Winner's Curse in Data-Driven Decision-Making: Evidence and Solutions

Sikun Xu

Olin Business School, Washington University in St. Louis, St. Louis, MO 63130, sikun@wustl.edu

Raphael Thomadsen

Olin Business School, Washington University in St. Louis, St. Louis, MO 63130, thomadsen@wustl.edu

Dennis J. Zhang

Olin Business School, Washington University in St. Louis, St. Louis, MO 63130, denniszhang@wustl.edu

Data-driven decision-making involves estimating the value associated with each possible decision and selecting the optimal estimated choice. This type of decision making is at the heart of a huge number of modern marketing applications, including ad creative choice, algorithm optimization, personalized targeting, A/B testing, pricing, and assortment optimization. In practice, it is crucial not only to estimate the optimal policy but also to accurately measure the incremental value or lift of that policy. In this paper, we first demonstrate theoretically that selecting the optimal policy based on estimated effects from data leads, on average, to overly optimistic evaluations of the policy value, a phenomenon known as the winner's curse. This is true no matter what best methodology is used to estimate the policy's effectiveness. We then empirically illustrate that the winner's curse arises in a wide range of key marketing applications, including A/B testing, personalized targeting, and counterfactual estimation using structural models, and that its magnitude can be substantial even within realistic parameter ranges commonly seen in the literature. Given the generality of this problem across diverse settings, we propose a correction method based on a non-continuous bootstrap approach designed to effectively mitigate the winner's curse in nearly all scenarios. Finally, we benchmark our proposed method against several existing context-specific solutions, demonstrating that our bootstrapbased correction consistently performs well and frequently outperforms alternative methods across important marketing contexts.

Key words:

1. Introduction

In the era of digital marketing, most marketing problems involve data-driven decision-making. Data-driven decision-making, as the name suggests, requires decision-makers first to estimate the value of each potential action based on data, and subsequently optimize the decision using these estimates. This procedure encompasses a wide range of marketing applications. For example, digital platforms regularly run thousands of A/B tests to refine their daily decisions. In each A/B test, platforms initially estimate the effectiveness of each treatment option and then choose the optimal treatment based on these outcomes. Similarly, traditional firms rely heavily on data-driven

methods for personalized advertising and promotional targeting. These firms first assess the impact of different advertisements or coupons across consumer subgroups, and then select the optimal advertisement or coupon for each segment of customers. Furthermore, many marketing researchers and practitioners utilize structural estimation and counterfactual simulations to evaluate policy alternatives. In structural estimation, model parameters are first estimated, and then, in various counterfactual scenarios—such as pricing or assortment decisions—optimal policies are determined based on these estimated models.

We refer to this process of first estimating the expected value of each option from data and subsequently selecting the optimal action as the "inference-then-optimize" framework. In practice, it is crucial not only to identify optimal decisions but also to obtain accurate estimates of the value associated with these decisions. In the case of A/B testing, accurately measuring the effects of chosen treatments is essential because firms need precise forecasts for internal tracking purposes, and potentially for public reporting, such as financial disclosures regarding anticipated outcomes. Similarly, in personalized advertising and promotions, significant fixed costs are often associated with collaboration with advertising platforms, prompting firms to demand reliable Return on Investment (ROI) estimates of their personalized targeting programs before committing resources. Additionally, from a human resource management perspective, understanding the actual impacts of these decisions is essential for accurately evaluating managers and employees responsible for these initiatives, thus ensuring fairness and transparency in performance appraisals and promotions. Similarly, in some cases there will be other operational costs, such as handling of ordering and inventory, that will depend on accurate forecasts.

Despite the importance of accurately reporting the optimal decision's value through the inferencethen-optimize procedure, this issue has received limited attention within the marketing literature, particularly concerning whether traditional methods in the literature and practice reliably estimate this value. This research therefore seeks to address the following questions: Does reporting the expected value of the optimal policy in the inference-then-optimize framework accurately capture the true value? If inaccuracies exist, how substantial might the biases be, and under which marketing scenarios are these biases likely to be significant? Finally, can we develop general solutions applicable across a broad spectrum of marketing contexts to correct this estimation bias?

To address these questions, we first theoretically demonstrate that under the general inference-then-optimize framework, the reported value of the optimal decision is inherently overoptimistic—even when the estimated value of each option is unbiased and consistent. The winner's curse arises from two key aspects of the inference-then-optimize framework. First, estimation errors inevitably occur in the inference stage, as perfect estimation of each option's value is typically unattainable, especially when data is limited. Second, the optimization stage requires selecting among various options based on their estimated values from the inference stage. When these two conditions are met, the conditional expected value of the chosen optimal option will differ from its unconditional expected value, thereby creating the winner's curse.

After establishing the general existence of the winner's curse, we focus on a specific and commonly encountered scenario involving two options, each with normally distributed values. In this scenario, we derive a theoretical expression for the winner's curse and demonstrate that its magnitude is proportional to the difference in expected returns between the two options and inversely proportional to their joint empirical variance. Specifically, the winner's curse becomes more pronounced when the two options yield similar expected values, making them difficult to distinguish, or when the options have high variance or limited observational data.

Having demonstrated the existence and comparative statics of the winner's curse theoretically, we next employ simulations to illustrate its presence across a broad range of scenarios. We begin with an A/B testing context and explore various extensions. For instance, experimental outcomes can be either continuous (e.g., customer spending or time spent on an online platform) or discrete (e.g., purchase versus non-purchase outcomes). We then examine personalized targeting problems, where different treatments are assigned to groups of heterogeneous consumers. A similar logic applies when choosing the "best" option for each segment based on data, which typically leads to an overestimation of aggregated targeting values across segments. Subsequently, we consider scenarios without predefined discrete segments, where the impact of each policy on outcomes depends continuously on observable customer features. This situation is common in marketing contexts, where firms utilize predictive covariates for personalized targeting. Here again, we confirm the presence of the winner's curse. Importantly, we show that the winner's curse is not only present but also substantial within parameter ranges frequently encountered in previous marketing research.

Given the widespread presence of the winner's curse across various marketing applications, we next propose a solution to address this problem. Although prior literature in biostatistics and economics has offered solutions to the winner's curse, these remedies are typically tailored to specific contexts, requiring substantial adaptation when applied to new settings. To overcome this limitation, we introduce a model-free, bootstrap-based policy value estimator, encompassing standard bootstrapping, as well as subsampling and perturbation methods, that systematically adjusts for upward bias. Analogous to addressing the winner's curse in auction theory, where bids are shaded to avoid overbidding, our approach involves adjusting ("shading") the policy value estimates to counteract over-optimism. The degree of correction required is based on the magnitude of the winner's curse, which we estimate using bootstrap methods. By resampling with replacement from the original dataset to create bootstrap samples, we reconstruct the policies within each bootstrap iteration and subsequently evaluate these policies on the original dataset. This procedure approximates the value of the focal policy under real-world conditions using the full sample.

We then benchmark our approach against several widely used methods in biostatistics and economics, including sample splitting, Bayesian shrinkage estimation (e.g., Johnstone and Silverman (2004), Efron (2011)), and selective inference (e.g., Rasines and Young (2020), Kuchibhotla et al. (2022), Andrews et al. (2024)). The first method, sample splitting, divides the data into two subsets: one subset is used to learn policies from data, and the other is reserved for policy evaluation. The second benchmark, Bayesian shrinkage estimation, evaluates the learned policies through posterior estimation. The third method, selective inference, models the distribution of the winning policy conditional on its selection (e.g., as a truncated normal distribution), typically relying on strong modeling assumptions about the underlying data-generating process.

After benchmarking our method against these widely discussed approaches from other literature, we demonstrate that our approach consistently performs well across various applications and parameter settings, frequently outperforming other methods, though not universally. Each of the benchmark methods exhibits significant drawbacks. First, sample splitting guarantees unbiased estimates of the reported policy value but sacrifices efficiency, as only part of the dataset is used for policy estimation while the remainder is used solely for evaluation. Second, Bayesian estimation can be applied broadly but relies heavily on correctly specified prior assumptions. Misspecification of the prior significantly diminishes performance. Lastly, selective inference performs effectively if the data generating process (DGP) assumed by the estimator matches the actual DGP; however, mismatches in these assumptions lead to poor performance.

Finally, although our approach generally performs well, the standard bootstrap method does not always fully correct for the winner's curse due to discontinuities in the optimization process. Furthermore, alternative approaches such as the m-out-of-n (e.g., Bickel et al. (1997), Chakraborty et al. (2013)) or numerical bootstrap (Hong and Li 2020) can theoretically handle these discontinuities but typically depend on hyperparameters that are sensitive and lack robust theoretical guidance for tuning. Considering the limitations inherent in all these methods, including ours, we conclude by discussing practical considerations that can help researchers and practitioners choose the most appropriate method for their specific contexts.

The paper is organized as the following. We start with the analytical analysis of winner's curse in Section 2. We develop an analytical formulation of winner's curse and provide a sufficient condition for it to hold. In Section 3, we discuss different potential remedies and propose our bootstrap-based policy value estimator in Section 3. Finally, we demonstrate winner's curse and the performance of various correction methods via extensive simulation under A/B testing and targeting contexts in Section 4 and Section 5.

2. The Winner's Curse Phenomena

In this section, we start by discussing winner's curse in a data-driven single-segment targeting problem. We show that winner's curse is a common phenomena that could exist in a wide range of inference-then-optimize problems. We then propose a bootstrap-based correction estimator for removing the winner's curse. We will also discuss other proposed methods in the litearture for alleviating winner's curse.

2.1. Winner's Curse of Single-Segment Targeting

Consider a targeting problem where we want to assign one of two treatments to a single segment of homogeneous consumers. Let Y_i denote the consumer's response of interest, such as purchase decision, spending, time-spent on platform, etc. In this section we consider continuous response, i.e., $Y_i \in \mathbb{R}$ for analytical clarity. However, the results generalize to discrete response and we will discuss them via simulation in later sections. Let $T_i \in \{a_1, a_2\}$ denote the treatment variable we need to decide. The causal effects of the two treatments are denoted as τ_1 and τ_2 respectively. Because we consider a single segment problem, the causal effects are the same for all individuals. Finally, the data generation process for consumer's response is modeled as (1):

$$Y_i = \tau_1 \mathbb{1}\{T_i = a_1\} + \tau_2 \mathbb{1}\{T_i = a_2\} + \epsilon_i.$$
(1)

where $\mathbb{1}\{\cdot\}$ is an indicator function and ϵ_i is idiosyncratic risk following mean-zero normal distribution with variance σ^2 .

Because the true treatment effects are unknown, solving the targeting problem follows an inference-then-optimize procedure. In the inference stage, we assume that the decision maker have access to some unbiased estimators for the treatment effects. Let $\hat{\tau}_1$ and $\hat{\tau}_2$ denote the unbiased estimators we constructed from a dataset $\mathcal{D} = \{(T_i, Y_i)\}_{i=1}^N$ with N observations. For example, when the dataset is collected from a randomized control trial, the common difference-in-mean or regression adjustment estimators are unbiased. Notably, even though the estimators are unbiased, they will have mean zero estimation error as long as the sample size N is finite. When we use the estimators as input to the optimization stage, such estimation error will cause over-optimistic policy value estimate.

In the optimization stage, we want to choose among the two treatments to maximize expected consumer's response (e.g., spending). The optimization problem is formulated as (2). $f(T_i, \hat{\tau}_{1:2})$ is the objective value given treatment assignment and parameter estimates. Let $\mathbf{T}^*(\hat{\tau}_{1:2})$ denote the optimal treatment decision constructed based on the estimated effects. We call it the data-driven decision. Note that the $\mathbf{T}^*(\hat{\tau}_{1:2}) = a_1$ if $\hat{\tau}_1 \geq \hat{\tau}_2$ and chooses a_2 otherwise.

$$\max_{T_i \in \{a_1, a_2\}} f(T_i, \hat{\tau}_{1:2}) = \hat{\tau}_1 \mathbb{1}\{T_i = a_1\} + \hat{\tau}_2 \mathbb{1}\{T_i = a_2\}.$$
(2)

The goal is to estimate the actual policy value of the data-driven decision. Using the notations above, the actual value of the data-driven decision can be written as $f(\mathbf{T}^*(\hat{\tau}_{1:2}), \tau_{1:2})$, meaning that we are evaluating the data-driven decision using the actual causal effects $\tau_{1:2}$. Because we do not have access to the actual causal effects, researchers and practitioners use the estimated effects to evaluate the data-driven decision, i.e., $f(\mathbf{T}^*(\hat{\tau}_{1:2}), \hat{\tau}_{1:2})$, which is also the optimal objective value from directly solving (2). The difference between the two values are summarized in Table 1.

Table 1 Comparison of Different Objective Values			
Targeting Value	Decision under Evaluation	Evaluation Environment	
$f(\mathbf{T}^*(\hat{\tau}_{1:2}), \hat{\tau}_{1:2})$	Data-Driven Decision	Estimated	
$f(\mathbf{T}^*(\hat{\tau}_{1:2}), \tau_{1:2})$	Data-Driven Decision	Actual	

 Table 1
 Comparison of Different Objective Values

We find that when we evaluate the data-driven decision using estimated treatment effects, we are overestimating the actual targeting value in expectation. The main cause is we are selecting treatment assignment based on estimated causal effects. Instead of picking the treatment with an actually larger effect, we might pick the one with inflated estimate due to estimation error. As a result, we will overestimate the targeting value. This is called the winner's curse phenomenon, which was first found in auction theory, where the winner of a common-value auction over-bids. Next, we formalize this idea. Suppose that the estimators follow $\hat{\tau}_j \sim \mathcal{N}(\tau_j, \sigma^2/N_j)$, where N_j is the number of individuals assigned with treatment a_j in the dataset. We can construct them by taking sample average of the responses from individuals assigned with treatment a_j . Although the estimators are unbiased and \sqrt{N} -consistent, basically the best property we can hope for, Proposition 1 shows that we still have winner's curse in a single-segment targeting problem.

PROPOSITION 1 (Winner's Curse in Single-Segment Targeting). Assume that we have unbiased and independent estimators $\hat{\tau}_j \sim \mathcal{N}(\tau_j, \sigma^2/N_j)$ for $j \in \{1, 2\}$ and that the DGP variance σ^2 is nonzero. Then, evaluating the data-driven decision using estimated effects $f(\mathbf{T}^*(\hat{\tau}_{1:2}), \hat{\tau}_{1:2})$ overestimates the actual value of the decision $f(\mathbf{T}^*(\hat{\tau}_{1:2}), \tau_{1:2})$ in expectation:

$$WC := \mathbb{E}\left[f(\mathbf{T}^{*}(\hat{\tau}_{1:2}), \hat{\tau}_{1:2}) - f(\mathbf{T}^{*}(\hat{\tau}_{1:2}), \tau_{1:2})\right] = \sigma_{s}\varphi(\frac{\Delta\tau}{\sigma_{s}}) > 0,$$
(3)

where $\sigma_s = \sqrt{\sigma^2/N_1 + \sigma^2/N_2}$ and $\Delta \tau = \tau_2 - \tau_1$.

Proof of Proposition 1 Let $\xi_j = \hat{\tau}_j - \tau_j$ denote the estimation error of the causal effect of treatment a_j . Then the winner's curse can be written as a function of estimation errors and the datadriven treatment decision:

$$WC = \mathbb{E}[\xi_1 \mathbb{1}\{\mathbf{T}^*(\hat{\tau}_{1:2}) = a_1\}] + \mathbb{E}[\xi_2 \mathbb{1}\{\mathbf{T}^*(\hat{\tau}_{1:2}) = a_2\}]$$
$$= \mathbb{E}[\xi_1 \mathbb{1}\{\hat{\tau}_1 \ge \hat{\tau}_2\}] + \mathbb{E}[\xi_2 \mathbb{1}\{\hat{\tau}_1 < \hat{\tau}_2\}].$$

Because only one treatment out of the two treatments can be chosen, the two treatment indicators must sum to one, i.e., $\mathbb{1}\{\hat{\tau}_1 \geq \hat{\tau}_2\} + \mathbb{1}\{\hat{\tau}_1 < \hat{\tau}_2\} = 1$. Furthermore, ξ_j have zero mean because of unbiasedness, so the winner's curse is:

$$WC = \mathbf{E}[(\xi_1 - \xi_2)\mathbb{1}\{\xi_1 - \xi_2 \ge \tau_2 - \tau_1\}].$$

Let $\xi := \xi_1 - \xi_2$, which follows $\mathcal{N}(0, \sigma_s^2)$ with sampling variance $\sigma_s^2 = \sigma^2 / N_1 + \sigma^2 / N_2$. Then, the winner's curse satisfies:

$$WC = \sigma_s \int_{\frac{\tau_2 - \tau_1}{\sigma_s}}^{\infty} z\varphi(z) dz = \sigma_s \varphi(\frac{\Delta \tau}{\sigma_s}) > 0.$$

Q.E.D.

Proposition 1 shows that even with unbiasedness and \sqrt{N} -consistent estimators, which are the best properties of a statistical estimator one can get in practice, we still overestimate the actual targeting value on average. In addition, the magnitude of winner's curse depends on the sampling variance σ_s^2 and the difference between the two treatments.

PROPOSITION 2. The magnitude of winner's curse in a single-segment targeting problem is the largest when the two treatment effects are the same. It decreases as the difference $|\Delta \tau|$ increases. The magnitude is also an increasing function of the sampling variance σ_s^2 while fixing other variables.

Proof of Theorem 2 First we show winner's curse increases with the sampling variance.

$$\frac{dWC}{d\sigma_s} = \varphi\left(\frac{\Delta\tau}{\sigma_s}\right) + \left(\frac{\Delta\tau}{\sigma_s}\right)^2 \varphi\left(\frac{\Delta\tau}{\sigma_s}\right) \ge 0.$$

Next, we show winner's curse decreases with the treatment effect difference $|\Delta \tau|$.

$$\frac{dWC}{d\Delta\tau} = -\frac{\Delta\tau}{\sigma_s}\varphi\left(\frac{\Delta\tau}{\sigma_s}\right).$$

When $\Delta \tau \leq 0$, $\frac{dWC}{d\Delta \tau} \geq 0$; When $\Delta \tau > 0$, $\frac{dWC}{d\Delta \tau} < 0$. That is, the magnitude of winner's curse is the largest when the two treatment effects are the same, i.e., $\Delta \tau = 0$. As the difference between the two treatments $|\Delta \tau|$ increases, winner's curse decreases.

Q.E.D.

Theorem 2 summarizes two major factors affecting the magnitude of winner's curse. First, when the causal effects of the two treatment arms get closer, it is increasingly hard to identify which one is larger and we are more likely to choose the wrong treatment. That is, the selection is harder when the two treatments are similar. Second, when the sampling variance is large, it is also difficult to identify which treatment is the better one. In addition, the magnitude of estimation error is also larger, leading to bigger winner's curse. Furthermore, the sampling variance σ_s^2 is affected not only by the noise level σ^2 in the DGP, but also by the balanced-ness of the dataset. Specifically, σ_s^2 is the smallest when $N_1 = N_2 = N/2$, i.e., when the dataset is balanced. The more imbalanced the dataset, the higher the sampling variance σ_s^2 and the winner's curse.

2.2. Winner's Curse of General Inference-then-Optimize

In this section, we consider winner's curse in a more general framework of data-driven decision making, namely inference-then-optimize. Let \mathcal{T} denote the space of candidate decisions (treatments) and Θ the space of demand parameters. In the single-segment targeting example, the treatment space is $\mathcal{T} = \{a_1, a_2\}$ and the parameter space is $\Theta = \mathbb{R}^2$. The objective function f is a mapping from $\mathcal{T} \times \Theta$ to the real line \mathbb{R} and the optimal decision operator \mathbf{T}^* is a measurable function from the parameter space \mathcal{T} .

Under the inference-then-optimize framework, we first establish an estimator from data and conduct optimization based on the estimator. Let $\theta_0 \in \Theta$ denote the true parameter of interest and $\hat{\theta} \in \Theta$ the estimator constructed from data. For example, $\theta_0 = (\tau_1, \tau_2)$ and $\hat{\theta} = (\hat{\tau}_1, \hat{\tau}_2)$ in the single-segment targeting problem. The data-driven decision comes from solving some optimization of the following form:

$$\mathbf{T}^*(\hat{\theta}) \in \underset{T \in \mathcal{T}}{\operatorname{arg\,max}} \left\{ f(T, \hat{\theta}) \mid h(T, \hat{\theta}) \ge 0, \ g(T, \hat{\theta}) = 0 \right\},$$

where h, g are some constraints the decision needs to satisfy. For example, in A/B testing, we only choose the experiments with significant results; in search engine marketing, we have a budget limit on bidding; in personalized pricing, we have an upper bound on the price we can set.

Winner's curse describes the average discrepancy $f(\mathbf{T}^*(\hat{\theta}), \hat{\theta}) - f(\mathbf{T}^*(\hat{\theta}), \theta_0)$ of evaluating the data-driven decision using estimated against actual parameters. Note that $\mathbf{T}^*(\hat{\theta})$ is a measurable function from the underlying probability space (Ω, \mathcal{F}, P) to the decision space \mathcal{T} . Let \mathcal{B} denote the σ -algebra on \mathcal{T} and μ denote the pushforward measure on $(\mathcal{T}, \mathcal{B})$ induced by $\mathbf{T}^*(\hat{\theta})$. Thus, we can write the winner's curse as the following.

THEOREM 1. Assume that the objective function $f: \mathcal{T} \times \Theta \mapsto \mathbb{R}$ is twice differentiable w.r.t. the second argument, then the winner's curse is

$$WC = E\left[f(\mathbf{T}^{*}(\hat{\theta}), \hat{\theta}) - f(\mathbf{T}^{*}(\hat{\theta}), \theta_{0})\right]$$

$$= \int_{t \in \mathcal{T}} \nabla_{\theta} f(t, \theta_{0})^{T} E\left[\hat{\theta} - \theta_{0} \mid \mathbf{T}^{*}(\hat{\theta}) = t\right] \mu(dt)$$

$$+ \frac{1}{2} \int_{t \in \mathcal{T}} E\left[(\hat{\theta} - \theta_{0})^{T} \nabla_{\theta}^{2} f(t, \theta_{0})(\hat{\theta} - \theta_{0}) \mid \mathbf{T}^{*}(\hat{\theta}) = t\right] \mu(dt)$$

$$+ O(E[\|\hat{\theta} - \theta\|^{3}]).$$

$$(4)$$

If the estimator $\hat{\theta}$ has negligible third or higher-order moments and

$$\nabla_{\theta} f(t,\theta_0)^T \mathbf{E} \left[\hat{\theta} - \theta_0 \mid \mathbf{T}^*(\hat{\theta}) = t \right] \ge -\frac{1}{2} \mathbf{E} \left[(\hat{\theta} - \theta_0)^T \nabla_{\theta}^2 f(t,\theta_0) (\hat{\theta} - \theta_0) \mid \mathbf{T}^*(\hat{\theta}) = t \right], \tag{5}$$

for μ -almost every $t \in \mathcal{T}$. Then, we have nonnegative winner's curse $WC \geq 0$.

Proof of Theorem 1 Let $wc(\hat{\theta}) := f(\mathbf{T}^*(\hat{\theta}), \hat{\theta}) - f(\mathbf{T}^*(\hat{\theta}), \theta_0)$, which is the discrepancy from one realization. Applying the tower property to the winner's curse we have:

$$WC = \int_{t \in \mathcal{T}} \mathbf{E}\left[wc(\hat{\theta}) | \mathbf{T}^*(\hat{\theta}) = t\right] \mu(dt).$$

Expanding the estimated policy value $f(t, \hat{\theta})$ on the second argument around the true parameter value θ_0 , the conditional expectation becomes:

$$\begin{split} \mathbf{E}[wc(\hat{\theta})|\mathbf{T}^{*}(\hat{\theta}) = t] = &\nabla_{\theta}f(t,\theta_{0})^{T}\mathbf{E}[\hat{\theta} - \theta_{0}|\mathbf{T}^{*}(\hat{\theta}) = t] \\ &+ \frac{1}{2}\mathbf{E}\left[(\hat{\theta} - \theta_{0})^{T}\nabla_{\theta}^{2}f(t,\theta_{0})(\hat{\theta} - \theta_{0})\right] + O(E[\|\hat{\theta} - \theta_{0}\|^{3}|\mathbf{T}^{*}(\hat{\theta}) = t]). \end{split}$$

Plugging the conditional expectation back to the integral of WC, we have (4). In addition, a sufficient condition for WC to be nonnegative is the conditional expectation $E[wc(\hat{\theta})|\mathbf{T}^*(\hat{\theta}) = t]$ being nonnegative a.e. t. Under negligible higher-order moments assumption, the sufficient condition is:

$$\nabla_{\theta} f(t,\theta_0)^T \mathbf{E} \left[\hat{\theta} - \theta_0 \mid \mathbf{T}^*(\hat{\theta}) = t \right] \ge -\frac{1}{2} \mathbf{E} \left[(\hat{\theta} - \theta_0)^T \nabla_{\theta}^2 f(t,\theta_0) (\hat{\theta} - \theta_0) \mid \mathbf{T}^*(\hat{\theta}) = t \right],$$

holds for μ -almost every $t \in \mathcal{T}$.

Q.E.D.

Theorem 1 provides a sufficient condition for a nonnegative winner's curse. Eq 4 shows that the winner's curse depends on the conditional bias and conditional estimation variance scaled by the curvature of the objective function w.r.t. the parameter. For example, if the objective function has increasing margin (positive curvature) over θ , such as quadratic treatment effects, it will amplify the impact of estimation variance and increase winner's curse.

In many applications we have a linear objective function. For example in targeting, the objective is a summation over the causal effects of the treated individuals. In A/B testing, the objective is the total causal effects of all the significant tests. Because the curvature is zero for a linear function, the sufficient condition for a nonnegative winner's curse is:

$$\nabla_{\theta} f(t,\theta_0)^T \mathbf{E} \left[\hat{\theta} - \theta_0 \ \middle| \ \mathbf{T}^*(\hat{\theta}) = t \right] \ge 0, \tag{6}$$

for μ -almost every $t \in \mathcal{T}$. Consider an A/B testing scenario where we have two experiments with true effects $\theta_0 = (\theta_{01}, \theta_{02})$ and we want to choose the one with a higher estimated effect to launch. Eq 6 is then $E[\hat{\theta}_1 - \theta_{01}|\hat{\theta}_1 \ge \hat{\theta}_2]$ if the first experiment is chosen and $E[\hat{\theta}_2 - \theta_{02}|\hat{\theta}_1 \le \hat{\theta}_2]$ otherwise. Lemma 1 shows that even if the estimators $\hat{\theta}$ are unbiased, conditioning on the fact that an experiment is chosen, its conditional bias is no longer zero. Thus, the sufficient condition is satisfied and we have a nonnegative winner's curse. LEMMA 1. Assume that we have independent estimators $\hat{\theta}_j \sim \mathcal{N}(\theta_{0j}, \sigma_j^2)$ for j = 1, 2. Then the bias of an estimator conditioning on it being selected is:

$$\mathbf{E}[\hat{\theta}_1 - \theta_{01} | \hat{\theta}_1 \ge \hat{\theta}_2] = \frac{\sigma_1^2}{\sigma_s} \frac{\varphi(\frac{\theta_1 - \theta_2}{\sigma_s})}{\Phi(\frac{\theta_1 - \theta_2}{\sigma_s})} > 0,$$

where $\sigma_s = \sqrt{\sigma_1^2 + \sigma_2^2}$, φ, Φ are the PDF and CDF of a standard normal distribution.

Proof of Theorem 1 Let $\xi_j = \hat{\theta}_j - \theta_{0j}$ and $\Delta \theta_0 = \theta_{01} - \theta_{02}$. Then, we have:

$$\begin{split} \mathbf{E}[\hat{\theta}_{1} - \theta_{01} | \hat{\theta}_{1} \geq \hat{\theta}_{2}] &= \mathbf{E}[\xi_{1} | \xi_{1} - \xi_{2} \geq -\Delta \theta_{0}] \\ &= \mathbf{E} \left\{ \mathbf{E}[\xi_{1} | \xi_{1} - \xi_{2}] | \xi_{1} - \xi_{2} \geq -\Delta \theta_{0} \right\} \end{split}$$

Note that because ξ_1 and $\xi_1 - \xi_2$ are jointly normal, the conditional expectation $E[\xi_1|\xi_1 - \xi_2]$ is linear and it is:

$$\mathbf{E}[\xi_1|\xi_1-\xi_2] = \frac{Cov(\xi_1,\xi_1-\xi_2)}{Var(\xi_1-\xi_2)}(\xi_1-\xi_2) = \frac{\sigma_1^2}{\sigma_s^2}(\xi_1-\xi_2).$$

Plugging back we have:

$$\mathbf{E}[\hat{\theta}_1 - \theta_{01}|\hat{\theta}_1 \ge \hat{\theta}_2] = \frac{\sigma_1^2}{\sigma_s^2} \mathbf{E}\left\{\xi_1 - \xi_2|\xi_1 - \xi_2 \ge -\Delta\theta_0\right\} = \frac{\sigma_1^2}{\sigma_s} \frac{\varphi(\frac{\Delta\theta}{\sigma_s})}{\Phi(\frac{\Delta\theta}{\sigma_s})}.$$

Q.E.D.

Lemma 1 conveys an important message: conditioning on selection, the distribution of estimators change. Thus, an unbiased estimator could become biased if we know that it is selected among other unbiased estimators. Although the precise impact of such bias on policy evaluation depends on the functional form of the objective function (see Eq 4), the takeaway is that we need to account for such biasedness to remove winner's curse.

3. Correcting the Winner's Curse

We showcase different methods for correcting the winner's curse in this section. We start from three major categories of remedies in the literature: sample splitting, Bayesian estimation, and selective inference. Then, we propose a bootstrap-based correction estimator that is easy to implement and lean in assumptions.

3.1. Sampling Splitting

The winner's curse comes from optimizing and evaluating using the same dataset, so sample splitting divides the data into separate subsets, with one portion for inference-then-optimize and the other for policy evaluation. Sample splitting does not suffer from winner's curse because we are using a hold-out sample for policy evaluation and it is easy to implement. However, splitting the dataset reduces data efficiency, which has two consequences. First, the quality of the data-driven policy is deteriorated, as we only use half of the data for estimation. Second, the variance of policy evaluation is large, hence the policy value estimate will be less reliable. The inefficiency problem is more prominent when we have a small dataset, which is exactly when the winner's curse is significant.

3.2. Bayesian/Shrinkage Estimators

Bayesian estimators solve the inference-then-optimize problem and then evaluate the data-driven policy using posterior mean of the demand parameters. Because the posterior mean naturally shrinks the Maximum Likelihood Estimation (MLE) results, Bayesian estimators can reduce winner's curse. Consider an A/B testing application where the treatment effect estimators $\hat{\theta}_j \in$ $\mathcal{N}(\theta_{0j}, \sigma_j^2)$ for $j = 1, \ldots, J$ experiments and the true mean follows the prior $\theta_{0j} \sim \mathcal{N}(m, s^2)$. Then the posterior mean of experiment j is:

$$\mathbf{E}_{post}\left(\hat{\theta}_{j}^{Bayes}\right) = \frac{s^{2}}{s^{2} + \sigma_{j}^{2}}\hat{\theta}_{j} + \frac{\sigma_{j}^{2}}{s^{2} + \sigma_{j}^{2}}m.$$
(7)

That is, the posterior mean shrinks the MLE estimates towards the prior mean, an example of which is the standard normal $\mathcal{N}(0,1)$. Notice that if we are uncertain about the MLE estimates, i.e., the variance σ_j^2 is large, the weights on the prior mean in Eq (7) will be larger and hence stronger shrinkage. Because of this shrinkage structure, the choice of prior parameters play a significant role in the performance of Bayesian estimators. If mis-specified, Bayesian estimators will either fail to reduce winner's curse or over-correct.

Empirical Bayesian methods provide an alterative to alleviate the critical reliance on prior specification. Empirical Bayes estimators either specify a functional form for prior distribution and estimate its parameters, or estimate the entire prior distribution nonparametrically. A common used prior model is the spike-and-slab prior in Eq 8, which contains an active component for no effects (spike) and and inactive component for nonzero effects (slab).

$$\theta_{0j} \sim \pi \delta_0 + (1 - \pi) \mathcal{N}(m, s^2), \tag{8}$$

where δ_0 is a Dirac function concentrated at zero (the spike) and π is the prior probability that there is no effect. The posterior mean in this case is then:

$$\mathbf{E}_{post}\left(\hat{\theta}_{j}^{EB}\right) = (1-\pi) \left[\frac{s^{2}}{s^{2}+\sigma_{j}^{2}}\hat{\theta}_{j} + \frac{\sigma_{j}^{2}}{s^{2}+\sigma_{j}^{2}}m\right].$$

Although the functional form is similar to the normal-prior-Bayesian estimator (7), the parameters (m, s) are estimated from data, hence, it provides more accurate shrinkage as we will see in later sections.

Empirical Bayesian provides an efficient shrinkage estimator for the demand parameters but the amount of reduction in winner's curse is not guaranteed. Because Bayes estimators are optimizationagnostic, its performance varies when the optimization problem differs. In the next section, we will show a variant of Bayes methods where the posterior distribution is conditioned on the optimization. Another drawback of empirical Bayes methods is its computational complexity. Solving empirical Bayes with spike-and-slab prior (8) needs using Expectation-Maximization algorithm, which could be computationally cumbersome, and it only gets worse when we have more complicated models.

3.3. Selective Inference

Selective inference, also known as post-selection inference, explicitly models how the selection (or optimization) process impacts a selected estimator and corrects for it. (Andrews et al. 2024) provides a median-unbiased estimator for the winner of several treatments or strategies. For example, the message with the highest lift in subscription rate among multiple candidate messages in advertising; The feature with the highest treatment effect among multiple randomized control trials.

Consider a set of candidate treatments $\mathcal{T} = \{a_1, \ldots, a_J\}$. For each option, we observe an estimate $\hat{\tau}_j$ assumed to be normally distributed: $\hat{\tau}_j \sim \mathcal{N}(\tau_{0j}, \sigma_j^2)$ The "winner" is defined as the option with the largest observed estimate:

$$j^* \in \operatorname*{arg\,max}_{j \in \{1, \dots, J\}} \hat{\tau}_j.$$

Due to the selection based on estimated effects $\hat{\tau}_j$'s, the observed value $\hat{\tau}_{j^*}$ is biased upward relative to the true effect τ_{j^*} .

To correct for the bias introduced by the selection, we condition on the event that $X(\hat{\theta})$ is the maximum. Let $L = \max_{j \neq j^*} \hat{\tau}_j$ denote the highest estimate among the non-selected options. Consequently, the distribution of $\hat{\tau}_{j^*}$ given $\hat{\tau}_{j^*} \ge L$ is that of a normal variable truncated below at L. Define the CDF of a normal distribution with mean μ and variance σ^2 truncated below at L as $F_{TN}(x;\mu,\sigma,L)$. To obtain a median-unbiased estimator for $\hat{\tau}_{j^*}$, we invert the median condition:

$$F_{TN}(\hat{\tau}_{i^*};\mu,\sigma_{i^*},L) = 0.5. \tag{9}$$

The unique solution $\hat{\mu}$ to this equation serves as the median-unbiased estimator for the true effect τ_{j^*} .

Because the selection process favors extreme values, the naive estimator $\hat{\tau}_{j^*}$ overstates the true effect. By considering the truncated normal distribution—reflecting that $\hat{\tau}_{j^*}$ is observed only when it exceeds *L*—the inversion corrects for the upward bias. Solving Eq (9) yields an estimator whose median equals τ_{j^*} , thus effectively countering the winner's curse. Bayesian selective inference follows a similar logic and condition the posterior of demand distributions on the selection event. Consider again the Bayesian estimators discussed in the last section, where the likelihood model for the demand estimator is $\hat{\theta}_j \sim \mathcal{N}(\theta_{0j}, \sigma_j^2)$ and the prior for θ_{0j} could either be standard normal of a spike-and-slab model estimated from data. Let S denote the selection event. Then the posterior model is:

$$f(\theta_{0j}|S,\hat{\theta}_j) \propto \frac{f(\hat{\theta}_j|\theta_{0j})f(\theta_{0j})}{\Pr(S|\theta_{0j})},\tag{10}$$

where the numerator is the original posterior distribution without conditioning on selection. The formulation Eq 10 assumes that the prior is not affected by selection, i.e., $f(\theta_{0j}|S) = f(\theta_{0j})$, which fits into many practical scenarios such as when choosing a treatment does not change its true effect.¹ $Pr(S|\theta_{0j})$ is the probability of selection given the prior parameters. For example, the probability of treatment a_j has the largest estimated effects among all treatments, or the probability that an A/B test has a positively significant result.

Selective inference typically enjoys theoretical guarantees such as unbiasedness. It not only provides unbiased point estimates, but also generates valid confidence intervals. However, because selective inference conditions for problem-specific selection rules, its estimators are not typically generalizable. For example, (Andrews et al. 2024) corrects for selecting the largest effect, (Andrews et al. 2022) corrects for selecting the k-largest effect, and (Andrews et al. 2021) corrects for selecting the largest absolute values. In the Bayesian selective inference example, the probability of selection $\Pr(S|\theta_{0j})$ is different for different optimization problems. Thus, selective inference methods can be mathematically intensive.

3.4. Bootstrap Correction

In this section, we introduce a bootstrap correction method for removing winner's curse, which is easy to implement and generally applicable. When we know that a statistic is bias, a natural way to debias it is subtracting the bias from it. Thus, the bias correction method for removing winner's curse is the following:

$$V(\mathbf{T}^*(\hat{\theta})) = f(\mathbf{T}^*(\hat{\theta}), \hat{\theta}) - WC$$
(11)

By taking expectation, we see that the correction estimator is unbiased: when averaging over all realizations of the training dataset \mathcal{D} , the error of approximating the true value of the data-driven policy $\mathbf{T}^*(\hat{\theta})$ with the correction estimator (11) is zero:

$$\mathbf{E}[V(\mathbf{T}^*(\hat{\theta}))] = \mathbf{E}[f(\mathbf{T}^*(\hat{\theta}), \theta_0)]$$

Note that when doing correction, we treat the data-driven decision as given, regardless of whether it is the "correct" decision. Our goal is to accurately estimate its value such that our estimation error is zero on average. We propose using bootstrap to estimate the winner's curse bias. Bootstrap (Efron (1979), Hall (1992)) is a family of resampling methods commonly used to approximate the limiting distribution of a statistic. Recall from Eq (4) that winner's curse occurs when we evaluate the data-driven decision $\mathbf{T}^*(\hat{\theta})$ using sample statistic $\hat{\theta}$ against population statistic θ_0 . Bootstrap approximates such discrepancy between sample and population statistics by resampling from the original dataset with replacement to form bootstrap samples. We first estimate the treatment effects and construct the bootstrap decision from the bootstrap samples, denoted as $\hat{\theta}^b$ and $\mathbf{T}^*(\hat{\theta}^b)$ respectively. Then, we can approximate the actual winner's curse by the discrepancy when we evaluate the bootstrap decision under the bootstrap estimate $\hat{\theta}^b$ and the sample estimate $\hat{\theta}$. Formally, let $b \in \{1, \ldots, B\}$ denote the index of B bootstrap samples. The bootstrap estimate for the winner's curse (4) is:

$$\hat{WC} = \frac{1}{B} \sum_{b=1}^{B} f(\mathbf{T}^{*}(\hat{\theta}^{b}), \hat{\theta}^{b}) - f(\mathbf{T}^{*}(\hat{\theta}^{b}), \hat{\theta})$$
(12)

The bootstrap correction estimator is computed by plugging the bootstrap-estimated winner's curse back into the bias correction formula:

$$\hat{V}(\mathbf{T}^*(\hat{\theta})) = f(\mathbf{T}^*(\hat{\theta}), \hat{\theta}) - \hat{WC}$$
(13)

Algorithm 1 summarizes the bias correction procedure. Note that we use a standard nonparametric bootstrap in this algorithm, meaning that when bootstrapping, we sample with replacement N observations from the original dataset \mathcal{D} . That is, each bootstrap sample has the same number of observations as the original dataset. Standard bootstrap is robust and does not have any hyperparameters to tune.

Validity of the bootstrap correction estimator (13) hinges on the convergence of standard nonparametric bootstrap. There are three conditions (Van der Vaart 2000) required for standard bootstrap to converge at the ideal rate. For the sake of simplicity, we remove the subscripts for the treatments temporally. The first condition is that the estimator for the treatment effect itself needs to be consistent, i.e., $\hat{\theta} \xrightarrow{p} \theta_0$, which is satisfied by most estimators seen in practice if used appropriately. The second condition is that the bootstrap estimator $\hat{\theta}^b$ converges conditionally in distribution to the same distribution as the original estimator $\hat{\theta}$. For example, if $\sqrt{N}(\hat{\theta} - \theta_0) \xrightarrow{d} T$ where T is the asymptotic distribution such as a mean-zero normal, then we must have $\sqrt{N}(\hat{\theta}^b - \hat{\theta}) \xrightarrow{d} T$ conditional on the dataset \mathcal{D} . This condition is satisfied by the standard bootstrap (Van der Vaart and Wellner 1996) and many other bootstrap methods. The last condition is some level of smoothness of the objective function f. Smoothness is necessary because when we use convergence of the demand parameters to build convergence of the winner's curse via (functional) Delta method, we cannot have jumps in the objective function f. The critical role of smoothness for the convergence of standard bootstrap is summarized Lemma 2.

Algorithm 1 Standard Bootstrap Correction

Require: Dataset \mathcal{D} with N observations; Number of bootstraps B.

- 1: Calculate sample estimate $\hat{\theta}$ from \mathcal{D}
- 2: Optimize for the data-driven decision $\mathbf{T}^*(\hat{\theta})$
- 3: Evaluate the decision using sample estimate $\hat{\theta}$ and get $f(\mathbf{T}^*(\hat{\theta}), \hat{\theta})$
- 4: for b = 1, ..., B do
- 5: Draw N observations with replacement from \mathcal{D} as a bootstrap sample \mathcal{D}^b
- 6: Calculate bootstrap estimate $\hat{\theta}^b$ from \mathcal{D}^b
- 7: Optimize for the bootstrap decision $\mathbf{T}^*(\hat{\theta}^b)$
- 8: Evaluate the bootstrap decision using bootstrap estimate $\hat{\theta}^b$ and get $f(\mathbf{T}^*(\hat{\theta}^b), \hat{\theta}^b)$
- 9: Evaluate the bootstrap decision using sample estimate $\hat{\theta}$ and get $f(\mathbf{T}^*(\hat{\theta}^b), \hat{\theta})$

10: end for

- 11: Calculate winner's curse estimate $\hat{WC} = \frac{1}{B} \sum_{b=1}^{B} f(\mathbf{T}^*(\hat{\theta}^b), \hat{\theta}^b) f(\mathbf{T}^*(\hat{\theta}^b), \hat{\theta})$
- 12: Calculate bootstrap-corrected policy value estimate $\hat{V}^{standard}(\mathbf{T}^*(\hat{\theta})) = f(\mathbf{T}^*(\hat{\theta}), \hat{\theta}) \hat{WC}$
- 13: return $\hat{V}^{standard}(\mathbf{T}^*(\hat{\theta}))$

LEMMA 2 (Theorem 3.1, Fang and Santos (2019)). Given an asymptotic normal estimator $r_N(\hat{\theta} - \theta_0) \xrightarrow{d} \mathcal{N}(0, \sigma^2)$ for some convergence rate $r_N \to \infty$ and a function f mapping from the parameter space to the real line that is Hadamard directional differentiable at τ , assuming that a bootstrap estimator $\hat{\theta}^b$ satisfies $r_N(\hat{\theta}^b - \hat{\theta}) \xrightarrow{d} \mathcal{N}(0, \sigma^2)$, i.e., weakly converging conditional on the empirical distribution over the dataset \mathcal{D} , then

$$r_N(f(\hat{\theta}^b) - f(\hat{\theta})) \xrightarrow{d} f'_{\theta_0}(\mathcal{N}(0, \sigma^2))$$

conditional on the empirical distribution if and only if f is fully Hadamard differentiable at τ .

However, winner's curse often violates the smoothness condition. Notice that the optimal decision operator $T^*(\cdot)$ is an argmax function of the demand parameters and the decision space \mathcal{T} is discrete in many scenarios like selecting the best treatment, so the data-driven decision is non-differentiable. For instance, if the treatment effect estimates vary slightly, the optimal treatment assignment may change completely. Thus, the winner's curse WC is a nonsmooth function of the demand parameters θ . As a result, the winner's curse estimator \hat{WC} calculated via standard bootstrap will converge slower than the estimators $\hat{\theta}$. Consequently, using standard bootstrap can only partially mitigate winner's curse in finite-sample regime.

A natural solution for nonsmoothness is by applying some smoothing techniques. We propose using two major categories of alternative bootstrap methods: subsampling and perturbation. *m*- out-of-*n* bootstrap (Bickel et al. (1997), Bickel and Sakov (2008), etc.) is an important and easyto-implement subsampling method. Instead of drawing bootstrap samples of the same size N as the original dataset \mathcal{D} , we draw bootstrap samples with size m < N. Convergence of *m*-out-of-*n* bootstrap requires the bootstrap sample size m satisfies m = o(N), i.e., m is a slower-growing function than N itself. One typical rule for calculating m that satisfies this condition is m = $int(N^{pow})$ given $pow \in (0, 1)$.

Numerical bootstrap (Hong and Li 2020) offers another perspective into smoothing. First, recall that we have winner's curse because we are approximating the actual performance of the plugin decision using estimated effects. To capture that discrepancy, we can rewrite the targeting value estimate $f(\mathbf{T}^*(\hat{\theta}), \hat{\theta})$ as the following:

$$f\left(\mathbf{T}^{*}(\hat{\theta}), \theta_{0} + \frac{1}{\sqrt{N}}\sqrt{N}(\hat{\theta} - \theta_{0})\right)$$

Applying the bootstrap principle and replacing $1/\sqrt{N}$ with an adjustable hyperparameter ϵ_N , we have:

$$f\left(\mathbf{T}^{*}(\hat{\theta}^{b}), \hat{\theta} + \epsilon_{N}\sqrt{N}(\hat{\theta}^{b} - \hat{\theta})\right)$$

The purpose of the hyperparameter ϵ_N is to control the randomness introduced in the bootstrap process. Notice that instead of evaluating the bootstrap decisions $\mathbf{T}^*(\hat{\theta}^b)$ using bootstrap treatment effect estimates $\hat{\theta}^b$ as in standard bootstrap, we use perturbed treatment effect estimates $\hat{\theta}^P = \hat{\theta} + \epsilon_N \sqrt{N}(\hat{\theta}^b - \hat{\theta})$ for evaluation. Convergence of numerical bootstrap requires the hyperparameter satisfying $\epsilon_N \sqrt{N} \to \infty$ and $\epsilon_N \downarrow 0$ as $N \to \infty$. A typical rule for determining ϵ_N that satisfies the conditions is $\epsilon_N = N^{pow}$ given $pow \in (-0.5, 0)$. Algorithm 2 illustrates the procedure for calculating a numerical bootstrap-based correction estimator.

4. Monte Carlo Simulation: A/B Testing

Using A/B tests to evaluate the impact of new features is an important practice across a wide range of online platforms, including e-commerce, social media, online learning, etc. When we select features based on estimated effects, we suffer from winner's curse, i.e., we overestimate the total effect of the selected features. We demonstrate the existence of winner's curse using Monte Carlo simulations and compare the performance of various correction methods.

The simulation is set up as the following. Consider J A/B tests, whose true effects τ_j are drawn from a normal distribution $\tau_j \sim \mathcal{N}(m, s^2)$ for $j = 1, \ldots, J$. After the true effects are drawn, they are fixed throughout the experiments. Each A/B test has N responses in the treated and control respectively. We assume that the response is a continuous variable, such as time-spent on the platform, following normal distribution $Y_i \sim \mathcal{N}(\tau_j, \sigma^2)$ for $i = 1, \ldots, N$.

Algorithm 2 Numerical Bootstrap Correction

Require: Dataset \mathcal{D} with N observations; Number of bootstraps B; Hyperparameter ϵ_N

- 1: Calculate sample estimate $\hat{\theta}$ from \mathcal{D}
- 2: Optimize for the data-driven decision $\mathbf{T}^*(\hat{\theta})$
- 3: Evaluate the decision using sample estimate $\hat{\theta}$ and get $f(\mathbf{T}^*(\hat{\theta}), \hat{\theta})$
- 4: for b = 1, ..., B do
- 5: Draw N observations with replacement from \mathcal{D} as a bootstrap sample \mathcal{D}^b
- 6: Calculate bootstrap estimate $\hat{\theta}^b$ from \mathcal{D}^b
- 7: Calculate perturbation $\hat{\theta}^{P,b} = \hat{\theta} + \epsilon_N \sqrt{N}(\hat{\theta}^b \hat{\theta})$
- 8: Optimize for the bootstrap decision $\mathbf{T}^*(\hat{\theta}^b)$
- 9: Evaluate the bootstrap decision using perturbed estimate $\hat{\theta}^{P,b}$ and get $f(\mathbf{T}^*(\hat{\theta}^b), \hat{\theta}^{P,b})$
- 10: Evaluate the bootstrap decision using sample estimates $\hat{\theta}$ and get $f(\mathbf{T}^*(\hat{\theta}^b), \hat{\theta})$

11: end for

- 12: Calculate winner's curse estimate $\hat{WC} = \frac{1}{B} \sum_{b=1}^{B} f(\mathbf{T}^{*}(\hat{\theta}^{b}), \hat{\theta}^{P,b}) f(\mathbf{T}^{*}(\hat{\theta}^{b}), \hat{\theta})$
- 13: Calculate bootstrap-corrected policy value estimate: $\hat{V}^{num}(\mathbf{T}^*(\hat{\theta})) = f(\mathbf{T}^*(\hat{\theta}), \hat{\theta}) \hat{WC}$

14: return $\hat{V}^{num}(\mathbf{T}^*(\hat{\theta}))$

The inference-then-optimize procedure for decision making is setup as the following. We estimate the treatment effect for each experiment using difference in mean. The resulting estimator hence follows a normal distribution $\hat{\tau}_j \sim \mathcal{N}(\tau_j, \sigma^2/N)$. We consider selecting the experiments with positive estimated effects. The corresponding optimization is formulated as the following, where the objective is the average effect of the selected tests.

$$\mathbf{T}^*(\hat{\tau}_{1:J}) \in \underset{T_j \in \{0,1\}}{\operatorname{arg\,max}} f(T_{1:J}, \hat{\tau}_{1:J}) = \frac{1}{J} \sum_{j=1}^J \hat{\tau}_j T_j.$$

We repeat draw R = 500 datasets from the DGP and conduct inference-then-optimize for each of them. The average performance of different estimators are summarized in Figure 1.

We see that without correction, we overestimate the policy performance. Standard bootstrap correction is able to reduce the winner's curse, but only partially. m-out-of-n and numerical bootstrap further removes the winner's curse, but risks slight over-correction. The reason is that the exact convergence speeds of the two components in Eq (13), the objective value and the winner's curse estimate, are different. If the objective value $f(\mathbf{T}^*(\hat{\tau}_{1:J}), \hat{\tau}_{1:J})$ converges faster than \hat{WC} , then we will over-correct slightly. The convergence speed is controlled by the hyperparameters m and ϵ_N in the two methods and we use $m = int(N^{0.95})$ and $\epsilon_N = N^{-0.45}$ throughout the paper.

Empirical Bayes also provide good remedies for winner's curse under the current setting. They provide similar policy value estimates as the bootstrap estimators and are close to the true value.



Figure 1 Estimated Average Value of Selecting A/B Tests with Positive Effects

Note. Red dashed line indicates the true policy value $f(\mathbf{T}^*(\hat{\tau}_{1:J}), \tau_{1:J})$; Blue bars represent the estimates from each method; Red bar on the right is the true value of the sample splitting policy. DGP parameters: prior m = s = 0.1, response noise level $\sigma = 1$, sample size N = 100. Estimators from left to right are: no correction, standard bootstrap correction, m-out-of-n bootstrap correction, numerical bootstrap correction, Bayes with normal prior, empirical Bayes with spike-and-slab prior, selection-adjusted Bayes with normal prior, selection-adjusted Bayes with spike-and-slab prior, and sample splitting.

On the other hand, Bayes estimator with a standard normal prior fails to reduce winner's curse. The reason is that the difference-in-mean estimators have much smaller variance compared to the prior variance As a result, the posterior mean (7) is dominated by the MLE estimates and there is effectively no shrinkage. Thus, empirical Bayes is a better shrinkage method for dealing with winner's curse.

Bayesian selective inference provides a conservative estimates for the policy value with an appropriate prior. Recall that Bayesian selective inference adjust the posterior distribution to the selection. So if the posterior itself is bad, adjustment cannot help, such as the case where we use standard normal distribution as the prior. The selection event in this scenario is choosing the positive effects, which has a probability of $Pr(\hat{\tau}_j > 0|\tau_j)$. After adjusting for it, the posterior effect becomes conservative, and hence the policy value estimate.

Lastly, sample splitting provides an unbiased estimate for its own policy. Because we use only half of the sample for learning the policy, it is different from the focal policy learned from the entire dataset. Because of the independence between the subsets of data, the policy value estimate is unbiased. However, comparing the red bar of sample splitting in Figure 1, we see that the average quality of the sample splitting policy is lower. In addition, the sample splitting estimate has a higher variance compared with other estimators like bootstrap. As we will show later, the lower-quality-decision issue could be more prominent in other settings.

Next, we evaluate the winner's curse magnitude of different estimators when we change the prior mean of the true effects (Figure 2). Because the true effects have an average of m, which translates to the amount of lift from null, we report the winner's curse as a percentage of it, i.e., WC/m. As the average true effect diminishes, it is harder the identify the A/B tests with a positive effect, and hence, the winner's curse increases. On the other hand, when the average true effect is large enough, there will be no winner's curse as it is easy to identify the positive tests. m-out-of-n and numerical bootstrap estimators show robustness across different level of m, while standard bootstrap can only partially mitigate winner's curse when the average effect is closer to null because the impact of non-smoothness is stronger. Empirical Bayes with a spike-and-slab prior turns conservative when the average effect is small because the spike component will get a higher proportion in the posterior as the estimator could not identify whether a test has a small but positive effect or a null effect.

Figure 2 Selecting A/B Tests with Positive Effects: Winner's Curse vs. Average True Effect



Lastly, we show winner's curse and the performance of different estimators under different selection rules. Figure 3 shows the policy value estimates for selecting the A/B tests with a positive and significant effects, and for selecting the tests with the largest 10% effects among all tests. First, winner's curse is persistent under different selection rules without correction. Second, empirical Bayes with a spike-and-slab prior performs worse because the posterior mean of the selected tests is higher. Last but not least, although sample splitting is unbiased for its own policy, it has a significantly lower quality than the focal policy.



Figure 3 Estimated Average Value of Selection with Positively Significant Effects & the Largest 10% of Effects

5. Monte Carlo Simulation: Personalized Targeting

We show winner's curse in a wide range of simulated targeting problems in this section and compare the performance of our bootstrap correction method against other benchmarks. We start from the single segment targeting problem discussed in Section 2. Then, we illustrate the universality of winner's curse and the generality of our bootstrap correction method in a more complex targeting setting with multiple consumer segments and budget limit. Finally, we consider a continuous segment scenario where we do not have access to the actual functional form of the DGP and can only approximate it via a mis-specified model like causal forest. In this case, we do not have an unbiased estimator and winner's curse is more severe, but we show that our bootstrap correction method can still resolve the overestimation issue.

5.1. Single Segment

In this section, we consider the single segment targeting problem with a continuous response as in (1). We use the simulation procedure summarized in Algorithm 3.

We first compare the winner's curse in a single segment targeting problem before and after corrected by standard bootstrap, the results of which are shown in Figure 4. We report winner's curse as percentages of the treatment effect difference $WC/\Delta\tau$ throughout the discussion. The left plot in Figure 4 shows that we overestimate the actual targeting value on average without correction. Although in some realizations, the estimated targeting value underestimates, potentially because we underestimate the causal effects of both treatments simultaneously, on average we are still over-optimistic. The right plot in Figure 4 shows that when we correct the winner's curse using standard bootstrap, the magnitude of overestimation drops significantly. Without correction, the average winner's curse is 71.49% (12.15%), while after correction with standard bootstrap, the average winner's curse is 18.95% (13.38%). Notably, bootstrap correction reduces the winner's curse to a statistically insignificant level without compromising the shape of the distribution and hugely increasing the standard error.

Algorithm 3 Simulation Procedure for Single Segment Targeting

- **Require:** DGP variance σ^2 ; Sample size N; Baseline treatment effect τ_1 ; Treatment effect difference $\Delta \tau$; Number of repeats R; Number of bootstraps B; m-out-of-n bootstrap parameter m; Numerical bootstrap parameter ϵ_N
- 1: for r = 1, ..., R do
- 2: Draw N customers from $\mathcal{N}(0, \sigma^2)$ and assign treatments (a_1, a_2) with propensity 0.5 to form the training dataset $\mathcal{D}_r = \{(T_i, Y_i)\}_{i=1}^N$ according to (1)
- 3: Estimate treatment effects $\hat{\tau}_{r,1:2}$ and solve (2) for the plugin decision $\mathbf{T}_r^*(\hat{\tau}_{1:2})$
- 4: Evaluate the plugin decision with no correction and get $f_r(\mathbf{T}^*(\hat{\tau}_{1:2}), \hat{\tau}_{1:2})$
- 5: Evaluate the plugin decision with standard, *m*-out-of-*n*, and numerical bootstrap correction as in Algorithm 1 and 2 and get $\hat{V}_r^{standard}(\mathbf{T}^*(\hat{\tau}_{1:2})), \hat{V}_r^{mn}(\mathbf{T}^*(\hat{\tau}_{1:2})), \hat{V}_r^{num}(\mathbf{T}^*(\hat{\tau}_{1:2}))$

6: end for

7: return No-correction and bootstrap-correction estimates



Figure 4 Histogram of Winner's Curse of Single-Segment Targeting

Note. The left-hand-side plot is the histogram of winner's curse without correction across R repeated experiments. The right-hand-side plot adds the histogram of the winner's curse after corrected by standard bootstrap. The average winner's curse of no correction (blue) is 71.49% (12.15%); The average winner's curse of standard bootstrap correction estimator (orange) is 18.95% (13.38%). The simulation parameters are the following: DGP variance $\sigma = 1$; Sample size N = 100; Baseline treatment effect $\tau_1 = 1$; Treatment effect difference $\Delta \tau = 0.1$; Number of repeats R = 5000; Winner's curse reported as percentages of treatment effect difference $WC/\Delta \tau$.

Next, we examine the winner's curse magnitude and the performance of standard bootstrap correction under different DGP parameters. Recall from Theorem 2 that the treatment effect difference $\Delta \tau$ and the sampling variance σ_s affect winner's curse. Thus, we fix the baseline treatment effect $\tau_1 = 1$ and the DGP variance $\sigma = 1$ and change $\Delta \tau$. Table 2 summarizes the results. When the difference between the two treatment arms is profound ($\Delta \tau = 0.5$), there is no winner's curse because it is easy to identify which treatment is a better one. When the difference is moderate, there is significant winner's curse and standard bootstrap can mitigate most of it. When the difference shrinks further to $\Delta \tau = 0.05$, the winner's curse worsens. In this case, standard bootstrap can only partially mitigate the winner's curse.

Table 2 Comparative Statics of Winner's Curse under Single Segment Targeting			
	$\Delta\tau=0.05$	$\Delta\tau=0.1$	$\Delta\tau=0.5$
No Correction	$156.27\%^{***}$	$71.49\%^{***}$	-0.02%
	(23.55%)	(12.15%)	(2.83%)
Standard Bootstran	$46.70\%^{*}$	18.98%	-2.49%
Standard Dootstrap	(26.10%)	(13.38%)	(3.05%)
<i>m</i> -out-of- <i>n</i> Bootstrap	25.74%	4.00%	-3.52%
	(26.17%) $(13.47%)$		(3.11%)
Numerical Bootstrap	18.62%	-0.91%	-3.11%
Tumericai Dooistiap	(26.84%)	(13.94%)	(3.11%)

***, **, and * represents 1%, 5%, and 10% significance level respectively. No stars meaning the value is statistically insignificant.

Standard bootstrap does not fully eliminate winner's curse when the treatment effects are close because the value function $f(\mathbf{T}^*(\cdot), \cdot)$ is non-differentiable when $\Delta \tau = 0$. As discussed in Section 2, nonsmoothness of the value function slows down the convergence of winner's curse estimates. That is, as we move closer to the non-differentiable point, we need more observations to identify which treatment has the higher effect. *m*-out-of-*n* and numerical bootstrap can bypass the smoothness assumption and speed-up the convergence. As a result, they both achieve statistically insignificant winner's curse.

The performance of *m*-out-of-*n* and numerical bootstraps are sensitive to hyperparameters. In the case of *m*-out-of-*n*, a smaller bootstrap sample size *m* means stronger smoothing as we are infusing more noise into the estimator. If the bootstrap sample size *m* approaches the original sample size *N*, *m*-out-of-*n* bootstrap correction will be the same as a standard bootstrap. On the other hand, reducing *m* arbitrarily can potentially lead to over-correction. Similarly, because ϵ_N in numerical bootstrap controls how much noise we introduce, a larger ϵ_N yields stronger smoothing and increasing it could lead to over-correction. Thus, it is important to determine the hyperparameters carefully.

Tuning the hyperparameters depend on the specific targeting problem. Intuitively, the closer we are to the non-differentiable point, the stronger smoothing we need. However, because the optimization landscape of different targeting problems vary drastically, it is challenging to design a fixed rule for determining the hyperparameters. Although there are some heuristics proposed in the

statistics literature (Bickel and Sakov (2008), Chakraborty et al. (2013), etc.), their performance is not theoretically grounded. Thus, we recommend the following rule for choosing the hyperparameters and reporting results. First, standard bootstrap mitigates a significant amount of winner's curse and does not require parameter-tuning, so we should always report the standard bootstrapcorrected estimates. Second, if decided to use *m*-out-of-*n* or numerical bootstraps without following a heuristic parameter-tuning method, one should either fix a set of hyperparameters before analyzing the data and stick to it throughout, or report all results under all tested hyperparameters to avoid *p*-hacking. In this paper, we use $m = int N^{0.95}$ and $\epsilon_N = N^{-0.45}$ for all experiments.

Next, we compare the bootstrap correction estimator with other methods proposed in the litearture in Table 3. In addition to continuous response (1), we also investigate binary response (e.g., purchase decision) modeled either as a Bernoulli random variable or as a logit choice. More specifically, if we model the purchase decision as a Bernoulli random variable, then the purchase probability is $Pr(Y_i = 1) = \tau_1 \mathbb{1}\{T_i = a_1\} + \tau_2 \mathbb{1}\{T_i = a_2\}$. If we model the purchase decision via logit choice model, then the purchase probability is $Pr(Y_i = 1) = \exp(u_i)/(1 + \exp(u_i))$ where utility $u_i = \tau_1 \mathbb{1}\{T_i = a_1\} + \tau_2 \mathbb{1}\{T_i = a_2\}$.

The first set of benchmark is sample splitting and cross validation. The idea of sample splitting is dividing the dataset into two, one for estimation and designing the optimal treatment policy, and the other one for evaluating the optimal policy. The advantage is that the evaluation is unbiased for the treatment policy calculated from the first half of the dataset. However, because we only use half of the dataset for optimization, the resulting policy is worse than the one designed using the entire dataset on average. In addition, sample splitting and cross validation are inefficient in the sense that their standard errors could be over 50% higher than no correction. Bootstrap can be thought as a more efficient way of doing sample splitting, which instead of splitting the dataset into two, we use subsampling to separate policy optimization and policy evaluation. Thus, bootstrap methods enjoy better policy design and smaller estimation variation.

The second set of benchmark is the Bayesian estimators. When using the Bayesian estimators, we follow the cited papers to select the optimal treatment based on frequentist estimates and evaluate it using the posterior mean of the selected treatment. Because Bayesian methods are sensitive to the choice of the prior distribution, we compare against two Bayesian methods, one with a normal prior and the other not requiring prior specification. The first benchmark uses standard Bayesian with a normal prior, which comes from an industry report from Amazon. It barely reduces winner's curse as seen in Table 3. The second benchmark is an empirical Bayes approach (Efron 2011), which relaxes the dependence of Bayesian estimation on a prior distribution but assumes normally distributed observations and requires a smooth density estimation of data. The additional

	Continuous Response	Bernoulli Response	Logit Response
No Correction	71.49%***	39.18%***	335.48%***
No Correction	(12.15%)	(8.99%)	(48.77%)
Standard Bootstran	18.95%	7.95%	$92.82\%^{*}$
Standard Dootstrap	(13.38%)	(9.55%)	(52.88%)
<i>m</i> -out-of- <i>n</i> Bootstrap	4.00%	1.54%	44.34%
m-out-or-n Dootstrap	(13.47%)	(9.55%)	(52.72%)
Numerical Bootstrap	-0.91%	-0.22%	29.88%
Numericai Dootstrap	(13.94%)	(9.72%)	(54.09%)
Empirical Bayes	-11.16%	N / A	N/Δ
(Efron 2011)	(83.22%)	11/11	N/A
Standard Bayes	$66.83\%^{***}$	$38.28\%^{***}$	$332.67\%^{***}$
(Kessler 2024)	(11.93%)	(8.95%)	(48.55%)
Conditional Selective Inference	-426.13%	-72.52%	69.12%
(Andrews et al. 2024)	(708.84%)	(42.40%)	(826.54%)
Sample Splitting	-1.25%	-1.59%	19.40%
Sample Spitting	(20.00%)	(13.70%)	(85.56%)
Cross Validation	-2.14%	-3.42%	70.95%
	(17.47%)	(11.96%)	(104.91%)

 Table 3
 Average Winner's Curse of Single-Segment Targeting

The average winner's curse and standard errors are calculated from R = 5000 repeated experiments; The DGP parameters for continuous response are the same as before: N = 100, $\tau_1 = 1$, $\Delta \tau = 0.1$, and $\sigma = 1$. The DGP parameters for Bernoulli and Logit response are N = 100, $\tau_1 = 0.1$, and $\tau_2 = 0.15$. The hyperparameters for m--out-of-n and numerical bootstraps are $m = int(N^{0.95})$ and $\epsilon_N = N^{-0.45}$ under all scenarios.

restrictions limit the use of their method on binary response scenarios. But even in the continuous response scenario, the method over-corrects and suffers from large standard errors.

Finally, we compare our results with the conditional selective inference method from (Andrews et al. 2024). This method tries to find the location of the largest treatment effect's distribution so that the observed largest effect is median unbiased (i.e., equally likely to over-and under-estimate) when evaluated under the same distribution truncated on the left by the second largest treatment effect. As seen in Table 3, this method has arbitrary winner's curse on average and has the largest standard errors. The main reason is because when the largest two treatment effects are close, this method is infeasible and leads to numerical errors. As a result, across the R = 5000 repeated experiments, whenever the two estimated treatment effects $\hat{\tau}_{1:2}$ are close, the correction estimate is arbitrary. In spite of this, the median winner's curse for the continuous response scenario is merely 3.12%, thanks to the fact that there is only a small number of cases where the two estimated effects are close. An additional limit of (Andrews et al. 2024) is that their normality assumption

could lead to inaccurate correction with binary response cases, where the median winner's curse is -4.19% and 300.31%.

5.2. Targeting with Multiple Segments and Budget Constraints

In this section, we consider a more practical and difficult setting where we have multiple consumer segments and face a budget constraints. First, we assume that there are S discrete consumer segments, of which the segmentation rule is known. We will discuss unknown segmentation in the next section. Consumers in different segments have different treatment effects. Let τ_{js} denote the causal effect of treatment a_j on a consumer from segment s. The data generation process for a consumer belonging to segment s is $y_i = \tau_{1s} \mathbb{1}\{T_i = a_1\} + \tau_{2s} \mathbb{1}\{T_i = a_2\} + \epsilon_i$. A more compact way of writing the DGP is:

$$y_i = \sum_{s=1}^{S} \sum_{j=1}^{2} \tau_{js} \mathbb{1}\{T_i = a_j\} \mathbb{1}\{i \text{ is in Segment } s\} + \epsilon_i$$

where $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$. In simulation, we assume that τ_{js} are drawn from $\mathcal{N}(\bar{\tau}_j, \sigma^2)$ where $\bar{\tau}_j$ denote the base causal effects for treatment a_j . More specifically, in the first step of the simulation, we will draw $\tau_{js} \sim \mathcal{N}(\bar{\tau}_j, \sigma_\tau^2)$ for $j \in \{1, 2\}$ and use them for all repeated experiments.

Second, after estimating the treatment effects $\hat{\tau}_{js}$ from the training dataset, we will have K out-of-sample customers to target. However, we only have the budget to assign treatment a_2 to c of them. Treatment a_1 can be assigned without limit. The optimization problem is then formulated below:

$$\max_{T_i \in \mathcal{T} = \{a_1, a_2\}} \frac{1}{K} \sum_{k=1}^K \sum_{s=1}^S \sum_{j=1}^2 \hat{\tau}_{js} \mathbb{1}\{T_k = a_j\} \mathbb{1}\{k \text{ is in Segment } s\}$$
(14)
s.t.
$$\sum_{k=1}^K \mathbb{1}\{T_k = a_2\} \le c$$

We report Winner's curse as percentages of the base treatment effect difference: $WC/(\bar{\tau}_2 - \bar{\tau}_1)$ in Table 5. First, note that although standard bootstrap removes a significant amount of winner's curse in the continuous and logit response cases, the resulting winner's curse remains statistically significant at 99% level due to a different optimization landscape. On the contrary, *m*-out-of*n* and numerical bootstrap eliminate winner's curse across all scenarios under the same set of hyperparameters.

(Andrews et al. 2024) does not apply to the constrained targeting problem (14) because it is designed for choosing a single winner out of some candidates. Selective inference methods are problem-specific in general. For example, (Andrews et al. 2021) corrects for the winner's curse when choosing the k-th largest value and (Andrews et al. 2022) applies to choosing a single winner based on absolute value. On the contrary, our bootstrap method is generally applicable to all optimization problems.

	Continuous Response	Bernoulli Response	Logit Response
No Correction	$36.38\%^{***}$	6.15%***	100.96%***
No Correction	(2.75%)	(1.19%)	(5.40%)
Standard Bootstrap	7.55%***	-0.91%	$22.59\%^{***}$
Standard Dootstrap	(2.85%)	(1.23%)	(5.49%)
mout of n Bootstrop	-0.92%	-1.26%	0.28%
<i>m</i> -out-or- <i>n</i> bootstrap	(2.87%)	(1.24%)	(5.44%)
Numerical Bootstrap	-2.81%	-1.33%	-6.07%
Numerical Dootstrap	(2.90%)	(1.23%)	(5.57%)
Empirical Bayes	$-22.87\%^{***}$	N / A	N / A
(Efron 2011)	(2.33%)	N/A	N/A
Standard Bayes	$13.21\%^{***}$	$5.03\%^{***}$	99.50%***
(Kessler 2024)	(2.70%)	(1.18%)	(5.38%)
Conditional Selective Inference	N / A	N / A	N / A
(Andrews et al. 2024)	N/A	N/A	N/A
Sample Splitting	1.31%	-0.38%	11.49%
	(3.93%)	(1.71%)	(8.46%)
Cross Validation	-3.44%	-2.98%	$15.87\%^{**}$
Cross valuation	(3.17%)	(1.34%)	(7.69%)

 Table 4
 Average Winner's Curse of Targeting with Multiple Segments and Budget Constraints

The average winner's curse and standard errors are calculated from R = 5000 repeated experiments; Training data has size N = 500; There are K = 100 out-of-sample customers to target and we can only assign c = 30 treatment a_2 . There are S = 5 discrete segments. The DGP parameters for continuous response are: $(\bar{\tau}_1, \bar{\tau}_2) = (1, 1.1), \sigma = 1$, and $\sigma_{\tau} = 0.1$. The DGP parameters for Bernoulli and Logit response are $(\bar{\tau}_1, \bar{\tau}_2) = (0.1, 0.15)$ and $\sigma_{\tau} = 0.1$. The hyperparameters for m--out-of-n and numerical bootstraps are $m = int(N^{0.95})$ and $\epsilon_N = N^{-0.45}$ under all scenarios.

5.3. Continuous Segmentation

In many (if not most) empirical settings, the causal effect is a continuous function of consumer characteristics instead of based on discrete segments. In addition, we do not know the functional form of the true data generation process. Thus, we examine the impact of a misspecified demand model on winner's curse as well as the robustness of our bootstrap correction estimator in this section.

We consider a linear specification (15) for the data generation process. Consumer characteristics is summarized in a univariate Gaussian variable X_i with follows standard normal distribution in the population. The causal effect is a linear function $\tau_j X_i$ of the characteristics. There are two candidate treatments (a_1, a_2) similar as before. The idiosyncratic noise ϵ_i follows $\mathcal{N}(0, \sigma^2)$.

$$Y_i = \tau_1 X_i \mathbb{1}\{T_i = a_1\} + \tau_2 X_i \mathbb{1}\{T_i = a_2\} + \epsilon_i$$
(15)

When the functional form of the DGP is unknown, two popular choices for estimating heterogeneous causal effects are the segment-based model and the causal forest (Wager and Athey 2018). A segment-based model first groups customers into segments using clustering algorithms or based on prior knowledge then estimates the treatment effect for each segment. We use a simple two-segment model with threshold zero for experiments. That is, customers with $X_i < 0$ belong to segment one and the ones with $X_i > 0$ belong to segment two. Causal forest can be thought of as a more data-driven way of segmenting customers but with a consistency guarantee. If the depth of the decision trees of a causal forest is upper bounded by one, then the causal forest is equivalent to the two-segment model. Increasing the maximum depth allowed for the decision trees introduces more flexibility but also risks over-fitting.

We compare the true and estimated treatment effects on part of the feature space in Figure 5. The models are trained from one realization of the dataset. We see that causal forest with max depth one (denoted as CF1 from now on) basically aligns with the two-segment model and provides a step-function approximation for the true treatment effect. Causal forest with max depth ten (denoted as CF10 from now on) has a much flexible approximation to the true treatment effect, but it also has much higher variance than CF1 due to overfitting. High estimation variance could be fatal when we use the models for estimating targeting values, as discussed in Theorem 2.



Figure 5 True vs. Estimated Heterogeneous Treatment Effects of Segment-Based Model and Causal Forest

Note. The black line indicates the true treatment effect $\tau_2 X_i - \tau_1 X_i$ as a function of consumer characteristics X_i . The blue line is the estimated effect from a two-segment model with segmentation threshold at 0. The orange (left) and green (right) lines are the estimated effects from causal forest models with max depth 1 and 10 respectively. The simulation parameters are: $(\tau_1, \tau_2) = (1, 1.1), \sigma = 1$, sample size N = 5000.

Next, we investigate the impact of different demand models on the winner's curse. We also include a demand model with the correctional functional form for comparison. We measure the winner's curse as a percentage of $\tau_2 - \tau_1$ and report the results in Table 5. First, even if the demand model has a correctly specified functional form as the data generation process, we still have a statistically significant winner's curse of 15.10%. Misspecified model exaggerates winner's curse. When we approximate the linear treatment effect function with a step function like the two-segment model or CF1, the winner's curse is almost three times the correct model. Furthermore, overfitting will make things even worse, such as the CF10 model in this case.

Table 5 Average Winner's Curse of Different Demand Models				
	Correct	Two-Segment	Causal Forest	Causal Forest
	Functional Form	Model	(Max Depth = 1)	(Max Depth = 10)
No Correction	$15.10\%^{***}$	$51.09\%^{***}$	43.84%***	$103.39\%^{***}$
	(1.46%)	(4.09%)	(4.81%)	(4.67%)
Standard Bootstrap	0.06%	$11.63\%^{***}$	1.30%	$22.96\%^{***}$
	(1.72%)	(4.31%)	(4.64%)	(4.60%)
m-out-of- n Bootstrap	$-4.75\%^{***}$	1.01%	-5.58%	$24.93\%^{***}$
	(1.76%)	(4.32%)	(5.04%)	(4.97%)
Numerical Bootstrap	$-5.42\%^{***}$	-2.75%	$-13.19\%^{***}$	-7.41%
	(1.84%)	(4.41%)	(4.79%)	(4.79%)

 Table 5
 Average Winner's Curse of Different Demand Models

The winner's curse is measured as percentages of $\tau_2 - \tau_1$. The simulation parameters are: $(\tau_1, \tau_2) = (1, 1.1), \sigma = 1$, sample size N = 500. The hyperparameters for *m*-out-of-*n* and numerical bootstraps are $m = int(N^{0.95})$ and $\epsilon_N = N^{-0.45}$ under all scenarios.

Bootstrap correction successfully correct most of the winner's curse while the exact performance depends on the demand model. First, standard bootstrap achieves insignificant winner's curse in the correction functional form and CF1 because they have smooth estimation function for the treatment effect. As a result, applying further smoothing via m-out-of-n or numerical bootstrap will over-correct. Second, the estimation function of CF10 has many sharp turns (on the left of Figure 5), serving effectively as non-differentiable points. Thus, standard bootstrap can only partially mitigate its winner's curse. Third, m-out-of-n and numerical bootstrap successfully eliminate winner's curse for the nonsmooth demand models, the two-segment model and CF10, with only one exception: m-out-of-n bootstrap for CF10. We conjecture that CF10 is a highly flexible model learned via bootstrap aggregation, so bootstrap with subsampling has very limited effect except for adding more trees in the causal forest effectively. In comparison, CF1 is a simple model with limited degree of freedom, so bootstrap with subsampling manages to introduce noise into the model to smooth the estimation function.

6. Conclusion

In this paper, we demonstrate the existence of winner's curse-over-optimistic policy value estimateexists in a wide range of data-driven marketing decision-making problems. It happens because when selecting decisions based on their estimated efficacy, a procedure we call inference-then-optimize, the algorithm tends to select the overestimated ones, leading to inflated policy value estimate. Winner's curse is a systemic issue behind almost all data-driven decision-making problems, even when the data-driven decision is itself correct. Because accurate forecast of policy values is critical for many subsequent investment decisions, such as how much resource a firm should allocate to launch the policy, it is significant to provide a fix for the winner's curse. We discuss some existing remedies, including sample splitting, Bayesian/shrinkage estimators, and selective inference, from the statistics and economics literature. We note that their effectiveness rely on some restrictive assumptions. We propose an easy-to-implement and assumption-lean bootstrap correction method, includes standard nonparametric bootstrap as well as subsampling and perturbation variants, that systematically corrects the upward bias from inference-then-optimize. Through extensive simulation studies across different marketing applications, we find that bootstrap-correction method demonstrate a consistent debiasing performance.

Endnotes

1. For a detailed discussion, please refer to (Rasines and Young 2020)

References

- Andrews I, Bowen D, Kitagawa T, McCloskey A (2022) Inference for losers. AEA Papers and Proceedings, volume 112, 635–640 (American Economic Association 2014 Broadway, Suite 305, Nashville, TN 37203).
- Andrews I, Kitagawa T, McCloskey A (2021) Inference after estimation of breaks. Journal of Econometrics 224(1):39-59.

Andrews I, Kitagawa T, McCloskey A (2024) Inference on winners. The Quarterly Journal of Economics 139(1):305–358.

Bickel P, Götze F, van Zwet W (1997) Resampling fewer than n observations: Gains, losses, and remedies for losses. *Statistica Sinica* 7(1):1–31.

Bickel PJ, Sakov A (2008) On the choice of m in the m out of n bootstrap and confidence bounds for extrema. *Statistica Sinica* 967–985.

Chakraborty B, Laber EB, Zhao Y (2013) Inference for optimal dynamic treatment regimes using an adaptive m-out-of-n bootstrap scheme. *Biometrics* 69(3):714–723.

Efron B (1979) Bootstrap methods: Another look at the jackknife. The Annals of Statistics 7(1):1–26.

Efron B (2011) Tweedie's formula and selection bias. Journal of the American Statistical Association 106(496):1602–1614.

Fang Z, Santos A (2019) Inference on directionally differentiable functions. The Review of Economic Studies 86(1):377–412.

Hall P (1992) The bootstrap and Edgeworth expansion (Springer Science & Business Media).

Hong H, Li J (2020) The numerical bootstrap. The Annals of Statistics 48(1):397-412.

Johnstone IM, Silverman BW (2004) Needles and straw in haystacks: Empirical bayes estimates of possibly sparse sequences . Kessler R (2024) Overcoming the winner's curse: Leveraging bayesian inference to improve estimates of the impact of features launched via a/b tests. *Amazon Science*.

Kuchibhotla AK, Kolassa JE, Kuffner TA (2022) Post-selection inference. Annual Review of Statistics and Its Application 9(1):505–527.

Rasines DG, Young GA (2020) Bayesian selective inference: Non-informative priors. arXiv preprint arXiv:2008.04584 .

Van der Vaart AW (2000) Asymptotic statistics, volume 3 (Cambridge university press).

Van der Vaart AW, Wellner JA (1996) Weak Convergence and Empirical Processes (Springer).

Wager S, Athey S (2018) Estimation and inference of heterogeneous treatment effects using random forests. Journal of the American Statistical Association 113(523):1228–1242.