}} r(\mathbf{s},\mathbf{a})\mid_{\mathbf{a}=\pi(\mathbf{s})}\right]\right)^2, C_v := R(K)\mathbb{E}_{\mathbf{s}\sim p(\mathbf{s})}\left[\frac{\mathbb{E}[r^2|\,\mathbf{s},\mathbf{a}=\pi(\mathbf{s})]}{\pi_b(\mathbf{a}=\pi(\mathbf{s})\mid\mathbf{s})}\right],$$

where the first term in Eq. (2) is the squared leading-order bias and the second term is the leading-order variance, C_b and C_v are constants related to the leading-order bias and variance, respectively, the expected reward is $r(\mathbf{s},\mathbf{a}) := \mathbb{E}[r|\,\mathbf{s},\mathbf{a}]$, $\nabla^2_{\mathbf{a}}$ denotes the Laplacian operator w.r.t. action $\mathbf{a}$, the roughness of the kernel is $R(K) := \int K(\mathbf{u})^2 d\mathbf{u}$. For simplicity, we assumed a Gaussian kernel and used the property $\int \mathbf{u}\mathbf{u}^\top K(\mathbf{u}) d\mathbf{u} = I$. The kernel used for the derivation satisfies $\int K(\mathbf{u}) d\mathbf{u} = 1$ and $\int \mathbf{u} K(\mathbf{u}) d\mathbf{u} = 0$.

In Section 4, we use the leading-order bias and variance [2] for the derivation of our proposed metric. We also use the optimal bandwidth [2] for analyzing the properties of kernel-based IS estimator with and without a metric.

3.3 Mahalanobis Distance Metric

Relaxing the Dirac delta target policy π to a kernel K prevents the kernel-based IS estimator from estimating 0 a.s. By using an isotropic kernel for the relaxation, the difference between the target and behavior actions in all directions are treated equally for measuring the similarities between the actions with a kernel in Eq. (1). However, to produce a more accurate OPE estimation, the difference between the two actions in some directions should be ignored relative to the others for measuring the similarity between the actions. For this, the Mahalanobis distance can be used. We define the Mahalanobis distance between two D_A-dimensional vectors $\mathbf{a}_i \in \mathbb{R}^{D_A}$ and $\mathbf{a}_j \in \mathbb{R}^{D_A}$ with the metric $A \in \mathbb{R}^{D_A \times D_A}$ as in Eq. (4). Applying the Mahalanobis distance metric A $(= LL^\top)$ to a kernel function can be seen as linearly transforming the kernel inputs with the transformation matrix L $(L^\top \mathbf{a} = \mathbf{z})$.

$$\|\mathbf{a}_i - \mathbf{a}_j\|_A := \sqrt{(\mathbf{a}_i - \mathbf{a}_j)^\top A (\mathbf{a}_i - \mathbf{a}_j)} = \|\mathbf{z}_i - \mathbf{z}_j\|, \ (A \succ 0,\ A^\top = A,\ |A| = 1) \tag{4}$$

Figure 1 illustrates a case where the Mahalanobis metric is locally learned at a given state for reducing the bias of a kernel-based IS estimate. The bias of the estimate is reduced by altering the shape of the kernel to produce a higher similarity measure on a behavior action that has a similar reward to the target action.

4 Local Metric Learning for Kernel-Based IS

4.1 Local Metric Learning via LOMSE Reduction

To reduce the LOMSE (Eq. (2)) of the kernel-based IS estimate by learning a metric, we first analyze the LOMSE when the optimal bandwidth (Eq. (3)) is applied to the kernel. In Figure 1 we illustrate

how the bias of an IS estimation is induced due to kernel relaxation. The bias can worsen in high dimensional spaces. To analytically show this, we adapt Proposition 4 in the work of Noh et al. [23] and show that the C_b contained in the squared leading-order bias (Eq. (2)) becomes a dominant term in the LOMSE of kernel-based IS OPE given an optimal bandwidth and as the action dimension increases.

Proposition 1. *(Adapted from Noh et al. [23]) For a high dimensional action space $D_A \gg 4$, and given optimal bandwidth h^* (Eq. (3)), the squared leading-order bias dominates over the leading-order variance in LOMSE. Furthermore,* $\mathrm{LOMSE}(h^*, N, D_A)$ *can be approximated by C_b in Eq. (2).*

$$\mathrm{LOMSE}(h^*, N, D_A) = N^{-\frac{4}{D_A+4}} \left(\left(\frac{D_A}{4}\right)^{\frac{4}{D_A+4}} + \left(\frac{4}{D_A}\right)^{\frac{D_A}{D_A+4}} \right) C_b^{\frac{D_A}{D_A+4}} C_v^{\frac{4}{D_A+4}} \approx C_b. \tag{5}$$

The proof is made by plugging in Eq. (3) to Eq. (2) and taking the limit $D_A \to \infty$. (See Appendix B.1.) Proposition 1 implies that in an environment with high dimensional action space, the MSE can be significantly reduced by the reduction of C_b contained in the squared leading-order bias (Eq. (2)). Therefore, we aim to reduce C_b, or, reduce the leading bias in a bandwidth-agnostic manner with a Mahalanobis distance metric that shortens the distance between the target and behavior actions in the direction where their corresponding rewards are similar.

As mentioned in Section 3.3, applying a state-dependent metric ($A(\mathbf{s}) = L(\mathbf{s})L(\mathbf{s})^\top$) to a kernel is equivalent to linearly transforming input vectors of the kernel. Therefore, C_b with a metric $A : \mathcal{S} \to \mathbb{R}^{D_A \times D_A}$ (i.e. $C_{b,A}$) should be analyzed with the linearly transformed actions $L(\mathbf{s})^\top \mathbf{a}$ (Derivation in Appendix A.3.), and we aim to find an optimal A that minimizes it:

$$\min_{\substack{A:\ A(s)\succ 0,\\ A(s)=A(s)^\top, |A(s)|=1\ \forall s}} C_{b,A} = \frac{1}{4}\mathbb{E}_{s\sim p(s)} \left[\mathrm{tr}\left(A(\mathbf{s})^{-1}\, \mathbf{H}_{\mathbf{a}}\, r(\mathbf{s},\mathbf{a})\big|_{a=\pi(\mathbf{s})} \right) \right]^2, \tag{6}$$

where $\mathbf{H}$ is the Hessian operator. In the derivation, we assume a Gaussian kernel and use the property $\int \mathbf{u}\mathbf{u}^\top K(\mathbf{u}) d\mathbf{u} = I$. Still, optimizing the function A in Eq. (6) itself is challenging since it requires considering the overall effect of each metric matrix $A(s)$ on the objective function. Therefore, we instead consider minimizing the following upper bound, which allows us to compute the closed-form metric matrix for each state in a non-parametric way:

$$\min_{\substack{A:\ A(s)\succ 0,\\ A(s)=A(s)^\top, |A(s)|=1\ \forall s}} U_{b,A} = \frac{1}{4}\mathbb{E}_{s\sim p(s)} \left[\mathrm{tr}\left(A(\mathbf{s})^{-1}\, \mathbf{H}_{\mathbf{a}}\, r(\mathbf{s},\mathbf{a})\big|_{a=\pi(\mathbf{s})} \right)^2 \right], \tag{7}$$

In the following Theorem 1, we introduce the optimal Mahalanobis metric matrix $A^*(\mathbf{s})$ that minimizes $U_{b,A}$.

Theorem 1. *(Adapted from Noh et al. [22]) Assume that the $p(r|\,\mathbf{s},\mathbf{a})$ is twice differentiable w.r.t. an action $\mathbf{a}$. Let $\Lambda_+(\mathbf{s})$ and $\Lambda_-(\mathbf{s})$ be diagonal matrices of positive and negative eigenvalues of the Hessian $\mathbf{H}_{\mathbf{a}}\, \mathbb{E}[r|\,\mathbf{s},\mathbf{a}]|_{\mathbf{a}=\pi(\mathbf{s})}$, $U_+(\mathbf{s})$ and $U_-(\mathbf{s})$ be matrices of eigenvectors corresponding to $\Lambda_+(\mathbf{s})$ and $\Lambda_-(\mathbf{s})$ respectively, and $d_+(\mathbf{s})$ and $d_-(\mathbf{s})$ be the numbers of positive and negative eigenvalues of the Hessian. Then the metric $A^*(\mathbf{s})$ that minimizes $U_{b,A}$ is:*

$$A^*(\mathbf{s}) = \alpha(\mathbf{s})\,[U_+(\mathbf{s})U_-(\mathbf{s})] \underbrace{\begin{pmatrix} d_+(\mathbf{s})\Lambda_+(\mathbf{s}) & 0 \\ 0 & -d_-(\mathbf{s})\Lambda_-(\mathbf{s}) \end{pmatrix}}_{=:M(\mathbf{s})} [U_+(\mathbf{s})U_-(\mathbf{s})]^\top, \tag{8}$$

where $\alpha(\mathbf{s}) := |M(\mathbf{s})|^{-1/(d_+(\mathbf{s})+d_-(\mathbf{s}))}$.

The proof involves solving a Lagrangian equation to minimize the square of the trace term in Eq. (7) with constraints on $A^*(\mathbf{s})$. (See Appendix B.2.) In the special case where the Hessian matrix contains both positive and negative eigenvalues for all states, the metric reduces the $C_{b,A}$ to zero. (See Appendix B.2.) $A^*(\mathbf{s})$ is locally computed at a state $\mathbf{s}$ and the corresponding target action $\pi(\mathbf{s})$ as the optimal metric is derived from the Hessian at that point $\mathbf{H}_{\mathbf{a}}\, r(\mathbf{s},\mathbf{a})|_{\mathbf{a}=\pi(\mathbf{s})}$. When the optimal metric is applied to the kernel, it measures the similarity between a behavior and target action using the Mahalanobis distance instead of the Euclidean distance, as shown in Figure 1 and in Eq. (9):

$$\begin{aligned} \|\mathbf{a}-\pi(\mathbf{s})\|^2_{A^*(\mathbf{s})} = \alpha(\mathbf{s}) &\sum_{i=1}^{d_+} \left\{ \sqrt{d_+(\mathbf{s})\lambda_{+,i}(\mathbf{s})}\mathbf{u}_{+,i}(\mathbf{s})^\top(\mathbf{a}-\pi(\mathbf{s})) \right\}^2 \\ &+ \alpha(\mathbf{s}) \sum_{i=1}^{d_-} \left\{ \sqrt{-d_-(\mathbf{s})\lambda_{-,i}(\mathbf{s})}\mathbf{u}_{-,i}(\mathbf{s})^\top(\mathbf{a}-\pi(\mathbf{s})) \right\}^2 \end{aligned} \tag{9}$$

where $\lambda_{+,i}(\mathbf{s})$ and $\lambda_{-,i}(\mathbf{s})$ are positive and negative eigenvalues of the Hessian $\mathbf{H_a}\, r(\mathbf{s},\mathbf{a})|_{\mathbf{a}=\pi(\mathbf{s})}$, and their corresponding eigenvectors are denoted as $\mathbf{u}_{+,i}(\mathbf{s})$ and $\mathbf{u}_{-,i}(\mathbf{s})$ respectively. The Mahalanobis distance is formed to weigh the difference between the behavior action $\mathbf{a}$ and the target action $\pi(\mathbf{s})$ w.r.t. the axes formed by the eigenvectors. $A^*(\mathbf{s})$ increases the distance between the actions in the direction of eigenvectors whose corresponding eigenvalues are large, and vice versa.

Now we analyze how the optimal metric affects the convergence rate of a kernel-based IS estimator w.r.t. data size N and the action dimension D_A similar to Theorem 3 in the work of Kallus and Zhou [2].

Theorem 2. *(Adapted from Kallus and Zhou [2]) Kernel-based IS estimator with the optimal metric $A^*(\mathbf{s})$ from Eq.* (8) *and the optimal bandwidth h^* in Eq.* (3) *is a consistent estimator in which convergence rate is equal or faster than the isotropic kernel-based IS estimator with h^*. When the Hessian $\mathbf{H_a}\, r(\mathbf{s},\mathbf{a})|_{\mathbf{a}=\pi(\mathbf{s})}$ has both positive and negative eigenvalues for all states, $A^*(\mathbf{s})$ applied estimator converges to the true policy value faster than the one without the metric by the rate of $\mathcal{O}(D_A^{-\frac{1}{2}})$ as the action dimension increases.*

The proof is made by analyzing the complexity of the sample size N and the action dimension D_A of the LOMSE with and without the proposed metric. (See Appendix B.3.)

4.2 Practical Algorithm

To avoid the degenerate case of having zero for all eigenvalues of the Hessian $\mathbf{H_a}\, r(\mathbf{s},\mathbf{a})|_{\mathbf{a}=\pi(\mathbf{s})}$, we add in the regularizer $\gamma(\mathbf{s})$, and also use $\beta(\mathbf{s})$ to make $|\hat{A}(\mathbf{s})| = 1$ as in Eq. (10) so that the metric applied kernel can fall back to an isotropic kernel in the degenerate case. (See Appendix C.1 for more details.) For the simplicity of the algorithm, we did not discard the components along the eigenvectors with zero eigenvalues ($U_0(\mathbf{s})$) but let the metric have longer relative bandwidths in those directions compared to the others by the regularizers. The KMIS metric $\hat{A}(\mathbf{s})$ with the regularizers is,

$$\hat{A}(\mathbf{s}) = \beta(\mathbf{s})\,[U_+(\mathbf{s})U_-(\mathbf{s})U_0(\mathbf{s})] \begin{pmatrix} d_+(\mathbf{s})\Lambda_+(\mathbf{s}) & 0 & 0 \\ 0 & -d_-(\mathbf{s})\Lambda_-(\mathbf{s}) & 0 \\ 0 & 0 & \mathbf{0} \end{pmatrix} [U_+(\mathbf{s})U_-(\mathbf{s})U_0(\mathbf{s})]^\top + \gamma(\mathbf{s})I \tag{10}$$

To apply the KMIS metric $\hat{A}(\mathbf{s})$ on a kernel-based IS estimation, we first fit a neural network reward regressor with the dataset D for the estimation of the Hessian matrix $\mathbf{H_a}\, r(\mathbf{s},\mathbf{a})|_{\mathbf{a}=\pi(\mathbf{s})}$. The reward regressor may have some error in the estimation of the Hessian and may have an adverse effect on our algorithm. (Effect analyzed in Appendix F.1.) Using the estimated Hessian, we linearly transform the kernel inputs with the transformation matrix $\hat{L}(\mathbf{s})$ ($\hat{A}(\mathbf{s}) = \hat{L}(\mathbf{s})\hat{L}(\mathbf{s})^\top$) in Eq. (11). Then given a kernel, and a bandwidth from the previous works on kernel-based IS that selects bandwidths [2][5], IS estimation can be made with the offline data D as in Algorithm 1.

$$\hat{L}(\mathbf{s}) = [U_+(\mathbf{s})U_-(\mathbf{s})U_0(\mathbf{s})] \left[\begin{pmatrix} \beta(\mathbf{s})d_+(\mathbf{s})\Lambda_+(\mathbf{s}) & 0 & 0 \\ 0 & -\beta(\mathbf{s})d_-(\mathbf{s})\Lambda_-(\mathbf{s}) & 0 \\ 0 & 0 & \mathbf{0} \end{pmatrix} + \gamma(\mathbf{s})I \right]^{\frac{1}{2}} \tag{11}$$

Algorithm 1 KMIS for Kernel-Based IS Estimation

Input: Offline data $D = \{\mathbf{s}_i, \mathbf{a}_i, r_i\}_{i=1}^N$, behavior policy π_b, deterministic target policy π, reward function parameters ϕ, bandwidth h, kernel K.
Output: Estimate of the target policy value $\hat{\rho}^K$

1: Fit the neural network regressor $r_\phi(\mathbf{s},\mathbf{a})$ with D
2: Compute $\mathbf{H_a}\, r_\phi(\mathbf{s},\mathbf{a})|_{\mathbf{a}=\pi(\mathbf{s})}$
3: Compute $\hat{L}(\mathbf{s})$ in Eq. (11) from $\mathbf{H_a}\, r_\phi(\mathbf{s},\mathbf{a})|_{\mathbf{a}=\pi(\mathbf{s})}$
4: Transform the kernel inputs: $\mathbf{z}_i = \hat{L}(\mathbf{s}_i)^\top (\mathbf{a}_i - \pi(\mathbf{s}_i))$, for all i
5: Compute the estimate: $\frac{1}{Nh^{D_A}} \sum_{i=1}^N K\left(\frac{\mathbf{z}_i}{h}\right) \frac{r_i}{\pi_b(\mathbf{a}_i|\mathbf{s}_i)}$

5 Experiments

In this section, we show that applying our KMIS metric reduces the MSEs of the kernel-based IS estimators with bandwidths selected from existing works [2, 5] on synthetic domains and Warfarin dataset [30]. For baselines, we use existing works on kernel-based IS [2, 5] and a direct method (DM). For synthetic domains, we also include a simple discretized OPE estimator [2] as a baseline. For the baselines of existing works of kernel-based IS, we use the work of Kallus and Zhou [2] and SLOPE [5] which are bandwidths selection algorithms for kernel-based IS. For the direct method (DM) we use a neural network reward regressor with a Gaussian output layer. The discretized OPE estimator [2] evenly discretizes the action space by 10 for each action dimension (resulting in 100 bins in 2-dimensional action spaces) for an IS estimation. The baselines are compared to our proposed KMIS metric applied kernel-based IS estimator with bandwidths selected from Kallus and Zhou's estimator or SLOPE. DM's reward regressor is used for estimating the Hessian $\mathbf{H_a}\, r(\mathbf{s}, \mathbf{a})|_{\mathbf{a}=\pi(\mathbf{s})}$ (Eq. (8)) required for KMIS metric computation and the optimal bandwidth selection of Kallus and Zhou's estimator (compute C_v and C_b in Eq. (3) with DM). We use DM for bandwidth selection of Kallus and Zhou's estimator instead of using kernel density estimation (KDE), which is used in the work of Kallus and Zhou [2], since using KDE to estimate the bandwidth is computationally expensive [17]. For all estimators, we use self-normalization as in the work of Kallus and Zhou [2] as it is known to reduce estimation variance significantly with the addition of a small bias and result in reduced MSE. We do not correct the boundary bias as opposed to the works of Kallus and Zhou [2], Su et al. [5] since we regard the bias induced by the boundaries are negligible compared to the bias reduced by our metric. For more details of the experiments, see Appendix E.

5.1 Synthetic Data

In the experiment with the synthetic data, we show that our KMIS metric can be applied to the offline data sampled from environments with various reward functions. With such offline data, we first show the MSEs of existing kernel-based IS estimators can be reduced by our KMIS metric. Furthermore, we empirically validate Proposition 1. Lastly, we visualize the learned KMIS metrics to see if the metrics are learned as we intended.

We prepare three synthetic domains which are quadratic reward domain, absolute error domain, and multi-modal reward domain. All synthetic domains use actions and states in $\mathbb{R}^2$. In the quadratic reward domain, the rewards of the offline data are sampled from a normal distribution, where the mean is from a quadratic function w.r.t. states and actions (for details, see Appendix E). Each dimension of states are independently and uniformly sampled in the range of $[-1, 1]$, actions are sampled from $\pi_b(\mathbf{a}\,|\,\mathbf{s}) = N(\mathbf{s} + 0.2I, 0.5^2 I)$, target policy is $\pi(\mathbf{s}) = \mathbf{s}$. If the metric is learned successfully from the data, the metric should form an ellipsoidal shape and should be the same for any state and action since the Hessian $\mathbf{H_a}\, r(\mathbf{s}, \mathbf{a})$ of the quadratic reward function is a constant.

In the absolute error domain, the offline data is sampled from the environment with deterministic rewards given states and actions $r(\mathbf{s}, \mathbf{a}) = -|0.5s_1 - a_1|$, where s_1 and a_1 are the first dimensions of the state and action vectors. Each dimension of both states and actions are independently and uniformly sampled in the range of $[-1, 1]$, and the target policy is $\pi(\mathbf{s}) = 0.5\mathbf{s}$. Since the absolute value function has points in its domain where it is not twice differentiable, and the Hessian $\mathbf{H_a}\, r(\mathbf{s}, \mathbf{a})$ is zero at the twice differentiable points, it is reasonable to think that the metric learning would fail in such cases. However, since we are dealing with finite samples and using a neural network reward regressor, we conjecture that some form of concave-shaped reward estimate can be learned by the reward regressor and make meaningful metric learning possible. If the conjecture is right and the metric learning is successful, the metric will be learned to have a relatively larger bandwidth in the direction of the dummy action dimension (a_2) that is unrelated to the reward.

Lastly, for the multi-modal reward domain, the multi-modal reward function is made with exponential functions and max operators similar to the multi-modal reward function introduced in the work of Haarnoja et al. [31]. Rewards are deterministic given states and actions. (For the details see Appendix E.) Each dimension of states and actions are independently and uniformly sampled from the range of $[-1, 1]$, and the target policy is $\pi(\mathbf{s}) = \mathbf{s} + \left[\begin{smallmatrix} 0.5 \\ 0 \end{smallmatrix}\right]$. Due to the max operators, there are points in the reward function domain where it is not twice differentiable. For the twice differentiable points, the reward function is designed to have varying Hessians $\mathbf{H_a}\, r(\mathbf{s}, \mathbf{a})$, thus, varying metrics should be learned for varying actions unlike the other synthetic domains. Also, since the Taylor

expansion terms of the reward function w.r.t. the target action have higher order terms beyond 2nd order, the bias of the kernel-based IS estimation without the metric will contain the terms ignored in the derivation of the squared leading-order bias in Eq. (2) (the derivation in Appendix A.1) which upper bound is minimized by the KMIS metric.

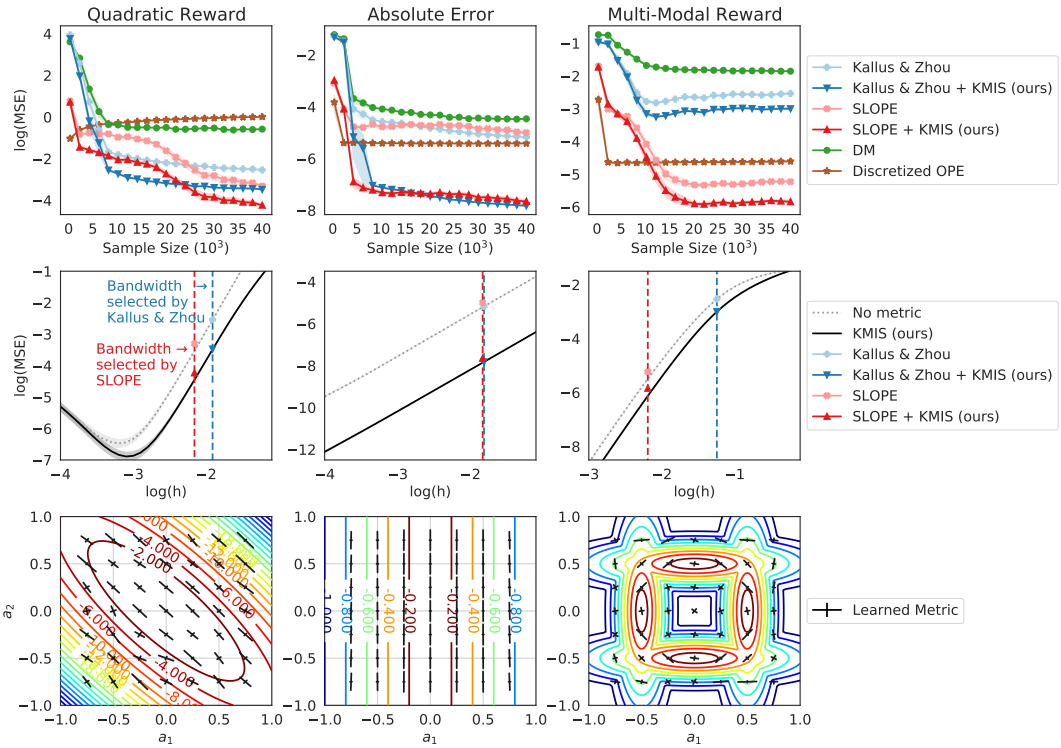


Figure 2: Experimental results on the synthetic domains. Columns starting from the left are results from the quadratic reward domain, absolute error domain, and multi-modal reward domain. The first row shows the performance of estimators as the data size is increased. The second row shows the amount of MSEs reduced by the KMIS metrics given various bandwidths with 40k data points. The second row also marks the results of the first row when the data size used in the estimation is 40k. For both first and second row figures, means and standard errors of squared errors from 100 trials are drawn. The last row visualizes the learned metrics from a trial along with a reward landscape.

The experimental results on the synthetic domains are shown in Figure 2. The first row of Figure 2 shows that for most cases, the KMIS metric reduces the MSEs of the kernel-based IS estimators. Also, the KMIS metric applied kernel-based IS estimators outperform DM, which is used by the KMIS algorithm to estimate the Hessians of a reward function for the metric learning. For the multi-modal reward domain, even though the bias contains terms ignored in the leading-order bias, the metrics that minimize the upper bound of squared leading-order bias also reduce MSE. The discretized OPE estimator performs worst in the quadratic reward domain for the sample size above 10k, but it performs better than some other estimators in the other domains. As it is unclear how to discretize the multi-dimensional action space [2], even though the same discretization rule is used for all synthetic domains, its performance varies from domain to domain.

The second row of Figure 2 shows that for most of the given bandwidths, the MSEs of the kernel-based IS estimations are reduced by the KMIS metrics. The gray dotted line denotes the MSEs of kernel-based IS estimators with given bandwidths and without a metric. The black line denotes the KMIS metric applied kernel-based IS estimators with given bandwidths. The dotted vertical lines present the average of the selected bandwidths by SLOPE (red) or Kallus and Zhou's estimator (blue), and the markers show the average MSEs of the estimators at the average bandwidths when 40k data points are used for estimation. The average MSEs of KMIS metric applied estimators with the same 40k data points are also marked. Since the selected bandwidths can vary for each run, the markers may not exactly lie on the black lines denoting the results of the fixed bandwidths.

The third row of Figure 2 shows learned KMIS metrics from a trial and the reward landscape when $\mathbf{s} = \left[\begin{smallmatrix}0\\0\end{smallmatrix}\right]$ for each synthetic domain. The learned metrics are drawn with black crosses. The metrics are not only drawn for a target action but also for other actions. For the metrics learned in the quadratic reward domain, the learned metrics are ellipsoidal and similar for most actions. Therefore, we can see that our metric is learned as intended with the Hessian $\mathbf{H_a}\, r(\mathbf{s}, \mathbf{a})$ provided by the reward regressor (DM). For the absolute error domain, the metrics are learned to ignore the dummy action dimension a_2 by having longer relative bandwidth in the direction of the action dimension while having shorter relative bandwidth in the direction of a_1 which is used for computing the reward. This result empirically verifies our conjecture made earlier, which suggests that our metric learning algorithm can be generally applied to the dataset where the true reward function is not twice differentiable at some actions, and the true Hessian is zero at the twice differentiable actions. The metrics learned on the multi-modal reward made with exponents verifies that our metric indeed learns varying metrics according to actions.

To empirically validate Proposition 1, the empirical squared bias, variance, and MSE ($\mathrm{MSE}[\widehat{\rho}^K] = \mathrm{Bias}[\widehat{\rho}^K]^2 + Var[\widehat{\rho}^K]$) of the Kallus and Zhou's estimator and the KMIS metric applied version of the estimator is observed on the modified absolute error domain with various number of dummy action dimensions. The modified domain can have additional dummy action dimensions where the value of each dummy dimension is independently and uniformly sampled in the range of $[-1, 1]$. The result in Figure 3 shows that the empirical squared bias of Kallus and Zhou's estimator is indeed the dominant term in the MSE in high action dimensions. As the action dimension is increased, both algorithms suffer from high bias. But the KMIS metric reduces the bias and shows lower MSE than the one without the metric. However, the KMIS metric applied estimator shows higher variance than the Kallus and Zhou's estimator. This increase in the variance due to the metric may come from the ignored components of the variance, or, the Hessian estimation error.

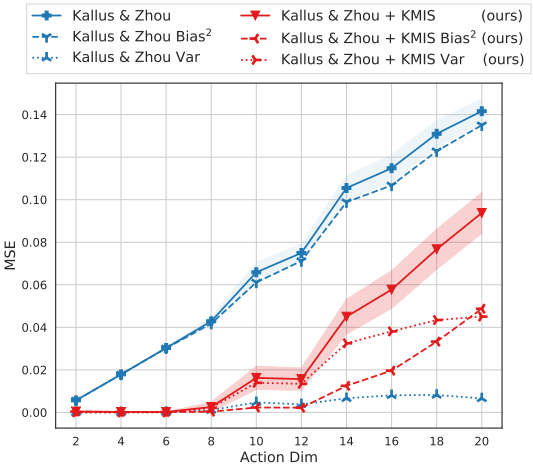
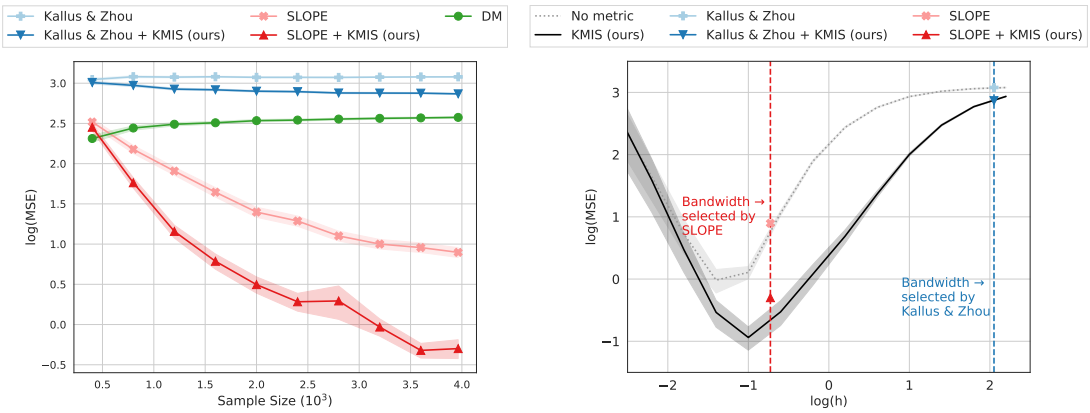


Figure 3: OPE performance on the modified absolute error domain with various number of action dimensions. Means and standard errors of squared errors were obtained from 100 trials with 40k samples. Empirical squared bias and variance of the estimates are also drawn.

5.2 Warfarin Data

To test our algorithm on a more realistic dataset, we test our algorithm on the Warfarin dataset [30]. Warfarin is a treatment that is commonly used for preventing blood clots. The dataset contains information on the patients, therapeutic doses, and the resulting outcomes. The outcomes of the treatments are reported in the international normalized ratio (INR), which measures how quickly the blood clots. We use a similar experimental setting used by Kallus and Zhou [2]. The 81 features of the patient information selected by Kallus and Zhou [2] is used as states $\mathbf{s}$. For the behavior action vectors, the first dimension is sampled from the normal distribution $N(\mu^* + \sigma^*\sqrt{0.5}z_{BMI}, (\sigma^*\sqrt{0.5})^2)$ truncated by the minimum and maximum therapeutic doses $a^*_{\min}$ and $a^*_{\max}$ in the data. z_{BMI} is the z score of patients' BMIs, μ^* and σ^* are the mean and the standard deviation of the therapeutic dosages, respectively. To test our algorithm we add a dummy action dimension sampled from a uniform distribution $a_2 \sim \mathrm{unif}[a^*_{\min}, a^*_{\max}]$. For the reward, since Kallus and Zhou [2] reported that the INR in the dataset is inadequate for testing OPE algorithms, we use their cost function as a negative reward function $r = -\max(|a_1 - a^*| - 0.1a^*, 0)$. The reward function is only dependent on the first action dimension. The target policy is $\pi(\mathbf{s}) = \left[\begin{smallmatrix}s_{BMI}\\0\end{smallmatrix}\right]$.

Experimental results in Figure 4a show that our metric reduces the MSE of the kernel-based IS estimate in Warfarin data that is more realistic than that of the synthetic domain. In the figure, we see that even though the MSE of the reward regressor is high, MSE of SLOPE and Kallus and Zhou's estimator is reduced with the KMIS metric. Similar to the results in the synthetic domains, the KMIS metric reduces MSE for almost all given bandwidths in Figure 4b.



(a) (b)

Figure 4: Experimental results on the Warfarin Data. (a) Performance of estimators on the Warfarin Data with increasing data size. The experiment is repeated for 300 trials, and means and standard errors of the squared errors are reported. (b) Reduction of MSEs by metrics at given bandwidths. The experiment is repeated for 100 trials. Means and standard errors of the squared errors are reported. The dotted vertical lines present the average of the bandwidths selected by Kallus and Zhou's estimator (blue) or SLOPE (red), and the markers show the average of the selected bandwidths and the MSEs of each estimator when all 3964 samples are used to obtain the OPE results. Since the selected bandwidth can vary for each run, the markers may not exactly lie on the black lines denoting the fixed bandwidth's results.

6 Conclusion

We presented KMIS, an algorithm for off-policy evaluation in contextual bandits for a deterministic target policy with multidimensional continuous action space. KMIS improves the kernel-based IS method by learning a local distance metric used in the kernel function per state, leading to a kernel-based IS OPE estimator that exploits Mahalanobis distance, rather than Euclidean distance. Based on the observation that the leading-order bias becomes a dominant term in the LOMSE when the action dimensionality is high, we derived an analytic solution to the optimal metric matrix that minimizes the upper bound of the leading-order bias. The optimal metric matrix is bandwidth-agnostic and computed using the Hessian of the learned reward function. Experimental results demonstrated that our KMIS significantly improves the performance of the kernel-based IS OPE across different bandwidth selection methods, outperforming baseline algorithms. As for future work, exploring algorithms that jointly optimize bandwidth and metric or extending to RL beyond contextual bandits case would be interesting directions to pursue.

Acknowledgments and Disclosure of Funding

Haanvid Lee, Yunseon Choi, and Kee-Eung Kim were supported by National Research Foundation (NRF) of Korea (NRF-2019R1A2C1087634), Field-oriented Technology Development Project for Customs Administration through National Research Foundation (NRF) of Korea funded by the Ministry of Science & ICT and Korea Customs Service (NRF-2021M3I1A1097938), Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2020-0-00940, Foundations of Safe Reinforcement Learning and Its Applications to Natural Language Processing; No.2022-0-00311, Development of Goal-Oriented Reinforcement Learning Techniques for Contact-Rich Robotic Manipulation of Everyday Objects; No.2019-0-00075, Artificial Intelligence Graduate School Program (KAIST); No.2021-0-02068, Artificial Intelligence Innovation Hub), and Electronics and Telecommunications Research Institute (ETRI) grant funded by the Korean government (22ZS1100, Core Technology Research for Self-Improving Integrated Artificial Intelligence System), and KAIST-NAVER Hypercreative AI Center. Byung-Jun Lee was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.2019-0-00079 , Artificial Intelligence Graduate School Program(Korea University)).

References

[1] Sheng Qiang and Mohsen Bayati. Dynamic pricing with demand covariates. *Available at SSRN 2765257*, 2016.

[2] Nathan Kallus and Angela Zhou. Policy evaluation and optimization with continuous treatments. In *International conference on artificial intelligence and statistics*, pages 1243–1251. PMLR, 2018.

[3] James McInerney, Benjamin Lacker, Samantha Hansen, Karl Higley, Hugues Bouchard, Alois Gruson, and Rishabh Mehrotra. Explore, exploit, and explain: personalizing explainable recommendations with bandits. In *Proceedings of the 12th ACM conference on recommender systems*, pages 31–39, 2018.

[4] Yuta Saito, Shunsuke Aihara, Megumi Matsutani, and Yusuke Narita. Open bandit dataset and pipeline: Towards realistic and reproducible off-policy evaluation. *arXiv preprint arXiv:2008.07146*, 2020.

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

[6] Miroslav Dudík, John Langford, and Lihong Li. Doubly robust policy evaluation and learning. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, page 1097–1104, Madison, WI, USA, 2011. Omnipress. ISBN 9781450306195.

[7] Lihong Li, Wei Chu, John Langford, and Xuanhui Wang. Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 297–306, 2011.

[8] Lu Wang, Andrea Rotnitzky, Xihong Lin, Randall E Millikan, and Peter F Thall. Evaluation of viable dynamic treatment regimes in a sequentially randomized trial of advanced prostate cancer. *Journal of the American Statistical Association*, 107(498):493–508, 2012.

[9] Mehrdad Farajtabar, Yinlam Chow, and Mohammad Ghavamzadeh. More robust doubly robust off-policy evaluation. In *International Conference on Machine Learning*, pages 1447–1456. PMLR, 2018.

[10] Miroslav Dudík, Dumitru Erhan, John Langford, and Lihong Li. Doubly robust policy evaluation and optimization. *Statistical Science*, 29(4):485–511, 2014.

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

[12] Doina Precup. *Temporal abstraction in reinforcement learning*. University of Massachusetts Amherst, 2000.

[13] 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.

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

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

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

[17] Hengrui Cai, Chengchun Shi, Rui Song, and Wenbin Lu. Deep jump learning for off-policy evaluation in continuous treatment settings. *Advances in Neural Information Processing Systems*, 34, 2021.

[18] Nashwa Masnoon, Sepehr Shakib, Lisa Kalisch-Ellett, and Gillian E Caughey. What is polypharmacy? a systematic review of definitions. *BMC geriatrics*, 17(1):1–10, 2017.

[19] Cairns S Smith, Ann Aerts, Paul Saunderson, Joseph Kawuma, Etsuko Kita, and Marcos Virmond. Multidrug therapy for leprosy: a game changer on the path to elimination. *The Lancet Infectious Diseases*, 17(9):e293–e297, 2017.

[20] Prasanta Chandra Mahalanobis. On the generalized distance in statistics. In *Proceedings of the National Institute of Sciences of India*, 1936.

[21] Roy De Maesschalck, Delphine Jouan-Rimbaud, and Désiré L Massart. The mahalanobis distance. *Chemometrics and intelligent laboratory systems*, 50(1):1–18, 2000.

[22] Yung-Kyun Noh, Byoung-Tak Zhang, and Daniel D Lee. Generative local metric learning for nearest neighbor classification. In *Advances in neural information processing systems*, pages 1822–1830, 2010.

[23] Yung-Kyun Noh, Masashi Sugiyama, Kee-Eung Kim, Frank Park, and Daniel D Lee. Generative local metric learning for kernel regression. *Advances in Neural Information Processing Systems*, 30, 2017.

[24] Kyle Colangelo and Ying-Ying Lee. Double debiased machine learning nonparametric inference with continuous treatments. *arXiv preprint arXiv:2004.03036*, 2020.

[25] Oleg V Lepski and Vladimir G Spokoiny. Optimal pointwise adaptive methods in nonparametric estimation. *The Annals of Statistics*, 25(6):2512–2546, 1997.

[26] Aurélien Bellet, Amaury Habrard, and Marc Sebban. A survey on metric learning for feature vectors and structured data. *arXiv preprint arXiv:1306.6709*, 2013.

[27] Jason V Davis, Brian Kulis, Prateek Jain, Suvrit Sra, and Inderjit S Dhillon. Information-theoretic metric learning. In *Proceedings of the 24th international conference on Machine learning*, pages 209–216, 2007.

[28] Jacob Goldberger, Geoffrey E Hinton, Sam Roweis, and Russ R Salakhutdinov. Neighbourhood components analysis. *Advances in neural information processing systems*, 17, 2004.

[29] Bac Nguyen, Carlos Morell, and Bernard De Baets. Large-scale distance metric learning for k-nearest neighbors regression. *Neurocomputing*, 214:805–814, 2016.

[30] International Warfarin Pharmacogenetics Consortium. Estimation of the warfarin dose with clinical and pharmacogenetic data. *New England Journal of Medicine*, 360(8):753–764, 2009.

[31] Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement learning with deep energy-based policies. In *International Conference on Machine Learning*, pages 1352–1361. PMLR, 2017.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]

(b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]

(c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]

(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

   (a) If your work uses existing assets, did you cite the creators? [Yes]

   (b) Did you mention the license of the assets? [N/A]

   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes]

   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]

   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...

   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Derivation Details

A.1 Leading-Order MSE

Bias of the Kernel-Based IS Estimation [2]

Define $\mathbb{E}_{\pi_b}[\cdot] := \mathbb{E}_{\mathbf{s}\sim \mathrm{p}(\mathbf{s}),\mathbf{a}\sim\pi_\mathrm{b}(\mathbf{a}|\mathbf{s}),\mathrm{r}\sim\mathrm{p}(\mathrm{r}|\mathbf{s},\mathbf{a})}[\cdot]$, then the bias of a kernel-based IS estimate is:

$$\begin{aligned}\mathrm{Bias}[\hat{\rho}^K] &= \mathbb{E}_{\pi_b}[\hat{\rho}^K] - \rho^{\pi_e} \\ &= \mathbb{E}_{\pi_b}\left[\frac{1}{Nh^{D_A}}\sum_{i=1}^{N} K\left(\frac{\mathbf{a}_i - \pi(\mathbf{s}_i)}{h}\right)\frac{r_i}{\pi_b(\mathbf{a}_i \mid \mathbf{s}_i)}\right] - \rho^{\pi}\end{aligned} \tag{12}$$

For the first term of Eq. (12),

$$\begin{aligned}&\mathbb{E}_{\pi_b}\left[\frac{1}{Nh^{D_A}}\sum_{i=1}^{N} K\left(\frac{\mathbf{a}_i - \pi(\mathbf{s}_i)}{h}\right)\frac{r_i}{\pi_b(\mathbf{a}_i \mid \mathbf{s}_i)}\right] \\ &= \frac{1}{h^{D_A}}\iiint p(\mathbf{s})p(r|\,\mathbf{s},\mathbf{a})K\left(\frac{\mathbf{a}-\pi(\mathbf{s})}{h}\right) r dr d\mathbf{a} d\mathbf{s} \\ &= \iiint rp(\mathbf{s})\left(p(r|\mathbf{s},\pi_e(\mathbf{s})) + \frac{h^2}{2}\mathbf{u}^\top\,\mathbf{H}_\mathbf{a}\,p(r|\mathbf{s},\mathbf{a})|_{\mathbf{a}=\pi_e(\mathbf{s})}\,\mathbf{u} + O\left(h^4\right)\right)K(\mathbf{u})d\mathbf{u}drd\mathbf{s} \\ &= \rho^\pi + \frac{h^2}{2}\iint rp(s)\,\mathrm{tr}\left(\left[\int \mathbf{u}\,\mathbf{u}^\top K(\mathbf{u})d\,\mathbf{u}\right]\mathbf{H}_\mathbf{a}\,p(r|\mathbf{s},\mathbf{a})|_{\mathbf{a}=\pi(\mathbf{s})}\right)drd\,\mathbf{s} + O\left(h^4\right) \\ &= \rho^\pi + \frac{h^2}{2}\iint rp(\mathbf{s})\,\nabla^2_\mathbf{a}p(r|\mathbf{s},\mathbf{a})\big|_{\mathbf{a}=\pi(\mathbf{s})}\,drd\,\mathbf{s} + O\left(h^4\right)\end{aligned}$$

$$\therefore \mathrm{Bias}[\hat{\rho}^K] = \frac{h^2}{2}\iint rp(\mathbf{s})\,\nabla^2_\mathbf{a}p(r|\mathbf{s},\mathbf{a})\big|_{\mathbf{a}=\pi(\mathbf{s})}\,drd\,\mathbf{s} + O\left(h^4\right) \tag{13}$$

where the following relations are used in the derivation,

$$\begin{aligned}&\int K(\mathbf{u})d\mathbf{u} = 1, \\ &\int \mathbf{u}K(\mathbf{u})d\mathbf{u} = 0, \\ &\kappa_2(K) := \int \mathbf{u}\mathbf{u}^\top K(\mathbf{u})d\mathbf{u} = I \text{ (By design on } K\text{)}, \\ &\mathbf{u} := \frac{\mathbf{a}-\pi(\mathbf{s})}{h}, \\ &\mathrm{h}^{D_A}d\mathbf{u} = d\mathbf{a},\end{aligned}$$

Taylor expansion of $p(r|\mathbf{s},\mathbf{a})$ at $\mathbf{a} = \pi(\mathbf{s})$,

$$\begin{aligned}p(r|\,\mathbf{s},\mathbf{a}) = p(r|\,\mathbf{s},\pi(\mathbf{s})) &+ (\mathbf{a}-\pi(\mathbf{s}))^\top\nabla_\mathbf{a}p(r|\,\mathbf{s},\mathbf{a})|_{\mathbf{a}=\pi(\mathbf{s})} \\ &+ \frac{1}{2}(\mathbf{a}-\pi(\mathbf{s}))^\top\,\mathbf{H}_\mathbf{a}\,p(r|\,\mathbf{s},\mathbf{a})|_{\mathbf{a}=\pi(\mathbf{s})}(\mathbf{a}-\pi(\mathbf{s})) + \ldots\end{aligned}$$

using the relation $h\mathbf{u} = \mathbf{a} - \pi(\mathbf{s})$,

$$\begin{aligned}p(r|\,\mathbf{s},\mathbf{a}) = p(r|\,\mathbf{s},\pi(\mathbf{s})) &+ h\mathbf{u}^\top\nabla_\mathbf{a}p(r|\,\mathbf{s},\mathbf{a})|_{\mathbf{a}=\pi(\mathbf{s})} \\ &+ \frac{h^2}{2}\mathbf{u}^\top\,\mathbf{H}_\mathbf{a}\,p(r|\,\mathbf{s},\mathbf{a})|_{\mathbf{a}=\pi(\mathbf{s})}\mathbf{u} + \ldots\end{aligned}$$

Variance of the Kernel-Based IS Estimation [2]

Define $\mathbb{E}_{\pi_b}[\cdot] := \mathbb{E}_{\mathbf{s}\sim \mathrm{p}(\mathbf{s}),\mathbf{a}\sim\pi_\mathrm{b}(\mathbf{a}|\mathbf{s}),\mathrm{r}\sim\mathrm{p}(\mathrm{r}|\mathbf{s},\mathbf{a})}[\cdot]$, then the variance of a kernel-based IS estimate is:

$$\begin{aligned}
\mathrm{Var}[\hat{\rho}^K] &= \mathrm{Var}\left[\frac{1}{Nh^{D_A}}\sum_{i=1}^{N} K\left(\frac{\mathbf{a}_i - \pi(\mathbf{s}_i)}{h}\right)\frac{r_i}{\pi_b(\mathbf{a}_i\,|\,\mathbf{s}_i)}\right] \\
&= \frac{1}{N^2}\times N \times \mathrm{Var}\left[\frac{1}{h^{D_A}}K\left(\frac{\mathbf{a}-\pi(\mathbf{s})}{h}\right)\frac{r}{\pi_b(\mathbf{a}\,|\,\mathbf{s})}\right] \\
&= \frac{1}{N}\left\{\mathbb{E}_{\pi_b}\left[\left(\frac{1}{h^{D_A}}K\left(\frac{\mathbf{a}-\pi(\mathbf{s})}{h}\right)\frac{r}{\pi_b(\mathbf{a}\,|\,\mathbf{s})}\right)^2\right]\right. \\
&\qquad\qquad \left. - \left(\mathbb{E}_{\pi_b}\left[\frac{1}{h^{D_A}}K\left(\frac{\mathbf{a}-\pi(\mathbf{s})}{h}\right)\frac{r}{\pi_b(\mathbf{a}\,|\,\mathbf{s})}\right]\right)^2\right\}
\end{aligned} \tag{14}$$

The second term of the Eq. (14) is,

$$\begin{aligned}
&\frac{1}{N}\left(\mathbb{E}_{\pi_b}\left[\frac{1}{h^{D_A}}K\left(\frac{\mathbf{a}-\pi(\mathbf{s})}{h}\right)\frac{r}{\pi_b(\mathbf{a}\,|\,\mathbf{s})}\right]\right)^2 \\
&\quad = \frac{1}{N}\left(\mathrm{Bias}(\hat{\rho}^K) + \rho^\pi\right)^2 \\
&\quad = \frac{1}{N}\left[\rho^\pi + \frac{h^2}{2}\iint rp(\mathbf{s})\,\nabla^2_{\mathbf{a}} p(r|\mathbf{s},\mathbf{a})\big|_{\mathbf{a}=\pi(\mathbf{s})}\,dr\,d\,\mathbf{s} + O\left(h^4\right)\right]^2 \\
\therefore\ &\frac{1}{N}\left(\mathbb{E}_{\pi_b}\left[\frac{1}{h^{D_A}}K\left(\frac{\mathbf{a}-\pi(\mathbf{s})}{h}\right)\frac{r}{\pi_b(\mathbf{a}\,|\,\mathbf{s})}\right]\right)^2 = O\left(\frac{1}{N}\right)
\end{aligned} \tag{15}$$

The first term of the Eq. (14) is,

$$\begin{aligned}
&\frac{1}{N}\mathbb{E}_{\pi_b}\left[\left(\frac{1}{h^{D_A}}K\left(\frac{\mathbf{a}-\pi(\mathbf{s})}{h}\right)\frac{r}{\pi_b(\mathbf{a}\,|\,\mathbf{s})}\right)^2\right] \\
&= \frac{1}{N}\mathbb{E}_{\mathbf{s}\sim p(\mathbf{s})}\left[\iint r^2\frac{p(r|\,\mathbf{s},\mathbf{a})}{h^{2D_A}\pi_b(\mathbf{a}\,|\,\mathbf{s})}K\left(\frac{\mathbf{a}-\pi(\mathbf{s})}{h}\right)^2 d\,\mathbf{a}\,dr\right]
\end{aligned}$$

By substituting $\mathbf{a} = h\,\mathbf{u} + \pi(\mathbf{s})$ $\Big($ which comes from $\mathbf{u} = \frac{\mathbf{a}-\pi(\mathbf{s})}{h}\Big)$,

$$= \frac{1}{N}\mathbb{E}_{\mathbf{s}\sim p(\mathbf{s})}\left[\iint r^2\frac{K(\mathbf{u})^2}{h^{D_A}}\frac{p(r|\,\mathbf{s},h\,\mathbf{u}+\pi(\mathbf{s}))}{\pi_b(h\,\mathbf{u}+\pi(\mathbf{s})|\,\mathbf{s})}d\,\mathbf{u}\,dr\right] \tag{16}$$

Let $g_r(\mathbf{s},\mathbf{a}) := \frac{p(r|\,\mathbf{s},\mathbf{a})}{\pi_b(\mathbf{a}\,|\,\mathbf{s})}$, and apply Taylor expansion at $\mathbf{a} = \pi(\mathbf{s})$,

$$\begin{aligned}
g_r(\mathbf{s},\mathbf{a}) = g_r(\mathbf{s},\pi(\mathbf{s})) &+ (\mathbf{a}-\pi(\mathbf{s}))^\top \nabla_{\mathbf{a}} g_r(\mathbf{s},\mathbf{a})\,|_{\mathbf{a}=\pi(\mathbf{s})} \\
&+ \frac{1}{2}(\mathbf{a}-\pi(\mathbf{s}))^\top\,\mathbf{H}_{\mathbf{a}}\,g_r(\mathbf{s},\mathbf{a})\,|_{\mathbf{a}=\pi(\mathbf{s})}\,(\mathbf{a}-\pi(\mathbf{s})) + O(h^3)
\end{aligned}$$

substitute $\mathbf{a} = h\,\mathbf{u} + \pi(\mathbf{s})$,

$$g_r(\mathbf{s}, h\,\mathbf{u} + \pi(\mathbf{s})) = g_r(\mathbf{s}, \pi(\mathbf{s})) + h\mathbf{u}^\top \nabla_a g_r(\mathbf{s}, \mathbf{a}) \mid_{\mathbf{a}=\pi(\mathbf{s})} + \frac{h^2}{2}\mathbf{u}^\top \mathbf{H}_\mathbf{a}\, g_r(\mathbf{s}, \mathbf{a}) \mid_{\mathbf{a}=\pi(\mathbf{s})} \mathbf{u} + O(h^3) \tag{17}$$

Plugging in Eq. (17) to Eq. (16),

$$\begin{aligned}
&\frac{1}{N}\mathbb{E}_{\mathbf{s}\sim p(\mathbf{s})}\left[\iint r^2 \frac{K(\mathbf{u})^2}{h^{D_A}} \frac{p(r|\,\mathbf{s}, h\,\mathbf{u} + \pi_e(\mathbf{s}))}{\pi_b(h\,\mathbf{u} + \pi(\mathbf{s})|\,\mathbf{s})} d\,\mathbf{u}\, dr\right] \\
&= \frac{1}{Nh^{D_A}}\mathbb{E}_{\mathbf{s}\sim p(\mathbf{s})}\left[\iint r^2 K(\mathbf{u})^2 \left(g_r(\mathbf{s}, \pi(\mathbf{s})) + \frac{h^2}{2}\mathbf{u}^\top \mathbf{H}_\mathbf{a}\, g_r(\mathbf{s}, \mathbf{a}) \mid_{\mathbf{a}=\pi(\mathbf{s})} \mathbf{u} + O(h^4)\right) d\,\mathbf{u}\, dr\right] \\
&= \frac{R(K)}{Nh^{D_A}}\mathbb{E}_{\mathbf{s}\sim p(\mathbf{s})}\left[\int_R \frac{r^2 p(r|\,\mathbf{s}, \mathbf{a} = \pi(\mathbf{s}))}{\pi_b(\mathbf{a} = \pi(\mathbf{s})|\,\mathbf{s})} dr\right] \\
&\qquad + \frac{1}{2Nh^{D_A-2}}\mathbb{E}_{\mathbf{s}\sim p(\mathbf{s})}\left[\int_R \int r^2 K(\mathbf{u})^2 \left(\mathbf{u}^\top \mathbf{H}_\mathbf{a}\, g_r(\mathbf{s}, \mathbf{a}) \mid_{\mathbf{a}=\pi(\mathbf{s})} \mathbf{u} + O(h^4)\right) d\,\mathbf{u}\, dr\right] \\
&\qquad \left(R(K) := \int K(\mathbf{u})^2 d\,\mathbf{u}\right) \\
&= \frac{R(K)}{Nh^{D_A}}\mathbb{E}_{\mathbf{s}\sim p(\mathbf{s})}\left[\int \frac{r^2 p(r|\,\mathbf{s}, \mathbf{a} = \pi(\mathbf{s}))}{\pi_b(\mathbf{a} = \pi(\mathbf{s})|\,\mathbf{s})} dr\right] + O\left(\frac{1}{Nh^{D_A-2}}\right)
\end{aligned} \tag{18}$$

From Eq. (15) and Eq. (18), the variance can be represented as,

$$\therefore \mathrm{Var}[\hat{\rho}^K] = \frac{R(K)}{Nh^{D_A}}\mathbb{E}_{\mathbf{s}\sim p(\mathbf{s})}\left[\frac{\mathbb{E}\left[r^2|\,\mathbf{s}, \mathbf{a} = \pi(\mathbf{s})\right]}{\pi_b(\mathbf{a} = \pi(\mathbf{s})|\,\mathbf{s})}\right] + O\left(\frac{1}{Nh^{D_A-2}}\right) \tag{19}$$

From the derived bias (Eq. (13)) and variance (Eq. (19)), MSE can be derived:

$$\begin{aligned}
\mathrm{MSE}[\hat{\rho}^K] &= \mathrm{Bias}[\hat{\rho}^K]^2 + \mathrm{Var}[\hat{\rho}^K] \\
&= \left(\frac{h^2}{2}\iint rp(\mathbf{s})\, \nabla^2_\mathbf{a} p(r|\mathbf{s}, \mathbf{a})\big|_{\mathbf{a}=\pi(\mathbf{s})}\, dr d\,\mathbf{s}\right)^2 + O\left(h^6\right) \\
&\qquad + \frac{R(K)}{Nh^{D_A}}\mathbb{E}_{\mathbf{s}\sim p(\mathbf{s})}\left[\frac{\mathbb{E}\left[r^2|\,\mathbf{s}, \mathbf{a} = \pi(\mathbf{s})\right]}{\pi_b(\mathbf{a} = \pi(\mathbf{s})|\,\mathbf{s})}\right] + O\left(\frac{1}{Nh^{D_A-2}}\right)
\end{aligned}$$

Assuming that $h \to 0$ and $\frac{1}{Nh^{D_A}} \to 0$ as $N \to \infty$,

$$\begin{aligned}
\mathrm{MSE}[\hat{\rho}^K] &\approx h^4 C_b + \frac{C_v}{Nh^{D_A}} \\
&=: \mathrm{LOMSE}[h, N, D_A] \\
C_b &:= \left(\frac{1}{2}\mathbb{E}_{s\sim p(s)}\left[\nabla^2_a \left(\mathbb{E}[r|s, a]\right) \mid_{\mathbf{a}=\pi(\mathbf{s})}\right]\right)^2, C_v := R(K)\mathbb{E}_{s\sim p(s)}\left[\frac{\mathbb{E}[r^2|s, a = \pi(s)]}{\pi_b(a = \pi(s)|s)}\right]
\end{aligned}$$

A.2 Optimal Bandwidth

The optimal bandwidth (h^*) that minimizes the leading-order MSE [2] is,

$$
\begin{aligned}
&\frac{d}{dh}(\text{LOMSE}(h, N, D_A)) = 4h^3 C_b - D_A h^{-D_A-1} N^{-1} C_v \\
&\quad 4(h^*)^3 C_b - D_A (h^*)^{-D_A-1} N^{-1} C_v = 0 \\
&\quad h^* = \left(\frac{D_A C_v}{N 4 C_b}\right)^{\frac{1}{D_A+4}}
\end{aligned}
$$

A.3 Derivation of Eq. (6)

For the derivation, we use the following relations:

$$
\begin{aligned}
p(r \mid \mathbf{s}, \mathbf{z}) &= \frac{p(r, \mathbf{s}, \mathbf{z})}{p(\mathbf{s}, \mathbf{z})} \\
&= \frac{p(r, \mathbf{s}, \mathbf{a}) \left|\frac{\partial \mathbf{a}}{\partial \mathbf{z}}\right|}{p(\mathbf{s}, \mathbf{a}) \left|\frac{\partial \mathbf{a}}{\partial \mathbf{z}}\right|} \\
&= p(r \mid \mathbf{s}, \mathbf{a})
\end{aligned} \tag{20}
$$

$$
\begin{aligned}
\mathbf{H}_{\mathbf{z}}\, p(r \mid \mathbf{s}, \mathbf{z})|_{\mathbf{z}=L(\mathbf{s})^\top \pi(\mathbf{s})} &= \frac{\partial^2}{\partial \mathbf{z}\, \partial \mathbf{z}^\top} p(r \mid \mathbf{s}, \mathbf{z}) \\
&= \frac{\partial}{\partial \mathbf{z}} \left(\frac{\partial}{\partial \mathbf{z}} p(r \mid \mathbf{s}, \mathbf{z})\right)^\top \\
&= \frac{\partial}{\partial \mathbf{z}} \left(L(\mathbf{s})^{-1} \frac{\partial}{\partial \mathbf{a}} p(r \mid \mathbf{s}, \mathbf{a})\right)^\top, \text{by Eq. (20)} \\
&= \frac{\partial \mathbf{a}}{\partial \mathbf{z}} \frac{\partial}{\partial \mathbf{a}} \left(\left(\frac{\partial}{\partial \mathbf{a}} p(r \mid \mathbf{s}, \mathbf{a})\right)^\top L(\mathbf{s})^\top\right) \\
&= L(\mathbf{s})^{-1} \frac{\partial^2}{\partial \mathbf{a}\, \partial \mathbf{a}^\top} p(r \mid \mathbf{s}, \mathbf{a}) L(\mathbf{s})^{-\top} \\
&= L(\mathbf{s})^{-1}\, \mathbf{H}_{\mathbf{a}}\, p(r \mid \mathbf{s}, \mathbf{a})\big|_{\mathbf{a}=\pi(\mathbf{s})}\, L^{-\top}(\mathbf{s})
\end{aligned} \tag{21}
$$

By changing $\mathbf{a}$ of C_b in Eq. (2) to $\mathbf{z}$ ($= L(\mathbf{s})^\top \mathbf{a}$), we get:

$$
\begin{aligned}
C_{b,A} &= \left(\frac{1}{2} \mathbb{E}_{\mathbf{s}\sim p(\mathbf{s})} \left[\nabla^2_{\mathbf{z}} r(\mathbf{s}, \mathbf{z})\,|_{\mathbf{z}=L(\mathbf{s})^\top \pi(\mathbf{s})}\right]\right)^2 \\
&= \left(\frac{1}{2} \iint r p(\mathbf{s})\, \mathrm{tr}\left(\mathbf{H}_{\mathbf{z}}\, p(r \mid \mathbf{s}, \mathbf{z})|_{\mathbf{z}=L(\mathbf{s})^\top \pi(\mathbf{s})}\right) dr d\mathbf{s}\right)^2 \\
&= \left(\frac{1}{2} \mathbb{E}_{\mathbf{s}\sim p(\mathbf{s})} \left[\mathrm{tr}\left(L(\mathbf{s})^{-1}\, \mathbf{H}_{\mathbf{a}}\, r(\mathbf{s}, \mathbf{a})\big|_{\mathbf{a}=\pi(\mathbf{s})}\, L(\mathbf{s})^{-\top}\right)\right]\right)^2, \text{ by Eq. (21)} \\
&= \left(\frac{1}{2} \mathbb{E}_{\mathbf{s}\sim p(\mathbf{s})} \left[\mathrm{tr}\left((L(\mathbf{s}) L(\mathbf{s})^\top)^{-1}\, \mathbf{H}_{\mathbf{a}}\, r(\mathbf{s}, \mathbf{a})\big|_{\mathbf{a}=\pi(\mathbf{s})}\right)\right]\right)^2 \\
\therefore C_{b,A} &= \left(\frac{1}{2} \mathbb{E}_{\mathbf{s}\sim p(\mathbf{s})} \left[\mathrm{tr}\left(A(\mathbf{s})^{-1}\, \mathbf{H}_{\mathbf{a}}\, r(\mathbf{s}, \mathbf{a})\big|_{\mathbf{a}=\pi(\mathbf{s})}\right)\right]\right)^2
\end{aligned}
$$

B Proofs

B.1 Proof of Proposition 1

Proposition 1. *(Adapted from Noh et al. [23]) For a high dimensional action space $D_A \gg 4$, and given optimal bandwidth h^* (Eq.* (3)*), the squared leading-order bias dominates over the leading-order variance in LOMSE. Furthermore,* $\mathrm{LOMSE}(h^*, N, D_A)$ *can be approximated by C_b in Eq.* (2)*.*

$$\mathrm{LOMSE}(h^*, N, D_A) = N^{-\frac{4}{D_A+4}} \left(\left(\frac{D_A}{4}\right)^{\frac{4}{D_A+4}} + \left(\frac{4}{D_A}\right)^{\frac{D_A}{D_A+4}} \right) C_b^{\frac{D_A}{D_A+4}} C_v^{\frac{4}{D_A+4}} \approx C_b. \tag{22}$$

Proof. By plugging in h^* in Eq. (3) to the leading-order bias and leading-order variance in Eq. (2) and by taking $D_A \to \infty$ (or, for $D_A \gg 4$), their ratio is:

$$\lim_{D_A \to \infty} \frac{(\text{leading-order bias})^2}{\text{leading-order var}} = \lim_{D_A \to \infty} \left(\frac{D_A}{4}\right) = \infty \tag{23}$$

Therefore, the squared leading-order bias dominates over the leading-order variance in the LOMSE in the high dimensional action space ($D_A \gg 4$).

By plugging in h^* Eq. (3) to the LOMSE in Eq. (2) we get,

$$\mathrm{LOMSE}(h^*, N, D_A) = N^{-\frac{4}{D_A+4}} \left(\left(\frac{D_A}{4}\right)^{\frac{4}{D_A+4}} + \left(\frac{4}{D_A}\right)^{\frac{D_A}{D_A+4}} \right) C_b^{\frac{D_A}{D_A+4}} C_v^{\frac{4}{D_A+4}}$$

By taking $D_A \to \infty$ (or, for $D_A \gg 4$),

$$\begin{aligned}
\lim_{D_A \to \infty} N^{-\frac{4}{D_A+4}} &= 1 \\
\lim_{D_A \to \infty} \left(\frac{4}{D_A}\right)^{\frac{4}{D_A+4}} &= 0 \\
\lim_{D_A \to \infty} C_b^{\frac{D_A}{D_A+4}} &= C_b \\
\lim_{D_A \to \infty} C_v^{\frac{4}{D_A+4}} &= 1 \\
\lim_{D_A \to \infty} \left(\frac{D_A}{4}\right)^{\frac{4}{D_A+4}} &= 1
\end{aligned}$$

$$\begin{aligned}
\therefore \mathrm{LOMSE}(h^*, N, D_A) &= N^{-\frac{4}{D_A+4}} \left(\left(\frac{D_A}{4}\right)^{\frac{4}{D_A+4}} + \left(\frac{4}{D_A}\right)^{\frac{D_A}{D_A+4}} \right) C_b^{\frac{D_A}{D_A+4}} C_v^{\frac{4}{D_A+4}} \\
&\approx C_b. \text{ (for } D_A \gg 4)
\end{aligned}$$

□

B.2 Proof of Theorem 1

We use the semi-definite programming solution presented in the work of Noh et al. [22] for computing the metric matrix $A^*(\mathbf{s})$ that minimizes the $\operatorname{tr}\left(A^{-1}(\mathbf{s})B(\mathbf{s})\right)^2$ for nearest neighbor classification. We use the solution to minimize the squared trace term in Eq. (7). In our case, we use $B(\mathbf{s}) := \mathbf{H}_{\mathbf{a}}\, \mathbb{E}[r|\, \mathbf{s}, \mathbf{a}]|_{\mathbf{a}=\pi(\mathbf{s})}$ which comes from Eq. (7).

Theorem 1. *(Adapted from Noh et al. [22]) Assume that the $p(r|\, \mathbf{s}, \mathbf{a})$ is twice differentiable w.r.t. an action $\mathbf{a}$. Let $\Lambda_+(\mathbf{s})$ and $\Lambda_-(\mathbf{s})$ be diagonal matrices of positive and negative eigenvalues of the Hessian $\mathbf{H}_{\mathbf{a}}\, \mathbb{E}[r|\, \mathbf{s}, \mathbf{a}]|_{\mathbf{a}=\pi(\mathbf{s})}$, $U_+(\mathbf{s})$ and $U_-(\mathbf{s})$ be matrices of eigenvectors corresponding to $\Lambda_+(\mathbf{s})$ and $\Lambda_-(\mathbf{s})$ respectively, and $d_+(\mathbf{s})$ and $d_-(\mathbf{s})$ be the numbers of positive and negative eigenvalues of the Hessian. Then the metric $A^*(\mathbf{s})$ that minimizes $U_{b,A}$ is:*

$$A^*(\mathbf{s}) = \alpha(\mathbf{s})\, [U_+(\mathbf{s})U_-(\mathbf{s})] \underbrace{\begin{pmatrix} d_+(\mathbf{s})\Lambda_+(\mathbf{s}) & 0 \\ 0 & -d_-(\mathbf{s})\Lambda_-(\mathbf{s}) \end{pmatrix}}_{=:M(\mathbf{s})} [U_+(\mathbf{s})U_-(\mathbf{s})]^\top\,, \tag{24}$$

where $\alpha(\mathbf{s}) := |M(\mathbf{s})|^{-1/(d_+(\mathbf{s})+d_-(\mathbf{s}))}$.

Proof. Define $B(\mathbf{s}) := \mathbf{H}_{\mathbf{a}}\, r(\mathbf{s}, \mathbf{a})|_{\pi(\mathbf{s})}$, then we want $A(\mathbf{s})$ that,

$$\begin{aligned} &\min_{A(\mathbf{s})} \left(\operatorname{tr}\left(A(\mathbf{s})^{-1}B(\mathbf{s})\right)\right)^2 \\ &\text{s.t. } |A(\mathbf{s})| = 1 \\ &\quad A(\mathbf{s}) \succ 0 \\ &\quad A(\mathbf{s}) = A(\mathbf{s})^\top \\ &\quad B(\mathbf{s}) = B(\mathbf{s})^\top \end{aligned} \tag{25}$$

In the cases where some of the eigenvalues of $B(\mathbf{s})$ are 0, the components along the directions of the corresponding eigenvectors can be discarded.

Minimizing the objective, or, minimizing $U_{b,A}$ is the same as minimizing $C_{b,A}$ under the condition that there are not both sets of states with positive and negative trace terms $\operatorname{tr}\left(A(\mathbf{s})^{-1}B(\mathbf{s})\right)$. The trace term is positive or negative when the eigenvalues of $B(\mathbf{s})$ other than 0 are all positive or negative, respectively. And the trace term becomes 0 when there are both negative and positive eigenvalues in the eigenvalues of $B(\mathbf{s})$. In cases where all the states have both positive and negative eigenvalues in the eigenvalues of $B(\mathbf{s})$, $C_{b,A} = U_{b,A} = 0$.

When there are both positive and negative eigenvalues in the eigenvalues of $B(\mathbf{s})$,

$$\begin{aligned} A^*(\mathbf{s}) &= \alpha(\mathbf{s})\, [U_+(\mathbf{s})U_-(\mathbf{s})] \begin{pmatrix} d_+(\mathbf{s})\Lambda_+(\mathbf{s}) & 0 \\ 0 & -d_-(\mathbf{s})\Lambda_-(\mathbf{s}) \end{pmatrix} [U_+(\mathbf{s})U_-(\mathbf{s})]^\top \\ B(\mathbf{s}) &= [U_+(\mathbf{s})U_-(\mathbf{s})] \begin{pmatrix} \Lambda_+(\mathbf{s}) & 0 \\ 0 & \Lambda_-(\mathbf{s}) \end{pmatrix} [U_+(\mathbf{s})U_-(\mathbf{s})]^\top \\ &\operatorname{tr}\left(A^*(\mathbf{s})^{-1}B(\mathbf{s})\right) \\ &= \frac{1}{\alpha(\mathbf{s})} \operatorname{tr}\left[[U_+(\mathbf{s})U_-(\mathbf{s})] \begin{pmatrix} \frac{\Lambda_+^{-1}(\mathbf{s})}{d_+(\mathbf{s})} & 0 \\ 0 & -\frac{\Lambda_-^{-1}(\mathbf{s})}{d_-(\mathbf{s})} \end{pmatrix} \begin{pmatrix} \Lambda_+(\mathbf{s}) & 0 \\ 0 & \Lambda_-(\mathbf{s}) \end{pmatrix} [U_+(\mathbf{s})U_-(\mathbf{s})]^\top \right] \\ &= \frac{1}{\alpha(\mathbf{s})} \left(\frac{d_+(\mathbf{s})}{d_+(\mathbf{s})} - \frac{d_-(\mathbf{s})}{d_-(\mathbf{s})} \right) \\ &= 0 \end{aligned}$$

When the eigenvalues of $B(\mathbf{s})$ other than 0 are all negative or positive, then we can solve the Lagrangian equation $F(\mathbf{s})$. We first solve for the case when the eigenvalues of $B(\mathbf{s})$ other than 0 are all positive,

$$
\begin{aligned}
F(\mathbf{s}) &= \operatorname{tr}\left(A(\mathbf{s})^{-1}B(\mathbf{s})\right)^2 - c(\mathbf{s})(|A(\mathbf{s})| - 1) \\
\frac{\partial F(\mathbf{s})}{\partial c(\mathbf{s})} &= |A(\mathbf{s})| - 1 = 0 \\
\frac{\partial F(\mathbf{s})}{\partial A(\mathbf{s})} &= -2\operatorname{tr}\left(A(\mathbf{s})^{-1}B(\mathbf{s})\right) A(\mathbf{s})^{-\top}B(\mathbf{s})^\top A(\mathbf{s})^{-\top} - c(\mathbf{s})|A(\mathbf{s})|A(\mathbf{s})^{-\top} = 0 \\
c(\mathbf{s})I &= -2\operatorname{tr}\left(A(\mathbf{s})^{-1}B(\mathbf{s})\right) A(\mathbf{s})^{-1}B(\mathbf{s}) \\
&\quad \left(\text{since } A(\mathbf{s}) = A(\mathbf{s})^\top,\ B(\mathbf{s}) = B(\mathbf{s})^\top,\ |A(\mathbf{s})| = 1\right)
\end{aligned} \tag{26}
$$

(i) Assume $c(\mathbf{s}) = 0$,

Then either $\operatorname{tr}\left(A(\mathbf{s})^{-1}B(\mathbf{s})\right) = 0$, or, $A(\mathbf{s})^{-1}B(\mathbf{s}) = \mathbf{0}$

We first check if $\operatorname{tr}\left(A(\mathbf{s})^{-1}B(\mathbf{s})\right) = 0$.

$$
\operatorname{tr}\left(A(\mathbf{s})^{-1}B(\mathbf{s})\right) = \operatorname{tr}\left(B(\mathbf{s})^{\frac{1}{2}}A(\mathbf{s})^{-1}B(\mathbf{s})^{\frac{1}{2}}\right) \tag{27}
$$

Since $B(\mathbf{s}) = U_+(\mathbf{s})\Lambda_+(\mathbf{s})U_+(\mathbf{s})^\top$, $B(\mathbf{s})^{\frac{1}{2}} = B(\mathbf{s})^{\frac{\top}{2}} = U_+(\mathbf{s})\Lambda_+(\mathbf{s})^{\frac{1}{2}}U_+(\mathbf{s})^\top$. And as $A(\mathbf{s}) \succ 0$, its inverse is also positive definite $A(\mathbf{s})^{-1} \succ 0$. Using these relations, we can show that the term inside the trace in Eq. (27) is a positive definite matrix.

$$
\begin{aligned}
\mathbf{a}^\top\left[B(\mathbf{s})^{\frac{1}{2}}A(\mathbf{s})^{-1}B(\mathbf{s})^{\frac{1}{2}}\right]\mathbf{a} &= \left(B(\mathbf{s})^{\frac{\top}{2}}\mathbf{a}\right)^\top A(\mathbf{s})^{-1}\left(B(\mathbf{s})^{\frac{\top}{2}}\mathbf{a}\right) \\
&> 0, \quad \forall\, \mathbf{a} \in \mathbb{R}^{D_A}\backslash\{\mathbf{0}\}
\end{aligned}
$$

Because $B(\mathbf{s})^{\frac{1}{2}}A(\mathbf{s})^{-1}B(\mathbf{s})^{\frac{1}{2}} \succ 0$, $\operatorname{tr}\left(B(\mathbf{s})^{\frac{1}{2}}A(\mathbf{s})^{-1}B(\mathbf{s})^{\frac{1}{2}}\right) > 0$. Then, from Eq. (27),

$$
\begin{aligned}
\operatorname{tr}\left(A(\mathbf{s})^{-1}B(\mathbf{s})\right) &> 0 \\
A(\mathbf{s})^{-1}B(\mathbf{s}) &\neq \mathbf{0}, \quad \text{as it's trace is positive}
\end{aligned}
$$

Therefore, the assumption is wrong, and $c(\mathbf{s}) \neq 0$

(ii) Assume $c(\mathbf{s}) \neq 0$,

Since the Lagrangian multiplier $c(\mathbf{s})$ and $\operatorname{tr}\left(A(\mathbf{s})^{-1}B(\mathbf{s})\right)$ in Eq. (26) are scalar values, $A(\mathbf{s})$ needs to be a scalar multiple of $B(\mathbf{s})$ (which is symmetric) to match the scalar multiple of identity matrix in the LHS of Eq. (26) while satisfying $A(\mathbf{s}) \succ 0$ and $|A(\mathbf{s})| = 1$. Therefore the $A^*(\mathbf{s})$ is,

$$
\therefore A^*(\mathbf{s}) = \alpha(\mathbf{s})U_+(\mathbf{s})\left[d_+(\mathbf{s})\Lambda_+(\mathbf{s})\right]U_+(\mathbf{s})^\top \tag{28}
$$

Similarly, we can also derive $A^*(\mathbf{s})$ for the case where all eigenvalues of $B(\mathbf{s})$ are negative. □

B.3 Proof of Theorem 2

Theorem 2. *(Adapted from Kallus and Zhou [2]) Kernel-based IS estimator with the optimal metric $A^*(\mathbf{s})$ from Eq. (8) and the optimal bandwidth h^* in Eq. (3) is a consistent estimator in which convergence rate is equal or faster than the isotropic kernel-based IS estimator with h^*. When the Hessian $\mathbf{H_a}\, r(\mathbf{s},\mathbf{a})|_{\mathbf{a}=\pi(\mathbf{s})}$ has both positive and negative eigenvalues for all states, $A^*(\mathbf{s})$ applied estimator converges to the true policy value faster than the one without the metric by the rate of $\mathcal{O}(D_A^{-\frac{1}{2}})$ as the action dimension increases.*

Proof. The MSE of a kernel-based IS estimation without a metric derived by Kallus and Zhou [2] in terms of action dimension D_A and data size of N is (from the derived bias and variance in Appendix A.1),

$$\mathrm{MSE}(h, N, D_A) = \underbrace{h^4 C_b + O\left(h^6\right)}_{\mathrm{Bias}[\hat{\rho}^K]^2} + \underbrace{\frac{C_v}{N h^{D_A}} + O\left(\frac{1}{N h^{D_A-2}}\right)}_{\mathrm{Var}[\hat{\rho}^K]}, \tag{29}$$

where, the MSE consists of squared bias and the variance of the estimate $\hat{\rho}^K$. Since the optimal bandwidth presented in Eq. (3) is $h^* = \mathcal{O}\left(\frac{D_A}{N}^{\frac{1}{D_A+4}}\right)$, the convergence of the MSE without a metric is:

$$\begin{aligned}
\mathrm{MSE}(h^*, N, D_A) &= \mathcal{O}\left(\left(\frac{D_A}{N}\right)^{\frac{4}{D_A+4}}\right) + \mathcal{O}\left(N^{-1}\left(\frac{D_A}{N}\right)^{\frac{-D_A}{D_A+4}}\right) && (30)\\
&= \mathcal{O}\left(\left(\frac{D_A}{N}\right)^{\frac{4}{D_A+4}}\right) + \mathcal{O}\left(\left(\frac{D_A}{N}\right)^{\frac{4}{D_A+4}}\frac{1}{D_A}\right), \quad (D_A \geq 2) && (31)\\
&= \mathcal{O}\left(\left(\frac{D_A}{N}\right)^{\frac{4}{D_A+4}}\right) && (32)
\end{aligned}$$

Since the MSE of the kernel-based IS estimator with h^* converges to 0 by $\mathcal{O}\left(\left(\frac{D_A}{N}\right)^{\frac{4}{D_A+4}}\right)$, The estimation approaches to the true policy value by the rate of $\mathcal{O}\left(\left(\frac{D_A}{N}\right)^{\frac{2}{D_A+4}}\right)$.

For the KMIS metric applied kernel-based IS estimator, its convergence rate can be computed similar to that of the isotropic kernel-based IS estimator regarding $C_{b,A}$ as a non-zero constant, then the resulting convergence rate is the same as that of the isotropic kernel-based IS estimator.

However, in the best case where the Hessians $\mathbf{H_a}\, r(\mathbf{s},\mathbf{a})|_{\mathbf{a}=\pi(\mathbf{s})}$ contain both negative and positive eigenvalues for all states, $C_{b,A^*} = 0$. Then, the MSE with the optimal metric and the optimal bandwidth $\mathrm{MSE}(h^*, A^*, N, D_A)$ converges to zero by:

$$\begin{aligned}
\mathrm{MSE}(h^*, A^*, N, D_A) &= (h^*)^4 \cancel{C_{b,A^*}} + O\left((h^*)^6\right) + \frac{C_v}{N (h^*)^{D_A}} + O\left(\frac{1}{N (h^*)^{D_A-2}}\right), && (33)\\
&= \mathcal{O}\left(\left(\frac{D_A}{N}\right)^{\frac{6}{D_A+4}}\right) + \mathcal{O}\left(\left(\frac{D_A}{N}\right)^{\frac{4}{D_A+4}}\frac{1}{D_A}\right) && (34)\\
&= \mathcal{O}\left(\left(\frac{D_A}{N}\right)^{\frac{4}{D_A+4}}\frac{1}{D_A}\right), (\text{ where, } N \gg D_A) && (35)
\end{aligned}$$

In the best case where there are both positive and negative eigenvalues for all states, the MSE of our metric applied kernel-based IS estimation converges to 0 faster than the one without the metric by

$\mathcal{O}(D_A^{-1})$. Therefore, our algorithm converges to the true policy value faster than the one without the metric by $\mathcal{O}(D_A^{-\frac{1}{2}})$ in the best case. □

C Algorithm Details

C.1 Regularizers for the KMIS Metric

$$X(\mathbf{s}) := [U_+(\mathbf{s})U_-(\mathbf{s})U_0(\mathbf{s})] \begin{pmatrix} d_+(\mathbf{s})\Lambda_+(\mathbf{s}) & 0 & 0 \\ 0 & -d_-(\mathbf{s})\Lambda_-(\mathbf{s}) & 0 \\ 0 & 0 & \mathbf{0} \end{pmatrix} [U_+(\mathbf{s})U_-(\mathbf{s})U_0(\mathbf{s})]^\top \tag{36}$$

To compute the regularizers $\beta(\mathbf{s})$ and $\gamma(\mathbf{s})$ in Eq. (10), first add a positive real coefficient $\epsilon(\mathbf{s})$ that is relatively smaller than the eigenvalues of $A^*(\mathbf{s})$ to $X(\mathbf{s})$. For the experiments, $\epsilon(\mathbf{s})$ was assigned to be the maximum value among eigenvalues in the Hessian of reward multiplied by 0.01.

$$Y(\mathbf{s}) := X(\mathbf{s}) + \epsilon(s)I \tag{37}$$

With the coefficient $\epsilon(\mathbf{s})I$ added to $X(\mathbf{s})$, compute the determinant of the resulting matrix $Y(\mathbf{s})$. To scale the metric matrix to have determinant of 1, We scale the matrix by multiplying $\beta(s) = \frac{1}{|Y(\mathbf{s})|^{(-1/D_A)}}$. Then the $\gamma(s) = \beta(s)\epsilon(s)$

D Additional Experiment

D.1 Experiment with Varying Noise Level in the Rewards

When the noise in the rewards increases, DM suffers the most as the function estimation becomes difficult as the noise increases. Other algorithms also show an increase in the MSE as the noise increases. The KMIS metric applied kernel-based IS estimator performs the best with the learned metric even though the DM which the KMIS uses to estimate the Hessians required for the metric learning performs the worst.

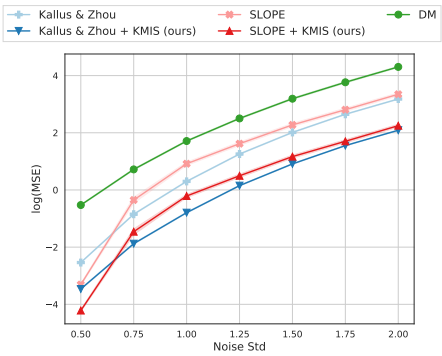


Figure 5: Performance of OPE algorithms with an increasing standard deviation of the Gaussian noise in the rewards in the quadratic reward domain. Means and standard errors of squared errors were obtained from 100 trials with 40k samples.

E Experiment Details

For all experiments, we used a neural network reward regressor with 2 fully-connected layers of 128 hidden units with tanh activations, and a Gaussian output layer. The neural network was trained with learning rate of 5e-4. The learning rate was chosen by conducting grid search over {1e-4, 5e-4, 1e-3}. The reward regressor was trained with early stopping rule where we stop training when the validation error does not decrease for 20 training epochs. And we used the trained weights that showed the lowest validation error during the training. For the train and validation split, 20% of the available offline data was used as the validation data, and the rest of the data was used for the training. For the estimation time, when we test our algorithm on the synthetic domain with absolute error and a dummy action dimension, our algorithm takes 40 seconds on average to make an OPE estimate with 20k offline data on the i7 CPU with 32GB RAM. The experimental results were made with 100 C2-standard-4 instances on the Google Cloud Platform where each instance has 4 virtual CPUs and 16GB of RAM. The code is available at https://github.com/haanvid/kmis.

Self-Normalization All estimators in the experiments are self-normalized as in the work of Kallus and Zhou [2]. Self-normalization is used because it reduces the estimation variance significantly with the addition of a small bias and yields a smaller MSE in result compared to the estimation without self-normalization [2, 13]. The self-normalized kernel-based IS estimator without a metric is shown in Eq. (38).

$$\hat{\rho}^{K}_{\text{norm}} = \frac{\sum_{i=1}^{N} \frac{r_i}{\pi_b(\mathbf{a}_i \mid \mathbf{s}_i)} K\left(\frac{\mathbf{a}_i - \pi(\mathbf{s}_i)}{h}\right)}{\sum_{i=1}^{N} \frac{1}{\pi_b(\mathbf{a}_i \mid \mathbf{s}_i)} K\left(\frac{\mathbf{a}_i - \pi(\mathbf{s}_i)}{h}\right)} \tag{38}$$

No Boundary Bias Correction Action space can be bounded, and the kernels used in importance sampling can be extended past the bounds. As the actions of the offline data are only observed inside the bounds, the kernels extending outside the bounds will induce bias. Previous works [2, 5] made a correction for the induced bias by truncating the kernel by the action bounds of each dimension and normalizing the kernel. However, we do not correct the boundary bias for all estimators in this work since we regard the boundary bias as negligible compared to the bias reduced by our locally learned metric. Removing the boundary bias correction makes kernel-based IS estimators simpler to use than those with corrections, as the implementation of boundary bias correction requires information on the action bounds, and the integration of the kernel within the action bounds.

Discretized OPE Estimator Discretized ope estimator discretizes the action space bounded by the minimum and maximum value of the behavior actions for each dimension with the given number of bins assigned for each dimension. In our experiment, we discretized each action dimension by 10 intervals resulting in 100 discretized bins in 2D action space. The IS ratio of the discretized OPE estimator is composed of the indicator function in the numerator and the behavior policy probability density function integrated over the bin in the denominator.

E.1 Synthetic Domains Experiments

For the synthetic domain, we used dropout after each hidden layer with dropout rate of 0.5. Behavior policy density values were clipped below 0.1 for the quadratic reward domain as in the works of Kallus and Zhou [2]. For absolute error domain and multi-modal reward domain, as their behavior policy density is uniform density, we did not apply behavior density value clipping. Clipping the behavior policy density value adds a small bias to the estimation, but reduces a significant variance, and therefore MSE is smaller compared to the estimations without the clipping [2]. For SLOPE, we select a bandwidth from geometrically spaced bandwidths $\left\{2^{-i} : i \in [1, 7], i \in \mathbb{N}\right\}$ as in the work of Su et al. [5].

Quadratic Reward Domain The conditional reward distribution given a state $\mathbf{s}$ and a action $\mathbf{a}$ is:

$$p(r|\, \mathbf{s}, \mathbf{a}) = N\left(r(\mathbf{s}, \mathbf{a}), 0.5^2\right), \tag{39}$$

where, $r(\mathbf{s}, \mathbf{a}) := \mathbb{E}[r|\, \mathbf{s}, \mathbf{a}]$. The mean reward of a quadratic reward domain given a state $\mathbf{s}$ and a action $\mathbf{a}$ is:

$$r(\mathbf{s}, \mathbf{a}) = -(\mathbf{s} - \mathbf{a})^\top \begin{bmatrix} 11 & 9 \\ 9 & 11 \end{bmatrix} (\mathbf{s} - \mathbf{a}) \tag{40}$$

Dummy Action Dimension Domain Each dimension of actions and states are sampled uniform randomly in the range of $[-1, 1]$. The absolute error which is only dependent on the first dimension of action is designed as shown below.

$$r(\mathbf{s}, \mathbf{a}) = -\left|0.5 s_1 - a_1\right| \tag{41}$$

where the reward is not sampled from a distribution but determined by states $\mathbf{s}$ and actions $\mathbf{a}$.

Multi-Modal Reward Domain Each dimension of actions and states are sampled uniform randomly in the range of $[-1, 1]$. The multi-modal reward with exponential functions is designed as shown below.

$$\begin{aligned}
f_1(\mathbf{s}, \mathbf{a}) &= \exp\left(-\left(\left(\frac{(s_1 - a_1) - 0.5}{0.25}\right)^2 + \left(\frac{(s_2 - a_2)}{1}\right)^2\right)\right) \\
f_2(\mathbf{s}, \mathbf{a}) &= \exp\left(-\left(\left(\frac{(s_1 - a_1) + 0.5}{0.25}\right)^2 + \left(\frac{(s_2 - a_2)}{1}\right)^2\right)\right) \\
f_3(\mathbf{s}, \mathbf{a}) &= \exp\left(-\left(\left(\frac{(s_1 - a_1)}{1}\right)^2 + \left(\frac{(s_2 - a_2) + 0.5}{0.25}\right)^2\right)\right) \\
f_4(\mathbf{s}, \mathbf{a}) &= \exp\left(-\left(\left(\frac{(s_1 - a_1)}{1}\right)^2 + \left(\frac{(s_2 - a_2) - 0.5}{0.25}\right)^2\right)\right) \\
r(\mathbf{s}, \mathbf{a}) &= -\max\left(f_1(\mathbf{s}, \mathbf{a}), f_2(\mathbf{s}, \mathbf{a}), f_3(\mathbf{s}, \mathbf{a}), f_4(\mathbf{s}, \mathbf{a})\right)
\end{aligned} \tag{42}$$

where the reward is not sampled from a distribution but determined by states $\mathbf{s}$ and actions $\mathbf{a}$.

E.2 Warfarin Data Experiments

For the warfarin data, we used L2 regularizer for the 2 hidden layers with the coefficient of 1e-1. The L2 regularizer coefficient was chosen by conducting grid search over {1e-5, 1e-4, 1e-3, 1e-2, 1e-1}. We selected the L2 regularization coefficient that has a lowest validation error (the offline dataset D was split into a train and validation dataset) for the reward regressor. Behavior policy density values below 1e-1 was clipped as in the work of Kallus and Zhou [2]. For SLOPE, we selected a bandwidth from $\left\{2^{-i} : i \in [-2, 7], i \in \mathbb{N}\right\}$ similar to the work of Su et al. [5].

F Theoretical Analysis

F.1 Effect of the Hessian Estimation Error on the Estimation Error and the Convergence Speed of the KMIS Metric Applied Kernel-Based IS Estimator

Error Analysis To reduce the squared leading-order bias $\mathrm{LOBIAS}(h, A(\mathbf{s}))^2$ in Eq. (43) (the equation can be derived with bias from Eq. (13), variance from Eq. (19), and the squared leading-order bias with metric from Eq. (6)) of a kernel-based IS estimator, KMIS learns the metric that minimizes the upper bound $U_{b,A}$ in Eq. (7). To minimize $U_{b,A}$, we derived the optimal metric matrix $A^*(\mathbf{s})$ given the true Hessian $H(\mathbf{s}) := \mathbf{H_a}\,\mathbb{E}[r|\,\mathbf{s},\mathbf{a}]|_{a=\pi(s)}$ in Appendix B.2. However, since we will be using estimated Hessian matrix $\widetilde{H}(\mathbf{s})$ acquired from a reward regressor, we analyze how the error in $\widetilde{H}(\mathbf{s})$ affects KMIS estimation error.

$$\mathrm{MSE}(h, A, N, D_A) = \underbrace{h^4 C_{b,A}}_{=:\mathrm{LOBIAS}(h,A)^2} + O\left(h^6\right) + \frac{C_v}{Nh^{D_A}} + O\left(\frac{1}{Nh^{D_A-2}}\right) \tag{43}$$

where,

$$C_{b,A} := \left(\frac{1}{2}\mathbb{E}_{\mathbf{s}\sim p(\mathbf{s})}\left[\mathrm{tr}\left(A(\mathbf{s})^{-1}H(\mathbf{s})\right)\right]\right)^2 \tag{44}$$

$$C_v := R(K)\mathbb{E}_{\mathbf{s}\sim p(\mathbf{s})}\left[\frac{\mathbb{E}[r^2|\,\mathbf{s},\mathbf{a}=\pi(\mathbf{s})]}{\pi_b(\mathbf{a}=\pi(\mathbf{s})|\,\mathbf{s})}\right], \text{where } C_{v,A} = C_v \tag{45}$$

In Appendix B.2, we proved that when $H(\mathbf{s})$ contains both positive and negative eigenvalues, the optimal metric makes the squared leading-order bias $\mathrm{LOBIAS}(h, A^*)^2 = 0$ by making the the trace term $\mathrm{tr}(A(\mathbf{s})^{-1}H(\mathbf{s}))$ zero. In this section we analyze the upper bound of $\mathrm{LOBIAS}(h, \widetilde{A})^2$ in a similar setting where the $H(\mathbf{s})$ contains both positive and negative eigenvalues and no zero eigenvalues. For analyzing, we use the following notations regarding the true Hessian $H(\mathbf{s})$, the optimal metric computed from the true Hessian $A^*(\mathbf{s})$, estimated Hessian from the reward regressor $\widetilde{H}(\mathbf{s})$ and the metric computed from it $\widetilde{A}(\mathbf{s})$:

$$H(\mathbf{s}) = [U_+(\mathbf{s})U_-(\mathbf{s})]\begin{pmatrix}\Lambda_+(\mathbf{s}) & 0\\ 0 & \Lambda_-(\mathbf{s})\end{pmatrix}[U_+(\mathbf{s})U_-(\mathbf{s})]^\top \tag{46}$$

$$A^*(\mathbf{s}) = \alpha(\mathbf{s})\,[U_+(\mathbf{s})U_-(\mathbf{s})]\begin{pmatrix}d_+(\mathbf{s})\Lambda_+(\mathbf{s}) & 0\\ 0 & -d_-(\mathbf{s})\Lambda_-(\mathbf{s})\end{pmatrix}[U_+(\mathbf{s})U_-(\mathbf{s})]^\top \tag{47}$$

$$\Lambda(\mathbf{s}) = \begin{pmatrix}\Lambda_+(\mathbf{s}) & 0\\ 0 & \Lambda_-(\mathbf{s})\end{pmatrix}$$

$$\Lambda_+(\mathbf{s}) = \begin{pmatrix}\lambda_{+,1}(\mathbf{s}) & 0 & \cdots\\ 0 & \lambda_{+,2}(\mathbf{s}) & \\ \vdots & & \ddots\end{pmatrix}$$

$$U_+(\mathbf{s}) = \begin{pmatrix}| & | & \\ \mathbf{u}_{+,1}(\mathbf{s}) & \mathbf{u}_{+,2}(\mathbf{s}) & \cdots\\ | & | & \end{pmatrix}$$

$$\alpha(\mathbf{s}) = \left\{\left(d_+(\mathbf{s})^{d_+(\mathbf{s})}\prod_{i=1}^{d_+(\mathbf{s})}\lambda_{+,i}(\mathbf{s})\right)\left((-1)^{d_-(\mathbf{s})}d_-(\mathbf{s})^{d_-(\mathbf{s})}\prod_{i=1}^{d_-(\mathbf{s})}\lambda_{-,i}(\mathbf{s})\right)\right\}^{-\frac{1}{d_+(\mathbf{s})+d_-(\mathbf{s})}} \tag{48}$$

$$\widetilde{H}(\mathbf{s}) = \left[\widetilde{U}_+(\mathbf{s})\widetilde{U}_-(\mathbf{s})\right] \begin{pmatrix} \widetilde{\Lambda}_+(\mathbf{s}) & 0 \\ 0 & \widetilde{\Lambda}_-(\mathbf{s}) \end{pmatrix} \left[\widetilde{U}_+(\mathbf{s})\widetilde{U}_-(\mathbf{s})\right]^\top \tag{49}$$

$$\widetilde{A}(\mathbf{s}) = \tilde{\alpha}(\mathbf{s}) \left[\widetilde{U}_+(\mathbf{s})\widetilde{U}_-(\mathbf{s})\right] \begin{pmatrix} d_+(\mathbf{s})\widetilde{\Lambda}_+(\mathbf{s}) & 0 \\ 0 & -d_-(\mathbf{s})\widetilde{\Lambda}_-(\mathbf{s}) \end{pmatrix} \left[\widetilde{U}_+(\mathbf{s})\widetilde{U}_-(\mathbf{s})\right]^\top \tag{50}$$

$$\widetilde{\Lambda}(\mathbf{s}) = \begin{pmatrix} \widetilde{\Lambda}_+(\mathbf{s}) & 0 \\ 0 & \widetilde{\Lambda}_-(\mathbf{s}) \end{pmatrix}$$

$$\begin{aligned} \widetilde{\alpha}(\mathbf{s}) &= \left\{ \left(d_+(\mathbf{s})^{d_+(\mathbf{s})} \prod_{i=1}^{d_+(\mathbf{s})} \widetilde{\lambda}_{+,i}(\mathbf{s}) \right) \left((-1)^{d_-(\mathbf{s})} d_-(\mathbf{s})^{d_-(\mathbf{s})} \prod_{i=1}^{d_-(\mathbf{s})} \widetilde{\lambda}_{-,i}(\mathbf{s}) \right) \right\}^{-\frac{1}{d_+(\mathbf{s})+d_-(\mathbf{s})}} \\ &= \left((-1)^{d_-(\mathbf{s})} d_+(\mathbf{s})^{d_+(\mathbf{s})} d_-(\mathbf{s})^{d_-(\mathbf{s})} |\widetilde{H}(\mathbf{s})| \right)^{-\frac{1}{D_A}} \\ &> 0 \end{aligned} \tag{51}$$

where we used eigendecomposition on $H(\mathbf{s})$. $\Lambda_+(\mathbf{s})$ and $\Lambda_-(\mathbf{s})$ are diagonal matrices containing positive and negative eigenvalues, respectively. $U_+(\mathbf{s})$ and $U_-(\mathbf{s})$ are matrices containing eigenvectors corresponding to $\Lambda_+(\mathbf{s})$ and $\Lambda_-(\mathbf{s})$, respectively. Similarly, $\widetilde{\Lambda}_+(\mathbf{s})$, $\widetilde{\Lambda}_-(\mathbf{s})$, $\widetilde{U}_+(\mathbf{s})$, $\widetilde{U}_-(\mathbf{s})$ are decomposed from $\widetilde{H}(\mathbf{s})$. We assume that $\widetilde{H}(\mathbf{s})$ correctly estimates the sign (positive, negative signs) of the eigenvalues in $H(\mathbf{s})$. Because of the assumption, there are $d_+(\mathbf{s})$ and $d_-(\mathbf{s})$ in Eq. (50) and Eq. (51) instead of $\tilde{d}_+(\mathbf{s})$ and $\tilde{d}_-(\mathbf{s})$. We quantify the error in $\widetilde{H}(\mathbf{s})$ with constants ϵ and η ($\epsilon \geq 0$, $\eta > 0$) by:

$$|\widetilde{\mathbf{u}}_{a,i}(\mathbf{s})^\top \mathbf{u}_{a,i}(\mathbf{s}) - 1| \leq \epsilon \tag{52}$$

$$|\widetilde{\mathbf{u}}_{a,i}(\mathbf{s})^\top \mathbf{u}_{b,j}(\mathbf{s})| \leq \epsilon, \text{ if } a \neq b, \text{ or, } i \neq j \tag{53}$$

$$\|\widetilde{\Lambda}(\mathbf{s})^{-1}\Lambda(\mathbf{s}) - I\| \leq \epsilon \tag{54}$$

$$0 < \frac{\lambda_{a,i}(\mathbf{s})}{\widetilde{\lambda}_{a,j}(\mathbf{s})} \leq \eta, \text{ if } i \neq j \tag{55}$$

$$-\eta \leq \frac{\lambda_{a,i}(\mathbf{s})}{\widetilde{\lambda}_{b,j}(\mathbf{s})} < 0, \text{ if } a \neq b \tag{56}$$

$$\text{where, } a, b \in \{+, -\}$$
$$i, j \in \{1, 2, 3, \cdots\}$$

The trace term $\mathrm{tr}(\widetilde{A}(\mathbf{s})^{-1}H(\mathbf{s}))$ in the $\mathrm{LOBIAS}(h, \widetilde{A})^2$ is,

$$\begin{aligned} &\mathrm{tr}(\widetilde{A}(\mathbf{s})^{-1}H(\mathbf{s})) \\ &= \frac{1}{\widetilde{\alpha}(\mathbf{s})} \mathrm{tr}\left([d_+(\mathbf{s})\widetilde{U}_+\widetilde{\Lambda}_+\widetilde{U}_+^\top - d_-\widetilde{U}_-\widetilde{\Lambda}_-\widetilde{U}_-^\top]^{-1}[U_+\Lambda_+U_+^\top + U_-\Lambda_-U_-^\top] \right) \qquad (57) \\ &= \frac{1}{\widetilde{\alpha}(\mathbf{s})} \mathrm{tr}\left(\frac{1}{d_+}\widetilde{U}_+\widetilde{\Lambda}_+^{-1}\widetilde{U}_+^\top U_+\Lambda_+U_+^\top + \frac{1}{d_+}\widetilde{U}_+\widetilde{\Lambda}_+^{-1}\widetilde{U}_+^\top U_-\Lambda_-U_-^\top \right. \qquad (58) \\ &\qquad \left. -\frac{1}{d_-}\widetilde{U}_-\widetilde{\Lambda}_-^{-1}\widetilde{U}_-^\top U_+\Lambda_+U_+^\top - \frac{1}{d_-}\widetilde{U}_-\widetilde{\Lambda}_-^{-1}\widetilde{U}_-^\top U_-\Lambda_-U_-^\top \right) \\ &= \underbrace{\frac{1}{\widetilde{\alpha}(\mathbf{s})d_+(\mathbf{s})} \left[\sum_{j=1}^{d_+(\mathbf{s})} \sum_{i=1}^{d_+(\mathbf{s})} \frac{\lambda_{+,i}(\mathbf{s})}{\widetilde{\lambda}_{+,j}(\mathbf{s})} (\widetilde{\mathbf{u}}_{+,j}(\mathbf{s})^\top \mathbf{u}_{+,i}(\mathbf{s}))^2 + \sum_{j=1}^{d_+(\mathbf{s})} \sum_{i=1}^{d_-(\mathbf{s})} \frac{\lambda_{-,i}(\mathbf{s})}{\widetilde{\lambda}_{+,j}(\mathbf{s})} (\widetilde{\mathbf{u}}_{+,j}(\mathbf{s})^\top \mathbf{u}_{-,i}(\mathbf{s}))^2 \right]}_{=:X(\mathbf{s})} \\ &\quad - \underbrace{\frac{1}{\widetilde{\alpha}(\mathbf{s})d_-(\mathbf{s})} \left[\sum_{j=1}^{d_-(\mathbf{s})} \sum_{i=1}^{d_+(\mathbf{s})} \frac{\lambda_{+,i}(\mathbf{s})}{\widetilde{\lambda}_{-,j}(\mathbf{s})} (\widetilde{\mathbf{u}}_{-,j}(\mathbf{s})^\top \mathbf{u}_{+,i}(\mathbf{s}))^2 + \sum_{j=1}^{d_-(\mathbf{s})} \sum_{i=1}^{d_-(\mathbf{s})} \frac{\lambda_{-,i}(\mathbf{s})}{\widetilde{\lambda}_{-,j}(\mathbf{s})} (\widetilde{\mathbf{u}}_{-,j}(\mathbf{s})^\top \mathbf{u}_{-,i}(\mathbf{s}))^2 \right]}_{=:Y(\mathbf{s})} \end{aligned} \tag{59}$$

where in the third equality (Eq.59) we used,

$$\mathrm{tr}(\widetilde{U}_a\widetilde{\Lambda}_a^{-1}\widetilde{U}_a^\top U_b\Lambda_b U_b^\top) = \sum_{j=1}^{d_a(\mathbf{s})}\sum_{i=1}^{d_b(\mathbf{s})}\frac{\lambda_{b,i}(\mathbf{s})}{\widetilde{\lambda}_{a,j}(\mathbf{s})}(\widetilde{\mathbf{u}}_{a,j}(\mathbf{s})^\top\mathbf{u}_{b,j}(\mathbf{s}))^2, \text{where, } a,b\in\{+,-\} \tag{60}$$

Expanding $X(\mathbf{s})$ we get,

$$X(\mathbf{s}) = \frac{1}{\widetilde{\alpha}(\mathbf{s})d_+(\mathbf{s})}\left[\sum_{i=1}^{d_+(\mathbf{s})}\frac{\lambda_{+,i}(\mathbf{s})}{\widetilde{\lambda}_{+,i}(\mathbf{s})}(\widetilde{\mathbf{u}}_{+,i}(\mathbf{s})^\top\mathbf{u}_{+,i}(\mathbf{s}))^2 + \sum_{i\neq j}^{(d_+(\mathbf{s}),d_+(\mathbf{s}))}\frac{\lambda_{+,i}(\mathbf{s})}{\widetilde{\lambda}_{+,j}(\mathbf{s})}(\widetilde{\mathbf{u}}_{+,j}(\mathbf{s})^\top\mathbf{u}_{+,i}(\mathbf{s}))^2 + \sum_{j=1}^{d_+(\mathbf{s})}\sum_{i=1}^{d_-(\mathbf{s})}\frac{\lambda_{-,i}(\mathbf{s})}{\widetilde{\lambda}_{+,j}(\mathbf{s})}(\widetilde{\mathbf{u}}_{+,j}(\mathbf{s})^\top\mathbf{u}_{-,i}(\mathbf{s}))^2\right] \tag{61}$$

We get the following inequality for $X(\mathbf{s})$ by applying Eq. (52-56):

$$\underbrace{\frac{1}{\widetilde{\alpha}(\mathbf{s})}\left\{(1-\epsilon)^3 - d_-(\mathbf{s})\eta\epsilon^2\right\}}_{=:X_l(\mathbf{s})} \leq X(\mathbf{s}) \leq \underbrace{\frac{1}{\widetilde{\alpha}(\mathbf{s})}\left\{(1+\epsilon)^3 + (d_+(\mathbf{s})-1)\eta\epsilon^2\right\}}_{=:X_u(\mathbf{s})} \tag{62}$$

Similarly, we get,

$$\underbrace{\frac{1}{\widetilde{\alpha}(\mathbf{s})}\left\{(1-\epsilon)^3 - d_+(\mathbf{s})\eta\epsilon^2\right\}}_{=:Y_l(\mathbf{s})} \leq Y(\mathbf{s}) \leq \underbrace{\frac{1}{\widetilde{\alpha}(\mathbf{s})}\left\{(1+\epsilon)^3 + (d_-(\mathbf{s})-1)\eta\epsilon^2\right\}}_{=:Y_u(\mathbf{s})} \tag{63}$$

The trace term $\mathrm{tr}(\widetilde{A}(\mathbf{s})^{-1}H(\mathbf{s}))$ $(= X(\mathbf{s}) - Y(\mathbf{s}))$ satisfies the following inequality:

$$X_l(\mathbf{s}) - Y_u(\mathbf{s}) \leq \mathrm{tr}(\widetilde{A}(\mathbf{s})^{-1}H(\mathbf{s})) \leq X_u(\mathbf{s}) - Y_l(\mathbf{s}) \tag{64}$$

Then the absolute value of the trace term satisfies the following inequality:

$$|\,\mathrm{tr}(\widetilde{A}(\mathbf{s})^{-1}H(\mathbf{s}))| \leq \max\{|X_u(\mathbf{s}) - Y_l(\mathbf{s})|, |X_l(\mathbf{s}) - Y_u(\mathbf{s})|\} \tag{65}$$

where, $|X_u(\mathbf{s}) - Y_l(\mathbf{s})|$, $|X_l(\mathbf{s}) - Y_u(\mathbf{s})|$ are,

$$|X_u(\mathbf{s}) - Y_l(\mathbf{s})| = \frac{1}{\widetilde{\alpha}(\mathbf{s})}\left\{6\epsilon + 2\epsilon^3 + \eta\epsilon^2(2d_+(\mathbf{s})-1)\right\} \tag{66}$$

$$|X_l(\mathbf{s}) - Y_u(\mathbf{s})| = \frac{1}{\widetilde{\alpha}(\mathbf{s})}\left\{6\epsilon + 2\epsilon^3 + \eta\epsilon^2(2d_-(\mathbf{s})-1)\right\}, \quad \text{where, } d_+(\mathbf{s}), d_-(\mathbf{s}) \geq 1, \epsilon \geq 0, \eta > 0 \tag{67}$$

By defining $d_{max}(\mathbf{s}) := \max\{d_+(\mathbf{s}), d_-(\mathbf{s})\}$, we can rewrite Eq. (65):

$$|\,\mathrm{tr}(\widetilde{A}(\mathbf{s})^{-1}H(\mathbf{s}))| \leq \frac{1}{\widetilde{\alpha}(\mathbf{s})}\left\{\eta\epsilon^2(2d_{max}(\mathbf{s})-1) + 6\epsilon + 2\epsilon^3\right\} \tag{68}$$

Then we get the following inequality for $\mathrm{LOBIAS}(h, \widetilde{A}(\mathbf{s}))^2$:

$$\mathrm{LOBIAS}(h, \widetilde{A})^2 \leq \left(\frac{h^2}{2\widetilde{\alpha}(\mathbf{s})}\right)^2\left[\eta\epsilon^2(2d_{max}(\mathbf{s})-1) + 6\epsilon + 2\epsilon^3\right]^2 \tag{69}$$

When there is no error in the Hessian estimation (when $\epsilon = 0$) acquired from the reward regressor, then the $\mathrm{LOBIAS}(h, \widetilde{A})^2$ becomes zero in Eq. (69). But when there is an error in the Hessian estimation (when $\epsilon > 0$), the KMIS estimator may have $\mathrm{LOBIAS}(h, \widetilde{A})^2$ up to $\left(\frac{h^2}{2\widetilde{\alpha}(\mathbf{s})}\right)^2\left[\eta\epsilon^2(2d_{max}(\mathbf{s})-1) + 6\epsilon + 2\epsilon^3\right]^2$.

Convergence Speed Analysis Now we analyze the convergence speed of KMIS estimator with $\widetilde{H}(\mathbf{s})$ with additional assumption that $|\widetilde{H}(\mathbf{s})|$ is bounded. By replacing in $\mathrm{LOBIAS}(h, A)^2$ in Eq. (43) with the derived upper bound of $\mathrm{LOBIAS}(h, \widetilde{A})^2$ in Eq. (69), and also using the relation $h^* = \mathcal{O}\left(\left(\frac{D_A}{N}\right)^{\frac{1}{D_A+4}}\right)$ from Eq. (3), we get:

$$\begin{aligned}
&\mathrm{MSE}(h^*, \widetilde{A}, N, D_A) \\
&\quad = (h^*)^4 \mathcal{O}\left(\left(\frac{d_{max}(\mathbf{s})}{\widetilde{\alpha}(\mathbf{s})}\right)^2 \eta^2 \epsilon^4\right) + \mathcal{O}\left((h^*)^6\right) + \frac{C_v}{N(h^*)^{D_A}} + \mathcal{O}\left(\frac{1}{N(h^*)^{D_A-2}}\right), \qquad (70) \\
&\quad = \mathcal{O}\left(\left(\frac{D_A}{N}\right)^{\frac{4}{D_A+4}} \left(\frac{d_{max}(\mathbf{s})}{\widetilde{\alpha}(\mathbf{s})}\right)^2 \eta^2 \epsilon^4\right) + \mathcal{O}\left(\left(\frac{D_A}{N}\right)^{\frac{4}{D_A+4}} \frac{1}{D_A}\right) \qquad (71) \\
&\quad \text{where, } N \gg D_A
\end{aligned}$$

As for the first term of Eq. (71),

$$\begin{aligned}
&\mathcal{O}\left(\left(\frac{D_A}{N}\right)^{\frac{4}{D_A+4}} \left(\frac{d_{max}(\mathbf{s})}{\widetilde{\alpha}(\mathbf{s})}\right)^2 \eta^2 \epsilon^4\right) \\
&\quad = \mathcal{O}\left(\left(\frac{D_A}{N}\right)^{\frac{4}{D_A+4}} (d_{max}(\mathbf{s}))^2 \underbrace{\left(d_+(\mathbf{s})^{d_+(\mathbf{s})} d_-(\mathbf{s})^{d_-(\mathbf{s})}\right)^{\frac{2}{D_A}}}_{=:Q(\mathbf{s})} \eta^2 \epsilon^4\right) \qquad (72) \\
&\quad \because |\widetilde{H}(\mathbf{s})| \text{ is bounded}
\end{aligned}$$

Assuming that $d_{max}(\mathbf{s}) = d_+(\mathbf{s})$, $Q(\mathbf{s})$ monotonically increases w.r.t. $d_+(\mathbf{s})$.

$$\begin{aligned}
Q(\mathbf{s}) &= \left(d_+(\mathbf{s})^{d_+(\mathbf{s})} (D_A - d_+(\mathbf{s}))^{(D_A - d_+(\mathbf{s}))}\right)^{\frac{2}{D_A}} \\
\frac{dQ(\mathbf{s})}{d(d_+(\mathbf{s}))} &= \frac{2}{D_A} d_+(\mathbf{s})^{\frac{2d_+(\mathbf{s})}{D_A}} (D_A - d_+(\mathbf{s}))^{\frac{2(D_A - d_+(\mathbf{s}))}{D_A}} \left(\ln \frac{d_+(\mathbf{s})}{D_A - d_+(\mathbf{s})}\right) \\
&\geq 0. \quad \left(\because D_A - 1 \geq d_+(\mathbf{s}) = d_{max}(\mathbf{s}) \geq \frac{D_A}{2}\right) \qquad (73)
\end{aligned}$$

The maximum of $Q(\mathbf{s})$ is (when $d_+(\mathbf{s}) = D_A - 1$),

$$Q(\mathbf{s}) = (D_A - 1)^{2 - \frac{2}{D_A}}. \qquad (74)$$

The minimum of $Q(\mathbf{s})$ is (when $d_+(\mathbf{s}) = \frac{D_A}{2}$),

$$Q(\mathbf{s}) = \left(\frac{D_A}{2}\right)^2. \qquad (75)$$

We can assume $d_{max}(\mathbf{s}) = d_-(\mathbf{s})$ and show that the maximum and minimum of $Q(\mathbf{s})$ are the same. Therefore,

$$\mathcal{O}(Q(\mathbf{s})) = (D_A)^2. \qquad (76)$$

With Eq. (76) and using the relation $D_A - 1 \geq d_{max}(\mathbf{s}) \geq \frac{D_A}{2}$ on Eq. (72),

$$\mathcal{O}\left(\left(\frac{D_A}{N}\right)^{\frac{4}{D_A+4}} \left(\frac{d_{max}(\mathbf{s})}{\widetilde{\alpha}(\mathbf{s})}\right)^2 \eta^2 \epsilon^4\right) = \mathcal{O}\left(\left(\frac{D_A}{N}\right)^{\frac{4}{D_A+4}} (D_A)^4 \eta^2 \epsilon^4\right) \qquad (77)$$

Plugging in Eq. (77) to Eq. (71),

$$\mathrm{MSE}(h^*, \widetilde{A}, N, D_A) = \mathcal{O}\left(\left(\frac{D_A}{N}\right)^{\frac{4}{D_A+4}} (D_A)^4 \eta^2 \epsilon^4\right) + \mathcal{O}\left(\left(\frac{D_A}{N}\right)^{\frac{4}{D_A+4}} \frac{1}{D_A}\right) \qquad (78)$$

When the estimated Hessian $\widetilde{H}(\mathbf{s})$ has no error (when $\epsilon = 0$), the convergence speed derived in Eq. (78) matches the convergence speed of KMIS estimator with a true Hessian derived in Appendix B.3. However, when there is an error in the estimated Hessian (when $\epsilon > 0$), KMIS estimator with the estimated Hessian has a slower convergence rate to a true policy value than the KMIS estimator with a true Hessian by the rate of $\mathcal{O}((D_A)^{\frac{5}{2}})$ (MSE of a KMIS estimator with the estimated Hessian converges to zero slower by the rate of $\mathcal{O}((D_A)^5)$ compared to the KMIS with a true Hessian).