Triply Robust Estimation for Continuous Treatments: Curvature-Aware Debiasing Beyond the $n^{-4/5}$ Barrier

Hongseok Namkoong
Columbia Business School
namkoong@gsb.columbia.edu

Isaac Scheinfeld
Columbia Business School
ils2124@columbia.edu

Yunbei Xu
National University of Singapore
yunbei@nus.edu.sg

Shunri Zheng
University of Illinois Urbana-Champaign
zshunri@gmail.com

Abstract

Continuous-treatment policy evaluation is a nonregular problem in which kernel smoothing typically induces an h^2 bias and the canonical mean-squared-error rate $n^{-4/5}$. We show that this barrier is not intrinsic. We propose a triply robust estimator that augments outcome regression and generalized propensity weighting with a higher-order nuisance component: local curvature of the conditional reward with respect to treatment. In pricing applications this extra nuisance is elasticity-aware, because the curvature of expected revenue encodes the own- and cross-price sensitivity primitives that determine marginal revenue. We derive a kernelized one-step expansion, finite-sample bias-variance formulas, and asymptotic normality for a cross-fitted version of the estimator. The main implication is that the leading smoothing bias depends on second-order nuisance error rather than on the curvature of the target regression itself. Under higher-order stochastic equicontinuity—a weaker requirement than imposing a global Hölder modulus on the regression function—and vanishing second-order nuisance error $r_n \to 0$, the estimator achieves $O(n^{-4/5} r_n^{2/5}) = o(n^{-4/5})$. We also establish a multidimensional extension and a formal policy-learning regret bound. Simulations and a semi-synthetic e-commerce pricing application show that the proposed estimator reduces bias and RMSE relative to existing continuous-treatment estimators.

1 Introduction

Continuous treatments arise whenever the policy variable is intrinsically graded rather than binary: price, dosage, advertising intensity, reserve levels, or portfolio weights. In such settings the econometrician observes outcomes only at the realized treatment level and must recover counterfactual outcomes at nearby treatment values. Continuous-treatment causal inference is therefore central to modern empirical work in economics, operations, marketing, and online experimentation [Kennedy et al., 2017, Kallus and Zhou, 2018, Colangelo and Lee, 2020, Wu et al., 2023].

This paper studies policy evaluation with a continuous treatment under unconfoundedness. The starting point is a familiar one: kernel localization is needed to target a point treatment value, and that localization creates an h^2 smoothing bias. Existing kernel-based doubly robust estimators therefore inherit the canonical nonparametric mean-squared-error rate $n^{-4/5}$ in the one-dimensional case. Our main question is whether that rate is unavoidable once one allows flexible nuisance estimation. We show that the answer is no.

Authors are listed in alphabetical order.

The key observation is that, for continuous treatments, the leading bias is governed by higher-order treatment derivatives. In discrete-treatment problems these derivative terms are absent, but in continuous-treatment problems they are the object that determines how quickly kernel localization can be debiased. This creates a third nuisance component beyond the conditional mean reward and the generalized propensity score. Our estimator exploits that extra structure, which is why we refer to it as *triply robust*.

The economic interpretation is particularly transparent in pricing. Let p denote price and let $q_0(p, x)$ be the conditional demand curve. Then the conditional mean revenue is $m_0(p, x) = pq_0(p, x)$, and

$$\partial_p^2 m_0(p, x) = 2\partial_p q_0(p, x) + p\partial_p^2 q_0(p, x).$$

Thus the curvature nuisance entering our estimator is *elasticity-aware*: it depends on the same local price-sensitivity primitives that govern own-price elasticity and marginal revenue. In multi-product pricing, mixed second derivatives play the analogous role for cross-price substitution. This link connects our econometric contribution to a large pricing literature in operations, marketing, and causal pricing analytics [den Boer, 2015, Reibstein and Gatignon, 1984, Guelman and Guillén, 2014].

Our estimator is constructed from a kernelized one-step expansion of the policy value functional. The resulting debiasing term differs from existing continuous-treatment estimators in one crucial respect: the kernel correction is evaluated at the realized treatment A, not only at the policy target $\theta(X)$. That change shifts the leading bias away from the curvature of the target regression itself and onto the curvature of the nuisance *error* process. This distinction is the source of the rate improvement.

The first theoretical contribution is a conditional bias-variance expansion for the cross-fitted triply robust estimator. The leading variance remains of order $(nh)^{-1}$, but the leading bias takes the form

$$\mathbb{E}[\Delta(\theta(X), X)\delta(\theta(X), X)] - \frac{\kappa h^2}{2}\mathbb{E}\Big[\partial_a^2\{\Delta(a, X)(1 - \delta(a, X))\}\big|_{a=\theta(X)}\Big],$$

up to a higher-order remainder. Relative to existing estimators, the decisive feature is that the h^2 term depends on curvature of the *nuisance error product* rather than curvature of m_0 itself.

The second contribution is an asymptotic theory that makes explicit what is needed for valid inference. We show that Assumptions 4.1 and 4.2 alone are not sufficient for asymptotic normality. A rigorous theorem also requires out-of-fold nuisance evaluation together with regularity conditions on the nuisance estimators. Once the estimator is cross-fitted, the summands are conditionally independent and a triangular-array central limit theorem applies. This yields $\sqrt{nh}$-normality after centering by the conditional bias term.

The third contribution is a rate-improvement result built around higher-order stochastic equicontinuity in the sense of Newey [1991]. We show that if the second-order nuisance error is $O(r_n)$ with $r_n \to 0$, then the mean-squared error is $O(n^{-4/5} r_n^{2/5})$, hence $o(n^{-4/5})$. This breaks the usual kernel barrier without assuming a stronger global Hölder modulus on the conditional mean function itself. The argument relies on higher-order local regularity of the *estimated* derivative process rather than on a fixed population smoothness class.

The paper also develops two extensions. First, we provide a multidimensional version of the estimator and establish its asymptotic normality under product kernels. Second, we formalize the connection between policy evaluation and policy learning by proving a regret bound for finite policy classes. The resulting bound makes clear that any improvement in policy-value estimation transfers directly to policy learning.

Empirically, we compare the triply robust estimator to IPW, the estimator of Colangelo and Lee [2020], and the local-linear estimator of Kennedy et al. [2017]. Simulations show systematically

lower bias and lower RMSE, especially when larger bandwidths are used. A semi-synthetic pricing application based on the JD.com data of Shen et al. [2020] delivers the same pattern: the proposed estimator becomes increasingly advantageous as the sample size grows.

Organization. Section 3 introduces the setup, the estimator, and the kernelized one-step expansion. Section 4 develops the identification assumptions, bias-variance decomposition, asymptotic normality, and the rate-improvement result under higher-order stochastic equicontinuity. Section 5 presents the multidimensional extension, the policy-learning result, and pricing examples that clarify the elasticity interpretation. Section 6 reports the simulation and pricing evidence. Section 7 concludes. Appendix A collects proofs and implementation details.

2 Preliminaries

2.1 Problem Setting

We observe i.i.d. data $Z_i = (X_i, A_i, Y_i)$, where $X_i \in \mathcal{X}$ is a vector of covariates, $A_i \in \mathcal{A} \subseteq \mathbb{R}$ is a continuously valued treatment, and $Y_i = Y_i(A_i)$ is the realized outcome. For each treatment level $a \in \mathcal{A}$, let $Y(a)$ denote the potential outcome under treatment a. A target policy is a measurable map $\theta : \mathcal{X} \to \mathcal{A}$, and its value is

$$V(\theta) = \mathbb{E}\big[Y(\theta(X))\big].$$

Under the standard unconfoundedness and overlap conditions introduced in Section 4,

$$V(\theta) = \mathbb{E}\big[m_0(\theta(X), X)\big], \qquad m_0(a, x) := \mathbb{E}[Y \mid A = a, X = x], \qquad f_0(a \mid x) := f_{A|X}(a \mid x).$$

Our goal is to estimate $V(\theta)$ from the observational sample without knowing either m_0 or f_0.

2.2 Pricing examples and elasticity-aware curvature

We focus the paper on pricing because it makes the extra nuisance component economically interpretable.

Example 2.1 (Single-product pricing)**.** In contextual pricing, the treatment is price $p \in \mathbb{R}_+$ and the outcome is revenue. Let $q_0(p, x)$ denote conditional demand. Then

$$m_0(p, x) = p\, q_0(p, x), \qquad \partial_p^2 m_0(p, x) = 2\partial_p q_0(p, x) + p\partial_p^2 q_0(p, x).$$

The extra curvature term therefore depends on the same local demand-slope primitives that determine own-price elasticity and marginal revenue. This is why the third nuisance component is naturally interpreted as *elasticity-aware*. Dynamic pricing and demand learning are standard in revenue management and operations [den Boer, 2015], while causal identification of price elasticity from observational data has also been emphasized in applied pricing settings [Guelman and Guillén, 2014].

Example 2.2 (Multi-product pricing)**.** Let $A = (p_1, \ldots, p_d) \in \mathbb{R}_+^d$ denote a vector of prices and let $q_{0j}(A, x)$ be demand for product j. For expected revenue

$$m_0(A, x) = \sum_{j=1}^{d} p_j q_{0j}(A, x),$$

the mixed partial derivative $\partial^2_{p_k p_\ell} m_0(A, x)$ loads on cross-price demand responses. Hence the Hessian of the conditional reward encodes substitution patterns across products. This is the object that matters for joint pricing whenever cross-elasticities are economically relevant [Reibstein and Gatignon, 1984].

3 The Triply Robust Estimator

Let $K : \mathbb{R} \to \mathbb{R}$ be a second-order kernel and let $h > 0$ be a bandwidth. For readability we suppress fold indices throughout the main text. In Sections 4 and 5, $\hat{m}$ and $\hat{f}$ should be understood as *out-of-fold* nuisance estimates produced by sample splitting or cross-fitting.

3.1 Estimator and basic comparison

We propose the triply robust estimator

$$\hat{V}^{\mathrm{TR}}(\theta) = \frac{1}{n}\sum_{i=1}^{n}\left[\hat{m}\big(\theta(X_i), X_i\big) + \frac{\frac{1}{h}K\left(\frac{A_i-\theta(X_i)}{h}\right)}{\hat{f}(A_i \mid X_i)}\Big(Y_i - \hat{m}(A_i, X_i)\Big)\right]. \tag{3.1}$$

The crucial difference relative to existing continuous-treatment estimators is that the kernel correction is evaluated at the realized treatment A_i and uses the residual $Y_i - \hat{m}(A_i, X_i)$. This is what allows the leading bias to depend on higher-order *nuisance error* rather than on the curvature of the target regression itself.

For comparison, Colangelo and Lee [2020] study the estimator

$$\hat{V}^{\mathrm{CL}}(\theta) = \frac{1}{n}\sum_{i=1}^{n}\left[\hat{m}\big(\theta(X_i), X_i\big) + \frac{\frac{1}{h}K\left(\frac{A_i-\theta(X_i)}{h}\right)}{\hat{f}(\theta(X_i) \mid X_i)}\Big(Y_i - \hat{m}\big(\theta(X_i), X_i\big)\Big)\right]. \tag{3.2}$$

The difference looks minor, but it changes the leading bias term and therefore the rate calculations in Section 4. In particular, when $\hat{m} = m_0$, our estimator is exactly unbiased conditional on the nuisance fits,

$$\mathbb{E}\Big[\hat{V}^{\mathrm{TR}}(\theta) \mid \hat{m} = m_0, \hat{f}\Big] = V(\theta), \tag{3.3}$$

whereas $\hat{V}^{\mathrm{CL}}$ continues to incur kernel smoothing bias because the residual correction is still localized at $\theta(X)$ rather than A.

The inverse-probability-weighted estimator of Kallus and Zhou [2018] is the special case obtained by setting $\hat{m} \equiv 0$ in (3.1). The local-linear estimator of Kennedy et al. [2017] uses a different debiasing device based on local polynomial fitting. Section 4 makes precise how the proposed estimator differs from both classes of methods.

3.2 Kernelized one-step expansion and derivation

Throughout the paper, define the nuisance errors

$$\Delta(a, x) := \hat{m}(a, x) - m_0(a, x), \qquad \delta(a, x) := 1 - \frac{f_0(a \mid x)}{\hat{f}(a \mid x)}.$$

Consider the policy-value functional

$$\Psi(P) := \mathbb{E}_P\big[m_P(\theta(X), X)\big].$$

A convenient way to motivate (3.1) is through a kernelized one-step expansion. The next result records the algebraic decomposition that drives the estimator.

Theorem 3.1 (Kernelized one-step expansion)**.** *For any two distributions P and $\bar{P}$ with conditional mean functions $m, \bar{m}$ and conditional treatment densities $f, \bar{f}$,*

$$\Psi(\bar{P}) - \Psi(P) = \int \varphi_h(z; \bar{P})\, d(\bar{P} - P)(z) + R_2(\bar{P}, P),$$

where

$$\varphi_h(z; P) = \frac{\frac{1}{h}K\Big(\frac{A-\theta(X)}{h}\Big)}{f(A \mid X)}\{Y - m(A, X)\} + m(\theta(X), X) - \Psi(P)$$

is mean zero under P, and

$$R_2(\bar{P}, P) = \int \Bigg[\bar{\Delta}(\theta(x), x)\bar{\delta}(\theta(x), x) + \Big(1 - \bar{\delta}(\theta(x), x)\Big)\bar{\Delta}(\theta(x), x) - \int \frac{1}{h}K\left(\frac{a - \theta(x)}{h}\right)\Big(1 - \bar{\delta}(a, x)\Big)\bar{\Delta}(a, x)\, da\Bigg] dP(x),$$

with $\bar{\Delta}(a, x) = \bar{m}(a, x) - m(a, x)$ and $\bar{\delta}(a, x) = 1 - f(a \mid x)/\bar{f}(a \mid x)$.

The decomposition shows that the leading remainder is not only the familiar doubly robust product $\bar{\Delta}\,\bar{\delta}$, but also an additional term comparing the localized average of $(1-\bar{\delta})\bar{\Delta}$ to its value at the target treatment. That extra term is exactly where second-order treatment derivatives enter. Applying the one-step correction to the plug-in estimator $\Psi(\hat{P})$ yields (3.1).

Theorem 3.1 also clarifies the source of the higher-order nuisance component. The bias is controlled by the treatment curvature of

$$g(a, x) := \Delta(a, x)\big(1 - \delta(a, x)\big),$$

not by the curvature of m_0 alone. In pricing applications, this is the formal reason the estimator adapts to elasticity information contained in the nuisance fits.

4 Theoretical Properties

Throughout this section, all expectations, variances, and stochastic orders are understood *conditional on the auxiliary sample used to estimate the out-of-fold nuisance functions.* This convention is standard in cross-fitted semiparametric theory and isolates the triangular-array argument from first-stage estimation noise.

4.1 Identification and kernels

Our identification target is

$$V(\theta) = \mathbb{E}\big[Y(\theta(X))\big] = \mathbb{E}\big[m_0(\theta(X), X)\big].$$

To connect this counterfactual quantity to observed data, we impose standard continuous-treatment identification conditions; see Gill and Robins [2001] for a general discussion.

Assumption 4.1.

(a) **Conditional independence.** A and $Y(a)$ are conditionally independent given X for every $a \in \mathcal{A}$.

(b) **Overlap.** $\inf_{a\in\mathcal{A}} \operatorname{ess\,inf}_{x\in\mathcal{X}} f_0(a \mid x) \geq c$ for some $c > 0$.

(c) **Smoothness of the observed law.** The joint density of (Y, A, X) is three-times differentiable with respect to a, and all derivatives up to order three are uniformly bounded.

(d) **Moment bounds.** $\mathrm{Var}(Y \mid A = a, X = x)$ and its derivatives with respect to a are uniformly bounded.

We use a second-order kernel to localize the treatment around the policy target.

Assumption 4.2 (Kernel)**.** The kernel K is bounded, differentiable, and symmetric, with

$$\int K(u)\,du = 1, \qquad \int uK(u)\,du = 0, \qquad \kappa := \int u^2 K(u)\,du \in (0,\infty), \qquad R(K) := \int K(u)^2\,du < \infty.$$

In addition, $\mu_3(K) := \int |u|^3|K(u)|\,du < \infty$ and $\int |K(u)|^3\,du < \infty$. For some constants $C, \bar{U} > 0$ and some $\nu > 1$, $|K'(u)| \leq C|u|^{-\nu}$ for $|u| > \bar{U}$.

4.2 Finite-sample bias and variance

Define

$$g(a,x) := \Delta(a,x)\big(1 - \delta(a,x)\big) = \Delta(a,x)\frac{f_0(a \mid x)}{\hat{f}(a \mid x)}.$$

The mean-squared error decomposes as

$$\mathbb{E}\Big[\big(\hat{V}^{\mathrm{TR}}(\theta) - V(\theta)\big)^2\Big] = \underbrace{\mathrm{Var}\Big(\hat{V}^{\mathrm{TR}}(\theta)\Big)}_{\text{Variance}} + \underbrace{\Big(\mathbb{E}\big[\hat{V}^{\mathrm{TR}}(\theta)\big] - V(\theta)\Big)^2}_{\text{Bias}^2}. \tag{4.1}$$

The next theorem gives the leading terms.

Theorem 4.3 (Bias and variance)**.** *Let Assumptions 4.1 and 4.2 hold. Then, conditional on the auxiliary sample used to estimate the nuisances,*

$$\begin{aligned}\text{Bias} &= B_{0,n}(\theta) - \frac{\kappa h^2}{2} B_{2,n}(\theta) + O(h^3),\\ \text{Variance} &= \frac{R(K)}{nh}\,\Gamma_n(\theta) + O\left(\frac{1}{n}\right),\end{aligned}$$

where

$$B_{0,n}(\theta) := \mathbb{E}\big[\Delta(\theta(X), X)\delta(\theta(X), X)\big], \qquad B_{2,n}(\theta) := \mathbb{E}\Big[\partial_a^2 g(a, X)\big|_{a=\theta(X)}\Big],$$

and

$$\Gamma_n(\theta) := \mathbb{E}_X\left[\frac{f_0(\theta(X) \mid X)}{\hat{f}(\theta(X) \mid X)^2}\mathbb{E}\big[(Y - \hat{m}(\theta(X), X))^2 \mid A = \theta(X), X\big]\right].$$

Theorem 4.3 makes the central point of the paper explicit: the leading h^2 bias is governed by the curvature of the nuisance-error product g, not by the curvature of m_0 itself.

Corollary 4.3A (Finite-sample MSE bound). Let the assumptions of Theorem 4.3 hold, and let

$$M_{3,n}(\theta) := \mathbb{E}\left[\sup_{|u|\le h} \left|\partial_a^3 g(\theta(X)+u, X)\right|\right].$$

Then

$$|\text{Bias}| \le |B_{0,n}(\theta)| + \frac{\kappa h^2}{2}|B_{2,n}(\theta)| + \frac{\mu_3(K)}{6} M_{3,n}(\theta) h^3,$$

and therefore

$$\mathbb{E}\Big[\big(\hat{V}^{\mathrm{TR}}(\theta) - V(\theta)\big)^2\Big] \le 3B_{0,n}(\theta)^2 + \frac{3\kappa^2}{4} B_{2,n}(\theta)^2 h^4 + \frac{\mu_3(K)^2}{12} M_{3,n}(\theta)^2 h^6 + \frac{R(K)}{nh}\,\Gamma_n(\theta) + O\left(\frac{1}{n}\right).$$

In particular, whenever $B_{0,n}(\theta) = O(h^2)$ and $B_{2,n}(\theta) = O(1)$, the worst-case rate remains $O(h^4 + (nh)^{-1})$.

4.3 Asymptotic normality

Assumptions 4.1 and 4.2 are not sufficient for asymptotic normality by themselves. A formal limit theorem also needs out-of-fold nuisance evaluation together with regularity conditions on the nuisance fits. We state those conditions here without creating additional numbered assumptions, so that Theorem 4.4 keeps its original numbering.

Additional conditions for Theorem 4.4.

(N1) **Cross-fitting.** The estimator is evaluated out of sample using a fixed number L of folds.

(N2) **Nuisance regularity.** With probability approaching one, $\inf_{a,x} \hat{f}(a \mid x) \ge c_0 > 0$, and the nuisance fits admit derivatives up to order three in a neighborhood of $\theta(\mathcal{X})$ that are bounded in probability.

(N3) **Small first-order remainder.** $B_{0,n}(\theta) = o_p(h^2)$.

(N4) **Third moments.** $\sup_{a,x} \mathbb{E}[|Y|^3 \mid A = a, X = x] < \infty$.

Theorem 4.4 (Asymptotic normality)**.** *Let Assumptions 4.1 and 4.2 and conditions (N1)–(N4) hold. If $h \to 0$ and $nh \to \infty$, then, conditional on the auxiliary sample,*

$$\sqrt{\frac{nh}{C_n(\theta)}}\left\{\hat{V}^{\mathrm{TR}}(\theta) - V(\theta) - b_n(\theta)\right\} \xrightarrow{d} N(0,1),$$

where

$$b_n(\theta) := B_{0,n}(\theta) - \frac{\kappa h^2}{2} B_{2,n}(\theta), \qquad C_n(\theta) := R(K)\,\Gamma_n(\theta).$$

In particular, if $B_{0,n}(\theta) = o_p((nh)^{-1/2})$ and $h^2 B_{2,n}(\theta) = o_p((nh)^{-1/2})$, then the estimator is centered at $V(\theta)$ up to an asymptotically negligible term.

The proof is a conditional triangular-array CLT. Cross-fitting is what makes that argument legitimate: without it, the summands depend on first-stage estimates trained on the same observations, and the simple Lyapunov proof could not go through.

4.4 Breaking the $n^{-4/5}$ barrier via higher-order stochastic equicontinuity

Theorem 4.3 implies that the canonical $n^{-4/5}$ rate is driven by the size of $B_{2,n}(\theta)$. The next result formalizes how faster rates arise when the second-order nuisance error is small.

Theorem 4.5 (Rates beyond $n^{-4/5}$)**.** *Suppose the conditions of Theorems 4.3 and 4.4 hold. Assume there is a deterministic sequence $r_n \to 0$ such that*

$$|B_{0,n}(\theta)| \le c_0 r_n h^2, \qquad |B_{2,n}(\theta)| \le c_1 r_n,$$

for constants $c_0, c_1 < \infty$. Then

$$\mathbb{E}\Big[\big(\hat{V}^{\mathrm{TR}}(\theta) - V(\theta)\big)^2\Big] = O\big(r_n^2 h^4\big) + O\big((nh)^{-1}\big).$$

Choosing

$$h \asymp (n r_n^2)^{-1/5}$$

yields

$$\mathbb{E}\Big[\big(\hat{V}^{\mathrm{TR}}(\theta) - V(\theta)\big)^2\Big] = O\big(n^{-4/5} r_n^{2/5}\big) = o\big(n^{-4/5}\big).$$

Theorem 4.5 is the paper's main rate statement. It shows that the usual kernel barrier is broken whenever the leading second-order nuisance error vanishes.

A convenient sufficient condition uses *stochastic equicontinuity* for the random derivative process; see Newey [1991] and Pötscher and Prucha [1994]. Let

$$G_n(a, x) := \partial_a^2 g(a, x).$$

If, for some $r_n \to 0$,

$$\sup_{|u| \le h} \mathbb{E}\big|G_n(\theta(X) + u, X) - G_n(\theta(X), X)\big| = o_p(r_n) \qquad \text{and} \qquad \mathbb{E}\big|G_n(\theta(X), X)\big| = O_p(r_n),$$

then $B_{2,n}(\theta) = O_p(r_n)$. This condition is weaker than imposing a global Hölder modulus on m_0 or f_0: it only requires a probabilistic local modulus for the *estimated second-order error process*. That is the right object for continuous-treatment inference with flexible first-stage learners.

4.5 Comparison with existing continuous-treatment estimators

The main contrast with the existing literature is easiest to summarize through the leading bias term.

Estimator	Leading variance	Leading h^2 bias depends on
IPW [Kallus and Zhou, 2018]	$(nh)^{-1}$	curvature of the target regression
CL [Colangelo and Lee, 2020]	$(nh)^{-1}$	curvature of the target regression
KMMS [Kennedy et al., 2017]	$(nh)^{-1}$	curvature of the target regression
TR (this paper)	$(nh)^{-1}$	curvature of the nuisance-error product g

This table is the substantive contribution of the triply robust construction. Existing estimators pay the canonical h^2 smoothing cost unless the target regression itself is exceptionally well behaved. Our estimator instead adapts to how well higher-order nuisance terms are learned. In pricing language, the estimator becomes effective when elasticity-relevant curvature is estimable even if the full reward surface is not globally smooth in a strong Hölder sense.

5 Extensions and pricing applications

5.1 Extension to multidimensional treatments

We now allow $A \in \mathbb{R}^d$ for fixed and low dimension d. Let

$$K_d(u) := \prod_{j=1}^{d} K(u_j), \qquad K_{d,h}(a-\theta(x)) := \frac{1}{h^d} K_d\left(\frac{a-\theta(x)}{h}\right).$$

The multidimensional triply robust estimator is

$$\hat{V}_d^{\mathrm{TR}}(\theta) = \frac{1}{n}\sum_{i=1}^{n}\left[\hat{m}\big(\theta(X_i), X_i\big) + \frac{K_{d,h}(A_i-\theta(X_i))}{\hat{f}(A_i \mid X_i)}\Big(Y_i - \hat{m}(A_i, X_i)\Big)\right]. \tag{5.1}$$

Because the product kernel has zero first moments coordinatewise, the same argument as in the one-dimensional case yields a diagonal Hessian bias term.

Theorem 5.1 (Asymptotic normality in the multidimensional setting)**.** *Let Assumptions 4.1 and 4.2 hold, and suppose the multidimensional analogue of conditions (N1)–(N4) in Section 4.3 holds for the cross-fitted estimator (5.1). If $h \to 0$ and $nh^d \to \infty$, then, conditional on the auxiliary sample,*

$$\sqrt{\frac{nh^d}{C_{n,d}(\theta)}}\left\{\hat{V}_d^{\mathrm{TR}}(\theta) - V(\theta) - b_{n,d}(\theta)\right\} \xrightarrow{d} N(0,1),$$

where

$$b_{n,d}(\theta) = B_{0,n,d}(\theta) - \frac{\kappa h^2}{2}\sum_{j=1}^{d}\mathbb{E}\Big[\partial^2_{a_j a_j} g(a,X)\big|_{a=\theta(X)}\Big],$$

with $B_{0,n,d}(\theta) = \mathbb{E}[\Delta(\theta(X),X)\delta(\theta(X),X)]$, and

$$C_{n,d}(\theta) = R(K)^d \mathbb{E}_X\left[\frac{f_0(\theta(X)\mid X)}{\hat{f}(\theta(X)\mid X)^2}\mathbb{E}\big[(Y-\hat{m}(\theta(X),X))^2 \mid A=\theta(X), X\big]\right].$$

The rate implication is immediate. In the worst case, balancing h^4 bias squared and $(nh^d)^{-1}$ variance yields the familiar $n^{-4/(d+4)}$ convergence rate. If the second-order nuisance term is $O(r_n)$ with $r_n \to 0$, then the same calculation as in Theorem 4.5 gives the faster rate $O\big(n^{-4/(d+4)} r_n^{2d/(d+4)}\big)$.

5.2 From Policy Evaluation to Policy Learning

We now connect policy evaluation accuracy to policy learning. Let

$$\theta^* \in \arg\max_{\theta\in\Theta} V(\theta), \qquad \hat{\theta} \in \arg\max_{\theta\in\Theta} \hat{V}^{\mathrm{TR}}(\theta).$$

The basic regret decomposition is deterministic:

$$V(\theta^*) - V(\hat{\theta}) \le 2\sup_{\theta\in\Theta}\left|\hat{V}^{\mathrm{TR}}(\theta) - V(\theta)\right|.$$

To obtain the usual $\log|\Theta|$ finite-class complexity term, one needs more than second moments. We therefore state the theorem under an explicit sub-Gaussian concentration condition on the centered policy-value errors.

Theorem 5.2 (Policy Learning Regret Bound)**.** *Suppose Θ is finite and that there exist deterministic sequences $b_n \geq 0$ and $\sigma_n \geq 0$ such that, for every $\theta \in \Theta$,*

$$\left|\mathbb{E}\left[\hat{V}^{\mathrm{TR}}(\theta) - V(\theta)\right]\right| \leq b_n,$$

and, for every $\lambda \in \mathbb{R}$,

$$\mathbb{E}\left[\exp\left\{\lambda\left(\hat{V}^{\mathrm{TR}}(\theta) - \mathbb{E}\left[\hat{V}^{\mathrm{TR}}(\theta)\right]\right)\right\}\right] \leq \exp\left(\frac{\lambda^2\sigma_n^2}{2}\right).$$

Then the policy $\hat{\theta} \in \arg\max_{\theta\in\Theta} \hat{V}^{\mathrm{TR}}(\theta)$ satisfies

$$\mathbb{E}\left[V(\theta^*) - V(\hat{\theta})\right] \leq 2b_n + 2\sigma_n\sqrt{2\log(2|\Theta|)}.$$

In particular, if $b_n = O(h^2)$ and $\sigma_n = O((nh)^{-1/2})$, then

$$\mathbb{E}\left[V(\theta^*) - V(\hat{\theta})\right] = O(h^2) + O\left(\sqrt{\frac{\log|\Theta|}{nh}}\right).$$

With the worst-case bandwidth choice $h = \Theta(n^{-1/5})$, this yields

$$\mathbb{E}\left[V(\theta^*) - V(\hat{\theta})\right] = O\left(\sqrt{\log|\Theta|}\, n^{-\frac{2}{5}}\right).$$

Remark. The $\log|\Theta|$ factor does *not* follow from the second-moment bound alone. It requires the explicit sub-Gaussian condition above, or some other exponential-tail control for the centered policy-value process. Without such a condition, the direct second-moment argument only gives a $\sqrt{|\Theta|}$ factor.

The proof of Theorem 5.2 can be found in Appendix A.3. The theorem shows that any improvement in policy-value estimation transfers directly to policy learning. When the leading bias term improves to $O(r_n h^2)$ with $r_n \to 0$, the same balancing argument gives the faster regret rate $O\left(\sqrt{\log|\Theta|}\, n^{-\frac{2}{5}} r_n^{\frac{1}{5}}\right)$.

5.3 Elasticity-aware nuisance modeling in pricing

The pricing interpretation is the most natural application of the triply robust construction because the higher-order nuisance has a direct economic meaning.

Example 1. Single-product contextual pricing

Let $p \in \mathbb{R}_+$ be price, $q_0(p,x)$ demand, and $m_0(p,x) = pq_0(p,x)$ expected revenue. The own-price elasticity is

$$\varepsilon(p,x) := -\frac{p\,\partial_p q_0(p,x)}{q_0(p,x)} = 1 - \frac{p\,\partial_p m_0(p,x)}{m_0(p,x)},$$

whenever $m_0(p,x) > 0$. The first derivative of revenue therefore determines elasticity, while the second derivative

$$\partial_p^2 m_0(p,x) = 2\partial_p q_0(p,x) + p\partial_p^2 q_0(p,x)$$

controls the local slope of marginal revenue. This is exactly the object entering the leading bias term of Theorem 4.3. In this sense, our third nuisance component is not a generic mathematical

add-on: it is the local price-sensitivity object that pricing models already use. This viewpoint aligns continuous-treatment causal inference with the pricing and demand-learning literature surveyed by den Boer [2015] and with causal formulations of price elasticity estimation such as Guelman and Guillén [2014].

Example 2. Multi-product pricing and cross-elasticities

Let $A = (p_1, \ldots, p_d)$ be a vector of prices and let $q_{0j}(A, x)$ be demand for product j. Expected revenue is

$$m_0(A, x) = \sum_{j=1}^{d} p_j q_{0j}(A, x).$$

The Hessian of m_0 loads on cross-price responses. For $k \neq \ell$,

$$\partial^2_{p_k p_\ell} m_0(A, x) = \partial_{p_\ell} q_{0k}(A, x) + \partial_{p_k} q_{0\ell}(A, x) + \sum_{j=1}^{d} p_j \partial^2_{p_k p_\ell} q_{0j}(A, x).$$

Hence the second-order nuisance terms encode the same substitution patterns that appear in cross-elasticity matrices and joint product-line pricing. Estimating these objects is essential whenever prices are set jointly rather than product by product [Reibstein and Gatignon, 1984]. In practice, the multidimensional triply robust estimator is attractive precisely because it can exploit structure in the cross-price derivative process while retaining robustness to first-stage misspecification.

6 Numerical Experiments

In this section we study the finite-sample performance of the proposed Triply Robust (TR) estimator and compare it with several existing off-policy estimators, including inverse probability weighting (IPW) and the continuous-treatment estimators of Colangelo and Lee [2020] (CL) and Kennedy et al. [2017] (KMMS). The empirical question is directly tied to the theory: does replacing the generic h^2 smoothing bias by second-order nuisance error let TR operate with larger bandwidths and lower RMSE? Our target parameter is the expected reward of a policy θ, defined as

$$V(\theta) = \mathbb{E}\left[Y \mid A = \theta(X), X\right] = \mathbb{E}\left[m_0(\theta(X), X)\right].$$

We consider two complementary settings. First, we use controlled simulations with known ground truth to isolate the bias–variance trade-off and to check whether TR indeed benefits from reduced smoothing bias under favorable structure. Second, we apply the estimators to a semi-synthetic pricing problem constructed from transaction-level data from JD.com [Shen et al., 2020]. This pricing application is especially natural for the paper's main message because treatment curvature is economically interpretable through elasticity and marginal-revenue structure.

6.1 Implementation details

We estimate the key nuisance components $m_0(a, x)$ and $f_{A|X}(a \mid x)$ using flexible machine learning methods. For $\hat{m}$, we use a random forest model due to its robustness and ability to capture nonlinear

relationships. The estimation of $\hat{f}$ is approached as a two-stage process: we first employ a random forest to estimate the policy value $\theta(X)$, and then apply Gaussian kernel density estimation to approximate the conditional treatment density around observed treatments. Hyperparameters in both stages are tuned using 5-fold cross-validation to balance flexibility and generalization.

For the competing estimators, we follow the implementation choices recommended in the original papers wherever possible. In particular, CL and KMMS are implemented with the same family of nuisance learners as TR so that performance differences can be attributed to the estimators themselves rather than to differences in first-stage modeling. A comprehensive description of our procedures, including specific hyperparameter configurations and additional diagnostics, is provided in Appendix A.5.2.

6.2 Simulation study

6.2.1 Design

Sample Size The simulations involve a training sample size of $N_1 = 500$ and an evaluation sample size of $N_2 = 500$. For each setting, we run five parallel experiments, each of which consists of 20 bootstrapped estimations so that we can get a better estimate of the bias and variance of the trained estimators. The performance of the estimators is measured using two key metrics: bias and root mean squared error (RMSE).

Dimensionality settings To demonstrate that our TR estimator's advantages over baseline methods are robust across different dimensionalities, we conduct experiments in both single-dimensional and ten-dimensional settings. In each case, we provide detailed specifications for the latent variable Z, behavior and evaluation policies, and the outcome function. The complete experimental designs are presented in Appendix A.5.

Outcome Function Specifications We employ two distinct outcome functions in our experiments to evaluate the performance of our estimators across different scenarios:

- **One-dimensional case:** For the one-dimensional setting, we use the following outcome function:
$$m_0(Z, A) = 4\sin(Z) - 4\sin(A) + (A - 4\pi)^2.$$
- **Ten-dimensional case:** For the ten-dimensional setting, we employ a more complex outcome function:
$$m_0(Z, A) = \frac{2}{5}\sum_{j=1}^{10}\sin(Z_j) - 4\sin(A) + \frac{1}{2}(A - 10\pi)^2.$$

These outcome functions are chosen based on the Fourier series theorem, which suggests that sinusoidal functions can approximate a wide range of continuous functions. This choice provides a reasonable proxy for continuous causal environments, allowing us to test our estimators under realistic yet mathematically tractable conditions.

6.2.2 Propensity score coverage

Coverage of behavior and evaluation policies To ensure robust estimation, it is crucial that the joint distribution of covariates and treatments (X, A) under the behavior policy adequately

covers that induced by the target policy. Figures 1 and 2 illustrate this coverage relationship in the one-dimensional case.

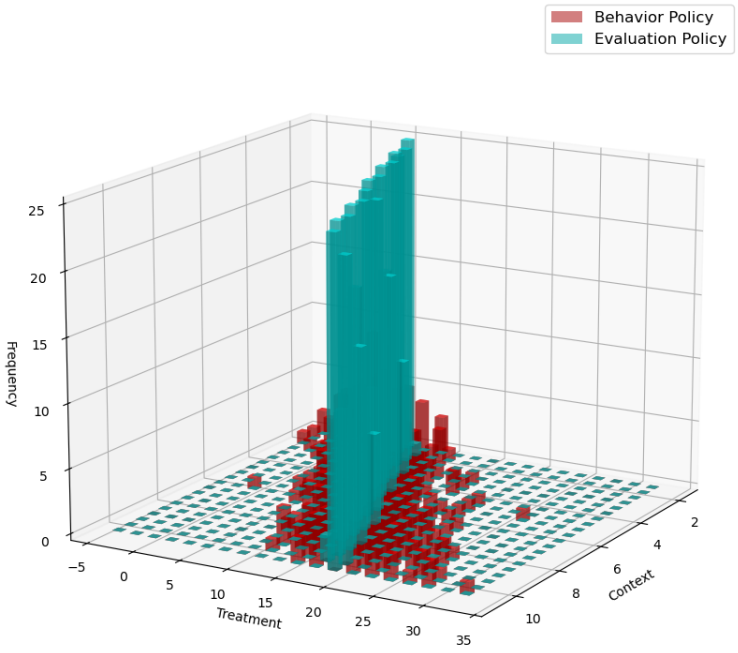

Figure 1: Empirical joint distribution of covariates and treatments induced by the behavior and evaluation policies in the one-dimensional simulation.

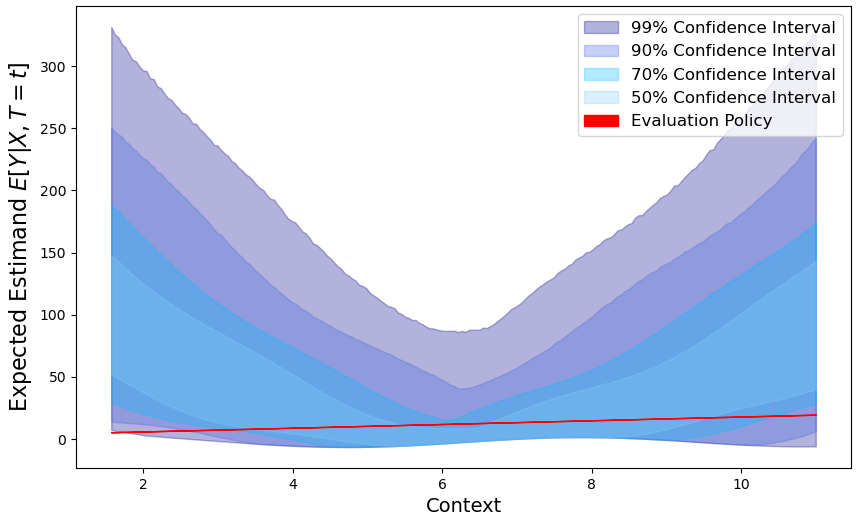

Figure 2: Overlap in (X, A) between behavior (blue) and evaluation (red) policies in the one-dimensional simulation design.

Figure 2 shows that the behavior policy induces a wide range of treatment values across the covariate space and that the evaluation policy places most of its mass in regions where the behavior policy has non-negligible support. The substantial overlap between the blue and red regions is essential for reliable off-policy evaluation, as it ensures that importance weights do not explode.

At the same time, the behavior policy exhibits areas of both high and low treatment density, corresponding to regions with larger and smaller effective sample sizes. This mix of dense and sparse areas creates a realistic test bed in which estimators must perform well under strong overlap while remaining stable in regions that require moderate extrapolation.

6.2.3 Estimator performance

We analyze the performance of different estimators in terms of RMSE and bias, comparing their behavior in both one-dimensional and ten-dimensional settings. Figures 3 and 4 report RMSE and bias as functions of the bandwidth in the one-dimensional case, while Figures 5 and 6 present the corresponding results in ten dimensions. The full results, including IPW, are provided in Appendix A.5.5[1].

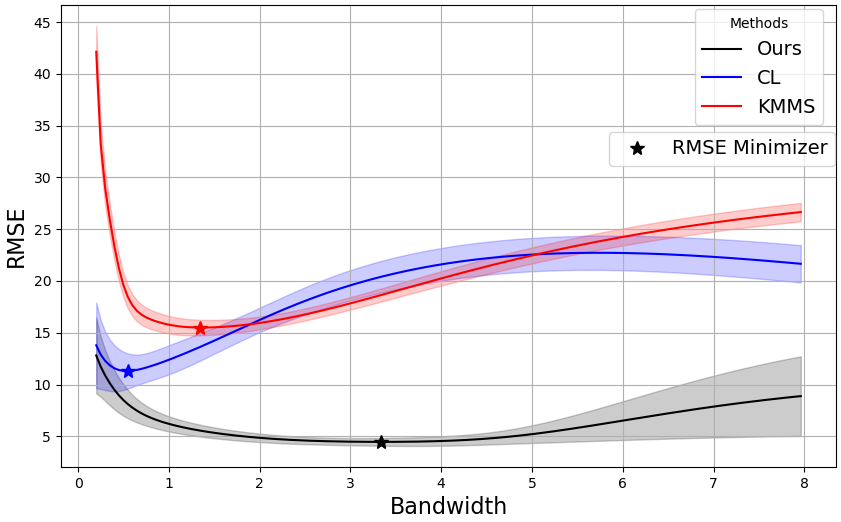

Figure 3: RMSE of TR, CL, and KMMS estimators across bandwidths in the one-dimensional simulation (IPW omitted due to much larger errors). TR attains the lowest RMSE at its optimal bandwidth and degrades more slowly away from the optimum.

[1]The IPW estimator consistently shows substantially higher RMSE and bias compared to the other methods across all bandwidth values in both dimensionality settings, hence its exclusion from the main figures.

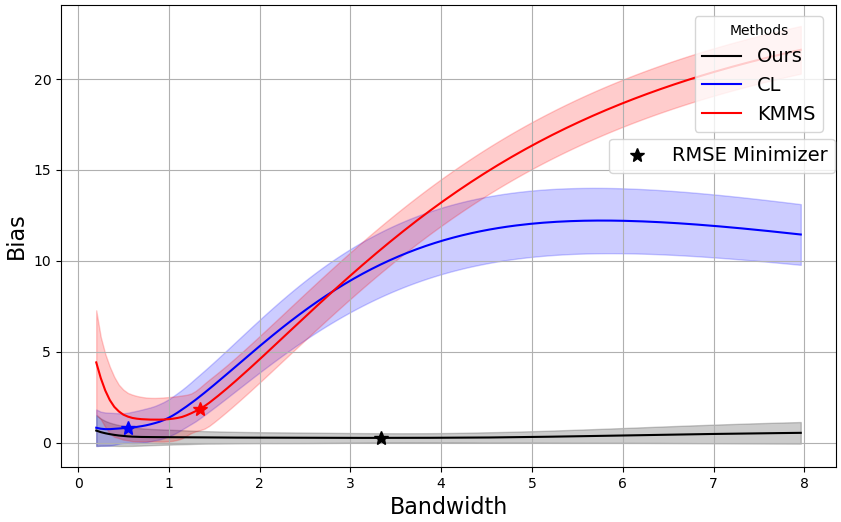

Figure 4: Bias of TR, CL, and KMMS estimators across bandwidths in the one-dimensional simulation (IPW omitted). TR exhibits smaller bias near the RMSE-minimizing bandwidth and can use slightly larger bandwidths without incurring large bias.

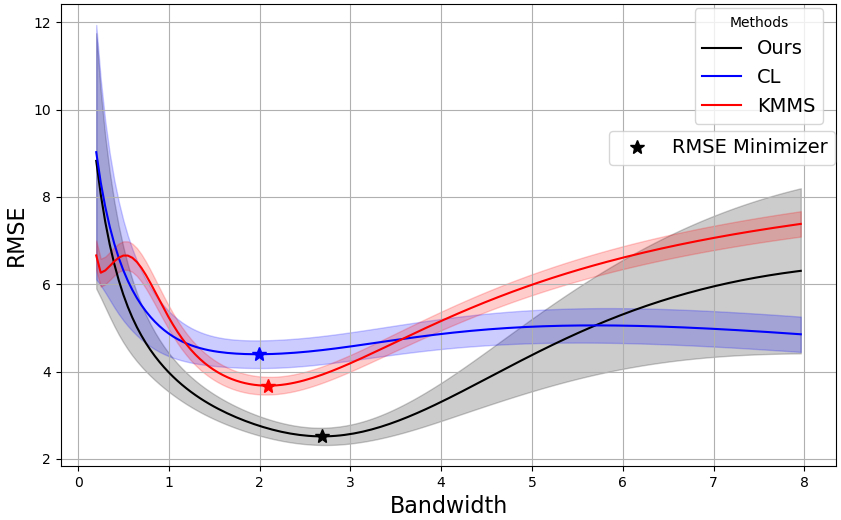

Figure 5: RMSE of TR, CL, and KMMS estimators across bandwidths in the ten-dimensional simulation (IPW omitted). As in one dimension, TR achieves the lowest minimal RMSE and shows greater robustness to bandwidth choice.

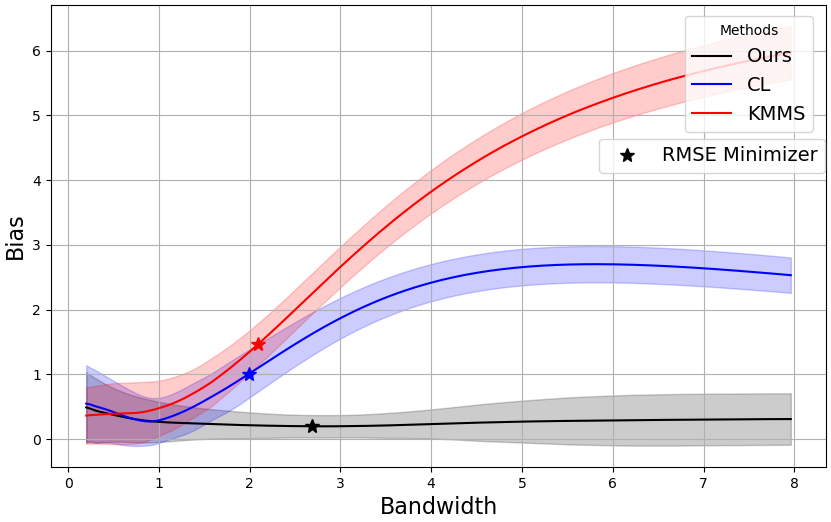

Figure 6: Bias of TR, CL, and KMMS estimators across bandwidths in the ten-dimensional simulation (IPW omitted). TR maintains comparatively small bias over a wider range of bandwidths, supporting its favorable bias–variance trade-off.

Across these designs, TR achieves the lowest RMSE and bias at the RMSE-minimizing bandwidth and also demonstrates greater robustness to bandwidth selection: its RMSE curves are flatter in a neighborhood around the optimum, and its bias increases more slowly as the bandwidth grows. This allows for a wider range of effective bandwidths and suggests that TR can be tuned less aggressively without sacrificing accuracy, potentially reducing the computational burden of bandwidth search in practice.

Furthermore, the theory in Section 4 implies that IPW, CL, and KMMS share the same generic h^2 smoothing-bias barrier, whereas TR replaces that leading term by second-order nuisance error. The simulations are consistent with this prediction while highlighting that TR attains smaller finite-sample bias, which in turn permits the use of larger bandwidths without a substantial increase in bias and yields a more favorable bias–variance trade-off.

6.2.4 No-overlap scenario

To further evaluate the robustness of our TR estimator, we examine its performance in scenarios with poor overlap between the behavior policy and the evaluation policy. This "No-Overlap Case" represents a challenging setting where all the estimators perform poorly. To illustrate the performance of various estimators in this challenging scenario, we present the results in Table 1.

As evident from Table 1, all estimators struggle in this setting, exhibiting high RMSE and bias due to the lack of overlap. Nevertheless, TR still achieves the lowest RMSE and bias, albeit by a small margin, and does so at a relatively large bandwidth. This is consistent with our earlier observations that TR can safely employ more smoothing, even in challenging designs. These results underscore both the fundamental importance of overlap and the incremental robustness gains that TR can offer when overlap is imperfect.

We include in Section A.5.6 of the Appendix a comprehensive analysis of estimator performance under poor coverage conditions, including detailed RMSE and bias comparisons across different bandwidths.

Method	RMSE	Bias	Optimal Bandwidth
Ours	**980.758**	**939.354**	**7.945**
Colangelo	980.774	939.366	4.621
IPW	1238.831	1206.441	**7.945**
KMMS	1084.013	978.716	4.417

Table 1: Comparison of RMSE, bias, and RMSE-minimizing bandwidths for different estimators in the no-overlap simulation scenario. All methods exhibit large errors, but TR attains the smallest RMSE and bias.

6.3 E-commerce pricing application

We next study a semi-synthetic pricing problem based on the JD.com E-Commerce Data Challenge dataset Shen et al. [2020], which contains transaction-level logs from a major Chinese platform. For each user–SKU interaction we construct a triplet (X, A, Y), where A is the displayed final unit price, Y is the realized revenue per interaction (which is zero when no purchase occurs), and X collects user and contextual features. Our goal is to estimate the expected revenue under a pricing policy θ,

$$V(\theta) = \mathbb{E}\big[Y \mid A = \theta(X), X\big],$$

interpreted as the average revenue that would be obtained if future prices followed θ under standard causal assumptions.

6.3.1 Experimental design

We evaluate the TR estimator on the JD.com E-Commerce Data Challenge dataset Shen et al. [2020] for March 2018, which provides transaction-level logs and associated user information. The data include orders (with anonymized user and SKU identifiers, timestamps, and both original and final unit prices), click interactions (timestamps of user–SKU page views), and a rich set of user-profile attributes such as membership level, tenure, subscription status, demographic variables, and purchase power. Combining orders with non-converting clicks and enriching them with user attributes enables us to approximate price–demand relationships beyond purchased baskets, which is crucial for off-policy evaluation of pricing.

6.3.2 Preprocessing and feature construction

We restrict attention to a focal SKU with sufficient price variability and volume to ensure overlap and stable estimation. For this SKU, we construct per-interaction examples by joining clicks and orders and define the treatment A as the displayed final unit price, the outcome Y as realized revenue per interaction (price times quantity, which may be zero if no purchase occurs), and the context X as the collection of user-profile attributes, including user level, tenure, subscription status, demographic covariates, purchase power, and city-level indicators. We remove extreme outliers in price and winsorize continuous covariates to mitigate the influence of rare extreme values.

6.3.3 Treatment, outcome, and policies

Non-purchase data generation A central challenge in real-world pricing applications is that we only observe purchases, not the full demand response at all possible prices. To address this limitation, we leverage the click data to construct a more comprehensive dataset that includes non-purchase interactions. Specifically, for each user–SKU interaction in the click data where no

purchase occurred, we create a non-purchase observation with $Y = 0$, aggregating multiple clicks within a day into a single non-purchase event.

Price structure The selected SKU exhibits favorable properties for pricing analysis: while the `original_unit_price` remains constant throughout the observation period, the `final_unit_price` varies across transactions due to discounts and promotions. This price variation provides the necessary treatment heterogeneity for reliable off-policy evaluation while maintaining a stable baseline reference price.

Treatment and outcome synthesis For non-purchase observations, we need to impute both the price shown to the user (treatment) and the resulting revenue (outcome). We employ a two-step approach:

1. **Price Imputation:** We first estimate what price each non-purchasing user would likely have been shown using ordinary least squares (OLS) regression. We model the relationship between customer features and the observed `final_unit_price` in the purchase data:

$$\text{final\_unit\_price}_i = \beta_0 + \sum_j \beta_j \cdot \text{feature}_{ij} + \epsilon_i$$

   For all purchase observations in our semi-synthetic approach, we assume that all prices have at least 5% variance of their intrinsic variance as Gaussian noise. Thus, the imputed price is:

$$A_{\text{purchase}} = 1.05 \times \widehat{\text{final\_unit\_price}}_{\text{OLS}} + \mathcal{N}(0, 0.05\sigma^2_{\text{price}})$$

   where σ^2_{price} is the intrinsic price variance.

2. **Revenue outcome:** For non-purchase observations, the revenue outcome is naturally zero, as no transaction occurred:

$$Y_{\text{non-purchase}} = 0$$

   During training, we treat the revenue outcome estimation as a multi-class classification problem. This approach is as effective as directly modeling the ground-truth outcome function $m(a, x)$ in this experiment, while more naturally capturing the discrete nature of purchase outcomes and leveraging the rich structure of user features.

This construction allows us to augment the purchase-only data with realistic non-purchase observations, creating a more complete picture of the price–demand relationship.

Behavior policy estimation Using the synthesized dataset combining both purchase and non-purchase observations, we estimate the behavior policy $\hat{\theta}_{\text{behavior}}(X)$ that generated the observed pricing decisions. We train a neural network to model the behavior policy as a Gaussian distribution, where the network outputs both the mean and variance parameters:

$$\hat{\theta}_{\text{behavior}}(X) \sim \mathcal{N}(\mu_{\text{NN}}(X), \sigma^2_{\text{NN}}(X))$$

This parameterization captures the platform's historical pricing strategy as a function of user characteristics, while accounting for the stochasticity in pricing decisions.

Evaluation policy generation To create a meaningful evaluation policy for our off-policy estimation task, we employ a reinforcement learning approach following the Adaptive KL Penalty Coefficient method from Proximal Policy Optimization Schulman et al. [2017]. The learned evaluation policy is optimized during training to guarantee a 2% increase in expected revenue compared to the behavior policy, while maintaining robust overlap with the behavior policy. In particular, it is constrained to preserve a high effective sample size (ESS) relative to the behavior policy, ensuring that our off-policy evaluation estimates are not dominated by a few observations with extreme importance weights.

6.3.4 Results

Choice of sample size To assess how estimator performance scales with data availability, we conduct 10 repeated experiments across sample sizes $n \in \{10{,}000, 50{,}000, 100{,}000, 200{,}000\}$. This design allows us to compare methods in both moderate and large-sample regimes that mirror realistic transaction volumes in modern online platforms, and to evaluate which estimators most effectively leverage additional data to improve accuracy.

Overlap and policy quality Across all repeated runs, the estimated behavior policy achieves effective sample size (ESS) very close to n, indicating minimal weight degeneracy and excellent overlap between the behavior and evaluation policies. This high ESS demonstrates that our evaluation policy maintains strong support overlap with the historical pricing policy, so that off-policy estimates are not dominated by a few observations with extreme importance weights.

Sample size	TR (Ours)	CL	KMMS	IPW
10k	5.670	**2.293**	21.949	13.524
50k	**2.598**	2.834	22.188	9.325
100k	**2.436**	2.804	22.259	9.823
200k	**2.091**	2.352	22.129	10.279

Table 2: Best RMSE of policy value estimates by estimator across sample sizes on the focal SKU. Lower is better.

At $n = 10{,}000$, CL attains a slightly lower RMSE than TR, consistent with the small-sample behavior observed in our simulations. From $n = 50{,}000$ onward, TR surpasses CL and maintains a widening advantage as n grows, while KMMS and IPW remain far less accurate across all sample sizes. This pattern aligns with TR's improved bias control and efficiency at moderate to large n.[2]

7 Conclusion and Future Directions

This paper studies continuous-treatment policy evaluation through a triply robust estimator with curvature-aware debiasing. In addition to the usual outcome-regression and generalized propensity-score corrections, the estimator incorporates a third nuisance component tied to treatment curvature. In pricing applications, this component is naturally elasticity-aware: it depends on the same local price-sensitivity objects that govern marginal revenue and cross-price substitution. More generally, the estimator shifts the leading kernel bias away from the curvature of the target regression itself and toward the curvature of the nuisance-error process.

[2]The IPW and KMMS estimators consistently show substantially higher RMSE and bias compared to TR and CL across all bandwidth values in this real-data setting, hence their exclusion from the main figures. Full per-estimator results are provided in the Appendix.

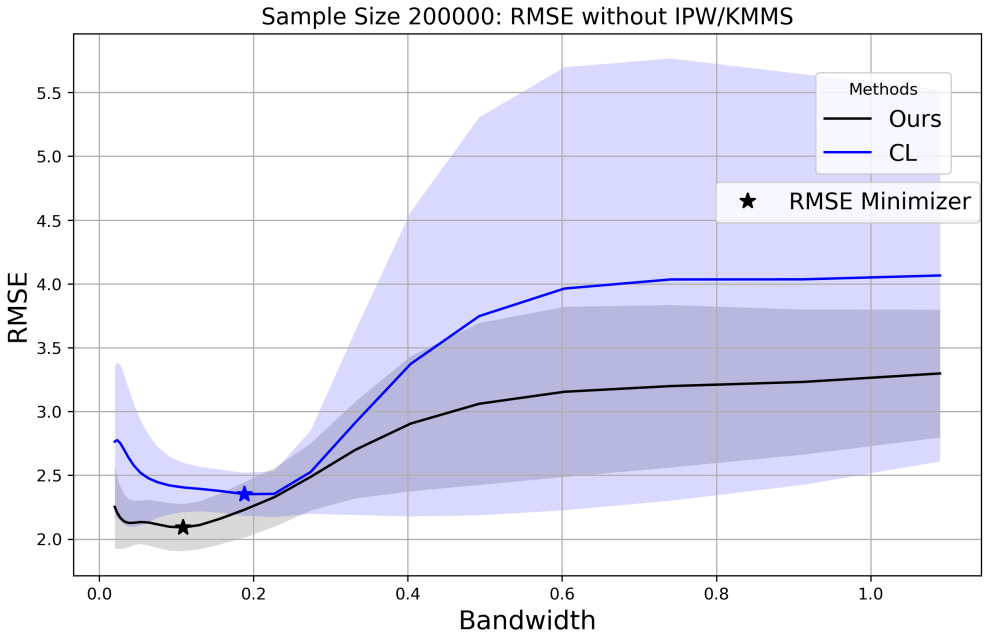

Figure 7: RMSE of TR and CL policy value estimators across bandwidths in the real-data pricing experiment at n=200k for the focal SKU (KMMS and IPW omitted due to much larger errors). TR achieves a lower minimal RMSE and is less sensitive to bandwidth choice than CL.

Our theory delivers three main implications. First, the canonical $n^{-4/5}$ rate is not intrinsic to one-dimensional continuous-treatment evaluation. When the second-order nuisance error is small, the mean-squared error improves to $o(n^{-4/5})$. Second, the relevant regularity requirement is not a blanket global Hölder condition on the population regression. Rather, the rate improvement is driven by higher-order stochastic equicontinuity of the estimated derivative process in the treatment direction. Third, valid asymptotic inference requires more than identification and kernel smoothness: it also depends on out-of-fold nuisance evaluation and sufficiently strong regularity conditions on the first-stage estimators.

These results suggest a broader lesson for continuous-treatment econometrics. The main obstacle is not kernel smoothing per se, but the way smoothing bias interacts with nuisance estimation. Once treatment curvature is modeled directly, debiasing can become materially sharper, and the conventional nonparametric rate barrier need not be binding.

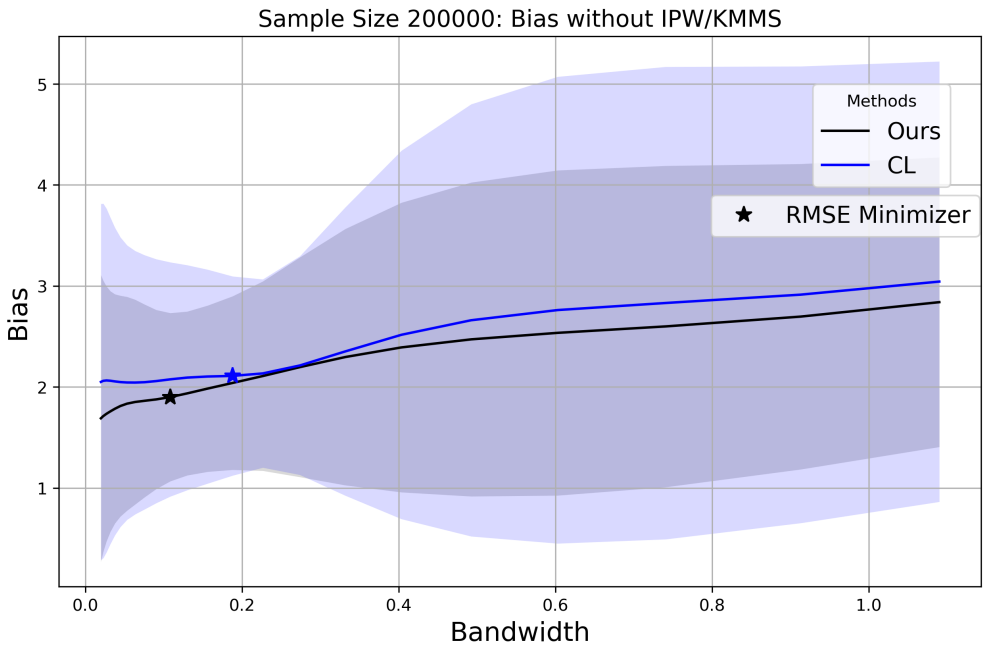

Figure 8: Bias of TR and CL policy value estimators across bandwidths in the real-data pricing experiment at n=200k for the focal SKU. TR attains comparable or smaller bias near the RMSE-minimizing bandwidth, indicating that its RMSE gains are not driven solely by variance reduction.

A Appendix

A.1 Notation dictionary

For convenience, we collect the main objects used throughout the paper.

- $m_0(a,x) = \mathbb{E}[Y \mid A = a, X = x]$: conditional mean reward.
- $f_0(a \mid x)$: conditional treatment density (generalized propensity score).
- $\Delta(a,x) = \hat{m}(a,x) - m_0(a,x)$ and $\delta(a,x) = 1 - f_0(a \mid x)/\hat{f}(a \mid x)$: nuisance estimation errors.
- $g(a,x) = \Delta(a,x)(1-\delta(a,x))$: nuisance-error product that determines the leading kernel bias.
- $V(\theta) = \mathbb{E}[m_0(\theta(X), X)]$: policy value.
- K, h, κ, and $R(K)$: kernel, bandwidth, second kernel moment, and squared-kernel norm.

A.2 Proofs

A.2.1 Distributional Taylor expansion

Proof of Theorem 3.1. Write

$$R_2(\bar{P}, P) = \Psi(\bar{P}) - \Psi(P) - \int \varphi_h(z;\bar{P})\, d(\bar{P} - P)(z).$$

Using that $\int \varphi_h(z;\bar{P})\, d\bar{P}(z) = 0$,

$$\begin{aligned} R_2(\bar{P}, P) &= \Psi(\bar{P}) - \Psi(P) + \int \varphi_h(z;\bar{P})\, dP(z) \\ &= \int \bar{m}(\theta(x), x)\, d\bar{P}(x) - \int m(\theta(x), x)\, dP(x) + \int \varphi_h(z;\bar{P})\, dP(z). \end{aligned} \tag{A.1}$$

Now

$$\begin{aligned} \int \varphi_h(z;\bar{P})\, dP(z) = \int \frac{f(a \mid x)}{\bar{f}(a \mid x)} \frac{1}{h} K\left(\frac{a - \theta(x)}{h}\right) \big(m(a,x) - \bar{m}(a,x)\big)\, da\, dP(x) \\ + \int \bar{m}(\theta(x), x)\, dP(x) - \int \bar{m}(\theta(x), x)\, d\bar{P}(x). \end{aligned} \tag{A.2}$$

Substituting (A.2) into (A.1) and collecting terms gives

$$\begin{aligned} R_2(\bar{P}, P) = \int \Bigg[\bar{\Delta}(\theta(x), x)\bar{\delta}(\theta(x), x) + \Big(1 - \bar{\delta}(\theta(x), x)\Big)\bar{\Delta}(\theta(x), x) \\ - \int \frac{1}{h} K\left(\frac{a - \theta(x)}{h}\right) \Big(1 - \bar{\delta}(a, x)\Big)\bar{\Delta}(a, x)\, da \Bigg] dP(x), \end{aligned}$$

which is the stated expression. □

A.2.2 Analysis of bias

Proof of the bias expansion in Theorem 4.3. Condition on the auxiliary sample used to estimate the nuisance functions. Then

$$\mathbb{E}\big[\hat{V}^{\mathrm{TR}}(\theta)\big] - V(\theta) = \mathbb{E}\big[\hat{m}(\theta(X), X) - m_0(\theta(X), X)\big] + \mathbb{E}\left[\frac{\frac{1}{h}K\left(\frac{A-\theta(X)}{h}\right)}{\hat{f}(A \mid X)}\{m_0(A, X) - \hat{m}(A, X)\}\right]$$
$$= \mathbb{E}\big[\Delta(\theta(X), X)\big] - \mathbb{E}\left[\frac{\frac{1}{h}K\left(\frac{A-\theta(X)}{h}\right)}{\hat{f}(A \mid X)}\Delta(A, X)\right].$$

Introduce $g(a, x) = \Delta(a, x)(1 - \delta(a, x)) = \Delta(a, x)f_0(a \mid x)/\hat{f}(a \mid x)$. Then

$$\mathbb{E}\big[\hat{V}^{\mathrm{TR}}(\theta)\big] - V(\theta) = \mathbb{E}\big[\Delta(\theta(X), X) - g(\theta(X), X)\big] + \mathbb{E}\left[g(\theta(X), X) - \int \frac{1}{h}K\left(\frac{a-\theta(X)}{h}\right) g(a, X)\, da\right]$$
$$= B_{0,n}(\theta) + \mathbb{E}\left[g(\theta(X), X) - \int \frac{1}{h}K\left(\frac{a-\theta(X)}{h}\right) g(a, X)\, da\right].$$

For each fixed x, Taylor's theorem with integral remainder gives

$$\int \frac{1}{h}K\left(\frac{a-t}{h}\right) g(a, x)\, da = g(t, x) + \frac{\kappa h^2}{2}\partial_a^2 g(t, x) + R_h(t, x),$$

where

$$|R_h(t, x)| \le \frac{\mu_3(K)}{6}h^3 \sup_{|u|\le h}\big|\partial_a^3 g(t + u, x)\big|.$$

Setting $t = \theta(X)$ and taking expectations yields

$$\mathbb{E}\big[\hat{V}^{\mathrm{TR}}(\theta)\big] - V(\theta) = B_{0,n}(\theta) - \frac{\kappa h^2}{2}B_{2,n}(\theta) + O(h^3),$$

which is the claimed expansion. The displayed remainder bound also proves Corollary 4.3A. □

A.2.3 Analysis of variance

Proof of the variance expansion in Theorem 4.3. Again condition on the auxiliary sample and write

$$\psi_i := \hat{m}(\theta(X_i), X_i) + \frac{\frac{1}{h}K\left(\frac{A_i-\theta(X_i)}{h}\right)}{\hat{f}(A_i \mid X_i)}\{Y_i - \hat{m}(A_i, X_i)\}, \qquad \hat{V}^{\mathrm{TR}}(\theta) = \frac{1}{n}\sum_{i=1}^{n}\psi_i.$$

The plug-in term $\hat{m}(\theta(X_i), X_i)$ has variance of order $O(1)$ and therefore contributes only $O(n^{-1})$ to the variance of the average. The leading contribution comes from the kernel-weighted residual

term. Using the law of iterated expectation,

$$\begin{aligned}
&\mathbb{E}\left[\left(\frac{\frac{1}{h}K\left(\frac{A-\theta(X)}{h}\right)}{\hat{f}(A \mid X)}\{Y-\hat{m}(A,X)\}\right)^2\right] \\
&\quad= \mathbb{E}_X\left[\int \frac{1}{h^2}K\left(\frac{a-\theta(X)}{h}\right)^2 \frac{f_0(a \mid X)}{\hat{f}(a \mid X)^2}\mathbb{E}\big[(Y-\hat{m}(a,X))^2 \mid A=a, X\big]da\right] \\
&\quad= \frac{1}{h}\mathbb{E}_X\left[\int K(u)^2 \frac{f_0(\theta(X)+hu \mid X)}{\hat{f}(\theta(X)+hu \mid X)^2}\mathbb{E}\big[(Y-\hat{m}(\theta(X)+hu,X))^2 \mid A=\theta(X)+hu, X\big]du\right] \\
&\quad= \frac{R(K)}{h}\Gamma_n(\theta)+O(1).
\end{aligned}$$

Since the ψ_i are conditionally independent across observations under cross-fitting,

$$\operatorname{Var}\big(\hat{V}^{\mathrm{TR}}(\theta)\big) = \frac{1}{n}\operatorname{Var}(\psi_1) = \frac{R(K)}{nh}\Gamma_n(\theta)+O\left(\frac{1}{n}\right),$$

which proves the theorem. □

A.2.4 Asymptotic normality

Proof of Theorem 4.4. Condition on the auxiliary sample used to estimate the nuisances. Under cross-fitting, the summands ψ_i are conditionally independent, so it is enough to verify Lyapunov's condition for the triangular array

$$\xi_{ni} := \psi_i - \mathbb{E}[\psi_i].$$

From Theorem 4.3,

$$s_n^2 := \sum_{i=1}^n \operatorname{Var}(\xi_{ni}) = n\operatorname{Var}(\psi_1) = \frac{n}{h}C_n(\theta)\{1+o(1)\}.$$

It remains to control the conditional third absolute moments. Because $\hat{f}$ is bounded away from zero with probability approaching one, K is bounded, and $\sup_{a,x}\mathbb{E}[|Y|^3 \mid A=a, X=x] < \infty$,

$$\begin{aligned}
\mathbb{E}\big[|\xi_{ni}|^3\big] &\lesssim 1+\mathbb{E}\left[\left|\frac{1}{h}K\left(\frac{A-\theta(X)}{h}\right)\frac{Y-\hat{m}(A,X)}{\hat{f}(A \mid X)}\right|^3\right] \\
&\lesssim 1+\mathbb{E}_X\left[\int \frac{1}{h^3}\left|K\left(\frac{a-\theta(X)}{h}\right)\right|^3 f_0(a \mid X)\,da\right] \\
&\lesssim 1+h^{-2}\int |K(u)|^3 du \lesssim h^{-2}.
\end{aligned}$$

Therefore

$$\frac{1}{s_n^3}\sum_{i=1}^n \mathbb{E}\big[|\xi_{ni}|^3\big] \lesssim \frac{nh^{-2}}{(nh^{-1})^{3/2}} = (nh)^{-1/2} \longrightarrow 0.$$

Lyapunov's central limit theorem yields

$$\frac{\hat{V}^{\mathrm{TR}}(\theta)-\mathbb{E}[\hat{V}^{\mathrm{TR}}(\theta)]}{\sqrt{\operatorname{Var}(\hat{V}^{\mathrm{TR}}(\theta))}} \xrightarrow{d} N(0,1).$$

Substituting the bias and variance expansions from Theorem 4.3 gives

$$\sqrt{\frac{nh}{C_n(\theta)}}\left\{\hat{V}^{\mathrm{TR}}(\theta) - V(\theta) - b_n(\theta)\right\} \xrightarrow{d} N(0,1),$$

which is the claim. □

A.2.5 Multidimensional treatment extension

Proof of Theorem 5.1. Let $K_d(u) = \prod_{j=1}^{d} K(u_j)$ and define the multidimensional kernel-weighted residual summand by

$$\psi_{i,d} := \hat{m}(\theta(X_i), X_i) + \frac{K_{d,h}(A_i - \theta(X_i))}{\hat{f}(A_i \mid X_i)}\{Y_i - \hat{m}(A_i, X_i)\}.$$

The proof is the same as in one dimension, with two changes.

First, for the bias term we use the multivariate Taylor expansion of $g(a, x)$ around $a = \theta(x)$:

$$\int K_d(u) g(\theta(x) + hu, x)\, du = g(\theta(x), x) + \frac{\kappa h^2}{2} \sum_{j=1}^{d} \partial^2_{a_j a_j} g(\theta(x), x) + O(h^3).$$

The mixed second derivatives disappear because K_d is a product of symmetric kernels with zero first moments.

Second, for the variance term,

$$\begin{aligned}
&\mathbb{E}\left[\left(\frac{K_{d,h}(A - \theta(X))}{\hat{f}(A \mid X)}\{Y - \hat{m}(A, X)\}\right)^2\right] \\
&\quad = \frac{1}{h^d}\mathbb{E}_X\left[\int K_d(u)^2 \frac{f_0(\theta(X) + hu \mid X)}{\hat{f}(\theta(X) + hu \mid X)^2}\mathbb{E}\big[(Y - \hat{m}(\theta(X) + hu, X))^2 \mid A = \theta(X) + hu, X\big] du\right] \\
&\quad = \frac{R(K)^d}{h^d}\Gamma_{n,d}(\theta) + o(h^{-d}).
\end{aligned}$$

Hence

$$\mathrm{Var}\big(\hat{V}_d^{\mathrm{TR}}(\theta)\big) = \frac{R(K)^d}{nh^d}\Gamma_{n,d}(\theta) + o\big((nh^d)^{-1}\big).$$

The same third-moment calculation gives

$$\sum_{i=1}^{n} \mathbb{E}\big[|\psi_{i,d} - \mathbb{E}\psi_{i,d}|^3\big] \lesssim nh^{-2d}, \qquad s^2_{n,d} \asymp nh^{-d},$$

so the Lyapunov ratio is $O((nh^d)^{-1/2}) \to 0$. The conditional CLT follows. □

A.3 Expected Regret in Policy Learning

Proof of Theorem 5.2: Define the policy-value error process

$$e_n(\theta) := \hat{V}^{\mathrm{TR}}(\theta) - V(\theta).$$

Because $\hat{\theta} \in \arg\max_{\theta\in\Theta} \hat{V}^{\mathrm{TR}}(\theta)$ and $\theta^* \in \arg\max_{\theta\in\Theta} V(\theta)$, we have the deterministic bound

$$\begin{aligned} V(\theta^*) - V(\hat{\theta}) &= V(\theta^*) - \hat{V}^{\mathrm{TR}}(\theta^*) + \hat{V}^{\mathrm{TR}}(\theta^*) - \hat{V}^{\mathrm{TR}}(\hat{\theta}) + \hat{V}^{\mathrm{TR}}(\hat{\theta}) - V(\hat{\theta}) \\ &\le |e_n(\theta^*)| + \left|e_n(\hat{\theta})\right| \\ &\le 2\sup_{\theta\in\Theta} |e_n(\theta)|. \end{aligned}$$

Taking expectations and writing

$$e_n(\theta) = \mu_n(\theta) + Z_n(\theta), \qquad \mu_n(\theta) := \mathbb{E}[e_n(\theta)], \qquad Z_n(\theta) := e_n(\theta) - \mathbb{E}[e_n(\theta)],$$

yields

$$\begin{aligned} \mathbb{E}\left[V(\theta^*) - V(\hat{\theta})\right] &\le 2\sup_{\theta\in\Theta} |\mu_n(\theta)| + 2\mathbb{E}\left[\max_{\theta\in\Theta} |Z_n(\theta)|\right] \\ &\le 2b_n + 2\mathbb{E}\left[\max_{\theta\in\Theta} |Z_n(\theta)|\right]. \end{aligned}$$

For any $\lambda > 0$,

$$\begin{aligned} \mathbb{E}\left[\max_{\theta\in\Theta} |Z_n(\theta)|\right] &\le \frac{1}{\lambda}\log\left(\sum_{\theta\in\Theta} \mathbb{E}\left[e^{\lambda|Z_n(\theta)|}\right]\right) \\ &\le \frac{1}{\lambda}\log\left(\sum_{\theta\in\Theta} \left\{\mathbb{E}\left[e^{\lambda Z_n(\theta)}\right] + \mathbb{E}\left[e^{-\lambda Z_n(\theta)}\right]\right\}\right) \\ &\le \frac{1}{\lambda}\log\left(2|\Theta| e^{\lambda^2\sigma_n^2/2}\right) \\ &= \frac{\log(2|\Theta|)}{\lambda} + \frac{\lambda\sigma_n^2}{2}. \end{aligned}$$

Optimizing the last display at

$$\lambda = \frac{\sqrt{2\log(2|\Theta|)}}{\sigma_n}$$

shows that

$$\mathbb{E}\left[\max_{\theta\in\Theta} |Z_n(\theta)|\right] \le \sigma_n\sqrt{2\log(2|\Theta|)}.$$

Therefore,

$$\mathbb{E}\left[V(\theta^*) - V(\hat{\theta})\right] \le 2b_n + 2\sigma_n\sqrt{2\log(2|\Theta|)}.$$

The stated rate follows by substituting $b_n = O(h^2)$ and $\sigma_n = O((nh)^{-1/2})$, and then taking $h = \Theta(n^{-1/5})$.

□

A.4 Algorithmic Details

A.4.1 Implementation Details of Estimators

In this subsection, we provide detailed implementation information for all the estimators benchmarked in our paper. This includes the original KMMS, our improved version of KMMS (KMMS-I), as well as the other estimators we compared against.

Inverse Probability Weighting (IPW) The IPW estimator, as described in Kallus and Zhou [2018], is implemented as follows:

Algorithm 1 Inverse Probability Weighting (IPW) Estimator

Input: Data $\{(X_i, A_i, Y_i)\}_{i=1}^n$, target policy θ, bandwidth h, kernel function $K(\cdot)$
Output: IPW estimate $\hat{V}^{\text{IPW}}$
procedure IPW($\{(X_i, A_i, Y_i)\}_{i=1}^n$, θ, h, $K(\cdot)$)
    Estimate propensity score $\hat{f}(A|X)$
    for $i = 1$ **to** n **do**
        Compute kernel weights $K_h(\theta(X_i), A_i) = \frac{1}{h}K(\frac{\theta(X_i)-A_i}{h})$
    end for
    $\hat{V}^{\text{IPW}}(\theta) = \frac{1}{n}\sum_{i=1}^{n} \frac{K_h(\theta(X_i),A_i)}{\hat{f}(A_i|X_i)} Y_i$
end procedure

Double Debiased Machine Learning (CL) The CL estimator, as described in Colangelo and Lee [2020], is implemented as follows:

Algorithm 2 Double Machine Learning (CL) Estimator

Input: Data $\{(X_i, A_i, Y_i)\}_{i=1}^n$, target policy θ, bandwidth h, kernel function $K(\cdot)$
Output: CL estimate $\hat{V}^{\text{CL}}$
procedure CL($\{(X_i, A_i, Y_i)\}_{i=1}^n$, θ, h, $K(\cdot)$)
    Estimate propensity score $\hat{f}(A|X)$
    Estimate outcome regression $\hat{m}(A, X)$
    for $i = 1$ **to** n **do**
        Compute kernel weights $K_h(\theta(X_i), A_i) = \frac{1}{h}K(\frac{\theta(X_i)-A_i}{h})$
    end for
    $\hat{V}^{\text{CL}} = \frac{1}{n}\sum_{i=1}^{n} \hat{m}(\theta(X_i), X_i) + \frac{K_h(\theta(X_i),A_i)}{\hat{f}(\theta(X_i)|X_i)}(Y_i - \hat{m}(\theta(X_i), X_i))$
end procedure

KMMS We have reimplemented the local linear kernel estimator with several improvements to enhance computational efficiency. Our implementation leverages vectorized operations, optimized matrix computations, and GPU acceleration to significantly reduce runtime, especially for large datasets. The use of GPU parallelization allows for faster processing of matrix operations and kernel computations, further improving the scalability of the algorithm for high-dimensional data and large sample sizes.

Algorithm 3 Improved Local Linear Kernel Estimator (KMMS-I)

Input: Data $\{(X_i, A_i, Y_i)\}_{i=1}^n$, target policy θ, bandwidth h, kernel function $K(\cdot)$
Output: Improved KMMS estimate $\hat{V}^{\text{KMMS}}_{\text{Improved}}$
procedure ImprovedKMMS($\{(X_i, A_i, Y_i)\}_{i=1}^n$, θ, h, $K(\cdot)$)
    Estimate propensity score $\hat{f}(A|X)$
    Estimate outcome regression $\hat{m}_0(A, X)$
    Estimate marginal treatment distribution $\hat{\varpi}(A)$
    Estimate marginal outcome regression $\hat{\mu}(A)$
    for $i = 1$ **to** n **do**
        Compute the pseudo outcome $\xi_i = \frac{Y_i - \hat{m}_0(A_i, X_i)}{\hat{f}(A_i|X_i)} \hat{\varpi}(A_i) + \hat{\mu}(A_i)$
        Kernel weight $K_h(\theta(X_i), A_i) = \frac{K(\frac{\theta(X_i) - A_i}{h})}{h}$
        Feature $g_i(\theta(X_i), A_i) = (1, \frac{\theta(X_i) - A_i}{h})^T$
    end for
    Solve the weighted least squares problem:
    $\hat{\beta} = \arg\min_\beta \sum_{i=1}^n K_h(\theta(X_i), A_i)(\xi_i - g_i(\theta(X_i), A_i)^T \beta)^2$
    Compute the improved KMMS estimate: $\hat{V}^{\text{KMMS}}_{\text{Improved}} = \frac{1}{n} \sum_{i=1}^n g_i(\theta(X_i), A_i)^T \hat{\beta}$
end procedure

Triply Robust (TR) Our proposed TR estimator is implemented as follows:

Algorithm 4 Triply Robust (TR) Estimator

Input: Data $\{(X_i, A_i, Y_i)\}_{i=1}^n$, target policy θ, bandwidth h, kernel function $K(\cdot)$
Output: TR estimate $\hat{V}^{\text{TR}}$
procedure TR($\{(X_i, A_i, Y_i)\}_{i=1}^n$, θ, h, $K(\cdot)$)
    Estimate propensity score $\hat{f}(A|X)$
    Estimate outcome regression $\hat{m}(A, X)$
    for $i = 1$ **to** n **do**
        Compute kernel weights $K_h(\theta(X_i), A_i) = \frac{1}{h} K(\frac{\theta(X_i) - A_i}{h})$
    end for
    $\hat{V}^{\text{TR}} = \frac{1}{n} \sum_{i=1}^n \hat{m}(\theta(X_i), X_i) + \frac{K_h(\theta(X_i), A_i)}{\hat{f}(A_i|X_i)} (Y_i - \hat{m}(A_i, X_i))$
end procedure

These implementations offer several advantages:

- Reduced computational complexity: By using separate estimators for nuisance parameters, we can save the estimated values for bootstrap inference.
- Enhanced flexibility: Our approach allows for more flexible estimation of nuisance parameters. This flexibility is particularly beneficial when dealing with complex or high-dimensional covariate spaces.
- Adaptability to real-world data: The use of non-parametric estimators for nuisance parameters allows for more adaptive modeling of these components, better capturing the nuances and complexities often present in real-life datasets.

In our empirical experiments, we maintained the statistical properties of the original estimators to ensure a fair comparison. The improvements we implemented were primarily focused on enhancing computational efficiency, particularly for large-scale applications.

A.5 Detailed Synthetic Settings and Results

A.5.1 Detailed Dimensionality Settings

Single-dimensional case In the single-dimensional scenario, the latent variable Z_i is drawn from a uniform distribution $Z_i \sim \text{Uniform}\left(\frac{\pi}{2}, \frac{7\pi}{2}\right)$, and the context variable is set as $X_i = Z_i$. The behavior policy is defined as $\theta_1(X) = 2X + \text{Normal}(0, 4)$, and the evaluation policy as $\theta_2(X) = 1.5X + \pi$. The outcome is modeled as $Y_i = m_0(Z_i, A_i) + \epsilon$, where the noise term $\epsilon \sim \text{Normal}(0, 1)$. The outcome function for this setting is given by:

$$m_0(Z, A) = 4\sin(Z) - 4\sin(A) + (A - 4\pi)^2.$$

Ten-dimensional case In the ten-dimensional scenario, the latent variable Z_i is drawn from a ten-dimensional uniform distribution, $Z_i \sim \text{Uniform}\left(\frac{7\pi}{2}, \frac{13\pi}{2}\right)^{10}$, and the context variable is set as $X_i = Z_i$. The behavior policy in this case is defined as $\theta_1(X_j) = 2\beta_j X_j + \text{Normal}(0, 2)$, and the evaluation policy as $\theta_2(X_j) = 1.5\beta_j X_j + \frac{5}{2}\pi$, where the coefficient vector $\beta = [0.05, 0.06, 0.07, 0.08, 0.09, 0.1, 0.11, 0.13, 0.14, 0.15]$. The outcome is again modeled as $Y_i = m_0(Z_i, A_i) + \epsilon$, with noise $\epsilon \sim \text{Normal}(0, 1)$. The outcome function in this setting is:

$$m_0(Z, A) = \frac{2}{5}\sum_{j=1}^{10} \sin(Z_j) - 4\sin(A_j) + \frac{1}{2}(A_j - 10\pi)^2.$$

A.5.2 Detailed Implementation of Estimators

Cross-Validation: We implement the Random Forest Breiman [2001] using the scikit-learn library in Python for estimating both $\hat{m}$ and $\hat{f}$. To ensure optimal performance, we employ k-fold cross-validation (with k=5) to tune the hyperparameters of the Random Forest estimator for both functions.

Estimating $\hat{m}$ In our experiments, we employ advanced machine learning techniques to estimate the response function $\hat{m}$. Specifically, we utilize the Random Forest algorithm through the scikit-learn library in Python. For our estimation of $\hat{m}$, we use the Root Mean Square Error (RMSE) as our primary evaluation metric.

Estimating $\hat{f}$ To estimate the propensity score function $\hat{f}$, we employ a two-step approach combining Random Forest regression and Gaussian Kernel density estimation. This method allows us to capture both the structured relationship between context and action, and the residual variability.

- **Step 1: Estimating Behavior Policy**
  - We use Random Forest regression to estimate the behavior policy $\hat{\theta}(X)$.
  - Similar to our approach for estimating $\hat{m}$, we utilize k-fold cross-validation (k=5) to tune the hyperparameters of the Random Forest estimator provided by scikit-learn.
- **Step 2: Residual Density Estimation**

- After obtaining $\hat{\theta}(X)$, we estimate the residual density of $A_i - \hat{\theta}(X_i)$ using a Gaussian Kernel.
- The bandwidth of the Gaussian Kernel is tuned via cross-validation.

We implement this two-step process using scikit-learn for the Random Forest regression and scipy for the Gaussian Kernel density estimation. The mean squared error (MSE) is used as the evaluation metric for cross-validation in the Random Forest step, while the log-likelihood is used for tuning the Gaussian Kernel bandwidth.

A.5.3 Hyperparameter Tuning and Nuisance Parameter Estimation

In our experiments, we employed Random Forests for estimating the nuisance parameters $\hat{m}$ and $\hat{f}$. The hyperparameters for these models were tuned using grid search with cross-validation to ensure optimal performance. For $\hat{m}$, we used a Random Forest regressor with the following hyperparameters tuned:

- `n_estimators`: {100, 200, 300}.
- `max_depth`: {3, 5, 7}.
- `min_samples_split`: {5, 10, 15, 20}.
- `min_samples_leaf`: {2, 4, 6, 8, 10}.

For $\hat{f}$, we employed a two-step approach combining Random Forest regression and Gaussian Kernel density estimation:

1. We first used a Random Forest regressor to estimate the behavior policy $\hat{\theta}(X)$, with the following hyperparameter grid:
   - `n_estimators`: {100, 200, 300}
   - `max_depth`: {3, 5, 7}
   - `min_samples_split`: {5, 10, 15, 20}
   - `min_samples_leaf`: {2, 4, 6, 8, 10}
2. We then estimated the residual density of $A_i - \hat{\theta}(X_i)$ using a Gaussian Kernel, with bandwidth tuned via cross-validation over {1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0, 10.0}.

The number of cross-validation splits used for both steps was 5.

To estimate $\hat{m}(a, x)$, we trained the Random Forest regressor on the entire dataset, using both the context X and the action A as features to predict the outcome Y. For $\hat{f}(a|x)$, we used a separate Random Forest regressor to estimate the propensity score directly. This Random Forest was trained on the context X as features to predict the action A, with the same hyperparameter tuning process as described for $\hat{m}(a, x)$. This approach allows for a flexible, non-parametric estimation of the propensity score that can capture complex relationships between the context and the action.

This two-step approach allows us to capture both the structured relationship between context and action, and the residual variability, potentially leading to more accurate propensity score estimates.

A.5.4 Evaluation of propensity score estimation:

We check the Effective Sample Size (ESS) as defined by Martino et al. [2016], aiming for an ESS of around 481 out of 500 in the single-dimensional case and 401 out of 500 in the 10-dimensional case, which indicates acceptable quality. Additionally, we compare the performance using the estimated propensity score against the oracle propensity score. Figure 9, 10, 11 and 12 shows that the performance of our model is not significantly improved by the oracle propensity score, implying the reliability of our estimated propensity score.

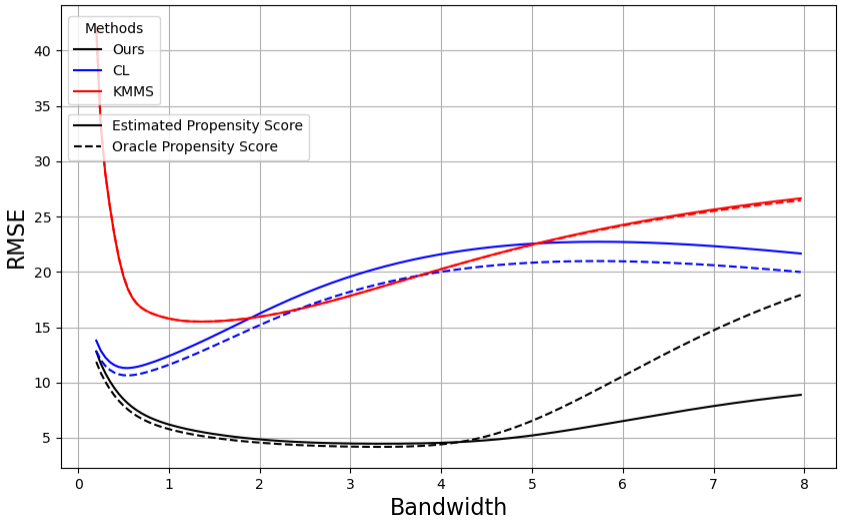

Figure 9: Comparison of different estimators with the oracle propensity score in the single-dimensional case (excluding IPW)

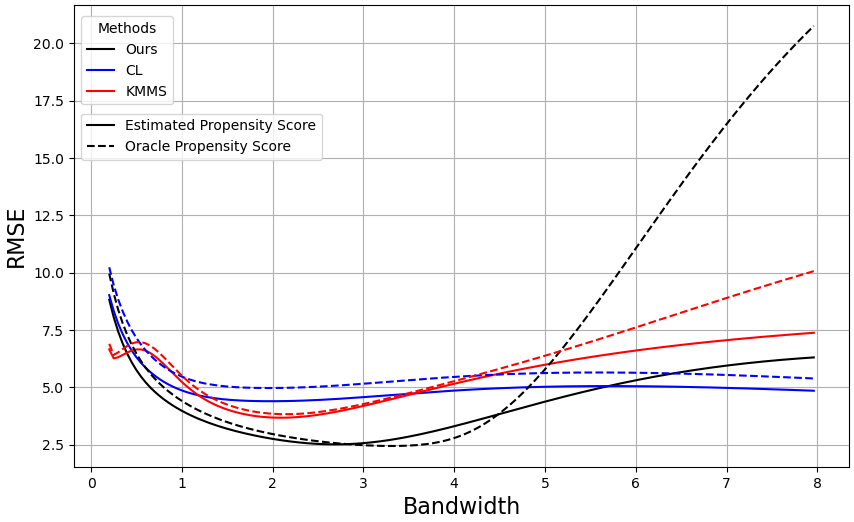

Figure 10: Comparison of different estimators with the oracle propensity score in the 10-dimensional case (excluding IPW)

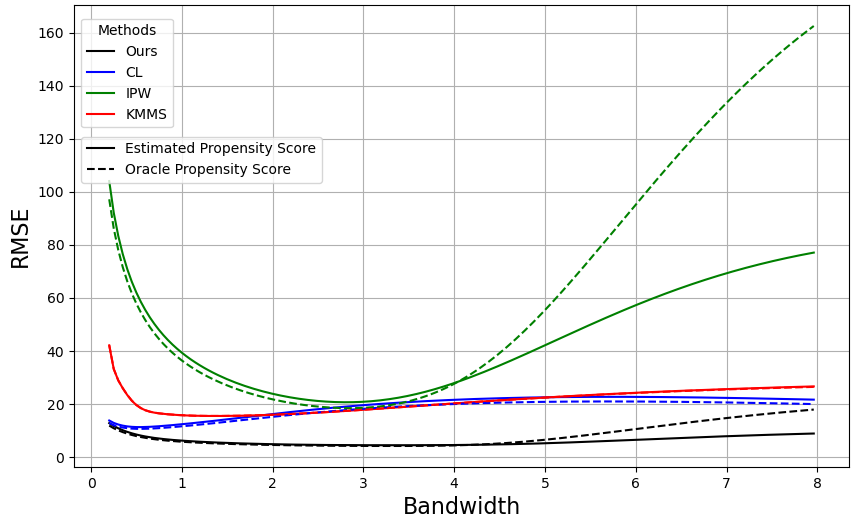

Figure 11: Comparison of different estimators with the oracle propensity score in the single-dimensional case

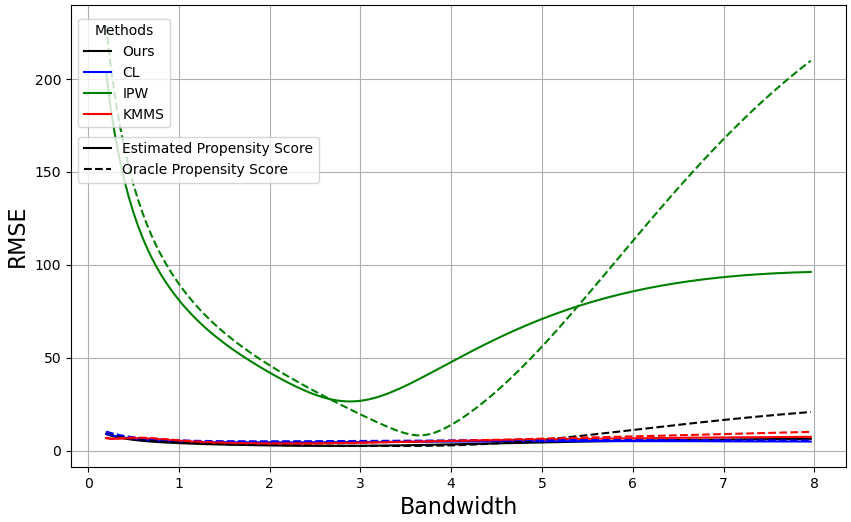

Figure 12: Comparison of different estimators with the oracle propensity score in the 10-dimensional case

Our experimental results align with the observations made in Robins and Greenland [1992] and Su et al. [2023], where the estimated propensity score sometimes outperforms the true propensity score in the IPW estimator. This counterintuitive phenomenon, known as the "propensity score paradox," occurs because the estimated propensity score can inadvertently correct for model misspecification in the outcome regression. In our experiments, we observed that using the estimated propensity score led to slightly lower RMSE compared to using the true propensity score, particularly for the IPW estimator.

Furthermore, the minor difference between the KMMS estimator and the KMMS estimator with true propensity score corresponds to the fact that the error term in KMMS is not related to δ, the estimation error of the propensity score.[3] This robustness to nuisance function estimation errors is a key advantage of the KMMS approach, as it allows for consistent estimation even when the propensity score model is misspecified or estimated with some error.

A.5.5 Detailed RMSE and Bias Analysis for All Estimators with Desirable Coverage

To provide a comprehensive view of the performance of all estimators, we present detailed RMSE and bias results for the IPW, CL, KMMS-I, and TR estimators in both single-dimensional and 10-dimensional cases.

[3]Note: The performance difference between KMMS-I and KMMS-I with oracle propensity score is visually subtle.

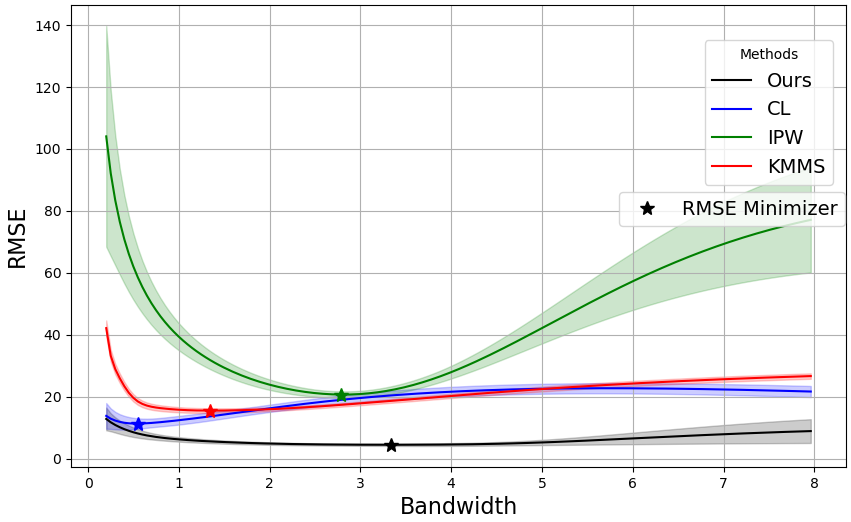

Figure 13: RMSE comparison of all estimators in the single-dimensional case

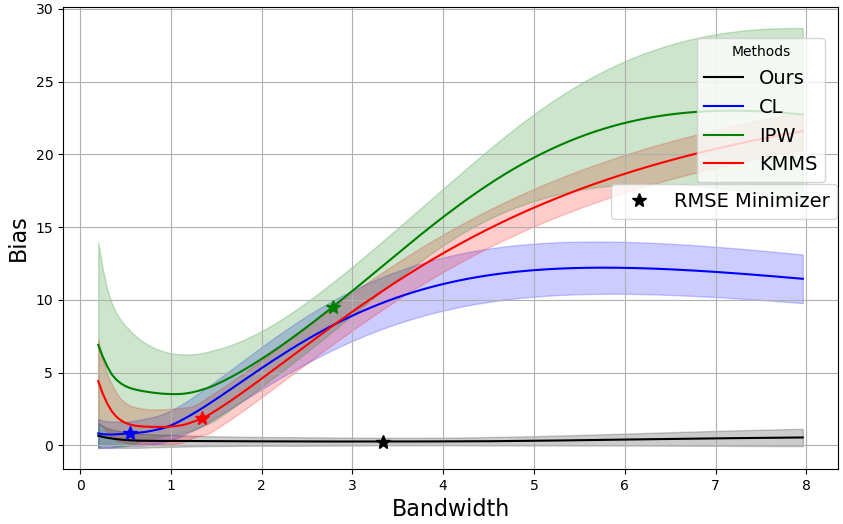

Figure 14: Bias comparison of all estimators in the single-dimensional case

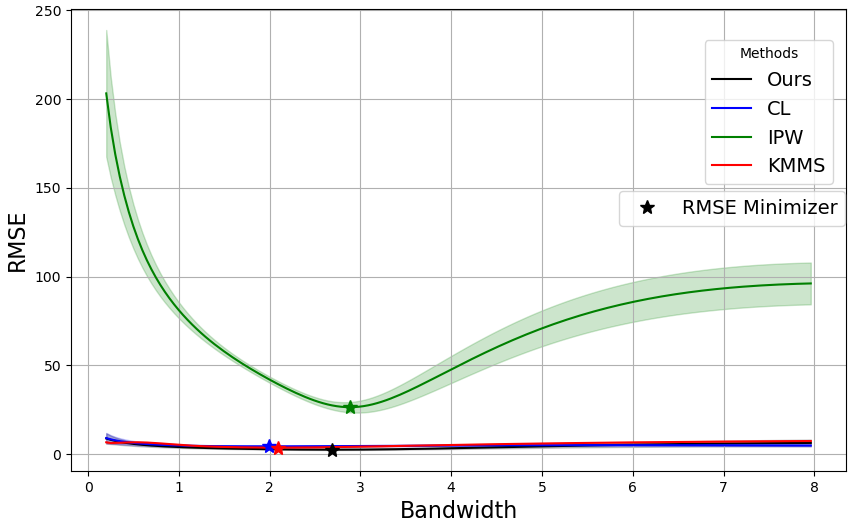

Figure 15: RMSE comparison of all estimators in the 10-dimensional case

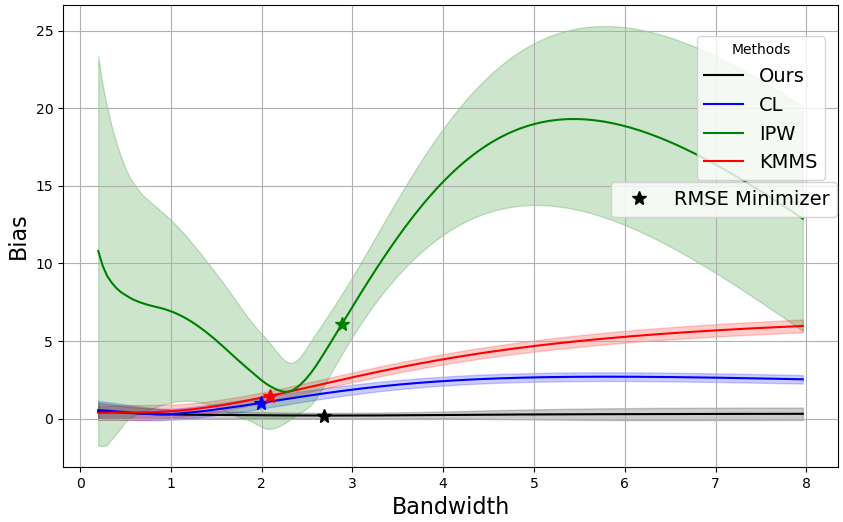

Figure 16: Bias comparison of all estimators in the 10-dimensional case

Figures 13, 14, 15, and 16 provide a comprehensive view of the performance of all estimators across different bandwidths. As observed in the previous analysis, the TR estimator consistently outperforms the other estimators in terms of both RMSE and bias reduction. The IPW estimator, as noted earlier, shows substantially higher RMSE and bias compared to the other methods across all bandwidth values in both dimensionality settings, as evident in these figures.

The CL and KMMS-I estimators demonstrate similar performance, particularly as the bandwidth increases, which aligns with our theoretical expectations. This similarity in performance between CL and KMMS-I at higher bandwidths aligns with our theoretical understanding, as both

estimators converge to similar forms in these conditions. However, it's important to note that when the bandwidth becomes large, we observe that the KMMS estimator becomes highly unstable. This instability is likely due to the sensitivity of the estimation of φ and μ to changes in the sample. Furthermore, we have observed that both KMMS exhibit unstable RMSE, which can be attributed to the sensitivity of the weight values in the weighted OLS to extreme bandwidths. As the bandwidth increases, these weight values can become highly volatile, leading to increased instability in the estimators' performance. The TR estimator's superior performance is evident in its ability to maintain lower RMSE and bias across a wider range of bandwidths, offering greater flexibility in the bias-variance trade-off. These results further reinforce the advantages of our proposed TR estimator, highlighting its robustness and efficiency in various dimensional settings and across different bandwidth choices.

A.5.6 Detailed RMSE and Bias Analysis for All Estimators with Poor Coverage

To further investigate the performance of our estimators under challenging conditions, we conducted simulations with poor coverage. This analysis focuses on the single-dimensional case, maintaining the same behavior policy and general settings as in Section A.5.5. However, we modified the evaluation policy to create a scenario with poor coverage. Specifically, we set the evaluation policy $\theta_2(X) = 1.5X - 10\pi$, which results in poor coverage over the evaluation area.

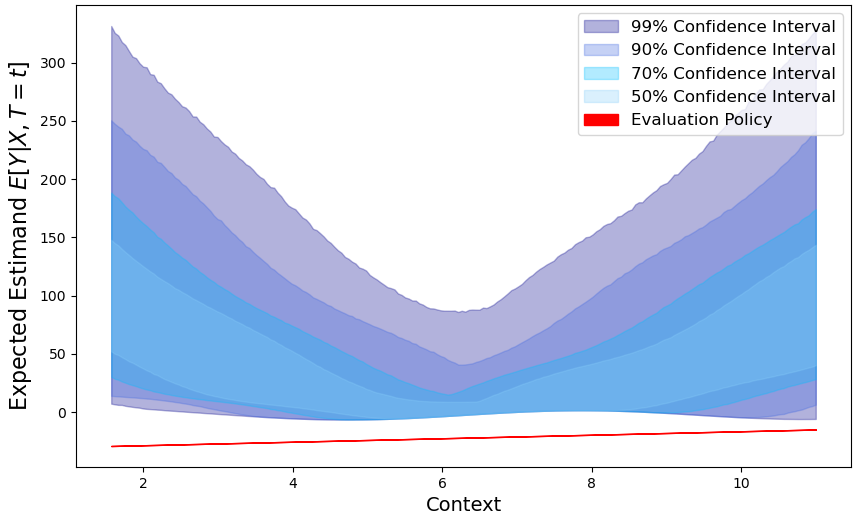

Figure 17: Coverage comparison of all estimators in the single-dimensional case with poor coverage

Figure 17 illustrates the coverage probabilities for all estimators in this challenging scenario. As evident from the graph, the poor coverage condition significantly impacts the performance of all estimators, leading to coverage probabilities that deviate substantially from the nominal 99% level.

This poor coverage scenario helps us understand the limitations of these estimators and highlights the importance of ensuring adequate overlap between the behavior and evaluation policies in off-policy evaluation tasks. It also underscores the need for careful consideration of policy design in practical applications to avoid situations where reliable estimation becomes challenging or impossible.

In the following figures, we present the RMSE and bias analysis for all estimators under this poor coverage condition:

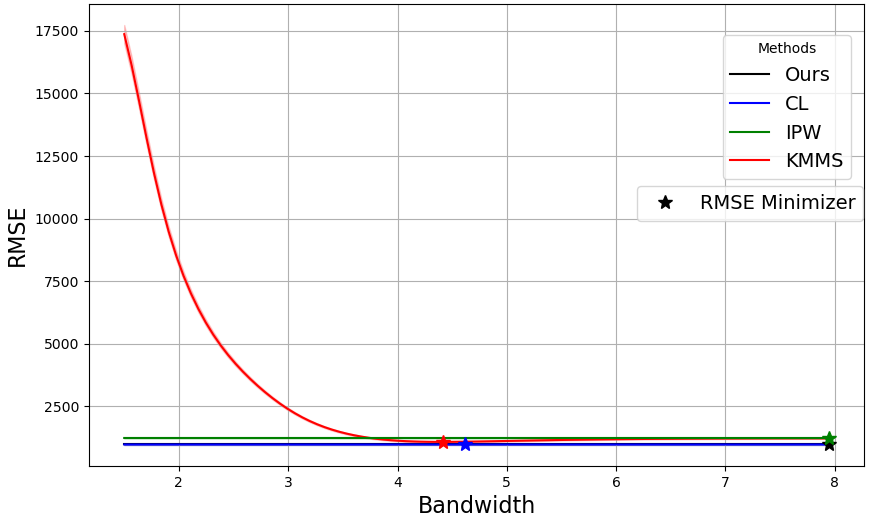

Figure 18: RMSE comparison of all estimators in the single-dimensional case with poor coverage

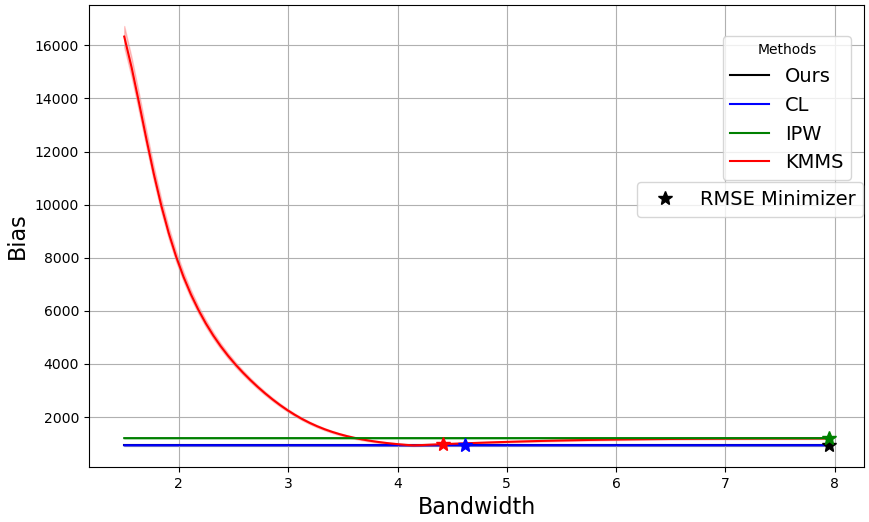

Figure 19: Bias comparison of all estimators in the single-dimensional case with poor coverage

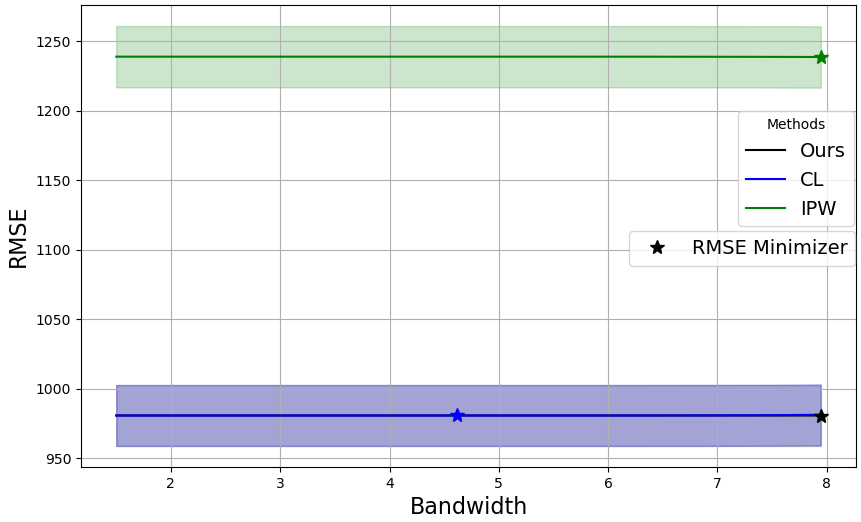

Figure 20: RMSE comparison of all estimators in the single-dimensional case with poor coverage (excluding KMMS and KMMS-I)

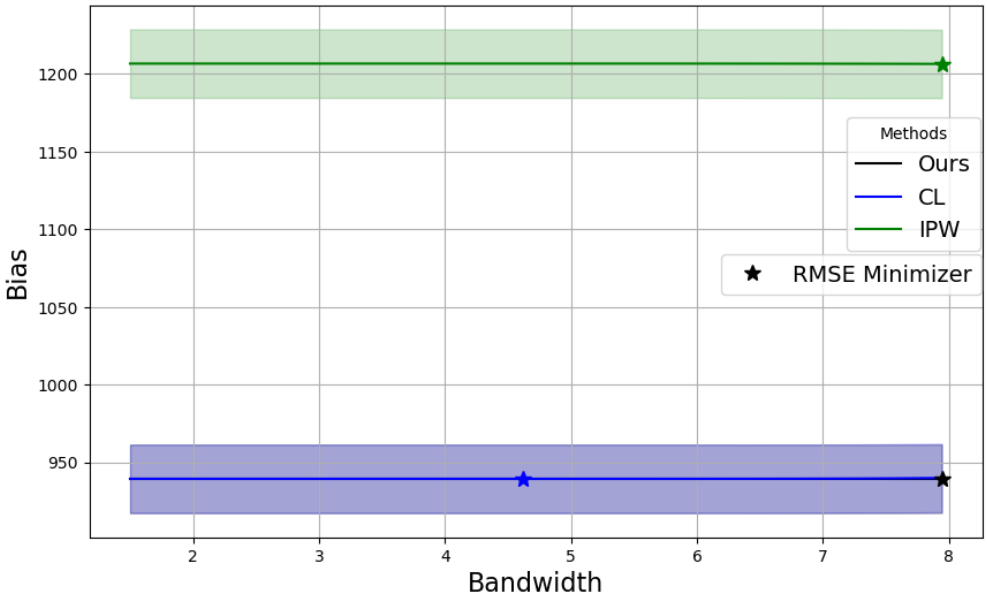

Figure 21: Bias comparison of all estimators in the single-dimensional case with poor coverage (excluding KMMS and KMMS-I)

Figures 18 and 19 show the RMSE and bias performance of all estimators under poor coverage conditions. These results demonstrate how the lack of adequate overlap between behavior and evaluation policies can lead to increased estimation errors and biases across all methods.

Despite the challenging conditions, the relative performance of the estimators remains consistent with our previous observations. The TR estimator continues to show better resilience in terms of

RMSE and bias reduction compared to other methods, although its absolute performance is notably affected by the poor coverage.

These findings emphasize the critical importance of ensuring sufficient overlap between behavior and evaluation policies in off-policy evaluation tasks. They also highlight the need for robust estimation methods that can maintain some level of reliability even under suboptimal conditions.

A.6 Detailed Semi-Synthetic Settings and Results

A.6.1 Data Source and Feature Construction

As described in Section 6, we evaluate our estimators on a semi-synthetic dataset constructed from the JD.com E-Commerce Data Challenge dataset Shen et al. [2020]. This real-world e-commerce dataset provides transaction-level logs from March 2018, capturing rich user-product interactions and pricing variations that enable realistic off-policy evaluation in a pricing context.

Feature Engineering We construct the context vector X from user-profile attributes and behavioral signals. The features are engineered to capture both demographic characteristics and engagement patterns that influence purchase decisions:

- **Behavioral Features:**
  - `click_count`: Number of clicks on the focal SKU within the same day, capturing same-day activity and engagement level.
  - `first_order_month`: The month of the user's first order, indicating customer tenure.
  - `months_since_first_order`: Continuous measure of customer lifetime, computed as the duration between the user's first order and the current interaction.
- **Demographic Features (One-Hot Encoded):**
  - `age`: User age group encoded as categorical features with the following mapping:

$$\text{age\_mapping} = \begin{cases} -1 & \text{U (Unknown age)} \\ 0 & \text{16-25} \\ 1 & \text{26-35} \\ 2 & \text{36-45} \\ 3 & \text{46-55} \\ 4 & \geq 56 \end{cases}$$

    These categories are then one-hot encoded to create separate binary features for each age group.
  - `gender`: User gender (one-hot encoded).
  - `marital_status`: User marital status (one-hot encoded).
- **Socioeconomic Features (One-Hot Encoded):**
  - `purchase_power`: User's purchasing capacity, capturing economic status. Since this is a discrete categorical variable in the dataset, it is one-hot encoded.
  - `city_level`: Tier classification of the user's city, reflecting urbanization level and market characteristics. This discrete variable is one-hot encoded.

– `education`: User education level. As a discrete categorical variable in the dataset, it is one-hot encoded.

These features collectively provide a comprehensive representation of user characteristics that influence both pricing sensitivity and purchase propensity. The combination of behavioral signals (clicks, tenure) and demographic attributes enables the estimators to capture heterogeneous treatment effects across different customer segments.

Price Range Filtering and Sample Construction In the raw JD.com dataset, we observed extreme pricing anomalies that posed significant challenges for policy learning. Specifically, some transactions involved substantial discounts where users purchased the focal SKU for prices lower than \$10. These extreme discount events, while representing valid historical transactions, create a critical issue for training pricing policies: models trained on such data may learn to predict negative or unrealistically low prices, which are not feasible in practical deployment scenarios.

To address this challenge and ensure that our learned behavior and evaluation policies produce economically viable pricing strategies, we restrict our analysis to a focused price range. Specifically, we filter the data to include only interactions where the final unit price falls within the range of \$31.9 to \$89.9 for the chosen SKU. This price range selection serves multiple purposes:

- It excludes extreme discount events that could lead to negative price predictions during policy optimization.
- It maintains sufficient price variability to enable meaningful off-policy evaluation while ensuring all prices remain within a realistic operational range.
- It provides adequate overlap between behavior and evaluation policies, as both policies are constrained to predict prices within this feasible region.

After applying this price filter and removing duplicate interactions, we obtain 773 unique user-context-price combinations from the real JD.com data. To construct our semi-synthetic dataset, we generate synthetic outcomes by first fitting an Ordinary Least Squares (OLS) regression model on these real features (context X and price A) to learn the underlying relationship, and then using this fitted model to generate outcomes Y. This approach creates a semi-synthetic dataset that preserves the realistic covariate distributions and pricing patterns from the real e-commerce platform while providing controlled, reproducible outcomes for rigorous evaluation. We repeat this outcome generation process 800 times to create semi-synthetic datasets, which are used for training both the behavior and evaluation policies to ensure robust performance across different data realizations.

It is important to note that the sample sizes reported in our experimental results (Section 6) refer to the number of observations used in each parallel estimation run for evaluating the policy value estimators, not the size of this base dataset.

A.6.2 Neural Network Architecture and Training

To model the pricing policies (both behavior and evaluation) on this semi-synthetic dataset, we employ fully connected neural networks. Unlike the Random Forest approach used in the synthetic experiments, the neural network architecture is specifically designed to capture both the expected pricing behavior and the uncertainty in pricing decisions through a probabilistic framework.

Dual-Output Network Architecture Our pricing policy is parameterized as a Gaussian distribution $\mathcal{N}(\mu_\theta(X), \sigma_\theta^2(X))$, where the mean $\mu_\theta(X)$ represents the expected price for a given context X, and the standard deviation $\sigma_\theta(X)$ captures the pricing uncertainty. To learn this distribution, we implement a dual-output neural network architecture:

- **Mean Network**: Predicts the mean price $\hat{\mu}_\theta(X)$ given the context features.
- **Standard Deviation Network**: Predicts the log-standard deviation $\log \hat{\sigma}_\theta(X)$, which is then exponentiated to ensure positivity: $\hat{\sigma}_\theta(X) = \exp(\log \hat{\sigma}_\theta(X))$.

The use of log-standard deviation (rather than directly predicting σ) is a standard practice in neural probabilistic modeling that ensures numerical stability and guarantees that the predicted standard deviation remains positive throughout training.

Loss Function and Optimization The network is trained by minimizing a composite loss function that combines probabilistic modeling with domain-specific regularization. The complete loss function consists of three components:

1. Negative Log-Likelihood (NLL) Loss: The primary objective is to maximize the likelihood of observed prices under the predicted Gaussian distribution. For a batch of training samples $\{(X_i, A_i)\}_{i=1}^n$ where A_i represents the observed price, the NLL loss is:

$$\mathcal{L}_{\text{NLL}}(\theta) = -\frac{1}{n} \sum_{i=1}^{n} \log \mathcal{N}(A_i; \hat{\mu}_\theta(X_i), \hat{\sigma}_\theta^2(X_i)) \tag{A.3}$$

Expanding the Gaussian log-likelihood:

$$\mathcal{L}_{\text{NLL}}(\theta) = \frac{1}{n} \sum_{i=1}^{n} \left[\log \hat{\sigma}_\theta(X_i) + \frac{(A_i - \hat{\mu}_\theta(X_i))^2}{2\hat{\sigma}_\theta^2(X_i)} + \frac{1}{2} \log(2\pi) \right] \tag{A.4}$$

2. Negative Price Penalty: To ensure the learned pricing policy produces economically viable prices, we add a penalty term that discourages negative price predictions:

$$\mathcal{L}_{\text{neg}}(\theta) = \lambda_1 \sum_{i=1}^{n} \max(0, -\hat{\mu}_\theta(X_i)) \tag{A.5}$$

where $\lambda_1 = 1.0$ is the penalty coefficient. This ReLU-based penalty activates only when the predicted mean price is negative, strongly discouraging the network from learning unrealistic pricing strategies.

3. Standard Deviation Regularization: To prevent the model from predicting unrealistically narrow or wide price distributions, we regularize the predicted standard deviation toward the empirical standard deviation of the training data:

$$\mathcal{L}_{\text{std}}(\theta) = \lambda_2 \left| \frac{1}{n} \sum_{i=1}^{n} \hat{\sigma}_\theta(X_i) - \sigma_{\text{data}} \right| \tag{A.6}$$

where σ_{data} is the standard deviation of observed prices in the training data and $\lambda_2 = 0.1$ is a regularization weight. This encourages the model to maintain calibrated uncertainty estimates that reflect the natural variability in the pricing data.

The complete training objective is the weighted sum of these three components:

$$\mathcal{L}_{\text{total}}(\theta) = \mathcal{L}_{\text{NLL}}(\theta) + \mathcal{L}_{\text{neg}}(\theta) + \mathcal{L}_{\text{std}}(\theta) \tag{A.7}$$

This composite loss function jointly optimizes prediction accuracy (through NLL), economic feasibility (through negative price penalty), and uncertainty calibration (through std regularization). We optimize this loss using the Adam optimizer Kingma and Ba [2015], which adapts the learning rate for each parameter based on first and second moment estimates of the gradients. This adaptive optimization is particularly beneficial for training neural networks with multiple outputs (mean and log-std) and multiple loss components that may have different convergence characteristics.

Hyperparameter Tuning To ensure optimal performance of the neural network pricing policies, we conduct a comprehensive grid search over several key hyperparameters:

- **Network Architecture**: We experiment with various hidden layer configurations to balance model capacity and generalization:
  - Single hidden layer: [64], [128]
  - Two hidden layers: [128, 64], [256, 128]

  These architectures range from relatively shallow networks to deeper configurations that can capture more complex nonlinear relationships between context features and pricing behavior.
- **Learning Rate**: We tune the Adam optimizer's learning rate over {1e-3, 5e-4, 1e-4} to control the step size during gradient descent. Smaller learning rates provide more stable convergence but may require longer training, while larger rates enable faster initial progress but risk overshooting optimal solutions.
- **Propensity Clipping Range**: To prevent numerical instability when computing inverse propensity weights, we clip predicted propensity densities to a minimum value. We search over {1e-5, 1e-3, 1e-1} to find a balance between preventing division by near-zero values and preserving the informativeness of the propensity estimates.
- **Regularization**: We do not employ dropout regularization in our final models, as early stopping mechanism provides sufficient regularization to prevent overfitting.

The best hyperparameter configuration is selected based on the negative log-likelihood on the validation set. This systematic tuning ensures that the learned pricing policies effectively capture the underlying relationships in the semi-synthetic data while maintaining good generalization properties.

Evaluation Policy Training While the behavior policy $\hat{\pi}_{\text{behavior}}(A|X)$ is trained directly on the semi-synthetic data to mimic the historical pricing strategy, the evaluation policy θ_{eval} is optimized using a different objective that balances policy improvement, KL constraint, and exploration. Specifically, we optimize the evaluation policy by maximizing an entropy-regularized, KL-penalized objective on the synthesized dataset:

$$\mathcal{L}(\theta) = \hat{\mathbb{E}}_{X,A}\left[\frac{\pi_\theta(A|X)}{\hat{\pi}_{\text{behavior}}(A|X)}\hat{R}(X,A)\right] - \beta \cdot \hat{\mathbb{E}}_X\left[D_{\text{KL}}(\hat{\pi}_{\text{behavior}}(\cdot|X)\|\pi_\theta(\cdot|X))\right] + \alpha \cdot \mathcal{H}(\pi_\theta) \quad \text{(A.8)}$$

where $\pi_\theta(A|X) = \mathcal{N}(A; \mu_\theta(X), \sigma_\theta^2(X))$ is the Gaussian evaluation policy being optimized, $\hat{\pi}_{\text{behavior}}(A|X)$ is the estimated behavior policy, $\hat{R}(X,A)$ is the estimated revenue (outcome), β is an adaptive KL penalty coefficient, and α is the entropy regularization weight. The three terms serve distinct purposes:

- **Policy objective**: The importance-weighted expected revenue $\hat{\mathbb{E}}_{X,A}\left[\frac{\pi_\theta(A|X)}{\hat{\pi}_{\text{behavior}}(A|X)}\hat{R}(X,A)\right]$ encourages the policy to maximize expected outcomes.
- **KL penalty**: The term $-\beta \cdot D_{\text{KL}}(\hat{\pi}_{\text{behavior}}(\cdot|X)\|\pi_\theta(\cdot|X))$ ensures the evaluation policy remains sufficiently close to the behavior policy to avoid extrapolation to regions with poor data coverage.
- **Entropy bonus**: The entropy term $\mathcal{H}(\pi_\theta) = \hat{\mathbb{E}}_X\left[\frac{1}{2}\log(2\pi e\sigma_\theta^2(X))\right]$ encourages exploration by rewarding policies with higher uncertainty, preventing premature convergence to overly deterministic strategies.

Adaptive Beta Controller: The KL penalty coefficient β is dynamically adjusted during training based on the measured KL divergence $d_t = \hat{\mathbb{E}}_X\left[D_{\text{KL}}(\hat{\pi}_{\text{behavior}}(\cdot|X)\|\pi_\theta(\cdot|X))\right]$ at iteration t. The controller maintains d_t within a target range $[d_{\text{lower}}, d_{\text{upper}}]$ using the following update rule:

$$\beta_{t+1} = \begin{cases} \min(\beta_t \cdot \lambda_{\text{inc}}, \beta_{\max}) & \text{if } d_t > d_{\text{upper}} \\ \max(\beta_t/\lambda_{\text{dec}}, \beta_{\min}) & \text{if } d_t < d_{\text{lower}} \\ \beta_t & \text{otherwise} \end{cases} \tag{A.9}$$

where λ_{inc} and λ_{dec} are the increase and decrease factors, respectively.

Hyperparameters: We train the evaluation policy with the following configuration:

- Training epochs: 500
- Batch size: Full batch
- Initial β: 1.4, α: 0.7
- Learning rate: 2e-5 (Adam optimizer)
- Beta controller parameters:
  - KL thresholds: $d_{\text{lower}} = 0.12$, $d_{\text{upper}} = 0.16$
  - Adjustment factors: $\lambda_{\text{inc}} = 1.70$, $\lambda_{\text{dec}} = 1.20$
  - Beta bounds: $\beta_{\min} = 0.80$, $\beta_{\max} = 3.5$
- **Early Stopping Criteria**: To prevent overfitting and ensure the evaluation policy maintains desirable properties, we employ a dual-criterion early stopping mechanism based on validation set performance:
  - **Revenue improvement threshold**: Training continues only if the validation revenue improves by at least 2%
  - **Coverage requirement**: The Effective Sample Size (ESS) on the validation set must be at least 0.92, ensuring sufficient overlap between the evaluation and behavior policies

  Training stops when the validation revenue meets the improvement threshold and the ESS also lands above the coverage requirement. This dual-criterion approach ensures that the learned evaluation policy not only achieves high expected revenue but also maintains adequate overlap with the behavior policy for reliable off-policy evaluation.

This adaptive mechanism, inspired by Schulman et al. [2017], balances the trade-off between policy improvement (maximizing expected revenue), staying within the support of the behavior policy (crucial for reliable off-policy evaluation), and maintaining sufficient exploration (via entropy regularization). The evaluation policy represents a potential alternative pricing strategy that the platform might consider deploying while maintaining reasonable overlap with historical pricing decisions. This approach ensures that our off-policy evaluation experiments reflect realistic scenarios where the evaluation policy is related to, but distinct from, the behavior policy.

A.6.3 Semi-Synthetic Experimental Results Across Sample Sizes

In this section, we present comprehensive experimental results evaluating the performance of different estimators on the semi-synthetic JD.com pricing dataset across varying sample sizes. To assess the scalability and finite-sample properties of each estimator, we conduct parallel experiments with four different sample sizes: $n \in \{10\text{k}, 50\text{k}, 100\text{k}, 200\text{k}\}$. For each sample size, we generate multiple independent datasets by sampling from the learned behavior policy and computing policy value estimates using IPW, CL, KMMS-I, and TR estimators.

The performance metrics we report are:

- **Root Mean Squared Error (RMSE)**: Measures the overall estimation accuracy, combining both bias and variance: $\text{RMSE} = \sqrt{\mathbb{E}[(\hat{V}(\theta) - V(\theta))^2]}$
- **Bias**: Measures systematic estimation error: $\text{Bias} = |\mathbb{E}[\hat{V}(\theta) - V(\theta)]|$

We present results both including and excluding the IPW and KMMS-I estimators to provide different perspectives on estimator performance. The exclusion of IPW and KMMS in some figures allows for clearer visualization of the performance differences between CL and TR estimators, which tend to have more similar performance characteristics.

Sample Size: $n = 10\text{k}$ Figures 22 and 23 show the performance of all estimators with a sample size of 10k observations. At this relatively small sample size, we observe that variance plays a significant role in the RMSE, and differences between estimators are most pronounced.

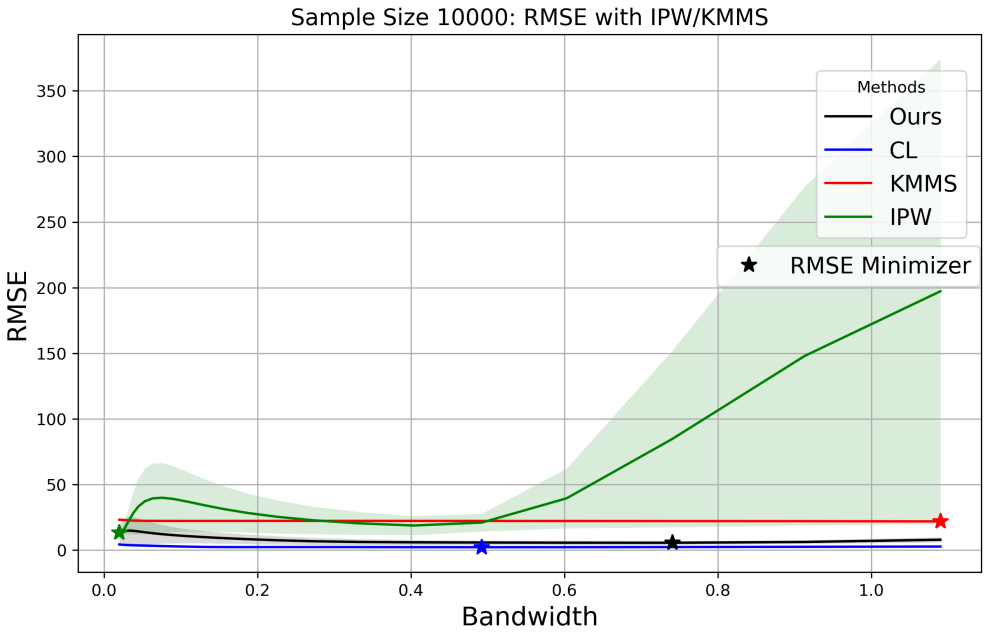

Figure 22: RMSE comparison for all estimators on semi-synthetic JD.com data ($n = 10$k)

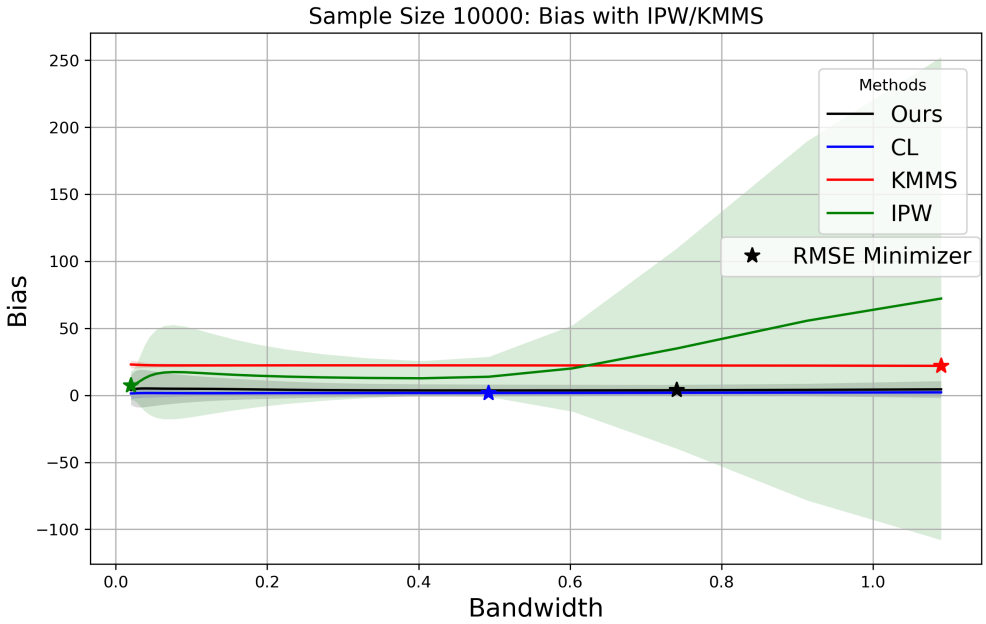

Figure 23: Bias comparison for all estimators on semi-synthetic JD.com data ($n = 10$k)

Figures 24 and 25 show the comparison excluding IPW and KMMS-I, providing a clearer view of the CL vs. TR performance.

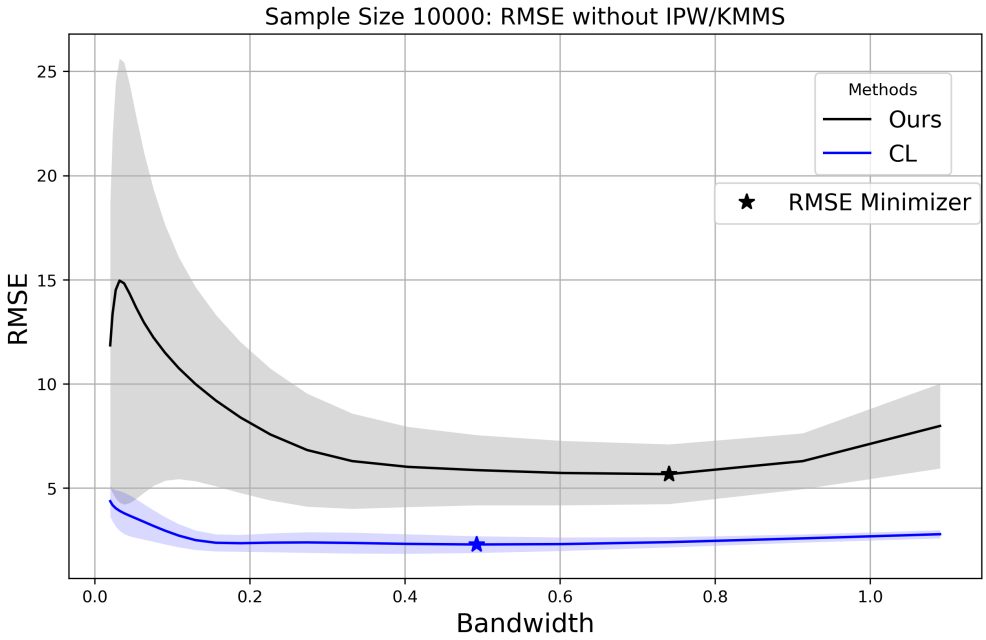

Figure 24: RMSE comparison for CL and TR estimators on semi-synthetic JD.com data ($n = 10$k)

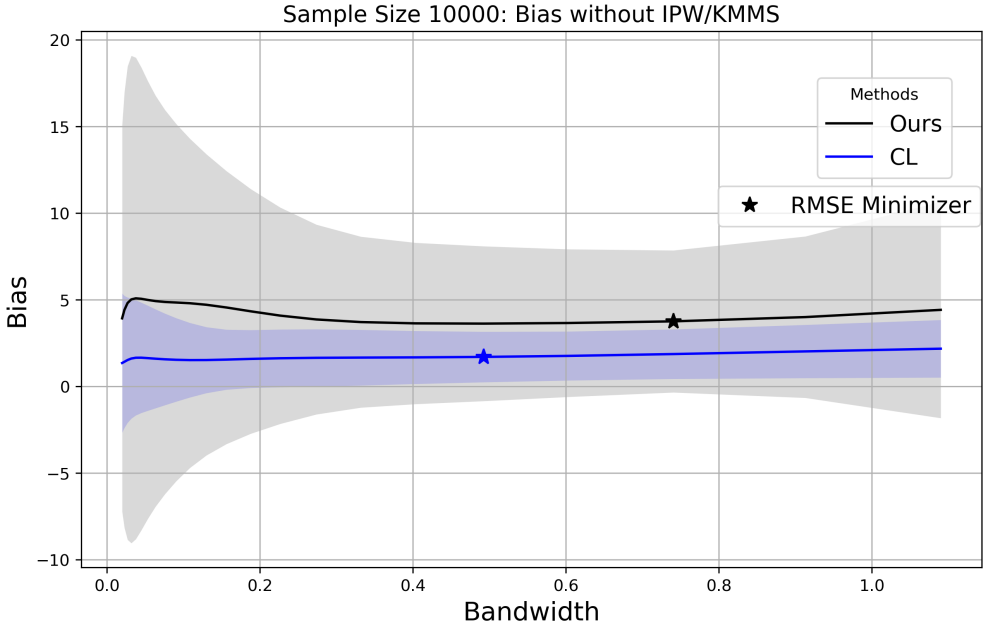

Figure 25: Bias comparison for CL and TR estimators on semi-synthetic JD.com data ($n = 10$k)

Sample Size: $n = 50$**k** With a sample size of 50k, the variance component decreases, and we begin to see clearer patterns in the bias-variance trade-off across different bandwidth choices.

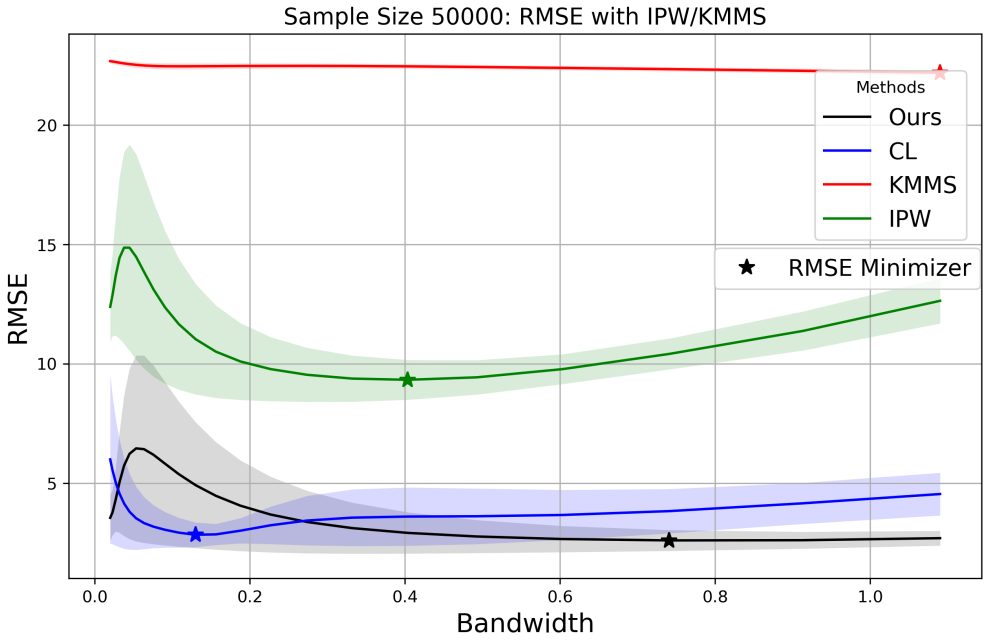

Figure 26: RMSE comparison for all estimators on semi-synthetic JD.com data ($n = 50$k)

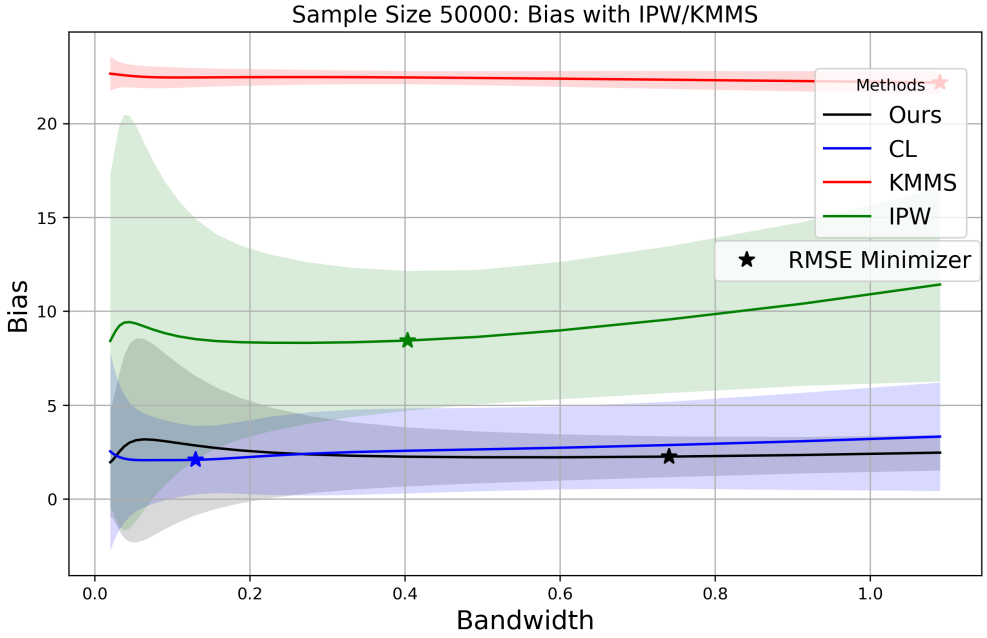

Figure 27: Bias comparison for all estimators on semi-synthetic JD.com data ($n = 50$k)

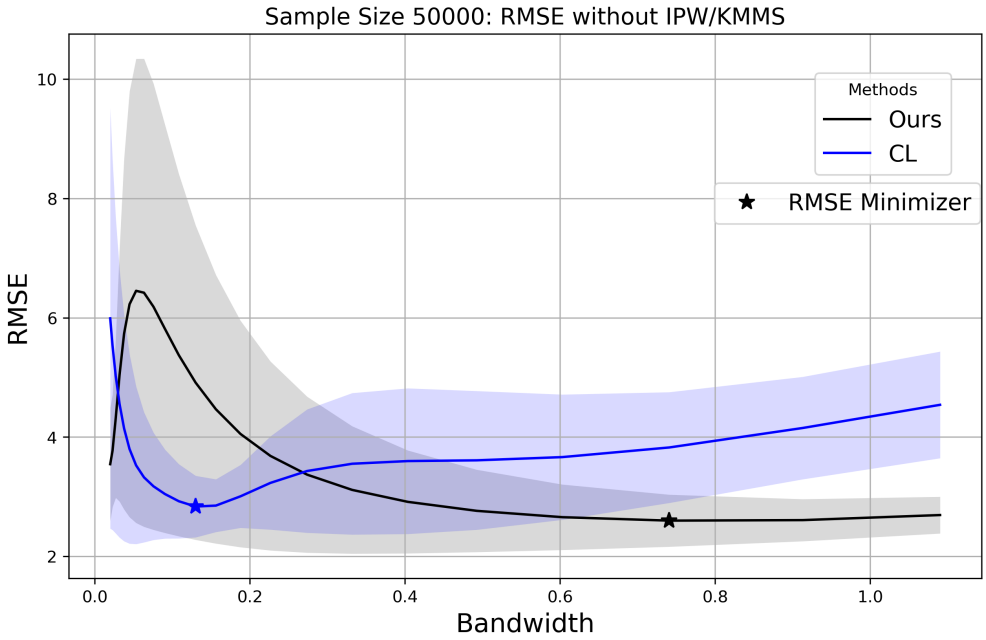

Figure 28: RMSE comparison for CL and TR estimators on semi-synthetic JD.com data ($n = 50$k)

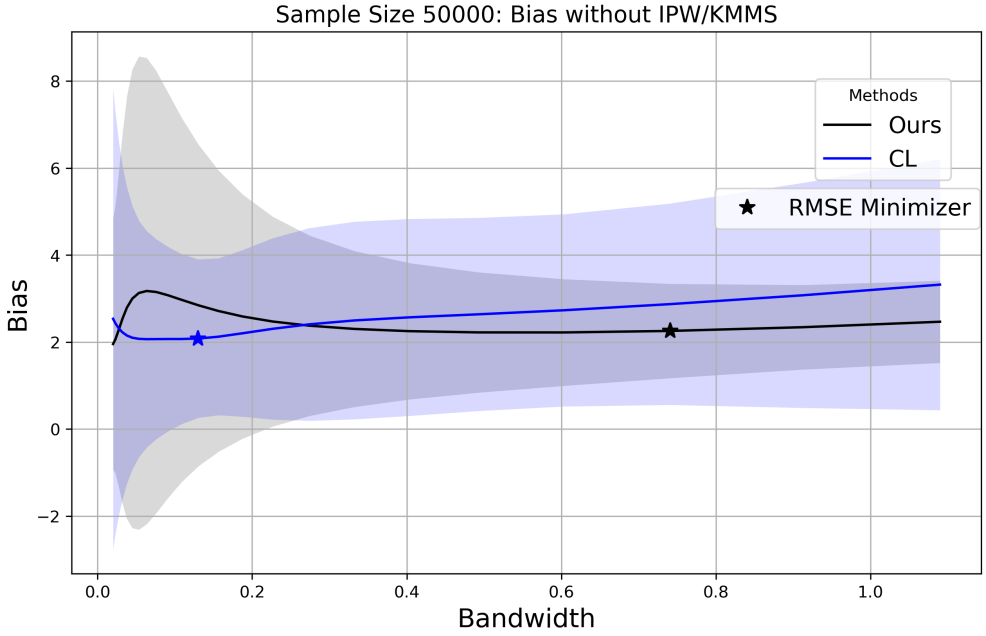

Figure 29: Bias comparison for CL and TR estimators on semi-synthetic JD.com data ($n = 50$k)

Sample Size: $n = 100$**k** At 100k observations, the asymptotic properties of the estimators become more apparent, and the bias terms dominate the MSE decomposition.

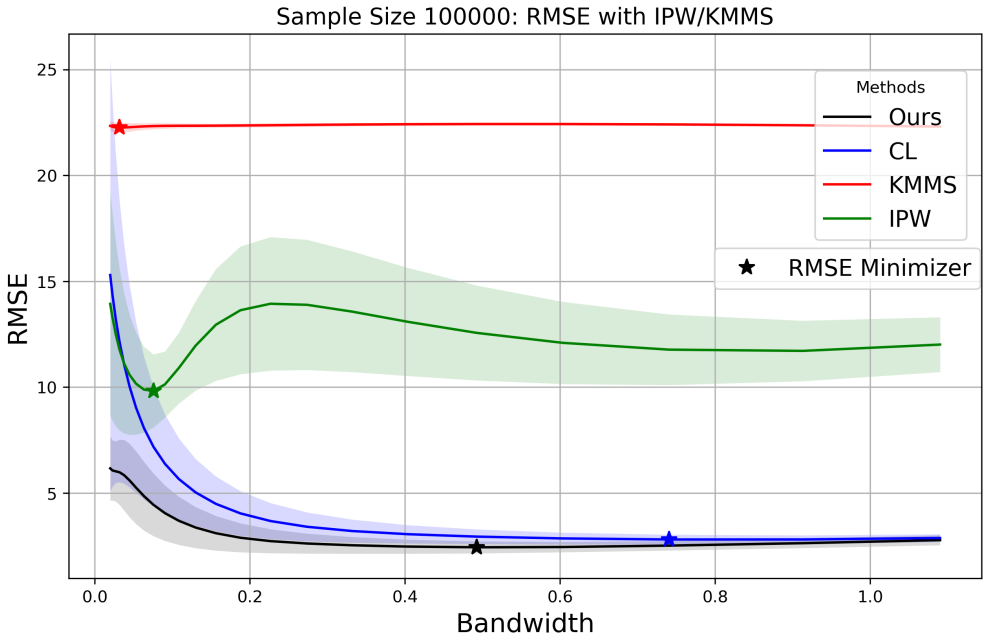

Figure 30: RMSE comparison for all estimators on semi-synthetic JD.com data ($n = 100$k)

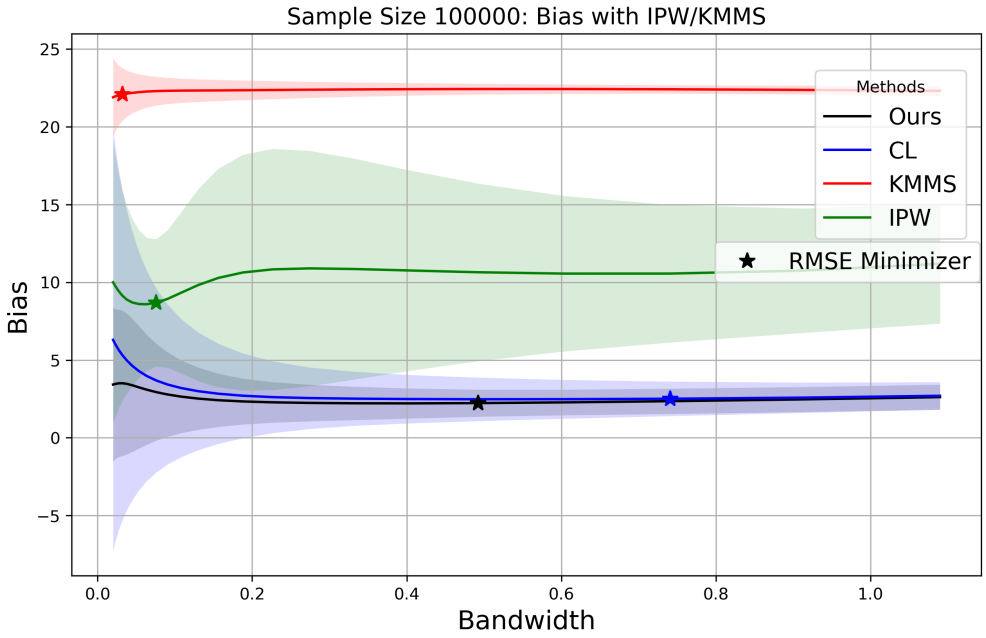

Figure 31: Bias comparison for all estimators on semi-synthetic JD.com data ($n = 100$k)

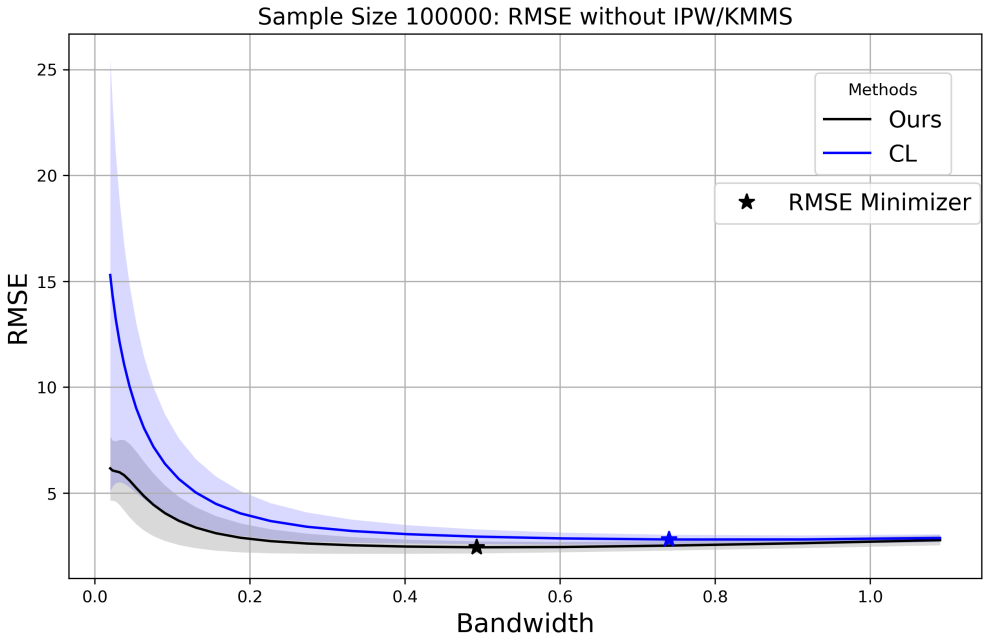

Figure 32: RMSE comparison for CL and TR estimators on semi-synthetic JD.com data ($n = 100$k)

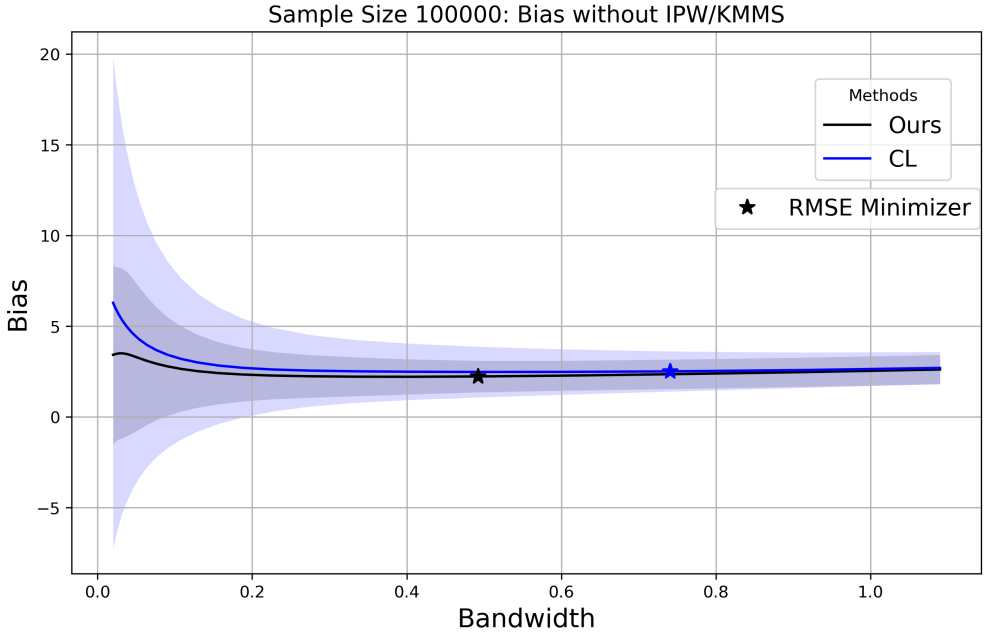

Figure 33: Bias comparison for CL and TR estimators on semi-synthetic JD.com data ($n = 100$k)

Sample Size: $n = 200$**k** With the largest sample size of 200k observations, we observe the most stable performance across all estimators, with minimal variance contribution to the overall error.

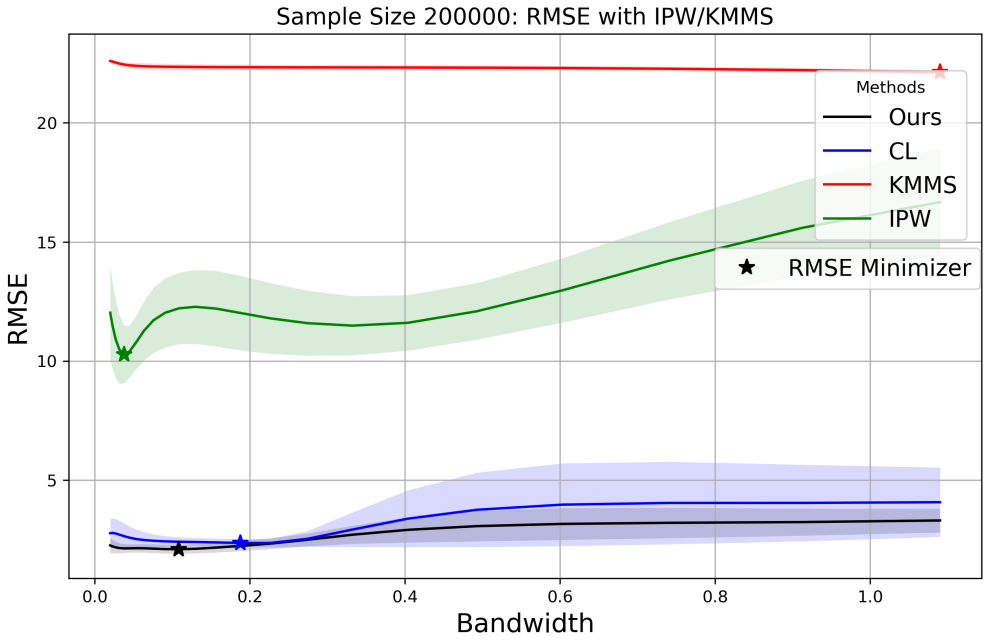

Figure 34: RMSE comparison for all estimators on semi-synthetic JD.com data ($n = 200$k)

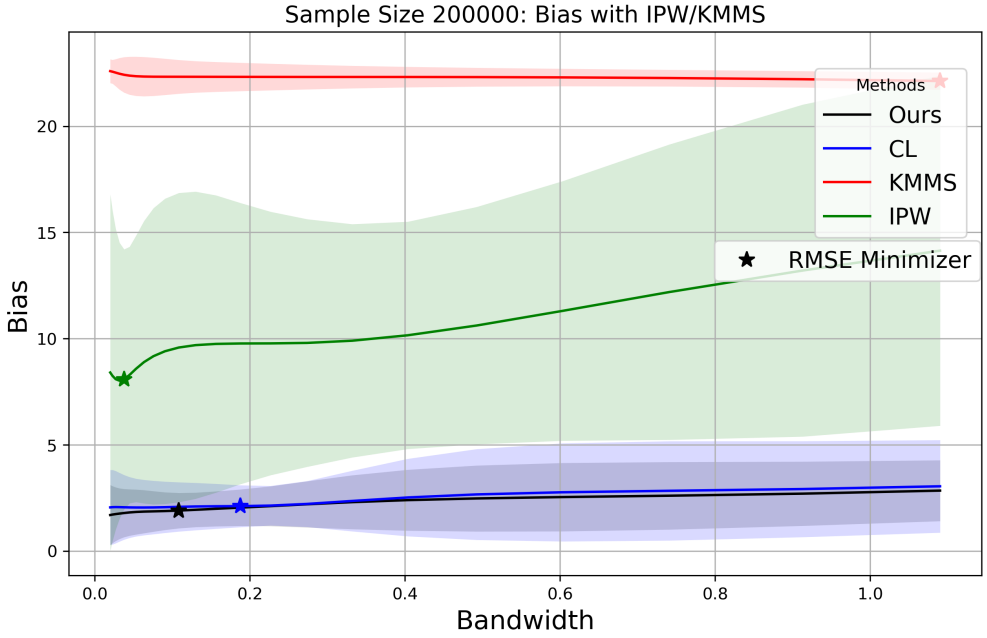

Figure 35: Bias comparison for all estimators on semi-synthetic JD.com data ($n = 200$k)

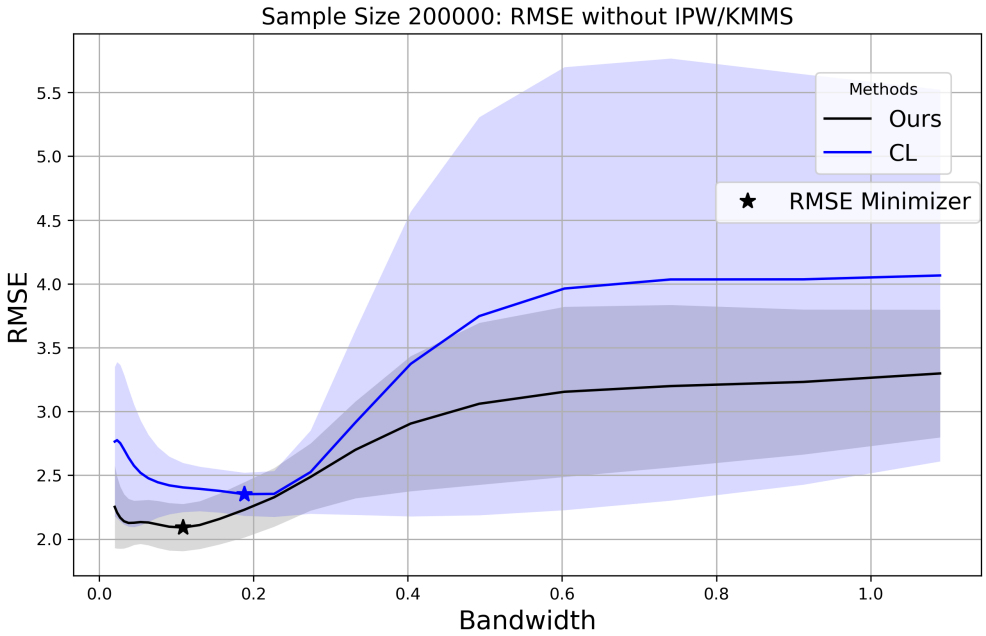

Figure 36: RMSE comparison for CL and TR estimators on semi-synthetic JD.com data ($n = 200$k)

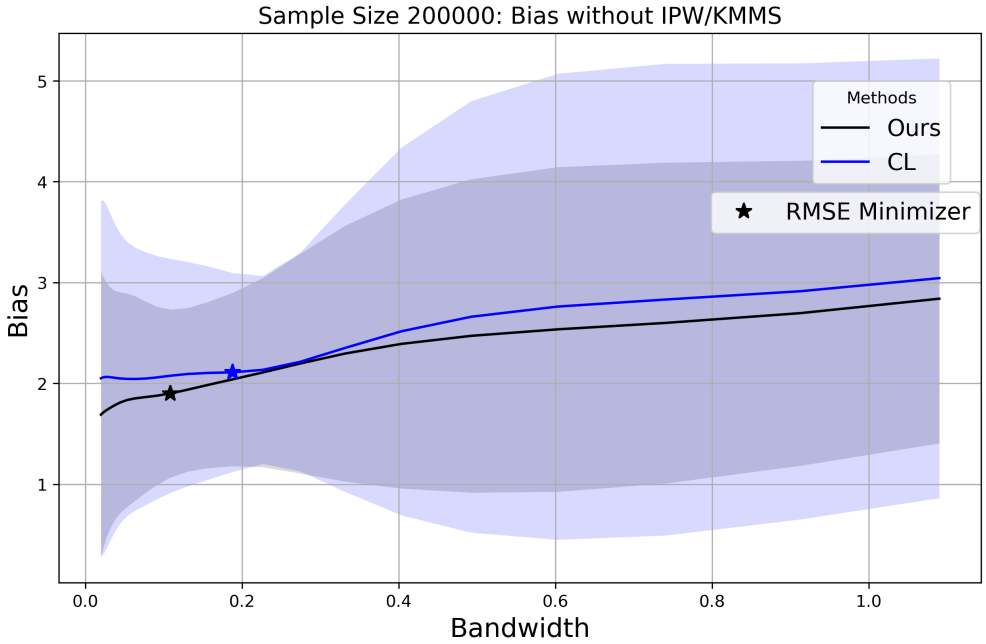

Figure 37: Bias comparison for CL and TR estimators on semi-synthetic JD.com data ($n = 200$k)

Summary of Findings Across Sample Sizes Across all sample sizes, we observe several consistent patterns:

- **TR estimator superiority**: The proposed TR estimator consistently achieves lower RMSE and bias compared to other methods across most sample sizes and bandwidth choices. This

advantage is particularly pronounced at smaller sample sizes where the triple robustness property provides substantial benefits.

- **Variance reduction with sample size**: As expected from asymptotic theory, the variance component of all estimators decreases as the sample size increases. This is evident from the convergence of RMSE curves at larger sample sizes.
- **Bandwidth selection robustness**: The TR estimator demonstrates greater robustness to bandwidth choice across the entire range of sample sizes. While other estimators show substantial performance degradation with suboptimal bandwidth choices, TR maintains relatively stable performance.
- **IPW performance**: The IPW estimator, which relies solely on propensity score weighting without outcome regression, shows substantially higher RMSE and bias across all sample sizes. This confirms the importance of incorporating outcome regression for bias reduction in off-policy evaluation.
- **Real-world applicability**: The consistent performance advantage of the TR estimator across realistic sample sizes.

These results provide strong empirical evidence for the effectiveness of the TR estimator in realistic semi-synthetic settings that preserve the complexity and distributional characteristics of real e-commerce pricing data.

References

Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001. ISSN 1573-0565. doi: 10.1023/A:1010933404324. URL `https://doi.org/10.1023/A:1010933404324`.

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

Arnoud V. den Boer. Dynamic pricing and learning: Historical origins, current research, and new directions. *Surveys in Operations Research and Management Science*, 20(1):1–18, 2015. doi: 10.1016/j.sorms.2015.03.001.

Richard D Gill and James M Robins. Causal inference for complex longitudinal data: the continuous case. *Annals of Statistics*, pages 1785–1811, 2001.

Leo Guelman and Montserrat Guillén. A causal inference approach to measure price elasticity in automobile insurance. *Expert Systems with Applications*, 41(2):387–396, 2014. doi: 10.1016/j.eswa.2013.07.059.

Nathan Kallus and Angela Zhou. Policy evaluation and optimization with continuous treatments. *arXiv preprint arXiv:1802.06037*, 2018.

Edward H Kennedy, Zongming Ma, Matthew D McHugh, and Dylan S Small. Non-parametric methods for doubly robust estimation of continuous treatment effects. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 79(4):1229–1245, 2017.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of the Third International Conference on Learning Representations*, 2015. URL `https://arxiv.org/abs/1412.6980`.

L. Martino, V. Elvira, and F. Louzada. Effective sample size for importance sampling based on discrepancy measures. *arXiv preprint arXiv:1602.03572*, 2016.

Whitney K. Newey. Uniform convergence in probability and stochastic equicontinuity. *Econometrica*, 59(4):1161–1167, 1991. doi: 10.2307/2938179.

Benedikt M. Pötscher and Ingmar R. Prucha. Generic uniform convergence and equicontinuity concepts for random functions: an exploration of the basic structure. *Journal of Econometrics*, 60(1–2):23–63, 1994. doi: 10.1016/0304-4076(94)90037-X.

David J. Reibstein and Hubert Gatignon. Optimal product line pricing: The influence of elasticities and cross-elasticities. *Journal of Marketing Research*, 21(3):259–267, 1984. doi: 10.1177/002224378402100303.

James M Robins and Sander Greenland. Estimating exposure effects by modelling the expectation of exposure conditional on confounders. *Biometrics*, 48(2):479–495, 1992.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. URL `https://arxiv.org/abs/1707.06347`.

Max Shen, Christopher S. Tang, Di Wu, Rong Yuan, and Wei Zhou. Jd.com: Transaction-level data for the 2020 msom data driven research challenge. *Manufacturing & Service Operations Management*, 26(1):2–10, 2020.

Fangzhou Su, Wenlong Mou, Peng Ding, and Martin J. Wainwright. High-dimensional analysis and bias correction for continuous treatment effects. *arXiv preprint arXiv:2303.17102*, 2023.

Yong Wu, Yanwei Fu, Shouyan Wang, and Xinwei Sun. Doubly robust proximal causal learning for continuous treatments. *arXiv preprint arXiv:2309.12819*, 2023.