

# A Matrix Completion Solution to the Problem of Ignoring the Ignorability Assumption

Omid Rafieian\*

Cornell Tech and Cornell University

---

\*The author thanks David Blei, Daria Dzyabura, Tesary Lin, Unnati Narang, Matt Osborne, Caio Waisman, and Hema Yoganarasimhan for detailed comments that have improved the paper. The author also thanks the participants of the 2023 UT Dallas FORMS conference and Temple University marketing seminars for their feedback. Please address all correspondence to: [or83@cornell.edu](mailto:or83@cornell.edu).

## Abstract

Digital platforms deliver numerous interventions to their users. One of platforms' main goals is to estimate the causal effect of these interventions. An ideal way to answer this question is to run a fully randomized experiment. However, the economic cost of such experiments is high, making alternative approaches based on observational data appealing to digital platforms. In this paper, we study the feasibility of using observational methods in the presence of algorithmic decision-making. Although the setting created by algorithmic decision-making satisfies the unconfoundedness assumption as the assignment rule is known, the overlap assumption is often violated because these algorithms generate deterministic recommendations. We theoretically show that the violation of overlap can substantially bias the estimates of the average treatment effect from observational data. We quantify this bias and discuss whether it is practically relevant in digital platforms. To address this issue, we propose a novel solution based on machine learning methods used for matrix completion that allows us to recover the average treatment effect estimates if the underlying space of treatment effects is low rank. We validate our theoretical results using synthetic and experimental data and discuss the implications.

**Keywords:** causal inference, machine learning, overlap assumption, unconfoundedness, digital platforms, observational methods

# 1 Introduction

Digital platforms deliver numerous interventions to their users every day. These interventions can take different forms, such as push notifications on mobile phones, content recommendations on streaming platforms, etc. At the core of this large-scale delivery of interventions are two elements: data collection and algorithmic decision-making. Digital platforms collect massive amounts of data from the users of their basic information such as demographics and their behavioral characteristics such as their past browsing history. These data are then given as inputs to algorithms that can efficiently process them and make real-time decisions, thereby allowing platforms to deliver interventions at a very large scale.

An important question that digital platforms and academic researchers want to know the answer to is the causal effect of these interventions. The gold standard in both research and practice is to use randomized controlled trials (RCT), where some users randomly receive the treatment, and some do not. This randomization, in turn, allows us to identify and estimate the causal effect of an intervention. However, running fully randomized experiments is not always in the interest of the platform because experiments can come at the expense of assigning a large group of users to sub-optimal interventions. Thus, it is crucially important for these platforms to estimate the causal effects of interventions with their existing observational data.

Both experimental and observational methods to estimate the causal effect of an intervention rely on a set of assumptions called strong ignorability of the treatment assignment. Strong ignorability assumption is a mix of two assumptions: (1) *unconfoundedness* of the treatment assignment, which states that conditional on observed covariates, assignment to the treatment is independent of potential outcomes, and (2) *overlap* or *positivity* of the treatment assignment, which assumes that the assignment to the treatment is probabilistic, that is, the propensity score of the treatment is a probability strictly between zero and one. The part that is often violated in observational studies is the unconfoundedness assumption. That is, there are unobserved confounding factors that affect both the treatment assignment and the outcome of interest. The presence of confounding, therefore, hampers researchers' ability to draw causal inference from observational studies.

What is different in digital platforms is that the unconfoundedness assumption is more plausible than in most settings. This is because the platform itself delivers the interventions to users. As such, given the output of the algorithm used for decision-making at the digital platform, assignment to a treatment is unconfounded. Even if the researcher does not have access to the algorithmic output but the data used for algorithmic decision-making, it is still possible to satisfy the unconfoundedness assumption by learning the underlying selection mechanism from data. This is increasingly an easier task with the development of methods that combine causal inference with machine learning

methods to capture complex confoundedness in the data. Thus, the presence of the exact output of the algorithm or high-dimensional data used for algorithmic decision-making serves as a strong motivation for using observational methods in the context of digital platforms.

What arises as an important challenge is an often-ignored part of the ignorability assumption: overlap or the requirement for the probabilistic assignment. Although algorithmic decision-making helps platforms better use their interventions, many of these algorithms only generate deterministic outputs. That is, one intervention will be shown with probability one, and the other interventions have zero probability of being shown. For example, the promotion offered by a ride-sharing app is the deterministic output of an algorithm. In these cases, the overlap assumption is violated, which leaves us with no theoretical guarantee for the estimated treatment effects. In this paper, we consider the case of a digital platform whose context satisfies the unconfoundedness assumption because the algorithmic outputs are readily available at the platform but violates the overlap assumption because of the deterministic assignment employed by the algorithms. To that end, we seek to answer the following sets of research questions:

1. How does the lack of overlap bias the estimates of average treatment effect in observational studies that satisfy the unconfoundedness assumption? Can the state-of-the-art model-based and model-free approaches overcome this challenge?
2. How likely is this lack of overlap to cause bias in the average treatment effect estimates from a practical standpoint?
3. What are the solutions to this problem, and under what assumptions do they work?

To answer these questions, we develop a simple framework that distinguishes between three regions in the data based on the treatment assignment: (1) probabilistic assignment, where the propensity score of the assignment is a number in the non-exclusive interval of  $(0, 1)$ , (2) deterministic assignment, where the treatment assignment happens deterministically with probability one, i.e., propensity score for the treatment is one, and (3) deterministic no-assignment, where the treatment assignment will not happen with probability one, i.e., the propensity score for the treatment is zero. As such, the only region that satisfies the overlap assumption is the one with the probabilistic assignment. We further define three conditional average treatment effects (CATE) for each of these three regions to allow for the possibility that these estimands are different at the population level. This allows us to say something concrete and testable about the magnitude of bias in our treatment effect estimates that is caused by the lack of overlap.

Our theoretical analysis first shows that the conditional average treatment effect for the regions with deterministic assignment is unidentified. We then consider the case where we use the data from

all three regions with a known propensity score and examine how well we can estimate the average treatment effect in this case. This mimics the setting at digital platforms where the propensity scores are either known ex-ante or can be estimated accurately. We also focus on the state-of-the-art model-based and model-free approaches to estimate the average treatment effects, such as double machine learning (Chernozhukov et al., 2018a) to ensure that a poor modeling choice does not drive the results of our analysis. Our analysis shows that all these methods can result in substantial bias due to the lack of overlap even when propensity scores are known. In cases where propensity scores need to be estimated, this bias can be considerably larger. On the bright side, our analysis shows that if the propensity scores are known, a large class of observational methods can recover the only identifiable causal estimand in the data, the conditional average treatment effect for the region with a probabilistic assignment. This finding allows us to quantify the magnitude of bias in a concrete manner and arrive at an important insight: the magnitude of bias in the ATE estimate can be arbitrarily large if the overlap assumption is violated. We then carry out a series of simulation experiments to verify our theoretical findings.

Next, we focus on the prevalence of this problem and ask the following question: to what extent will this issue arise in practical contexts? In principle, if the assignment probability is a function of the conditional average treatment effect for an observation, the lack of overlap likely results in large biases in the estimates of average treatment effects. The problem is that if the digital platform is also interested in optimizing the same causal estimand, the optimal strategy for them is to assign interventions based on scores that are related to users' conditional average treatment effects (Shalit et al., 2017; Wager and Athey, 2018). We consider two well-known cases in practice: (1) digital promotions and (2) advertising auctions. In the former, platforms usually use predictive models that estimate users' responsiveness to promotions and deterministically assign users with responsiveness above a certain threshold to promotions, so points at the right-tail of CATE distribution are more likely to belong to the deterministic assignment region. In the latter case, advertisers' bids usually reflect their CATE for each impression, so an advertiser with a very low CATE will never win the auction, which creates deterministic no-assignment at the left tail of CATE distribution. In both cases, the conditional average treatment effect of the probabilistic region is not the same as the average treatment effect for the entire population because the deterministic regions are likely selected from the tails of the CATE distribution. We theoretically characterize these cases and use simulation to demonstrate that even small associations between CATE and deterministic assignment can generate large biases in ATE estimates.

Once we establish the existence and prevalence of the lack of overlap in observational studies involving digital platforms and the challenges it pose, we focus on the potential solutions for this

problem. We propose a framework that formulates the unidentifiability of the conditional average treatment effect for the overlap-violating regions of the data as a missing data problem. Although we cannot fix this problem with a single study at hand, we can potentially use the information across studies to help with this missing data problem. In particular, if we have multiple studies with different treatments (e.g., price discount in one study and push notification for a loyalty program in another) whose individualized effects come from a low-rank space, we can use matrix completion methods to impute the conditional average treatment effect for the overlap-violating regions. In particular, we set CATE estimates from the overlap-violating regions as question marks in a matrix and only estimate CATE for units whose assignment is probabilistic. We then exploit the variation among those entries in the matrix to complete the matrix for the deterministic regions. The intuition for this approach is as follows: if there are a few factors that collectively determine CATE for each study, we can exploit similarities across users and across treatments to identify those factors and impute CATEs for units that belong to overlap-violating regions. Once we complete the matrix for the parts that were formerly unidentified, we can correct the bias in the ATE estimates.

We deliver a series of simulation studies to establish the performance of our proposed algorithm. We consider a wide range of deterministic assignment problems that may arise in real settings. Each case corresponds to a specific missingness pattern in the estimated CATE matrix due to identifiability issues. To that end, we consider two types of missingness patterns: (1) random and (2) CATE-dependent. When missingness is at random, we show that both Double ML (or other conventional ATE estimation approaches) and our proposed method are able to recover ATE across studies. However, our proposed method has lower error as it exploits the variation across studies. In settings with CATE-dependent missingness, we simulate cases where observations with higher or lower CATEs are more likely to have a deterministic assignment or no-assignment, which resemble settings for promotion assignment and advertising auctions. We show that the ATE estimates under conventional approaches such as Double ML are largely biased. However, our proposed method can reliably recover the true ATE.

Finally, we use actual data to evaluate the performance of our proposed algorithm. We design a game with 20 questions and randomize the deals provided in each question separately. We further include parts in our experiment that collect rich pre-treatment variables that allow us to estimate heterogeneous treatment effects. We run our experiment on Mechanical Turk to recruit participants. Our experiment design provides us with 20 experiments with their ground truth ATEs, which allows us to examine how our proposed algorithm performs in recovering the true ATEs relative to the state-of-the-art benchmarks. We consider three separate outcome variables and a series of assignment scenarios that vary in their level of adversariality and difficulty in recovering ATE.

Our results consistently show substantially better performance by our proposed algorithm than the benchmarks.

In sum, our paper makes several contributions to the literature. Methodologically, we present a comprehensive study of the overlap assumption and theoretically characterize the context in digital platforms that use algorithmic decision-making. In particular, we quantify the bias caused by the lack of overlap in a variety of contexts and propose a novel machine-learning solution that views the identification challenge as a missing data problem and combines heterogeneous treatment effect estimation with matrix completion to recover the treatment effects. From a substantive and practical viewpoint, we identify an important challenge for the digital platforms that employ algorithmic decision-making. While most of the applied causal inference literature is focused on satisfying unconfoundedness using state-of-the-art causal machine learning methods, we show that the fundamental problem in digital platforms is, in fact, the overlap violation. We further discuss empirical contexts where this problem may arise, such as digital promotions and advertising. Overall, our proposed algorithm is fairly general and can be applied to many contexts, specifically those in digital settings where platforms deliver numerous interventions that have common factors and satisfy the low-rank requirements. Thus, we expect our framework to be valuable for platforms that want to utilize their existing observational data and researchers who access the data from such platforms.

## **2 Related Literature**

Broadly, our paper relates to the causal inference literature that aims to estimate treatment effects (Neyman, 1923; Imbens and Rubin, 2015). Following the influential paper by (Rosenbaum and Rubin, 1983), much of this literature focuses on a set of assumptions known as the strong ignorability of the treatment assignment, which is a combination of two assumptions: unconfoundedness and overlap. While the unconfoundedness assumption has received considerable attention in the literature, the overlap assumption has often been viewed as a more straightforward assumption to be satisfied in real settings. As such, less attention has been paid to the overlap assumption in prior studies on causal inference, with a few notable exceptions that focus on various aspects of the overlap assumption, such as studying sample trimming strategies (Crump et al., 2009; Ma and Wang, 2020; D’Amour et al., 2021), extra assumptions that help recover causal estimands for overlap-violating regions (Nethery et al., 2019), and quantifying the uncertainty in overlap-violating regions of observational data (Jesson et al., 2020). Motivated by the context of algorithmic decision-making in digital platforms and the prevalent violation of this assumption in such contexts, we study the overlap assumption – how it arises and what theoretical implications it has for treatment effect estimates. We contribute to this literature by characterizing the bias induced by the lack of overlap

and identifying cases where the lack of overlap can be detrimental in the sense that conventional solutions such as using more competent causal machine learning models and sample trimming do not solve the problem. We further add to this literature by proposing a machine learning approach based on matrix completion that imposes low-rank assumptions on the treatment effects space to help correct this bias.

Second, our paper relates to the literature on the growing intersection of machine learning and causal inference. In recent years, a series of papers combined the insights from the causal inference literature with the flexibility and scalability of machine learning models in learning patterns from data to develop new methods to estimate causal estimands such as average treatment effect (Belloni et al., 2014; Chernozhukov et al., 2018a; Athey et al., 2018) or conditional average treatment effect (Shalit et al., 2017; Athey et al., 2019; Chernozhukov et al., 2018b; Nie and Wager, 2021). In marketing, many recent papers used these methods in a variety of application domains such as personalized promotions (Simester et al., 2020a,b), customer relationship management (Ascarza, 2018), personalized free-trial (Yoganarasimhan et al., 2022), ad targeting and sequencing (Rafieian and Yoganarasimhan, 2021; Rafieian, 2022), video advertising format (Rafieian et al., 2023), and personalized versioning (Goli et al., 2022b). We add to this literature in two separate ways. First, we theoretically characterize the performance of causal machine learning methods when the overlap assumption is violated. Second, we propose a machine learning algorithm that exploits the similarities between the treatments in the treatment space and overcomes the issue of overlap violation under certain assumptions.

Finally, our paper relates to the literature on matrix completion. Although the popularity of these models stems from the Netflix Prize for movie recommendation (Bennett et al., 2007), the application of matrix completion models is much broader to any setting where the underlying structure of matrix with missing data is low-rank (Mazumder et al., 2010). The relevance and success of matrix completion models motivated a large stream of theoretical work that establish the main theoretical guarantees of these models (Candès and Recht, 2009; Candès and Tao, 2010; Recht, 2011; Gross, 2011; Negahban and Wainwright, 2011). Recent work has focused on the intersection of matrix completion and causal inference and found useful applications (Kallus et al., 2018; Athey et al., 2021; Agarwal et al., 2021). Our work adds to this literature by formulating the unidentifiability of the overlap-violating parts of data as a missing data problem and applying matrix completion models to exploit cross-study variation and recover the true causal parameters. Specifically, we bring the recent advancements in CATE estimation to the matrix completion problem to help utilize the rich information in the covariate space.



### 3 Algorithmic Decision-making

#### 3.1 Problem Definition

We first formally define our problem. Consider a general case where a digital platform delivers interventions to observation units. The observation unit is often a user in digital platforms. When an observation unit is available to receive the intervention, the platform chooses from the set of all interventions, which is denoted by  $\mathcal{W}$  in our problem. For example, this set can be the list of different ads to show to the user. For observation  $i$ , let  $W_i$  denote the intervention delivered to the user, and  $X_i$  denote the vector of observable characteristics from the super set  $\mathcal{X}$ . As customary in digital platforms, the vector of characteristics  $X_i$  is often high-dimensional with detailed information about the user such as demographics and past user history, as well as contextual factors such as the timestamp of the observation.

In order to determine which intervention to deliver in each observation, digital platforms generally use an algorithm that scalably uses the feature vector  $X_i$  and returns an intervention that optimizes the platform’s objective. For any intervention  $w \in \mathcal{W}$ , we characterize this algorithmic policy as a function  $\pi_w : \mathcal{X} \rightarrow [0, 1]$ , where  $\pi_w(X_i)$  determines the probability that the platform chooses intervention  $w$  in observation  $i$ . The function  $\pi_w$  is the same as the propensity score function in the causal inference literature. Digital platforms often have direct access to this function.

Once the intervention is delivered, the platform collects the outcome of interest  $Y_i$  for observation  $i$ . This outcome is defined based on the problem under the study. For example, this outcome can be clicks or usage for push notifications. Following the potential outcomes framework, we define  $Y_i(w)$  for each  $w \in \mathcal{W}$  as the potential outcome we would have observed under intervention  $w$ . For simplicity and greater consistency with the causal inference literature, we focus our analysis on the binary case with one treatment and one control group.<sup>1</sup> As such,  $W_i = 1$  means that observation  $i$  has received the treatment, whereas  $W_i = 0$  refers to the case where observation  $i$  has received the control. Hence, for each observation  $i$ , there are two potential outcomes  $Y_i(0)$  and  $Y_i(1)$ . With this notation in place, we now define two estimands that researchers and practitioners often want to estimate as follows:

**Definition 1.** *The Average Treatment Effect (ATE) is denoted by  $\tau^*$  and defined as follows:*

$$\tau^* = \mathbb{E}[Y_i(1) - Y_i(0)], \quad (1)$$

*where the expectation is taken over the entire population.*

---

<sup>1</sup>The results are easily generalizable to the case with multiple treatment levels.

The Conditional Average Treatment Effect (CATE) is the same as ATE conditional on a certain value of the covariate vector. We denote CATE as  $\tau^*(x)$  and define it as follows:

$$\tau^*(x) = \mathbb{E}[Y_i(1) - Y_i(0) \mid X_i = x]. \quad (2)$$

The prior literature on causal inference has proposed a wide variety of methods to estimate ATE and CATE (Imbens and Rubin, 2015). These methods require a set of assumptions known as (1) *Stable Unit Treatment Value Assumption (SUTVA)*, and (2) *Strong Ignorability of Treatment Assignment*. SUTVA states that there is a single version of each treatment, and the units do not interfere with each other. In digital settings where treatments are well-defined with a single version and a unit's treatment status, and action is isolated in the sense that it does not change the treatment status of other units, SUTVA would be more plausible. In this paper, we consider the cases where SUTVA holds to exclusively focus on cases where the ignorability assumption is violated.<sup>2</sup>

The second set of assumptions is known as *Strong Ignorability* assumption, which is defined in the seminal paper by Rosenbaum and Rubin (1983) as follows:

**Definition 2.** *The assignment to treatment is strongly ignorable given the observed covariates  $X_i$ , if we have:*

- *Unconfoundedness: The potential outcomes are independent of the treatment assignment conditional on observed covariates:*

$$\{Y_i(1), Y_i(0)\} \perp\!\!\!\perp W_i \mid X_i, \quad (3)$$

*which is known as the unconfoundedness assumption and referred to with other names such as selection on observables, conditional exogeneity, etc.*

- *Overlap: The assignment to the treatment is probabilistic, that is:*

$$0 < \Pr(W_i = 1 \mid X_i) < 1, \quad (4)$$

*where  $\Pr(W_i = 1 \mid X_i)$  is the same as the propensity score when  $w = 1$ , that is,  $\pi(X_i)$ .<sup>3</sup> This assumption is often referred to as the overlap or positivity assumption and guarantees that the assignment to the treatment is not deterministic.*

The strong ignorability assumption serves as the foundation for studies of causal inference. The

<sup>2</sup>A series of recent studies show cases where SUTVA is violated in digital settings. Please see Goli et al. (2022a) for a great summary of these cases.

<sup>3</sup>For brevity, instead of  $\pi_1(X_i)$ , we use  $\pi(X_i)$ .

most common challenge in observational studies is often the unobservability of the assignment rule, which results in the confoundedness of the treatment. That is, there is an unobservable variable  $Z_i$  that affects both the treatment assignment and the outcome, thereby resulting in selection bias in the estimates of the average treatment effect.

The key difference in digital platforms that employ algorithmic decision-making is that the assignment rule is often fully observable. That is, the platform can easily store the  $X_i$  used for algorithmic decision-making and the output of the algorithm  $\pi(X_i)$ , which is shown to be sufficient to satisfy the unconfoundedness assumption (Rosenbaum and Rubin, 1983). Hence, observational studies on digital platforms do not suffer from the well-known confoundedness or endogeneity problem since there is no selection on unobservables. What makes these observational studies challenging is the commonly ignored part of the strong ignorability assumption, which requires the treatment assignment to be probabilistic. Although the probabilistic assignment is plausible in more traditional studies without algorithmic decision-making in the background, algorithms used by digital platforms to deliver interventions are often deterministic. That is,  $\pi(X_i)$  can be equal to zero or one depending on  $X_i$ .

Our goal in this paper is to study the consequences of the lack of overlap in observational studies on digital platforms. As such, we can formally define the problem as follows:

**Definition 3.** *Consider a digital platform that uses data  $\mathcal{D} = \{Y_i, W_i, X_i, \pi(X_i)\}$ . The main estimands the platform wants to estimate are the average treatment effect (ATE) for the entire population and conditional average treatment effects (CATE) for each value of the vector of covariates.*

Following the formal definition of our problem in Definition 3, our primary goals in this paper are to (1) quantify the magnitude of bias due to this overlap violation, (2) identify the link between this bias and the algorithm used by the platform, and (3) discuss potential solutions to overcome this problem.

## 3.2 Analysis

In this section, we theoretically analyze how the lack of overlap can lead to biased estimates of the average treatment effect (ATE). We start by showing the identification problem with the lack of overlap in observational data in §3.2.1. We then examine how the model-based approaches such as double machine learning perform in estimating the ATE in §3.2.2. Finally, we focus on model-free approaches such as importance sampling and theoretically derive their properties in §3.2.3.

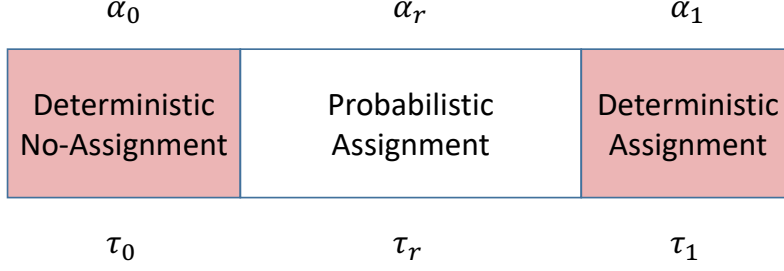


Figure 1: Different regions based on the type of assignment.

### 3.2.1 Identification Challenge

It is well-known that the violation of the overlap assumption can bias ATE estimates. In this section, we illustrate this point by presenting a simple framework that we can use for our subsequent analysis. To do so, we first introduce a new notation that captures the difference between different parts of the covariate space. In particular, we focus on the conditional average treatment effect for three separate groups of observation units as shown in Figure 1:

- *Probabilistic assignment region*: For observations where  $0 < \pi(X_i) < 1$ , we define  $\tau_r = \mathbb{E}[Y_i(1) - Y_i(0) \mid 0 < \pi(X_i) < 1]$ , which is the average treatment effect for the observations that have a probabilistic assignment. We denote the fraction of such observations in our data by  $\alpha_r$ .
- *Deterministic no-assignment region*: For observations where  $\pi(X_i) = 0$ , we define  $\tau_0 = \mathbb{E}[Y_i(1) - Y_i(0) \mid \pi(X_i) = 0]$ , which is the average treatment effect for observations that certainly receive the control. We denote the fraction of such observations in our data by  $\alpha_0$ .
- *Deterministic assignment region*: For observations where  $\pi(X_i) = 1$ , we define  $\tau_1 = \mathbb{E}[Y_i(1) - Y_i(0) \mid \pi(X_i) = 1]$ , which is the average treatment effect for observations that certainly receive the treatment. We denote the fraction of such observations in our data by  $\alpha_1$ .

Now, we can define the average treatment effect as  $\tau^* = \alpha_r \tau_r + \alpha_0 \tau_0 + \alpha_1 \tau_1$ , where  $\alpha_r + \alpha_0 + \alpha_1 = 1$ . This decomposition allows us to highlight where the deterministic assignment creates a problem. Suppose that the digital platform wants to use data  $\mathcal{D}$  to estimate  $\tau_1$ . The problem is that for this slice of the population, the treatment variable is perfectly correlated with the propensity score, that is,  $W_i = \pi(X_i) = 1$ . The same problem is present in identifying  $\tau_0$ , since there is no residual variation in treatment. Thus, we can write the following lemma:

**Lemma 1.** *The conditional average treatment effects  $\tau_1$  and  $\tau_0$  are unidentifiable given data  $\mathcal{D}$ .*

In light of Lemma 1, the only identifiable piece of  $\tau^*$  is  $\tau_r$ . We now want to see how this

identification problem manifests itself in both model-based and model-free approaches to estimate causal estimands.

### 3.2.2 Model-based Approaches to Estimate ATE

There are many model-based approaches one could use to estimate ATE from observational data. The traditional approach is to use a linear regression that projects the outcome on the treatment variable as well as other controls and estimates the average treatment effect. These methods work well if the confoundedness in the treatment assignment is captured by a linear combination of covariates. However, in many high-dimensional settings, the assignment has more complex patterns, which makes linear controls inadequate in accounting for observed confoundedness. Further, the relationship between other covariates and the outcome can also follow a non-linear pattern. These limitations, in turn, attracted a growing body of work that brings machine learning methods to causal inference in order to increase the flexibility and robustness of model-based methods to estimate ATE (Belloni et al., 2014; Chernozhukov et al., 2018a). Many of these methods are now considered state-of-the-art methods for estimating the ATE. Our goal is to quantify the magnitude of bias when we use these methods to estimate the causal estimands.

We present a general framework to study model-based approaches. Let  $\mu_w(x) = \mathbb{E}[Y_i(w) \mid X_i = x]$  denote the underlying population model for the conditional potential outcomes for any  $w$ . We can write:

$$Y_i(w) = \mu_0(X_i) + \tau^*(X_i)w + \epsilon_i(w), \quad (5)$$

where  $\epsilon_i(w)$  denotes the structural error term for any value of the treatment  $w \in \{0, 1\}$ . Unconfoundedness implies that  $\mathbb{E}[\epsilon_i(W_i) \mid X_i, W_i] = 0$ . We further define function  $m$  as the conditional mean function such that  $m(x) = \mathbb{E}[Y \mid X = x]$ . We can now write the following decomposition:

$$Y_i - m(X_i) = (W_i - \pi(X_i)) \tau^*(X_i) + \epsilon_i(W_i), \quad (6)$$

which holds because  $m(X_i) = \mu_0(X_i) + \tau^*(X_i)\pi(X_i)$ . This decomposition – which is first proposed by Robinson (1988) for estimating partially linear models – serves as a foundation for model-based approaches to estimate ATE or CATE that use machine learning models for causal inference. The key insight is that we can use machine learning models to flexibly learn nuisance functions  $m(X_i)$  and  $\pi(X_i)$ , and then feed these estimates into an objective function to estimate causal estimands. We can define this objective function as follows:

$$\tau^*(\cdot) = \underset{\tau}{\operatorname{argmin}} \mathbb{E} \left[ (Y_i - m(X_i) - (W_i - \pi(X_i)) \tau(X_i))^2 \right]. \quad (7)$$

The double machine learning (DML) approach estimates both nuisance functions using machine learning models and then estimates the ATE using a version of the objective function above, where there is only one  $\tau(X_i)$  for the population (Chernozhukov et al., 2018a). A series of methods use this decomposition to estimate heterogeneous treatment effects or CATE by using random forests (Athey et al., 2019), or more broadly, any loss minimization method (Nie and Wager, 2021; Chernozhukov et al., 2018b). We now use this objective function to prove the following proposition:

**Proposition 1.** *Suppose that there is a digital platform that has access to data  $\mathcal{D} = \{Y_i, W_i, X_i, \pi(X_i)\}$ , where  $\pi(X_i)$  is known, but takes values zero and one for parts of the population. The estimated average treatment effect (ATE)  $\hat{\tau}$  under any method that uses the objective function in Equation (7) converges to  $\tau_r$  in probability, that is:*

$$\hat{\tau} \xrightarrow{p} \tau_r \quad (8)$$

*Proof.* See Web Appendix A.1. □

This proposition shows that state-of-the-art model-based approaches such as double machine learning estimate  $\tau_r$  as the ATE when the propensity is known. As such, to the extent that  $\tau_r$  is different from  $\tau^*$ , the estimate for the ATE would be biased. Given that  $\tau_r$  appears in the equation for  $\tau^*$ , the question is if there is any bound for the magnitude of bias. In light of Proposition 1, we know that the magnitude of bias is  $|\tau^* - \hat{\tau}| \xrightarrow{p} |(\alpha_r - 1)\tau_r + \alpha_0\tau_0 + \alpha_1\tau_1|$  such that  $\alpha_r + \alpha_0 + \alpha_1 = 1$ , which allows us to further simplify this expression to the following:

$$|\tau^* - \hat{\tau}| \xrightarrow{p} |\alpha_0(\tau_0 - \tau_r) + \alpha_1(\tau_1 - \tau_r)|. \quad (9)$$

This simplification highlights the fact that if the treatment effects for the deterministic regions are the same as the treatment effect for the probabilistic region, there will be no bias. However, it is easy to imagine scenarios where the difference in  $\tau_1$ ,  $\tau_0$ , and  $\tau_r$  creates a substantial bias in estimates of the average treatment effect. In fact, for any constant  $c$ , we can find  $\tau_0$  and  $\tau_1$  such that  $|\alpha_0(\tau_0 - \tau_r) + \alpha_1(\tau_1 - \tau_r)| = c$ , which implies that we can have any magnitude of bias. We present this intuition in the following corollary:

**Corollary 1.** *The magnitude of bias can be any arbitrary amount if either  $\alpha_0$  or  $\alpha_1$  is non-zero.*

While Corollary 1 shows that the bias can be of any magnitude if we have a deterministic assignment in our population, the bright side is that all the methods that use the objective function in Equation (7) are able to recover the only identifiable part of  $\tau^*$ . That is, the presence of deterministic assignment for parts of the population does not result in biased estimates of the region with the

probabilistic assignment. Hence, the researcher can rely on the estimates as consistent estimators of the true population parameters for the region with the probabilistic assignment.<sup>4</sup>

### 3.2.3 Model-free Approaches to Estimate ATE

In §3.2.2, we show that model-based approaches to estimate ATE fail to recover the true ATE. However, one could argue that the bias comes from outcome modeling. To address this issue, we discuss model-free approaches to estimate the ATE that directly use the realized outcomes without modeling them. The foundation for these approaches is the idea of importance sampling proposed by Horvitz and Thompson (1952) in their seminal paper. The idea is to weight each observation by its inverse propensity score, which gives us the following estimator for the ATE:

$$\hat{\tau}_{\text{IPS}} = \frac{1}{N} \left( \sum_{i=1}^N Y_i \left( \frac{W_i}{\pi(X_i)} - \frac{1 - W_i}{1 - \pi(X_i)} \right) \right), \quad (10)$$

where the first term  $W_i/\pi(X_i)$  weights the observations that received the treatment by the inverse probability of that assignment, and the second term  $(1 - W_i)/(1 - \pi(X_i))$  weights the observations that did not receive the treatment. This estimator estimates the average treatment effect by subtracting an estimate of what would have happened if everyone had received the control from an estimate of what would have happened if everyone had received the treatment. It is a model-free approach because we do not need any model of the outcome to estimate our causal estimand.

In the absence of full overlap, a drawback of this approach becomes immediately apparent. For observations with deterministic assignment, the denominator in one of the terms is zero, which makes the overall estimator undefined. The conventional solution is to use sample trimming, wherein we drop observations with a deterministic assignment. As a result, this approach only relies on the  $\alpha_r$  fraction of observations with the probabilistic assignment. We can show the following proposition:

**Proposition 2.** *Suppose that there is a digital platform that has access to data  $\mathcal{D} = \{Y_i, W_i, X_i, \pi(X_i)\}$ , where  $\pi(X_i)$  is known, but takes values zero and one for parts of the population. The ATE estimator based on Equation (10) that drops observations with a deterministic assignment converges in probability to  $\tau_r$ , that is:*

$$\hat{\tau}_{\text{IPS}} \xrightarrow{p} \tau_r \quad (11)$$

---

<sup>4</sup>It is important to notice that this is only the case when propensity scores are known, which allows the optimizer to ignore overlap-violating elements of the objective function because  $W_i - \pi(X_i) = 0$ . The situation can be different if the propensity scores are to be estimated because  $W_i - \pi(X_i)$  in the objective function can be a very small number unequal to zero, which may largely bias the ATE estimate as the optimizer attempts to minimize the loss in Equation (7) by assigning presumably large weights to  $\tau^*$ .

*Proof.* See Web Appendix A.2. □

Similar to Proposition 1, Proposition 2 guarantees that the Inverse Propensity Scoring (IPS) estimator recovers the treatment effect for the probabilistic region. As such, Corollary 1 holds for this proposition too, indicating that the bias is a function of two unidentifiable elements  $\tau_1$  and  $\tau_0$ .<sup>5</sup>

### 3.3 Practical Relevance

In light of our theoretical analysis, we know that the lack of overlap can substantially bias the estimates of the average treatment effects. An important question is whether this is just a theoretical possibility that is not practically important. In other words, do we expect the bias term  $|\alpha_0(\tau_0 - \tau_r) + \alpha_1(\tau_1 - \tau_r)|$  to be large in real settings? Part of the rationale for the trimming approaches that are widely used in the literature is that  $\tau_0$  and  $\tau_1$  are not different from  $\tau_r$ . Here we ask the following question: is this homogeneity assumption (i.e.,  $\tau_0 = \tau_r = \tau_1$ ) correct in digital platforms?

To the extent that  $\pi(x)$  is a function of  $\tau^*(x)$ , we expect  $\tau_0$  and  $\tau_1$  to be different from  $\tau_r$ . The problem is that, in many cases, the objective function in the algorithm used by the digital platform is directly influenced by CATE, that is of interest to the researcher. We discuss two prime examples of such settings in practice:

- *Promotions:* In the context of promotions, many digital platforms use algorithmic scores and thresholding rules to assign users to promotions (Shi et al., 2022). That is, users with a score above a certain threshold will deterministically receive the promotion, which creates a case of *deterministic assignment*. The rest of the users will either be assigned to the probabilistic assignment or even deterministic no-assignment, depending on the context. The algorithmic scores are generally measured using supervised learning models that use the responsiveness of users. As such, there is some positive correlation between CATEs and belonging to the deterministic assignment region.
- *Advertising Auctions:* Digital ads are sold through auctions. In such settings, advertisers place bids per impression and win only when their submitted bid is the highest among all bidders. The advertiser’s submitted bid per impression for a user is a function of the CATE of that ad for the user (Waisman et al., 2019). The auction setting implies that an ad could never reach a certain user if the CATE for that user is too low because there will always be advertisers with higher bids for that user. This creates a form of *deterministic no-assignment*: there are some users in the control condition who could have never seen the ad because of their low valuation for the advertiser. Therefore, there will be a negative correlation between CATEs and belonging to the deterministic no-assignment region.

<sup>5</sup>Like Footnote 4, if propensity scores need to be estimated, it is crucial to perform trimming as weights are unequal but very close to 0 and 1 can heavily bias the estimates for ATE (Crump et al., 2009).



The examples above characterize practical settings where deterministic assignment happens in a way that violates the homogeneity of treatment effects across regions, i.e., we have  $\tau_0 \neq \tau_r \neq \tau_1$ . In particular, in the examples above, we expect to have  $\tau_0 \leq \tau_r \leq \tau_1$ , and therefore a large bias in any observational approach to estimate the ATE. We now formalize this intuition in the following proposition:

**Proposition 3.** *Let  $\tau(X_i)$  denote the CATE for observation unit  $i$ . We have:*

1. *If  $\tau(X_i)$  and belonging to the deterministic assignment region (i.e.,  $\mathbb{1}(\pi(X_i) = 1)$ ) are positively correlated, then we have  $\tau_1 \geq \tau^*$ .*
2. *If  $\tau(X_i)$  and belonging to the deterministic no-assignment region (i.e.,  $\mathbb{1}(\pi(X_i) = 0)$ ) are negatively correlated, then we have  $\tau_0 \leq \tau^*$ .*

*Proof.* See Web Appendix A.3. □

Proposition 3 is important because it shows that even a small correlation can link to a violation of  $\tau_0 \neq \tau_r \neq \tau_1$ . Therefore, unless we have a strong reason to believe that  $\tau_0 = \tau_r = \tau_1$ , the assumption is that the equality does not hold, especially in digital platforms that use algorithmic decision-making.

### 3.4 Simulation Experiments

#### 3.4.1 General Overlap Violation

In this section, we conduct simulation experiments with the general case of algorithmic decision-making as presented in section 3. In these cases, the platform has data:  $\mathcal{D} = \{Y_i, W_i, X_i, \pi(X_i)\}$  and wants to estimate effect of treatment  $W_i$  on  $Y_i$ . However, the assignment to  $W_i$  is through an algorithm  $\pi$  that can be partially deterministic. For every simulation, we need to set the number of observations  $N$ , matrix of covariates  $X_{N \times D}$ , CATE parameters  $\{\tau_0, \tau_1, \tau_r\}$  and their corresponding proportions  $\{\alpha_0, \alpha_1, \alpha_r\}$ . To generate data in our simulation experiments, we use the following step-by-step procedure:

- *Step 1:* We construct the treatment variable  $W_i$ . First, we define a raw score  $\pi^*(X_i)$  that can be any arbitrary function. We use the following function:  $\pi^*(X_i) = 1/(1 + e^{\sin(10X_{i,1}X_{i,2})})$ , to ensure that it is a complex function to learn. We then transform the score function  $\pi^*(X_i)$  into the propensity score function  $\pi(X_i)$ , such that the top  $\alpha_1$  and bottom  $\alpha_0$  fraction of  $\pi^*(X_i)$  values will be assigned to the deterministic assignment region ( $\pi(X_i) = 1$ ) and deterministic no-assignment region ( $\pi(X_i) = 0$ ), respectively. Let  $p_0$  and  $p_1$  denote the  $\alpha_0$  and  $1 - \alpha_1$

percentiles of  $\pi^*(X_i)$  values, respectively. We can write:

$$\pi(X_i) = \begin{cases} 0 & \text{if } \pi^*(X_i) < p_0 \\ (\pi^*(X_i) - p_0)/(p_1 - p_0) & \text{if } p_0 \leq \pi^*(X_i) \leq p_1 \\ 1 & \text{if } \pi^*(X_i) > p_1 \end{cases} \quad (12)$$

Once we have  $\pi(X_i)$  values, we generate treatment variables.

- *Step 2:* We use appropriate CATE value from  $\{\tau_0, \tau_1, \tau_r\}$  to calculate the outcome as follows:

$$Y_i = g(X_i) + W_i \left( \mathbb{1}(\pi(X_i) = 0)\tau_0 + \mathbb{1}(\pi(X_i) = 1)\tau_1 + \mathbb{1}(0 < \pi(X_i) < 1)\tau_r \right) + \epsilon_i, \quad (13)$$

where  $g(X_i) = X_{i,1}X_{i,2}$  is the nuisance function and structural error terms are drawn from a Standard Normal distribution, i.e.,  $\epsilon_i \sim \mathcal{N}(0, 1)$ .

We simulate data under different sets of parameters and estimate the Average Treatment Effect (ATE) using the following approaches: (1) Plain Mean Difference ( $\hat{\tau}_{MD}$ ), where we measure the mean difference between the treated and control units, (2) Ordinary Least Squares with controls  $\hat{\tau}_{OLS}$ , where we regress  $Y_i$  on  $W_i$ ,  $X_{i,1}$ ,  $X_{i,2}$ , and  $\pi(X_i)$ , (3) Double Machine Learning ( $\hat{\tau}_{DML}$ ), where we use a two-fold cross-validated random forest to estimate the nuisance function and directly plug-in the propensity scores, and (4) Inverse Propensity Score ( $\hat{\tau}_{IPS}$ ), where we directly use the actual propensity scores and estimate the ATE using Equation (10).

We present the results of our simulation experiments in Table 1. Each row in our table presents one simulation experiment, and the first three columns show the parameters needed to simulate data. All rows use the same set of CATE parameters, but the proportions generate different scenarios. In the first row, we only have deterministic assignment and probabilistic assignment, similar to the context of promotion assignments in many digital platforms where responsive users deterministically receive the promotion. In the second row, we only have deterministic no-assignment and probabilistic assignment, which resembles the case in advertising auctions where for some users, the propensity score is exactly equal to zero. The third case has two types of deterministic assignment but no probabilistic assignment, which is similar to cases where there is no built-in randomization by the platform, and the algorithm used is fully deterministic. The fourth row relates to a case where all three regions are represented, which commonly arises in settings where some thresholding rules are applied, such as promotions. With the exception of the third row, where the treatment effects are not identified in OLS, DML, and IPS cases due to lack of residual variation in treatments, these three methods can recover the treatment effect for the probabilistic region. However, as shown in Table 1, all these methods fail to recover the true ATE. The magnitude of the bias is the same as

$N$	$\{\tau_0, \tau_1, \tau_r\}$	$\{\alpha_0, \alpha_1, \alpha_r\}$	<i>True ATE</i> ( $\tau^*$ )	<i>Estimated ATE</i>			
				( $\hat{\tau}_{MD}$ )	( $\hat{\tau}_{OLS}$ )	( $\hat{\tau}_{DML}$ )	( $\hat{\tau}_{IPS}$ )
$10^4$	$\{1, 8, 2\}$	$\{0.4, 0.0, 0.6\}$	1.6	2.15	1.96	1.94	1.92
$10^4$	$\{1, 8, 2\}$	$\{0.0, 0.5, 0.5\}$	5.0	6.45	1.89	1.96	2.02
$10^4$	$\{1, 8, 2\}$	$\{0.5, 0.5, 0.0\}$	4.5	8.08	NA	NA	NA
$10^4$	$\{1, 8, 2\}$	$\{0.1, 0.4, 0.5\}$	4.3	6.06	2.13	2.05	2.00

Table 1: Estimates of Average Treatment Effects (ATE) using different estimators when propensity scores are known.

Corollary 1. Together, the results in Table 1 confirm the theoretical results presented in Propositions 1 and 2.

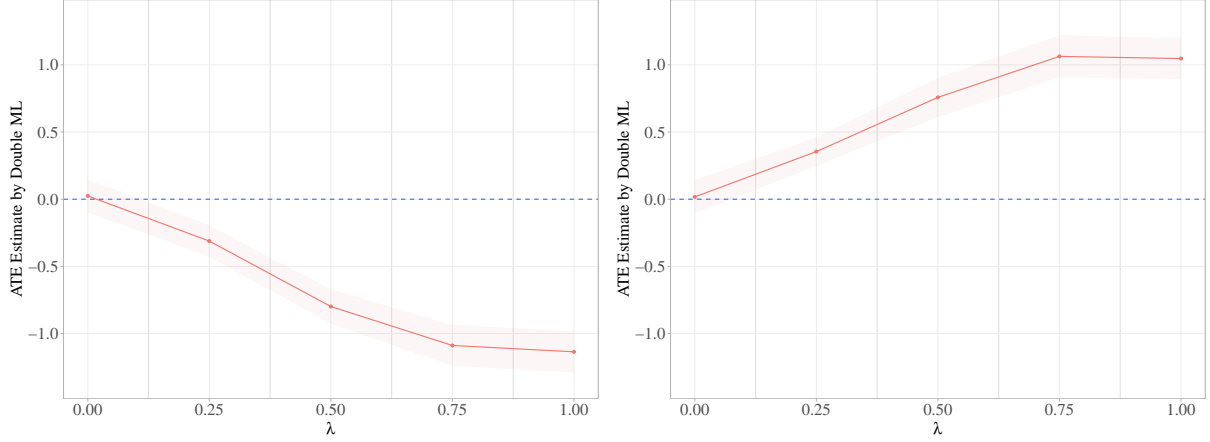
### 3.4.2 Special Cases Inspired by Real-World Settings

We now focus on special cases of interest where the propensity scores and CATEs are related, as discussed in §3.3. In particular, we focus on the following two cases, where (1) the right side of the CATE distribution is more likely to be assigned to deterministic assignment (e.g., promotions), and (2) the left side of the CATE distribution is more likely to be assigned to deterministic no-assignment (e.g., ad auctions). For our simulations, we simplify the former case to  $\{\alpha_0 = 0, \alpha_1 = 0.5, \alpha_r = 0.5\}$ , and the latter case to  $\{\alpha_0 = 0.5, \alpha_1 = 0, \alpha_r = 0.5\}$ . To operationalize these cases in our simulations, we start with the CATE function  $\tau(X_i)$  and set it arbitrarily. We then create scores  $\pi^*(X_i)$  as a function of CATEs as follows:

$$\pi^*(X_i) = \lambda\tau(X_i) + (1 - \lambda)u_i, \quad (14)$$

where  $u_i$  is random noise and  $\lambda$  controls for the level of correlation between  $\pi^*(X_i)$  and  $\tau(X_i)$ . To balance the noise, we set  $u_i$  as a shuffled version of  $\tau(X_i)$ . Once we have  $\pi^*(X_i)$ , we can simulate the data using the procedure in §3.4.1. We want to examine how the bias will change when we increase  $\lambda$  and make CATEs and propensity scores more correlated.

We set  $\tau(X_i) \sim \text{Poisson}(3)$  and run the simulations for a grid of  $\lambda \in \{0, 0.25, 0.50, 0.75, 1\}$  for both cases. For each value of  $\lambda$ , we repeat the simulation 20 times and calculate the average and standard deviation of the estimated ATE. Figure 2 shows the ATE estimates by Double ML against the values of  $\lambda$ . The dotted line demonstrates the true ATE in all cases. As shown in both figures, the bias in ATE estimates increases as the propensity scores and CATE values become more positively correlated. Importantly, even weaker associations still result in large biases, which



(a) Case 1: Deterministic assignment for higher CATEs (b) Case 2: Deterministic no-assignment for lower CATEs

Figure 2: Estimated ATE by Double ML when CATE and propensity scores are positively correlated.

suggests that even the use of algorithms poor in performance can still produce large biases. The results in Figure 2 can also explain why observational methods perform poorly in digital contexts where platforms use algorithmic decision-making (Gordon et al., 2022).

## 4 Observational Solution to Overlap Violation

In the previous section, we presented the challenge digital platforms face due to the lack of overlap in observational studies. The problem stems from the deterministic outputs of algorithms that are used for decision-making in these platforms. Our theoretical analysis shows the extent to which observational methods can produce largely biased and inconsistent estimates of the average treatment effect when the overlap assumption is violated.

In this section, we seek to find a solution to this challenge. As such, our goal is to use the existing data to recover the average treatment effects. In this section, we explicitly state our assumptions and data requirements and discuss a novel solution based on machine learning methods. We first formally define our problem in §4.1, where we present the identification problem caused by the lack of overlap as a missing data problem. Next, in §4.2, we present our solution to the problem and the assumptions under which this solution works. Finally, in §4.3, we present a series of simulated experiments to show how our model performs under different scenarios.

### 4.1 Lack of Overlap as a Missing Data Problem

As discussed earlier, the fundamental problem with the deterministic assignment is one of identification. In light of Lemma 1, we know that with the current set of assumptions, the parameters  $\tau_1$  and

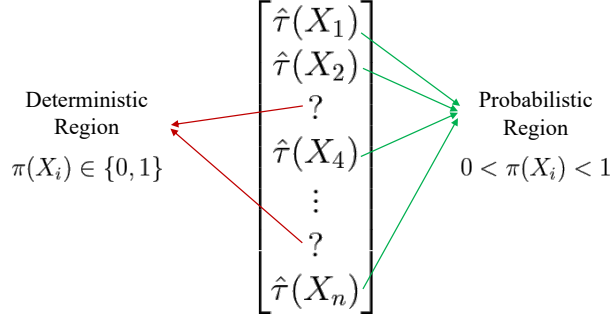


Figure 3: An illustration of the missing data problem due to the overlap violation.

$\tau_0$  cannot be identified because there is no variation in the treatment variable when accounting for the propensity score. In general, we can write the conditional average treatment effect as follows:

$$\tau^*(x) = \mathbb{E}[Y_i(1) - Y_i(0) \mid X_i = x] = \mu_1(X_i) - \mu_0(X_i), \quad (15)$$

where  $\mu_w(x)$  is the population function for potential outcomes conditional on  $x$  when assigned to treatment  $w$ . From a learning standpoint, if one of the two treatment states could have never been generated in the data, no model can estimate the corresponding  $\mu$  function. For example, if a unit with covariates  $X_i$  could have never received the treatment, we have no observation in our data to estimate  $\mu_1(X_i)$ . As such, the problem caused by the lack of overlap is one of missing data. That is, for a single treatment, the vector of CATE estimates has missing values for observations in the deterministic regions. Figure 3 visualizes this insight, where the CATE estimates are question marks for observations where the overlap assumption is violated.

We now turn to the question of what variation would allow us to impute these question marks. From our earlier results, we know that with only the data of a single treatment, it is not possible to identify these question marks. However, we argue that having the data on a set of other treatments for the same set of observation units (e.g., users) can potentially help. That is, instead of exploiting the within-study variation, we can exploit between-study variation. Such a setting is common among digital platforms that deliver different treatments at a large scale. Motivated by this insight, we define the problem of the digital platform as follows:

**Definition 4.** Consider a digital platform that has data from multiple studies indexed by  $j$  from 1 to  $J$ . Each study involves a binary treatment variable denoted by  $W^{(j)}$ , where the value for the  $i^{\text{th}}$  observation is either zero or one, i.e.,  $W_i^{(j)} \in \{0, 1\}$ . For each study  $j$ , the platform has the data  $\mathcal{D}^{(j)} = \{Y_i^{(j)}, W_i^{(j)}, X_i, \pi^{(j)}(X_i)\}$ , which collectively makes the data  $\mathcal{D}_T = \bigcup_{j=1}^J \mathcal{D}^{(j)}$ . The

platform's goal is to recover the following matrix:

$$\mathcal{T} = \begin{bmatrix} \tau^{(1)}(X_1) & \tau^{(2)}(X_1) & \dots & \tau^{(J)}(X_1) \\ \tau^{(1)}(X_2) & \tau^{(2)}(X_2) & \dots & \tau^{(J)}(X_2) \\ \vdots & \vdots & \ddots & \vdots \\ \tau^{(1)}(X_N) & \tau^{(2)}(X_N) & \dots & \tau^{(J)}(X_N) \end{bmatrix}, \quad (16)$$

where  $\tau^{(j)}(X_i)$  is the CATE from the treatment in study  $j$  for observation unit  $i$ . Formally, we can define this estimand as follows:

$$\tau^{(j)}(X_i) = \mathbb{E}[Y_i^{(j)}(1) - Y_i^{(j)}(0) \mid X_i]. \quad (17)$$

If the digital platform achieves the objective in Definition 4, it can recover the average treatment effect for the treatment in each study.

A few points are worth noting about the setting and data requirements presented in Definition 4. First, treatments in different studies can be different. For example, the treatment in study  $j$  and  $k$  can be whether a user receives a certain movie recommendation and whether a user receives a free-trial offer. One could imagine this as different interventions the platform made over time.<sup>6</sup> Second, for each study, we need to have the same set of observation units that form rows in the matrix in Equation 16. As such, one user can be assigned to multiple treatments (e.g., both the movie recommendation and the free trial in the example above). Third, it is important to emphasize that this data requirement is not excessive, as companies often run numerous different treatments over a short period of time.

## 4.2 Solution Concept

We now present our solution to the problem presented in Definition 4. We first propose the algorithm used for obtaining all the CATE values in Equation (16) in §4.2.1. We then discuss the assumptions that we need for identification in §4.2.2.

### 4.2.1 Proposed Algorithm

Before we present our algorithm, we need to define some model preliminaries. As mentioned earlier, the goal of our algorithm is to estimate CATE for all the elements in the matrix despite the overlap violation. To do so, we first need to know which elements we cannot estimate with the conventional

---

<sup>6</sup>If studies were concurrent, there is the possibility of interference. In our study, we assume no interference.

methods to estimate CATE. Therefore, we define the propensity matrix as follows:

$$\Pi = \begin{bmatrix} \pi^{(1)}(X_1) & \pi^{(2)}(X_1) & \dots & \pi^{(J)}(X_1) \\ \pi^{(1)}(X_2) & \pi^{(2)}(X_2) & \dots & \pi^{(J)}(X_2) \\ \vdots & \vdots & \ddots & \vdots \\ \pi^{(1)}(X_N) & \pi^{(2)}(X_N) & \dots & \pi^{(J)}(X_N) \end{bmatrix}, \quad (18)$$

where each element  $\Pi_{i,j}$  denotes the propensity score for the treatment in study  $j$  for unit  $i$ , i.e.,  $\Pi_{i,j} = \pi^{(j)}(X_i) = \Pr(W_i^{(j)} = 1 \mid X_i)$ . As such, the deterministic regions for each treatment are defined as rows where the propensity score is either zero or one. We know that the conditional average treatment effect is unidentified for these units. Thus, we define a feasibility matrix  $F$  that takes value one only when the assignment is probabilistic; that is, the propensity score is strictly between zero and one. As such, we can write each element of this matrix as follows:

$$F = \begin{bmatrix} \mathbb{1}(0 < \pi^{(1)}(X_1) < 1) & \mathbb{1}(0 < \pi^{(2)}(X_1) < 1) & \dots & \mathbb{1}(0 < \pi^{(J)}(X_1) < 1) \\ \mathbb{1}(0 < \pi^{(1)}(X_2) < 1) & \mathbb{1}(0 < \pi^{(2)}(X_2) < 1) & \dots & \mathbb{1}(0 < \pi^{(J)}(X_2) < 1) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbb{1}(0 < \pi^{(1)}(X_N) < 1) & \mathbb{1}(0 < \pi^{(2)}(X_N) < 1) & \dots & \mathbb{1}(0 < \pi^{(J)}(X_N) < 1) \end{bmatrix}. \quad (19)$$

The feasibility matrix  $F$  determines the scope of our CATE estimation. That is, if for treatment  $j$  in unit  $i$ , we have  $F_{i,j} = 0$ , Lemma 1 implies that we cannot identify  $\tau^{(j)}(X_i)$ . However, if  $F_{i,j} = 1$ , we can use conventional CATE estimators to estimate  $\tau^{(j)}(X_i)$ , because  $\pi^{(j)}(X_i)$  is probabilistic and the setting satisfies the unconfoundedness assumption. Therefore,  $F$  determines what is identifiable and transforms the problem in Definition 4 into a matrix completion problem, where we have an estimated CATE matrix  $\hat{\mathcal{T}}^{\text{incomplete}}$  and each element  $[i, j]$  is defined as follows:

$$\hat{\mathcal{T}}_{i,j}^{\text{incomplete}} = \begin{cases} \hat{\tau}^{(j)}(X_i) & \text{if } F_{i,j} = 1 \\ ? & \text{if } F_{i,j} = 0 \end{cases} \quad (20)$$

As shown in Equation (20),  $F$  determines the question marks in our matrix completion task. We now have an incomplete matrix  $\hat{\mathcal{T}}^{\text{incomplete}}$ , where the incomplete elements are the overlap-violating regions. If the underlying matrix  $\mathcal{T}$  is low-rank, we can use conventional matrix decomposition techniques to impute the question marks. This procedure exploits the similarities in the joint space of units and treatments. We denote this new completed matrix by  $\hat{\mathcal{T}}^{\text{complete}}$ . Algorithm 1 presents the details of our proposed approach.

The output of this algorithm is a complete matrix  $\hat{\mathcal{T}}^{\text{complete}}$  where all the elements are imputed.

---

**Algorithm 1** Matrix Completion for CATE Estimation

---

**Input:**  $\mathcal{D}_T$  ▷ From Definition 4  
**Output:**  $\hat{\mathcal{T}}^{\text{complete}}$

```
1:  $F \leftarrow \mathbb{1}(0 < \Pi < 1)$ 
2: for  $j = 1 \rightarrow J$  do
3:    $\hat{\tau}^{(j)} \leftarrow \text{learnCATE}(Y_i^{(j)}, W_i^{(j)}, \{X_i, \pi^{(j)}(X_i)\})$  ▷ Can be any CATE learner
4:   for  $i = 1 \rightarrow N$  do
5:      $\hat{\mathcal{T}}_{i,j}^{\text{incomplete}} \leftarrow \hat{\tau}^{(j)}(X_i)$ 
6:     if  $F_{i,j} = 0$  then
7:        $\hat{\mathcal{T}}_{i,j}^{\text{incomplete}} \leftarrow ?$ 
8:     end if
9:   end for
10: end for
11:  $\hat{\mathcal{T}}^{\text{complete}} \leftarrow \text{Complete}(\hat{\mathcal{T}}^{\text{incomplete}})$ 
```

---

This complete matrix can then be used to estimate the ATE from the data. For each treatment in study  $j$ , we can recover the average treatment effect as follows:

$$\hat{\tau}^{(j)} = \frac{1}{N} \sum_{i=1}^N \hat{\mathcal{T}}_{i,j}^{\text{complete}}. \quad (21)$$

If the matrix  $\mathcal{T}$  is low-rank,  $\hat{\tau}^{(j)}$  is a bias-corrected version of the ATE for treatment  $j$ . If the matrix is not low-rank, the approach is still effective at reducing bias in the ATE estimates.

#### 4.2.2 Assumptions and Identification

We now discuss the assumptions that we need for the matrix completion approach to recover the true average treatment effects. At a high level, our identification claim is that for each observation unit in an overlap-violating region ( $F_{i,j} = 0$ ), if we have enough cross-study variation, we can exploit the similarities in the data to impute the conditional average treatment effect for that observation unit. The following example helps illustrate the intuition behind our identification. Suppose that the treatment assignment in study  $j$  is deterministic for user  $i$ . As such, the CATE for this entry ( $\tau_i^{(j)}$ ) cannot be identified using the data for study  $j$ . Now, suppose that there is another treatment  $j'$  that has a probabilistic assignment for unit  $i$ , so we can estimate the CATE of  $j'$  for unit  $i$ . If the two treatments exhibit very similar patterns for the units where they can both feasibly estimate the CATE, we can use the CATE of  $j'$  for unit  $i$  to impute the CATE of  $j$  for unit  $i$ . Similarly, if there is another unit  $i'$  that has a probabilistic assignment for treatment  $j$  and is very similar to unit  $i$  for most treatments, we can use this similar unit's CATE to impute the missing entry for unit  $i$ .



The simple example above only illustrates what kind of variation we use in our method. However, such exact similarities may be difficult to find in reality, especially if we have to search on a case-by-case basis. Therefore, for this method to work, we need a more systematic way to capture the similarities in the space of treatments. This is why we use a matrix completion approach that has been widely used for collaborative filtering. In our setting, we have an incomplete and noisy version of the true CATE matrix. The entries are noisy because the estimated CATE will have some errors. The identification task at hand is to identify the complete CATE matrix and estimate ATEs. To perform this task with standard matrix completion algorithms, we need assumptions on (1) the rank of the matrix, (2) the missingness pattern, and (3) noise in the observed entries. We present these three assumptions and discuss each in this section.

**Assumption 1.** *The underlying CATE matrix  $\mathcal{T}$  is low-rank; that is, for  $R \ll \min(N, J)$ , there exist two matrices  $P_{N \times R}$  and  $Q_{J \times R}$  such that  $\mathcal{T} = PQ^T$ .*

At a very high level, this assumption suggests that the user response exhibits some common patterns across different treatments. More specifically, Assumption 1 implies that CATE values across studies come from a linear combination of a few factors that are defined at the individual level. In that sense, this assumption is close to those commonly made in the structural economics literature that imposes a micro-foundation that allows only a few factors to drive user behavior. For example, in the context of promotional treatments, we expect a few structural parameters to determine most of the treatment effects, such as users' price sensitivity, search cost, etc. We illustrate this insight formally in the following equation:

$$\mathcal{T} = \begin{bmatrix} \overbrace{ps(X_1)}^{\text{price sensitivity}} & \overbrace{sc(X_1)}^{\text{search cost}} & \dots \\ ps(X_2) & sc(X_2) & \dots \\ \vdots & \vdots & \ddots \\ ps(X_N) & sc(X_N) & \dots \end{bmatrix} \times \begin{bmatrix} \overbrace{w_1^{(1)}}^{\text{Study 1 Weights}} & \overbrace{w_1^{(2)}}^{\text{Study 2 Weights}} & \dots & \overbrace{w_1^{(J)}}^{\text{Study J Weights}} \\ w_2^{(1)} & w_2^{(2)} & \dots & w_2^{(J)} \\ \vdots & \vdots & \ddots & \vdots \end{bmatrix},$$

where factors include individual-level primitives such as price sensitivity and search cost that can be any complex function of covariates, and weights determine how much these factors matter in driving the treatment effect for each study. We need to stress that a greater homogeneity and commonality in the structure of different studies makes the low-rank assumption more suitable. For example, if studies are completely unrelated, the low-rank assumption will be less realistic. In other words, more than only a few factors determine the treatment effects across all studies. However, the homogeneity of studies is a condition that is likely satisfied in most digital platforms as interventions likely share some common characteristics.

More generally, we can view the low-rank assumption in our setting through the structure of the CATE matrix. Let  $X_{N \times D}$  denote the covariate matrix where each row represents a user and each column represents a covariate. The CATE from treatment  $j$  for unit  $i$  is  $\tau^{(j)}(X_i)$ , which is a function of the covariates. For each treatment  $j$ , there is a  $D$ -dimensional vector of coefficients  $\beta^{(j)}$  that determine the CATE value such that  $\tau^{(j)}(X_i) = \beta^{(j)} X_i^T$ . This linear approximation is reasonable as  $D$  can be large. Now, we can write the CATE matrix  $\mathcal{T}$  as follows:

$$\mathcal{T} = XB^T, \quad (22)$$

where  $B$  is a  $J \times D$  matrix where each column is the vector of coefficients for CATE for a specific treatment. For the low-rank assumption to be satisfied, we need matrix  $B$  to be low-rank. If the studies have similar characteristics, we expect weights in each row of  $B$  to be correlated, thereby making the matrix low-rank. Suppose there are two matrices  $U_{J \times R}$  and  $V_{D \times R}$  such that  $B = UV^T$ . In this case,  $\mathcal{T} = XUV^T$ , where  $XV$  maps the high-dimensional covariates into  $R$  factors, and  $U$  contains the weights for these factors in the different studies.

Apart from structural reasons for the suitability of low-rank assumption in the context of digital platforms, the insights from the prior literature suggest that the low-rank assumption performs remarkably well in a wide range of domains, especially when large-scale matrices are available. This insight is formally characterized in Udell and Townsend (2019) who show that under general conditions that the function generating the high dimensional  $N \times J$  matrix is analytic piece-wise, the rank grows as  $O(\log(N + J))$ .

The second set of assumptions for matrix completion to work relates to the missingness pattern. In our setting, feasibility matrix  $F$  produces the missingness pattern in the CATE matrix. Most of the prior theoretical literature on matrix completion assumes fully random missingness to derive theoretical results on the recovery of the matrix (Candès and Recht, 2009; Mazumder et al., 2010; Chen et al., 2019). More recent papers extend these theoretical results to specific non-random missingness patterns (Ma and Chen, 2019; Athey et al., 2021; Agarwal et al., 2021). In general, a common factor in all this literature is to assume that the missingness pattern does not affect the identification of factors. We make the following assumption and validate it through our simulations.

**Assumption 2.** *The missingness pattern  $F$  is such that factors can be identified using the observed entries.*

Intuitively, the missingness pattern needs to be such that we can jointly exploit the similarities between users and between treatments. As such, if the data are missing for an entire column, there is no way to recover the parameters for that column. Likewise, if the data are entirely missing for a

row, the matrix completion approach cannot exploit the similarities in any way. Thus, although the missingness pattern can be non-random, enough entries are needed for each row and each column.

Finally, since our task at hand is completing a noisy matrix, we need to impose some structure on the noise added to entries. The entries in our incomplete matrix are CATE estimates with a non-random sample of data. In general, we have  $\mathcal{T} = \hat{\mathcal{T}} + E$ , where  $E$  is the error in CATE estimates. We impose the following assumption on the calibratedness of our CATE estimates:

**Assumption 3.** *The error in CATE estimates is mean zero and independent and identically distributed in the matrix, such that we have  $\mathbb{E}[\hat{\mathcal{T}} \mid P, Q] = \mathbb{E}[\hat{\mathcal{T}} \mid P, Q, F]$ .*

This assumption implies that the missingness pattern does not affect the consistency of CATE estimates at the user level. This assumption holds in cases with a sufficiently large  $N$  where for each point, there are sufficient close neighbors in the data. However, in small data settings, CATE estimates can be more sensitive to adversarial missingness patterns.

In summary, Assumptions 1–3 are standard assumptions in matrix completion literature and characterize an environment where we can apply Algorithm 1 and recover ATEs. Notably, these assumptions are specifically likely to be satisfied in the context of digital platforms because studies have some commonality that creates a low-rank environment (Assumption 1), and the scale of the user base is large enough to make the missingness pattern more suitable (Assumption 2) and CATE estimates more calibrated (Assumption 3).

### 4.3 Simulation Experiments

In this section, we deliver a series of simulation experiments using synthetic data to validate our proposed algorithm. We consider a variety of cases that reflect real-world scenarios. Each scenario corresponds to a certain missingness pattern (random or non-random) and the extent of missingness. To show how our proposed algorithm performs, we need to make a ground truth CATE matrix  $\mathcal{T}$  with  $N$  rows that represent users and  $J$  columns that represent studies. As discussed earlier in §4.2.2, we can define the ground truth CATE matrix as the product of the covariate matrix  $X_{N \times D}$  and the transpose of the coefficient matrix  $B_{J \times D}$  as follows:

$$\mathcal{T} = XB^T. \quad (23)$$

We further decompose matrix  $B$  to control the rank of the CATE matrix for our experiment. We define two matrices  $U_{J \times R}$  and  $V_{D \times R}$  where we have:

$$B = UV^T, \quad (24)$$

where  $R$  controls the rank of the  $N \times J$  CATE matrix. Intuitively, we can interpret  $XV$  as  $R$  principal components (factors) that collectively define a CATE for the treatment in a specific study through some weights. These weights are specified for each of the  $J$  studies in matrix  $U$ . Together,  $XVU^T$  gives us the underlying CATE matrix  $\mathcal{T}$ , which is low-rank.

Our goal in the simulation experiments is to generate data  $\mathcal{D}_T$ , as defined in Definition 4. This data set is the union of data sets corresponding to each study  $j$ . To generate  $\mathcal{D}_T$ , we need three inputs: (1) CATE matrix  $\mathcal{T}$  that determines the treatment effect for each observation, (2) propensity matrix  $\Pi$  as defined in (18), and (3) nuisance matrix  $\mathcal{G}$  that determines the relationship between covariates and the outcome. We can use these three inputs and simulate  $\mathcal{D}_T^{\text{sim}}$  using the following procedure:

- *Step 1:* We use  $\Pi$  to simulate  $W_i^{(j)}$  for each unit  $i$  in each study  $j$ .
- *Step 2:* With the treatment variable realized, we can simulate the outcome as follows:

$$Y_i^{(j)} = \mathcal{G}_{i,j} + W_i^{(j)}\mathcal{T}_{i,j} + \epsilon_{i,j}, \quad (25)$$

where  $\mathcal{G}_{i,j}$  is the nuisance part of the outcome,  $W_i^{(j)}\mathcal{T}_{i,j}$  is the treatment effect given (if any), and  $\epsilon_{i,j} \sim \mathcal{N}(0, 1)$ .

- *Step 3:* For each study  $j$ , we can construct data set  $\tilde{\mathcal{D}}^{(j)} = \{Y_i^{(j)}, W_i^{(j)}, X_i, \pi^{(j)}(X_i)\}$ . The union of  $\tilde{\mathcal{D}}^{(j)}$  for all  $j$ 's will give us the  $\mathcal{D}_T^{\text{sim}}$ .

For our simulations, we use  $N = 1000$ ,  $D = 50$ , and  $J = 100$ . We set the rank of the CATE matrix as  $R = 10$  and generate two random matrices  $X_{N \times D}$  and  $B_{J \times D}$ , where each element of each matrix comes from  $\mathcal{N}(0, 1)$ . We generate another coefficient matrix  $G_{J \times D}$  from  $\mathcal{N}(0, 1)$  to generate the nuisance matrix  $\mathcal{G}$  as follows:

$$\mathcal{G} = XB^T. \quad (26)$$

What varies across our simulation experiments is the missingness patterns that is operationalized by  $\Pi$ . We formalize this matrix in the following sections. Once we generate the data  $\mathcal{D}_T^{\text{sim}}$  in a specific condition, we can apply our proposed method in Algorithm 1. Because we want to compare the performance of our proposed method with conventional methods such as Double ML, we use U-learner as our CATE estimator that directly uses Robinson's decomposition (Robinson, 1988).

In our simulation scenarios, we distinguish between two forms of missingness: random and non-random. Random missingness is similar to cases where platforms choose a very small but random subsample of their users and run the experiment. In these cases, we expect all the conventional methods such as Double ML to work well. Non-random missingness scenarios are those where we expect to see the difference between our proposed method and Double ML. We expect our method

to perform better than Double ML methods in these cases.

#### 4.3.1 Random Missingness Pattern

We start with the most well-known missingness pattern used in matrix completion problems: missing-completely-at-random (MCAR). In this scenario, each entry in the matrix has a uniform probability  $p$  of being missing. As such, each element in our feasibility matrix takes value zero with probability  $p$ , and one with probability  $1 - p$ . Specifically, we can write the propensity scores as follows:

$$\Pi_{i,j} = \begin{cases} 0 & \text{with prob } p/2 \\ 1/2 & \text{with prob } 1 - p \\ 1 & \text{with prob } p/2 \end{cases} \quad (27)$$

Random missingness of the elements in a matrix highlights the key intuition behind trimming approaches: if the overlap-violating regions are selected at random, then conventional models can recover the average treatment effect (ATE) as discussed earlier in §3.3. We consider two cases with low and high missingness at random, using  $p = 0.25$  and  $p = 0.75$  for the low and high missingness cases, respectively. Figure 4 shows the estimates obtained by Double ML (shown in red) and our proposed algorithm (shown in green) and compares them with the ground truth (black line). This figure reveals an important insight: even in the random missingness case where the conventional methods can theoretically recover the true parameters, our proposed algorithm performs better than conventional methods, especially for more sparse data. We demonstrate this point using richer simulation studies in Appendix B.1.

#### 4.3.2 CATE-Dependent Missingness: Case of Algorithmic Decision-Making

In real-world scenarios, we do not expect to have a random missingness pattern. As discussed in §3.3, we expect the overlap-violating regions to be correlated with the CATE in the context of algorithmic decision-making. This case is more troublesome as the conventional approaches can produce arbitrarily biased estimates. Our goal is to see how our proposed approach performs under these scenarios. The first step is to define a procedure that induces such CATE-dependent missingness patterns. To do so, we present the  $\lambda$ -adversariality score as follows:

**Definition 5.** Let  $s_{i,j}$  denote the percentile of CATE for user  $i$  in study  $j$  in the distribution of CATEs in study  $j$ , i.e.,  $s_i^{(j)} = (\sum_{i=1}^N \mathbb{1}(\tau^{(j)}(X_i) > \tau^{(j)}(X_k)))/N$ . We define  $\lambda$ -adversariality scores as follows:

$$r_{i,j}^{(\lambda)} = \lambda s_{i,j} + (1 - \lambda) u_{i,j}, \quad (28)$$

where  $u_{i,j} \sim \mathcal{U}(0, 1)$  adds some noise to the percentile of the CATE estimate for user  $i$  in study  $j$ ,

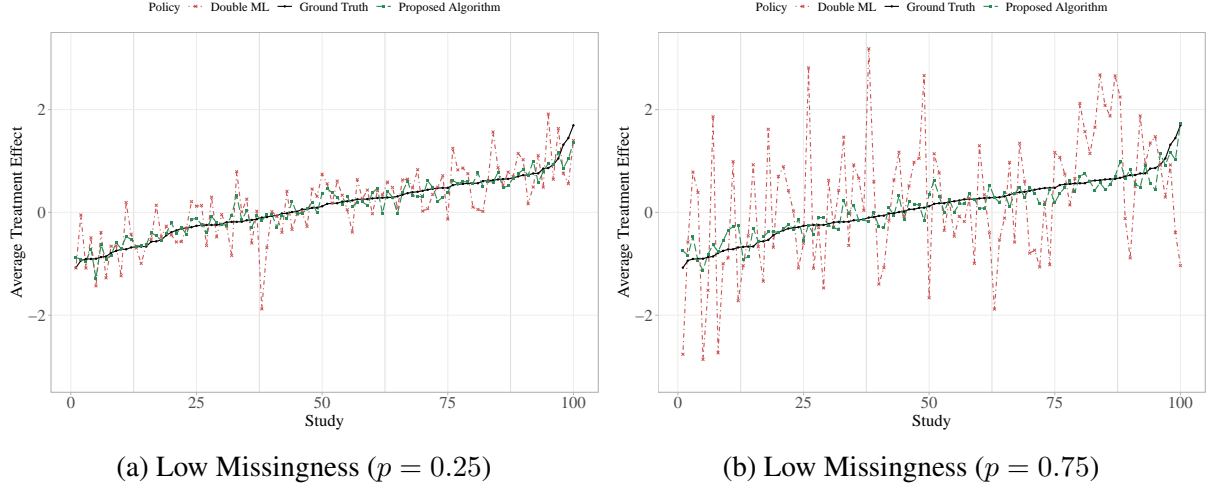


Figure 4: The performance of our proposed algorithm and conventional observational method when missingness is at random with a uniform probability. Each figure represents a level of missingness.

and  $\lambda$  controls the extent to which adversariality score uses the signal.

For the fully adversarial case, we can use  $\lambda = 1$ . However, we expect some imperfectness in scores that determine regions with deterministic assignment, which is why we use the definition above to be able to manipulate the level of adversariality of data. In general, higher values of  $r_{i,j}^{(\lambda)}$  are associated with higher CATE values. We use this score variable to design different scenarios where higher or lower CATEs are systematically missing. To that end, we consider the following types of missingness:

- *Right-tail missingness of CATE*: The first scenario is a missingness pattern whereby elements with higher CATE are more likely to have a deterministic assignment. We use the  $\lambda$ -adversariality score and generate this missingness pattern as follows:

$$\Pi_{i,j} = \begin{cases} 1/2 & \text{if } r_{i,j}^{(\lambda=0.25)} \leq c \\ 1 & \text{if } r_{i,j}^{(\lambda=0.25)} > c \end{cases}, \quad (29)$$

where  $c$  can be any threshold that determines what proportion of data is missing. We use a median split that creates 50% missingness. As shown in Equation (29), units with a higher adversariality score than the threshold will be in the deterministic assignment region. We expect this pattern to make conventional methods downward biased as the right tail of the treatment effect distribution is censored.

- *Left-tail missingness of CATE*: The second scenario is a CATE-dependent missingness pattern

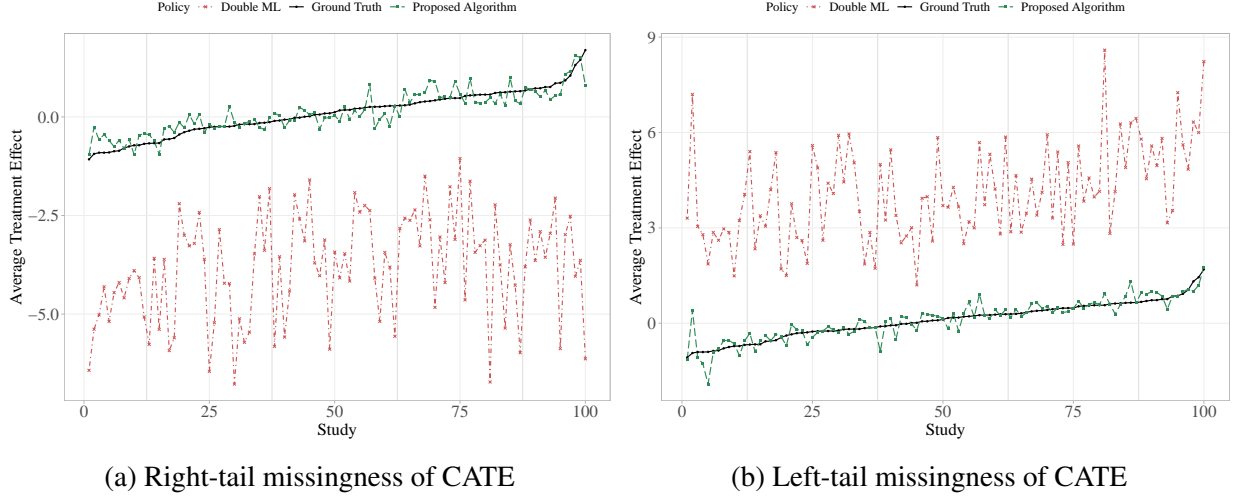


Figure 5: The performance of our proposed algorithm and conventional observational method when missingness depends on CATE.

that happens at the lower end of the CATE distribution. We use the same insight as before for constructing the missingness pattern based on the  $\lambda$ -adversariality score. We can write:

$$\Pi_{i,j} = \begin{cases} 0 & \text{if } r_{i,j}^{(\lambda=0.25)} \leq c \\ 1/2 & \text{if } r_{i,j}^{(\lambda=0.25)} > c \end{cases}, \quad (30)$$

where we use the median of  $r_{i,j}^{(\lambda=0.25)}$  values as the threshold to induce 50% missingness. In this case, units with a lower adversariality score than the threshold will be in the deterministic no-assignment region.

Figure 5 compares the performance of our proposed algorithm with that of conventional methods such as Double ML in recovering ATE. As shown in the two figures, when the missingness is more profound at the right (left) tail of the CATE distribution, conventional methods underestimate (overestimate) the true ATE. However, in both cases, our proposed algorithm is able to recover the true parameters as it exploits the between-study variation and the low-rank structure of the CATE matrix. In Appendix B.2, we consider a wider set of adversarial cases, specifically by changing the  $\lambda$  parameter and increasing the level of adversariality. The result of this practice confirms the main insight in this section: although conventional methods fail to recover the true ATE, our proposed algorithm can accurately recover the true parameters.

## 5 Empirical Application

So far, we have demonstrated the performance of our proposed algorithm using synthetic data. In this section, we use data from a real experiment to examine how our algorithm performs in real-world settings. We first describe the design of our experiment in §5.1. We then show some descriptive statistics on the data in §5.2. Finally, in §5.3, we perform our analysis and compare the performance of our algorithm with other benchmarks.

### 5.1 Experiment

To apply and evaluate our proposed algorithm, we need our data to have a few certain features. First, our matrix completion algorithm requires multiple studies (columns) for the same group of individuals (rows). This requirement imposes a challenge as most available data sets have either a single study or multiple studies on non-overlapping groups of individuals.<sup>7</sup> Second, we need to know the ground truth Average Treatment Effect (ATE) for each study to be able to evaluate how well our algorithm performs when the overlap assumption is violated. Third, we need a setting with observable heterogeneity in treatment effects since the lack of overlap only biases the estimates of the ATE when there is heterogeneity in treatment effects, as shown in Corollary 1. As such, we need a setting where we expect heterogeneity in treatment effects that we can identify with the vector of covariates.

We want to design an experiment that satisfies the requirements discussed above. To do so, we create a quiz game in Qualtrics, where participants are asked a series of multiple-choice questions. We design an approximation game where each question provides information about an object’s size and asks participants to guess the size of another object in the image. The reward for a correct answer is 1 point, which is equivalent to \$0.10. However, users can choose an easier version of the question with only two choices at a lower reward. For each question, we randomize the amount of reward for the easy version, such that users are randomly assigned to a better or worse deal. Figure 6 illustrates the two experimental conditions in one of the questions. As shown in this figure, users are asked to approximate the full length of the video given that the amount watched is 10 seconds. In the original (difficult) version of the question, users see four choices that they can choose from. The fifth option in both conditions is a deal whereby users can answer an easier version of the question at a lower reward. However, users are randomly assigned to two different rewards for the easy version, specifically, in this case, 0.5 points or 0.8 points.

We design a total of 20 questions in this game. All questions are about the geometric approxima-

---

<sup>7</sup>It is worth emphasizing that this requirement is easily satisfied in most digital platform settings as they deliver interventions on the same user base.



**Question 1 (Reward = 1)**

If the amount watched is 10 seconds (red area), what is the length of the total video?



57 seconds

61 seconds

53 seconds

42 seconds

Make it easier by removing 2 choices;  
**New Reward = 0.5**

Control

**Question 1 (Reward = 1)**

If the amount watched is 10 seconds (red area), what is the length of the total video?



61 seconds

42 seconds

53 seconds

57 seconds

Make it easier by removing 2 choices;  
**New Reward = 0.8**

Treatment

Figure 6: Experimental conditions in Question 1 of the game

tion of objects given limited inputs. For each question, users are randomly assigned to the better or worse deal, which means that over the course of playing the game, one user can be assigned to both control and treatment multiple times. The randomization for each question is independent of other questions. We then collect outcomes per question, such as choice of the easy version, choice of the correct answer, and reward earned. Theoretically, we expect the appeal of the treatment condition (a higher reward deal) in each question to be moderated by users' perceived level of expertise, price sensitivity, and risk preferences. As such, we collect information on these variables along with users' demographic characteristics to collect a rich set of covariates for the purpose of estimating heterogeneous treatment effects. We describe the full details of our experiment in Appendix C.1 and present a theoretical analysis of where heterogeneity in treatment effects comes from in Appendix C.2.

Overall, the design of our experiment satisfies all three features we discussed earlier: (1) we have 20 studies (questions) for the same set of users, (2) we know the ground truth ATE because we randomize the treatment assignment for each question, and (3) we collect a rich set of covariates on participants skills, risk preferences, and demographics, many of which likely determine the heterogeneity in treatment effects. An important advantage of our experiment is that it follows the structure in many popular quiz apps that monetize based on in-app purchases.

## 5.2 Data

We ran our experiment on 1511 participants recruited from Mechanical Turk (MTurk). Each participant was paid a flat rate of \$1 for the completion of the survey and \$0.10 bonus for each reward point. The maximum reward for 20 questions in our study is 20 points, which translates into a maximum bonus of \$2 for users. To increase the reliability of our analysis, we drop 139 observations generated by users who spent less than five minutes on the experiment.<sup>8</sup> Further, we drop an additional 21 users for whom the treatment status is missing in some questions due to some minor technical issues in the data collection phase. This gives us a sample of 1351 users with a full record of treatment status, actions, and performance measures.

We now share some summary statistics about our data and experiment in Table 2. As shown in this table, the average time users spent on the entire study is over 11 minutes, with a very wide range. Given the 50-50 split in each experiment, the average number of assignments to the treatment condition (better deal) is 10.02, which is half of the questions. We also notice that all users have been assigned to both treatment and control conditions. The average number of times the easy version is picked is 4.22, which indicates that only 20% of times users chose the easy version. However, the large standard deviation and the existence of users who only chose the difficult or easy version throughout reveal considerable heterogeneity in user behavior. The number of correct answers to these questions is 8.86 out of 20. The rewards earned average at 7.85, or equivalently  $7.85 \times \$0.10 = \$0.785$ . The majority of rewards earned come from difficult questions as users choose the difficult version more often. We present the analog of Table 2 for pre-treatment variables in Appendix §C.3.

<b>Variable</b>	<b>Mean</b>	<b>SD</b>	<b>Min</b>	<b>Median</b>	<b>Max</b>
<i>Duration (in seconds)</i>	698.05	364.31	301	604.0	5472
<i>Number of Treatment Assignments</i>	10.02	2.20	3	10	17
<i>Number of Easy Questions Picked</i>	4.22	5.26	0	2	20
<i>Number of Correct Answers</i>	8.86	3.07	1	9	20
<i>Rewards Earned</i>	7.85	2.77	1	7.6	20
<i>Rewards Earned from Difficult Questions</i>	6.40	3.49	0	6.0	20
<i>Rewards Earned from Easy Questions</i>	1.45	2.02	0	0.5	11

Table 2: Summary statistics of user actions and performance measures.

Finally, we run extensive randomization checks to ensure that all our experiments have been implemented properly. As a simple test, we regress the treatment variable on all the pre-treatment covariates available up to the treatment assignment. We find that the F-tests for all 20 regressions

<sup>8</sup>It is worth emphasizing that our results are robust to the inclusion of these users.

fail to reject the null hypothesis, which indicates that the pre-treatment variables do not explain the treatment assignment (please see Appendix C.4).

### 5.3 Performance Analysis

In this section, we compare the performance of our proposed algorithm to the existing state-of-the-art benchmarks for estimating ATE. We focus on two adversarial cases shown in §4.3.2, where the deterministic assignment is more profound at either the right or left tail of the CATE distribution. We expect ATE recovery to be more challenging in these cases for all methods. Our goal is to establish a proof-of-concept that our proposed algorithm performs significantly better than the existing benchmarks in recovering ATE.

To perform our analysis, we need to focus on an outcome and generate the adversarial missingness pattern. Since we have the ground truth ATEs, we can then examine how well each method recovers ATEs under adversarial missingness. We use *Reward* as our outcome of interest for each question as it is the ultimate outcome in our study. However, we present our results for the other two outcomes in Appendix D: *Easy* (whether the user chooses the easy version) and *Correct* (whether the user answers the question correctly).

To induce the missingness patterns, we use the  $\lambda$ -adversariality notion in Definition 5. We obtain the ground truth CATE matrix by estimating CATE on *Reward* for all units for each question in our experiment (please see Appendix D.1). We can do that since we have full randomization of the intervention (low vs. high price of the easy version of the question). We then obtain  $\lambda$ -adversariality scores using  $\lambda = 0.25$  and generate both right- and left-tail missingness patterns as shown in Equations (29) and (30). Once we have the corresponding propensity matrix for each case, we generate the feasibility matrix. Our methods cannot use the data from the observations that are not feasible. Although there are many versions of conventional methods and our proposed algorithm, we focus on a single version for each here and present the rest in Appendix D.2. For each missingness matrix, we apply the following methods:

- *Double ML*: For each question, we drop the observations that are not feasible according to the missingness pattern and estimate the ATE using Double ML. To estimate the nuisance functions, we use all the pre-treatment variables as covariates and Random Forest with two-fold cross-validation as learners. Please see the details in Appendix D.2. Overall, this gives us 20 estimated ATEs for questions in our experiment that serve as our benchmark against which we compare the performance of our proposed algorithm.
- *Proposed Algorithm*: For each question, we drop the observations that are not feasible and estimate CATE using a U-learner with XGBoost as the learner. We then form the incomplete CATE matrix with entries only for feasible elements. We use softImpute as our matrix comple-

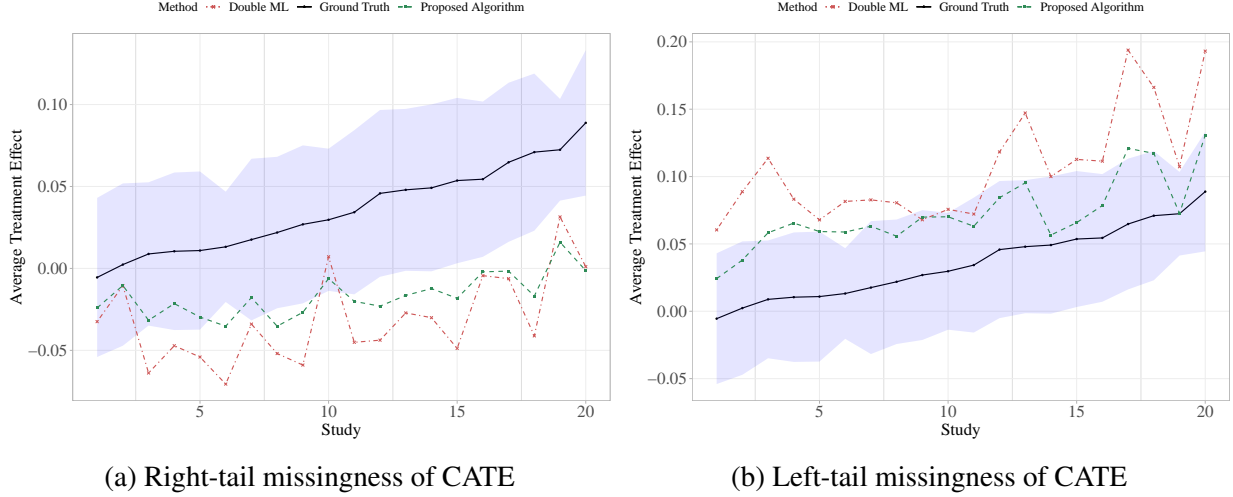


Figure 7: Model comparison with CATE-dependent missingness using data from a real experiment.

tion algorithm and tune its regularization parameter using validation. Please see the details in Appendix D.2. We complete the matrix as in Algorithm 1 and recover the ATEs.

We evaluate the performance of these two algorithms in estimating ATE across questions using the ground truth, as shown in Figure 7. Since we estimate the ground truth from data, we present the 95% confidence intervals around the ATE estimates. Figure 7a demonstrates the case with right-tail missingness. As shown in this figure, our proposed method performs better than Double ML in recovering ATE. Using Root Mean Squared Error (RMSE) between the estimates by each model and the ground truth ATEs, we find that our proposed algorithm has a 22% lower RMSE than Double ML. The performance of our algorithm is even better with left-tail missingness of CATE values, as shown in Figure 7b. Specifically for the example presented in this figure, our algorithm reduces the RMSE of Double ML by 47%. The worse performance in the case of right-tail missingness is likely due to the fact that in a small data setting like ours, missingness at the right tail of the CATE distribution substantially reduces the signal in the data used by the CATE estimator. This results in poor CATE estimation, which affects the overall performance of our approach.

We now repeat this practice for other levels of adversariality by changing  $\lambda$ . We consider five values for  $\lambda$  for either right or left tail missingness: 0, 0.25, 0.50, 0.75, and 1. The case of  $\lambda = 0$  is the same as a random missingness case, as it only uses random noise to generate the missingness pattern. For each case, we repeat the practice 20 times and present the average RMSE in Figure 8, with the ribbon showing the average plus and minus the standard deviation. As expected, higher levels of adversariality increase the error for both methods. However, our proposed algorithm consistently outperforms the existing benchmarks at all levels of adversariality. In Appendix D.3, we show the

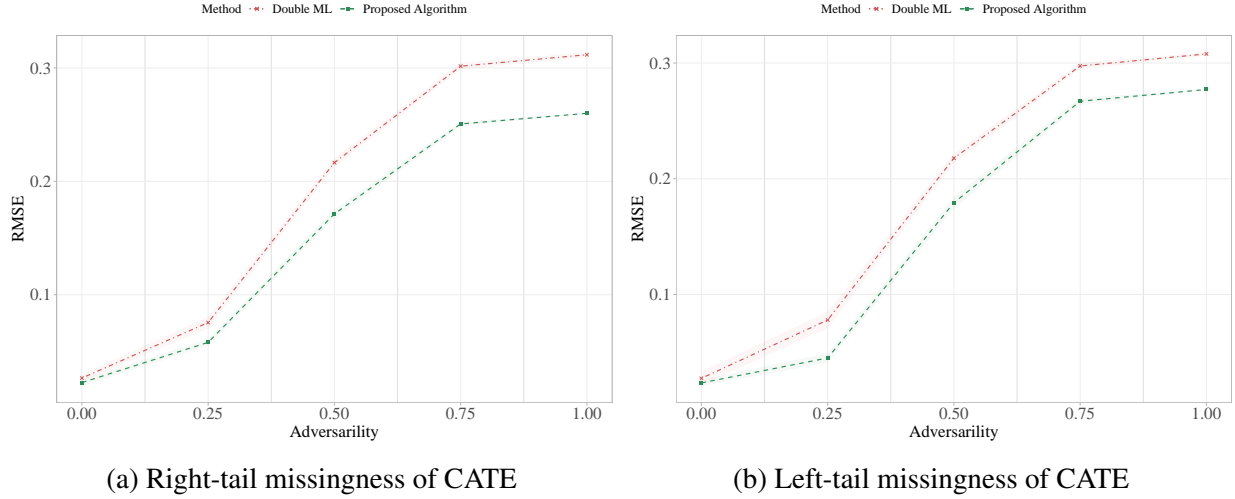


Figure 8: RMSE of different methods with CATE-dependent missingness at different levels of adversarity using data from a real experiment.

full results for the other two outcomes, a wider set of benchmarks, other versions of our proposed algorithm, and other sets of covariates to learn heterogeneity in treatment effects. All the results show substantial improvement from using our proposed algorithm over the existing benchmarks.

In summary, our results provide a proof-of-concept for our proposed algorithm and its use in cases where assignment to treatment can be partially deterministic. Given that CATE estimators generally require a large  $N$  to create a local variation at every part of data and matrix completion methods require a large  $J$  to fully exploit the low-rank structure, the superior performance of the proposed algorithm compared to the benchmarks in a small-data setting is particularly promising. It is worth noting that both a large number of users ( $N$ ) and studies ( $J$ ) are conditions that are commonly satisfied in digital platforms. Thus, we expect larger gains from our proposed algorithm in the context of digital platforms.

## 6 Conclusion

Digital platforms use algorithmic decision-making to deliver interventions to their users at a very large scale. An important goal for both practitioners and academic researchers is to identify the causal effect of such interventions. The gold standard answer to this question is to run randomized experiments. However, these experiments are often too costly, thereby giving rise to observational methods that use platforms' existing data without incurring experimentation costs. We examine this problem using the well-established potential outcomes framework (Holland, 1986). Observational studies generally require an important assumption called strong ignorability of the treatment assignment, which comprises two parts: unconfoundedness of the treatment assignment and overlap.

While much of the prior applied and methodological literature focused on the former, the latter received considerably less attention. We show that in digital platforms, this is, in fact, the overlap assumption that is not satisfied because the output of algorithmic recommendations is often deterministic. We theoretically show that the lack of overlap can be detrimental to the validity of an observational study. We quantify the bias term and argue that we expect the bias caused by the lack of overlap to be large in most digital platforms. Lastly, we formulate the identification problem caused by the lack of overlap as a missing data problem and propose a matrix completion solution that is often considered for such challenges. We show that if the platform has data on many treatments for the same units of population and the space of treatment effects is low-rank, we can recover the true average treatment effect.

There are several contributions that our paper makes to the literature. First, we present a comprehensive study of overlap violation in observational studies. We show how the lack of overlap can bias the estimates of average treatment effects from observational studies that ignore this assumption. Second, our paper provides important insights to practitioners. We show that the data from digital platforms that use algorithms to make decisions suffer from an often ignored part of the ignorability assumption: the overlap assumption. We show that this problem is generally prevalent in digital platforms. Finally, we provide a solution to this problem that can correct the bias caused by the lack of overlap if the platform has access to the data for numerous interventions and the underlying space of treatment effects is low-ranked. Our proposed algorithm can be used by digital platforms to utilize their existing observational data and by researchers who access a platform's data that suffer from the issue of deterministic assignment.

## References

- A. Agarwal, M. Dahleh, D. Shah, and D. Shen. Causal matrix completion. *arXiv preprint arXiv:2109.15154*, 2021.
- E. Ascarza. Retention futility: Targeting high-risk customers might be ineffective. *Journal of Marketing Research*, 55(1):80–98, 2018.
- S. Athey and S. Wager. Estimating treatment effects with causal forests: An application. *arXiv preprint arXiv:1902.07409*, 2019.
- S. Athey, G. W. Imbens, and S. Wager. Approximate residual balancing: debiased inference of average treatment effects in high dimensions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 80(4):597–623, 2018.
- S. Athey, J. Tibshirani, S. Wager, et al. Generalized random forests. *The Annals of Statistics*, 47(2):1148–1178, 2019.
- S. Athey, M. Bayati, N. Doudchenko, G. Imbens, and K. Khosravi. Matrix completion methods for causal panel data models. *Journal of the American Statistical Association*, 116(536):1716–1730, 2021.
- A. Belloni, V. Chernozhukov, and C. Hansen. Inference on treatment effects after selection among high-dimensional controls. *The Review of Economic Studies*, 81(2):608–650, 2014.
- J. Bennett, S. Lanning, et al. The netflix prize. In *Proceedings of KDD cup and workshop*, volume 2007, page 35. New York, 2007.
- E. J. Candès and B. Recht. Exact matrix completion via convex optimization. *Foundations of Computational mathematics*, 9(6):717–772, 2009.
- E. J. Candès and T. Tao. The power of convex relaxation: Near-optimal matrix completion. *IEEE Transactions on Information Theory*, 56(5):2053–2080, 2010.
- Y. Chen, J. Fan, C. Ma, and Y. Yan. Inference and uncertainty quantification for noisy matrix completion. *Proceedings of the National Academy of Sciences*, 116(46):22931–22937, 2019.
- V. Chernozhukov, D. Chetverikov, M. Demirer, E. Duflo, C. Hansen, W. Newey, and J. Robins. Double/debiased machine learning for treatment and structural parameters, 2018a.
- V. Chernozhukov, M. Demirer, E. Duflo, and I. Fernandez-Val. Generic machine learning inference on heterogeneous treatment effects in randomized experiments, with an application to immunization in india. Technical report, National Bureau of Economic Research, 2018b.
- R. K. Crump, V. J. Hotz, G. W. Imbens, and O. A. Mitnik. Dealing with limited overlap in estimation of average treatment effects. *Biometrika*, 96(1):187–199, 2009.
- A. D’Amour, P. Ding, A. Feller, L. Lei, and J. Sekhon. Overlap in observational studies with high-dimensional covariates. *Journal of Econometrics*, 221(2):644–654, 2021.

- A. Goli, A. Lambrecht, and H. Yoganarasimhan. A bias correction approach for interference in ranking experiments. *Available at SSRN 4021266*, 2022a.
- A. Goli, D. G. Reiley, and H. Zhang. Personalized versioning: Product strategies constructed from experiments on pandora. Working Paper, 2022b.
- B. R. Gordon, R. Moakler, and F. Zettelmeyer. Close enough? a large-scale exploration of non-experimental approaches to advertising measurement. *arXiv preprint arXiv:2201.07055*, 2022.
- D. Gross. Recovering low-rank matrices from few coefficients in any basis. *IEEE Transactions on Information Theory*, 57(3):1548–1566, 2011.
- P. W. Holland. Statistics and causal inference. *Journal of the American statistical Association*, 81(396):945–960, 1986.
- D. G. Horvitz and D. J. Thompson. A generalization of sampling without replacement from a finite universe. *Journal of the American statistical Association*, 47(260):663–685, 1952.
- G. W. Imbens and D. B. Rubin. *Causal inference in statistics, social, and biomedical sciences*. Cambridge University Press, 2015.
- A. Jesson, S. Mindermann, U. Shalit, and Y. Gal. Identifying causal-effect inference failure with uncertainty-aware models. *Advances in Neural Information Processing Systems*, 33:11637–11649, 2020.
- N. Kallus, X. Mao, and M. Udell. Causal inference with noisy and missing covariates via matrix factorization. *Advances in neural information processing systems*, 31, 2018.
- W. Ma and G. H. Chen. Missing not at random in matrix completion: The effectiveness of estimating missingness probabilities under a low nuclear norm assumption. *Advances in neural information processing systems*, 32, 2019.
- X. Ma and J. Wang. Robust inference using inverse probability weighting. *Journal of the American Statistical Association*, 115(532):1851–1860, 2020.
- R. Mazumder, T. Hastie, and R. Tibshirani. Spectral regularization algorithms for learning large incomplete matrices. *The Journal of Machine Learning Research*, 11:2287–2322, 2010.
- S. Negahban and M. J. Wainwright. Estimation of (near) low-rank matrices with noise and high-dimensional scaling. *The Annals of Statistics*, 39(2):1069–1097, 2011.
- R. C. Nethery, F. Mealli, and F. Dominici. Estimating population average causal effects in the presence of non-overlap: The effect of natural gas compressor station exposure on cancer mortality. *The annals of applied statistics*, 13(2):1242, 2019.
- J. Neyman. Sur les applications de la théorie des probabilités aux expériences agricoles: Essai des principes. *Roczniki Nauk Rolniczych*, 10:1–51, 1923.
- X. Nie and S. Wager. Quasi-oracle estimation of heterogeneous treatment effects. *Biometrika*, 108



- (2):299–319, 2021.
- O. Rafieian. Optimizing user engagement through adaptive ad sequencing. Technical report, Working paper, 2022.
- O. Rafieian and H. Yoganasimhan. Targeting and privacy in mobile advertising. *Marketing Science*, 2021.
- O. Rafieian, A. Kapoor, and A. Sharma. Multi-objective personalization of the length and skippability of video advertisements. *Available at SSRN 4394969*, 2023.
- B. Recht. A simpler approach to matrix completion. *Journal of Machine Learning Research*, 12(12), 2011.
- P. M. Robinson. Root-n-consistent semiparametric regression. *Econometrica: Journal of the Econometric Society*, pages 931–954, 1988.
- P. R. Rosenbaum and D. B. Rubin. The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1):41–55, 1983.
- U. Shalit, F. D. Johansson, and D. Sontag. Estimating individual treatment effect: generalization bounds and algorithms. In *International Conference on Machine Learning*, pages 3076–3085. PMLR, 2017.
- A. Shi, D. Zhang, T. Chan, H. Hu, and B. Zhao. Using algorithmic scores to measure the impacts of targeting promotional messages. *Available at SSRN*, 2022.
- D. Simester, A. Timoshenko, and S. I. Zoumpoulis. Efficiently evaluating targeting policies: Improving on champion vs. challenger experiments. *Management Science*, 66(8):3412–3424, 2020a.
- D. Simester, A. Timoshenko, and S. I. Zoumpoulis. Targeting prospective customers: Robustness of machine-learning methods to typical data challenges. *Management Science*, 66(6):2495–2522, 2020b.
- M. Udell and A. Townsend. Why are big data matrices approximately low rank? *SIAM Journal on Mathematics of Data Science*, 1(1):144–160, 2019.
- S. Wager and S. Athey. Estimation and inference of heterogeneous treatment effects using random forests. *Journal of the American Statistical Association*, 0(0):1–15, 2018. doi: 10.1080/01621459.2017.1319839.
- C. Waisman, H. S. Nair, C. Carrion, and N. Xu. Online causal inference for advertising in real-time bidding auctions. *arXiv preprint arXiv:1908.08600*, 2019.
- H. Yoganasimhan, E. Barzegary, and A. Pani. Design and evaluation of optimal free trials. *Management Science*, 2022.

# Appendices

## A Proofs

### A.1 Proof of Proposition 1

*Proof.* Let  $\mathcal{I}_r$  denote the set of observations that have probabilistic assignment. We denote the total number of these observations by  $N_r$ . From Chernozhukov et al. (2018a), we know that:

$$\operatorname{argmin}_{\tau} \frac{1}{N_r} \sum_{i \in \mathcal{I}_r} (Y_i - m(X_i) - (W_i - \pi(X_i)) \tau)^2 \xrightarrow{p} \tau_r. \quad (\text{A.31})$$

We now want to show that the RHS of Equation (A.31) is the same as what any methods optimizing Equation (7) would estimate. We can write:

$$\begin{aligned} \hat{\tau} &= \operatorname{argmin}_{\tau} \frac{1}{N} \sum_{i=1}^N (Y_i - m(X_i) - (W_i - \pi(X_i)) \tau)^2 \\ &= \operatorname{argmin}_{\tau} \frac{1}{N} \sum_{i \in \mathcal{I}_r} (Y_i - m(X_i) - (W_i - \pi(X_i)) \tau)^2 \\ &\quad + \sum_{i \notin \mathcal{I}_r} (Y_i - m(X_i) - (W_i - \pi(X_i)) \tau)^2 \\ &= \operatorname{argmin}_{\tau} \frac{1}{N} \sum_{i \in \mathcal{I}_r} (Y_i - m(X_i) - (W_i - \pi(X_i)) \tau)^2 + \sum_{i \notin \mathcal{I}_r} (Y_i - m(X_i))^2 \\ &= \operatorname{argmin}_{\tau} \frac{1}{N} \sum_{i \in \mathcal{I}_r} (Y_i - m(X_i) - (W_i - \pi(X_i)) \tau)^2 \\ &= \operatorname{argmin}_{\tau} \frac{1}{N_r} \sum_{i \in \mathcal{I}_r} (Y_i - m(X_i) - (W_i - \pi(X_i)) \tau)^2, \end{aligned} \quad (\text{A.32})$$

where the second line is a simple decomposition based on the observations with probabilistic and deterministic assignment, the fourth line is because  $W_i - \pi(X_i) = 0$  for observations with deterministic assignment, the fifth line drops the term  $\sum_{i \notin \mathcal{I}_r} (Y_i - m(X_i))^2$  because it is invariant of  $\tau$ , and the sixth line changes  $1/N$  to  $1/N_r$  because it is invariant of  $\tau$ . Now if we combine the result of Equation (A.32) with that of Equation (A.31), the proof is complete.  $\square$

### A.2 Proof of Proposition 2

*Proof.* The proof is straightforward and directly follows from the fact that we can only use non-deterministic propensity scores. As a result, we only focus on the observations in the probabilistic region. Therefore, the proof directly follows Horvitz and Thompson (1952).  $\square$

### A.3 Proof of Proposition 3

*Proof.* For the proof, we only show the first one, since the second one follows the same logic. We start by proving the following lemma:

**Lemma 2.** *We have  $\mathbb{E}[\mathbb{1}(\pi(X_i) = 1)\tau(X_i)] = P(\pi(X_i) = 1)\mathbb{E}[\tau(X_i) \mid \pi(X_i) = 1]$ .*

For brevity in our proof, we first define  $Q_i = \mathbb{1}(\pi(X_i) = 1)$ . We can now write:

$$\begin{aligned}
\mathbb{E}[\mathbb{1}(\pi(X_i) = 1)\tau(X_i)] &= \mathbb{E}[Q_i\tau(X_i)] \\
&= \mathbb{E}[\mathbb{E}[Q_i\tau(X_i) \mid Q_i]] \\
&= \mathbb{E}[Q_i\mathbb{E}[\tau(X_i) \mid Q_i]] \\
&= P(Q_i = 1)(1)\mathbb{E}[\tau(X_i) \mid Q_i = 1] + P(Q_i = 0)(0)\mathbb{E}[\tau(X_i) \mid Q_i = 0] \\
&= P(Q_i = 1)\mathbb{E}[\tau(X_i) \mid Q_i = 1] \\
&= P(\pi(X_i) = 1)\mathbb{E}[\tau(X_i) \mid \pi(X_i) = 1]
\end{aligned} \tag{A.33}$$

Now, we use this lemma to prove that if  $\tau(X_i)$  and belonging to the deterministic assignment region (i.e.,  $\mathbb{1}(\pi(X_i) = 1)$ ) are positively correlated, then we have  $\tau_1 \geq \tau^*$ . We can write:

$$\begin{aligned}
\tau_1 &= \mathbb{E}[\tau(X_i) \mid \pi(X_i) = 1] \\
&= \frac{P(\pi(X_i) = 1) \mathbb{E}[\tau(X_i) \mid \pi(X_i) = 1]}{P(\pi(X_i) = 1)} \\
&= \frac{\mathbb{E}[\mathbb{1}(\pi(X_i) = 1)\tau(X_i)]}{\mathbb{E}[\mathbb{1}(\pi(X_i) = 1)]} \\
&\geq \frac{\mathbb{E}[\mathbb{1}(\pi(X_i) = 1)]\mathbb{E}[\tau(X_i)]}{\mathbb{E}[\mathbb{1}(\pi(X_i) = 1)]} \\
&= \mathbb{E}[\tau(X_i)] \\
&= \tau^*,
\end{aligned} \tag{A.34}$$

where the fourth line comes from the fact that the two variables are positively correlated.  $\square$

## B Additional Simulation Experiments for the Proposed Algorithm

### B.1 Random Missingness Pattern

In this section, we consider the missing-completely-at-random setting but focus on a richer set of probabilities. We consider four different values for  $p$ : 0.2, 0.4, 0.6, and 0.8. We apply our Double ML and our proposed algorithm to the data to estimate the Average Treatment Effect for each study  $j$ .

Figure 4 shows four figures, each corresponding to a certain missingness level  $p$ . The x-axis presents studies as sorted by their true Average Treatment Effect (ATE). As shown in these figures,

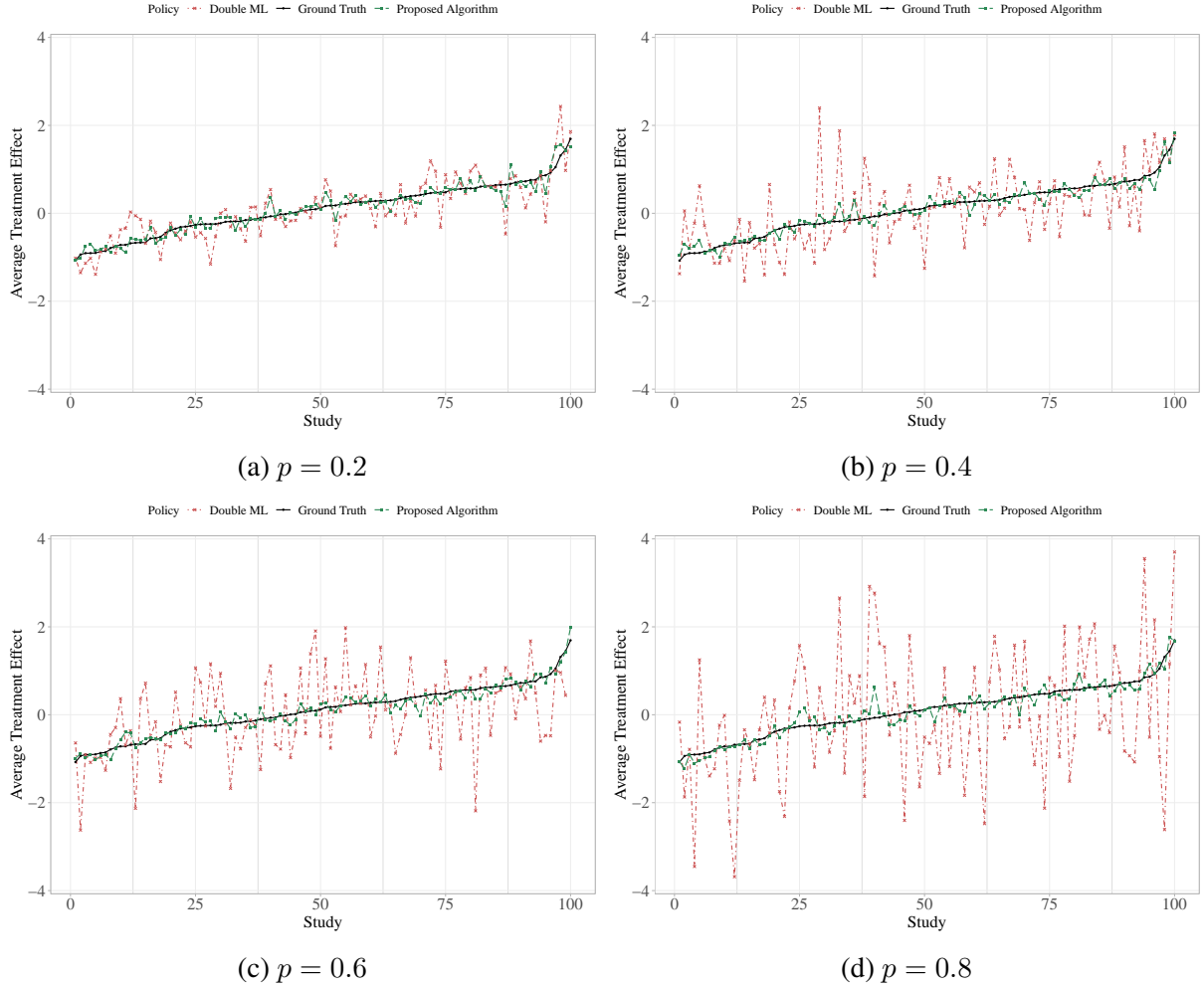


Figure A.1: The performance of our proposed algorithm and conventional observational method when missingness is at random with a uniform probability. Each figure represents a level of missingness.

conventional approaches like DML can all recover the true ATE, as expected. However, our proposed algorithm is more accurate even in these cases due to the fact that it uses data from other studies, especially in cases where a large portion of the elements of the CATE matrix is missing ( $p = 0.8$ ).

## B.2 CATE-Dependent Missingness

In §4.3.2, we present the results only for one case of  $\lambda$ -adversariality where  $\lambda = 0.25$ . We now extend that analysis to consider different levels of adversarial missingness at both the right and left tails of the CATE distribution. For simulations, we follow the procedures provided in §4.3.2 and use  $\lambda \in \{0.25, 0.50, 0.75, 1.00\}$  for both cases of right-tail and left-tail missingness of CATE. Figure A.2 shows the performance of our algorithm and Double ML when missingness happens at the right

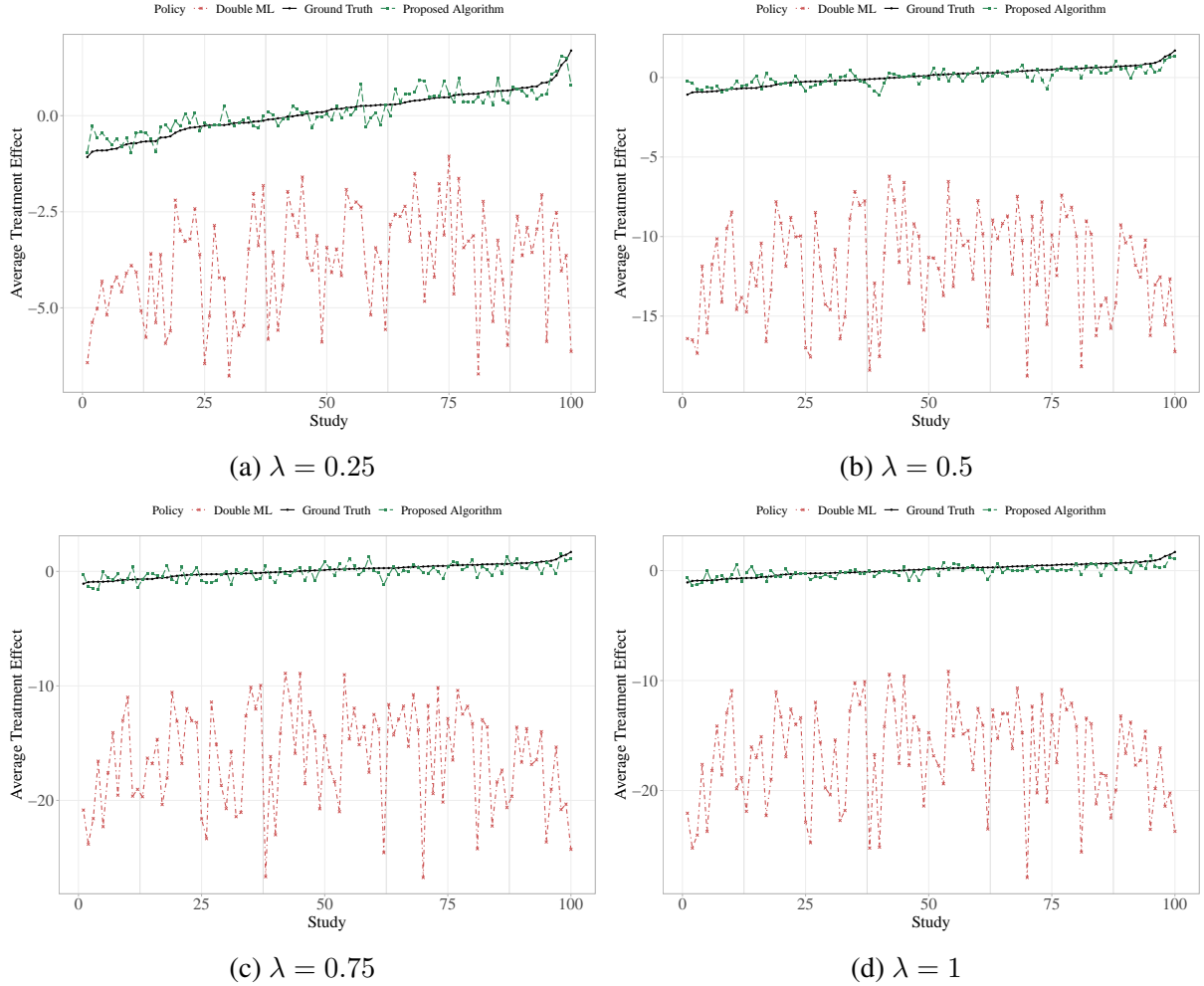


Figure A.2: The performance of our proposed algorithm and conventional observational method when we have a right-tail missingness of CATE at different levels of adversariality.

tail of the CATE distribution. As clearly shown in all four instances, conventional observational methods such as Double ML exhibit very large biases, whereas our method is able to accurately recover the true ATEs under extreme adversarial cases.

We then focus on the missingness cases for the left tail of the CATE distribution and perform the same simulation practice. We present the result of this practice in Figure A.3. The same pattern emerges: although conventional methods are largely biased in recovering ATEs, our proposed method demonstrates robustness to extreme adversarial cases. Even when the adversariality is set at  $\lambda = 1$ , our proposed method performs substantially better than the benchmark.

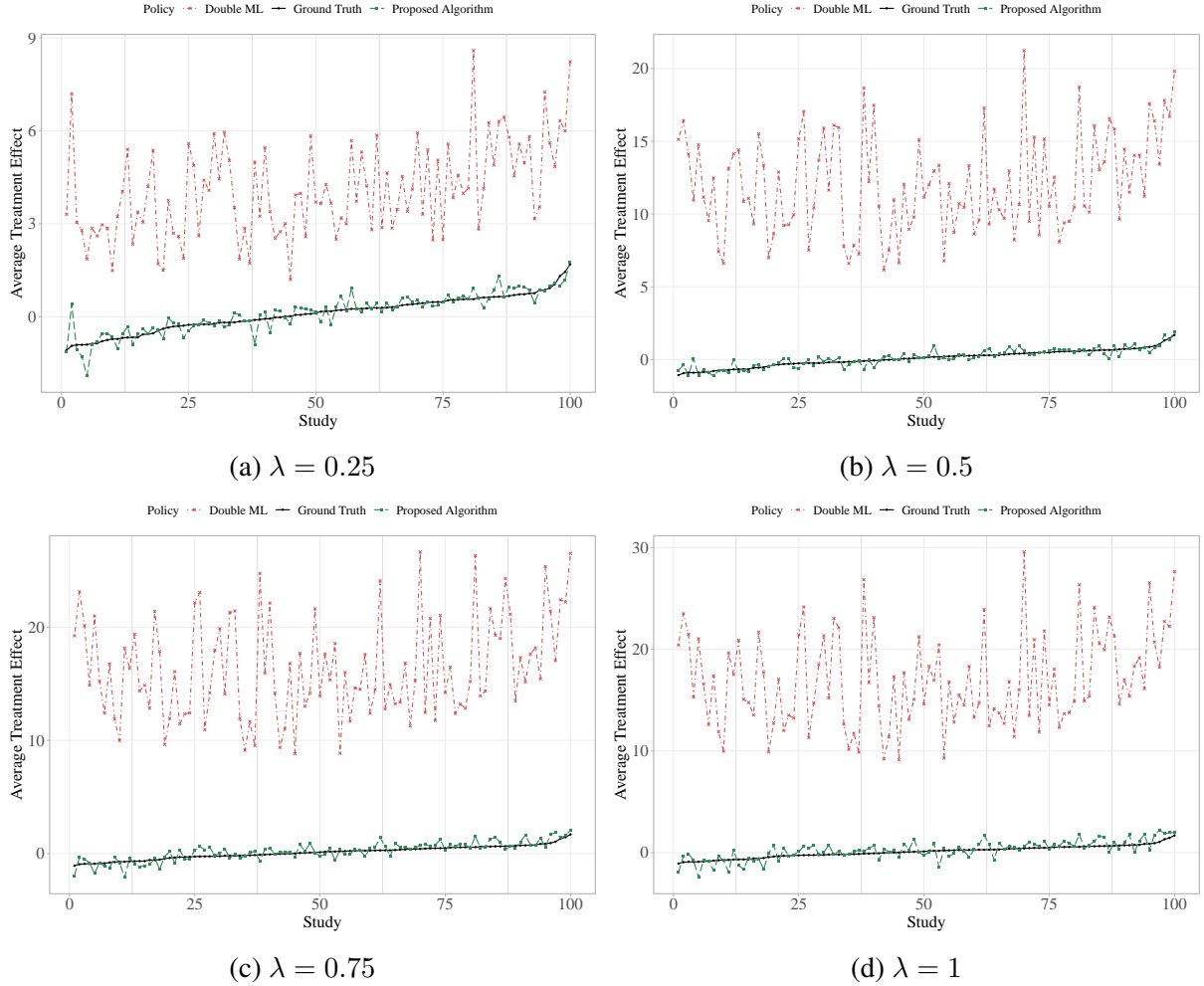


Figure A.3: The performance of our proposed algorithm and conventional observational method when we have a left-tail missingness of CATE at different levels of adversariality.

## C Experiment

### C.1 Experiment Design

In this section, we present the details of our experiment design. Our experiment timeline has four parts. The first part contains pre-treatment data collection about users' risk-taking, price sensitivity, and expertise (see §C.1.1). In the second part, we present the instructions of our experiment with a demo question that does not affect participants' final reward but informs them about the game (see §C.1.2). The third part of our experiment presents the main component of our experiment and contains all 20 approximation questions that users can answer and receive rewards (see §C.1.3). In the last part, we collect some of the users' demographic information (see §C.1.4). The following

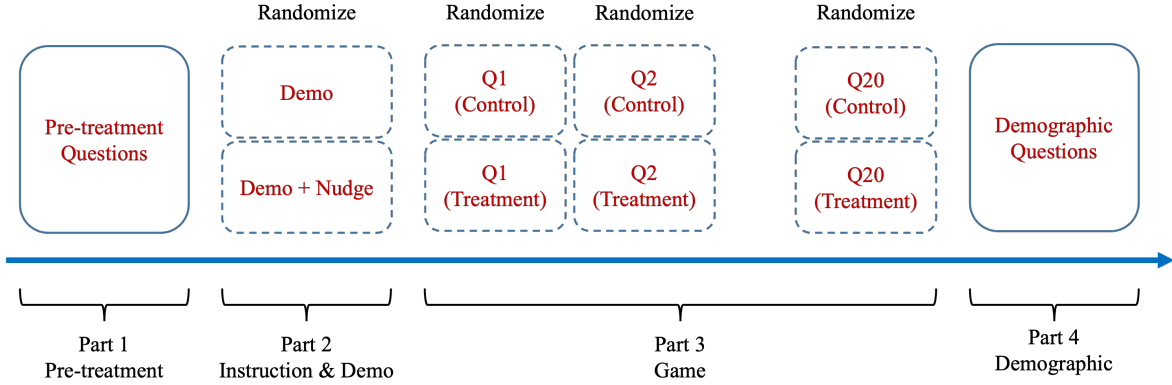


Figure A.4: Timeline of our experiment

sections present each of these parts.

Figure A.4 shows an overview of the timeline of our experiment and our randomization design. As shown in this figure, we implement independent randomization in Part 2 (instruction) and Part 3 (main game) of the experiment, which gives us a  $2^{21}$  factorial design. The randomization for each question in Part 3 allows us to have 20 separate studies with different ground truth average treatment effects, which allows us to perform our proposed algorithm and evaluate its performance relative to the benchmarks.

### C.1.1 Pre-treatment Questions

In this section, we present the first set of questions that participants in our experiment answered. As presented in Table A1, we ask two sets of questions before the start of the game. The first set contains five questions that ask participants to share how much they agree with different statements on a Likert scale from 1 to 5. The second set has two questions that measure participants' attitudes toward risk. We expect these variables to explain some of the heterogeneity in treatment effects, as they measure price sensitivity, risk preferences, and expertise of users.

Table A1: Pre-treatment questions of the experiment

Question	Choices
Please indicate how much you agree with the following statements:	
<i>PTQ1</i> : I am willing to spend more time in grocery shopping to make sure I am purchasing products with the best price deals.	Likert scale from 1 (completely disagree) to 5 (completely agree)
<i>PTQ2</i> : I am willing to take risks in general.	Likert scale from 1 (completely disagree) to 5 (completely agree)
<i>PTQ3</i> : Streaming platforms offer more value than their subscription cost.	Likert scale from 1 (completely disagree) to 5 (completely agree)

*PTQ4*: I am confident in my mathematical skills.

Likert scale from 1 (completely disagree) to 5 (completely agree)

*PTQ5*: I am confident in my geometry skills.

Likert scale from 1 (completely disagree) to 5 (completely agree)

---

Please answer the following investment questions:

*PTQ 6*: Imagine that someone gives you \$20. You can invest any portion of this amount in a risky lottery with the following two outcomes:

- With a 50% probability, your investment triples.
- With a 50% probability, you lose your investment.

Which one of the following options do you pick?

- Keep \$20.
- Keep \$15 and invest \$5 in lottery.
- Keep \$10 and invest \$10 in lottery.
- Keep \$5 and invest \$15 in lottery.
- Invest all \$20 in lottery.

*PTQ 7*: Imagine that someone gives you \$100. You can invest any portion of this amount in a risky lottery with the following two outcomes:

- With a 50% probability, your investment triples.
- With a 50% probability, you lose your investment.

Which one of the following options do you pick?

- Keep \$100.
  - Keep \$75 and invest \$25 in lottery.
  - Keep \$50 and invest \$50 in lottery.
  - Keep \$25 and invest \$75 in lottery.
  - Invest all \$100 in lottery.
- 

### C.1.2 Demo Question

Before participants start playing the game, we place a demo question to inform them about the game instructions. We first randomly assign participants to two versions of instructions. In one version, we include a nudge that encourages participants to carefully consider the deal offered in the easy version. We include this component in our design to create more observed heterogeneity in treatment effects, as users who receive a nudge (pro tip) are more likely to calculate the risk and benefits involved. Later in our analysis, we use the dummy variable for whether the participant has received the nudge as a covariate around which we want to estimate the treatment effect heterogeneity. The instructions and the demo question are presented in Table A2. For the demo question, we do not randomize the rewards for users, as this question was just presented for instructional purposes. The correct answer to the Demo Question is 59 seconds.

Table A2: Instructions and the demo question

---

Question	Control Condition	Treatment Condition
----------	-------------------	---------------------

---



### Instructions

In this survey, you participate in a game of approximating the size of objects. Below are the instructions and rules of the game:

- There are 20 questions in total.
- In each question, we provide some information about the size of some objects in a picture and ask you to guess the size of other objects in the same picture. You will see a demo question in the next page.
- You first see the difficult version of the question with 4 choices. If you correctly answer this version of the question, you will earn 1 point reward. However, you have the opportunity to choose an easy version of the question with 2 choices at a lower reward.
- You first see the difficult version of the question with 4 choices. If you correctly answer this version of the question, you will earn 1 point reward. However, you have the opportunity to choose an easy version of the question with 2 choices at a lower reward.
  - **Pro Tip:** Not every easy version is worth it. Carefully evaluate the new reward for the easy version of the question. For example, if you have a 25% chance of answering the difficult version and 50% chance of answering the easy version with 0.4 reward, sticking with the difficult version is the right strategy. However, if your chance of answering the easy question increases more substantially (e.g., to 75%), it is worth switching to the easy version.
- Your points accumulate as you play. At the end of the game, you will receive 10 cents for each point received. Thus, you can earn up to \$2 depending on how many points you earn in this study.

Let's try a Demo Question to see how the game is like.

#### Demo Question

If the amount watched is 40 seconds (red area), what is the length of the total video?



Pro Tip Excluded

Pro Tip Included




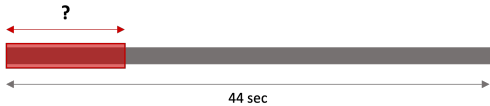
- |   |   |
|---|---|
| • 70 seconds  | • 70 seconds  |
| • 59 seconds  | • 59 seconds  |
| • 48 seconds  | • 48 seconds  |
| • 65 seconds  | • 65 seconds  |
| • Make it easier by removing 2 choices;<br>New Reward = 0.6 | • Make it easier by removing 2 choices;<br>New Reward = 0.6 |

### C.1.3 Main Questions

We now present the questions in our game with the choices provided in the control and treatment conditions in Table A3. The first column of Table A3 represents the questions, including both the text and figure for each. The second and third columns show the choices provided in the control and treatment conditions for each question, respectively. The only difference between control and treatment conditions for each question is in the offer provided: the treatment condition always offers a better deal for the easy version of the question. The first choice shown in columns 2 and 3 is the

correct answer to each question, and the third and fourth choices are those that would be removed if the user chooses the easy version of the question. The order of choices is randomized when shown to users. The reward is calculated based on the question answered by the question (easy vs. difficult) and the correctness of the answer.

Table A3: Main questions of the experiment

Question	Control Condition	Treatment Condition
<p><i>Question 1</i></p> <p>If the amount watched is 10 seconds (red area), what is the length of the total video?</p> 	<ul style="list-style-type: none"> <li>• 53 seconds</li> <li>• 61 seconds</li> <li>• 57 seconds</li> <li>• 42 seconds</li> <li>• Make it easier by removing 2 choices;</li> </ul> <p>New Reward = 0.5</p>	<ul style="list-style-type: none"> <li>• 53 seconds</li> <li>• 61 seconds</li> <li>• 57 seconds</li> <li>• 42 seconds</li> <li>• Make it easier by removing 2 choices;</li> </ul> <p>New Reward = 0.8</p>
<p><i>Question 2</i></p> <p>If the amount watched is 2 minutes and 12 seconds (red area), what is the length of the total video?</p> 	<ul style="list-style-type: none"> <li>• 6 min and 29 sec</li> <li>• 6 min and 2 sec</li> <li>• 5 min and 42 sec</li> <li>• 7 min and 8 sec</li> <li>• Make it easier by removing 2 choices;</li> </ul> <p>New Reward = 0.3</p>	<ul style="list-style-type: none"> <li>• 6 min and 29 sec</li> <li>• 6 min and 2 sec</li> <li>• 5 min and 42 sec</li> <li>• 7 min and 8 sec</li> <li>• Make it easier by removing 2 choices;</li> </ul> <p>New Reward = 0.6</p>
<p><i>Question 3</i></p> <p>If the amount watched is 5 minutes (red area), what is the length of the total video?</p> 	<ul style="list-style-type: none"> <li>• 7 min and 22 sec</li> <li>• 8 min and 2 sec</li> <li>• 7 min and 49 sec</li> <li>• 7 min and 39 sec</li> <li>• Make it easier by removing 2 choices;</li> </ul> <p>New Reward = 0.4</p>	<ul style="list-style-type: none"> <li>• 7 min and 22 sec</li> <li>• 8 min and 2 sec</li> <li>• 7 min and 49 sec</li> <li>• 7 min and 39 sec</li> <li>• Make it easier by removing 2 choices;</li> </ul> <p>New Reward = 0.7</p>
<p><i>Question 4</i></p> <p>If the total video is 44 seconds, how much of the video has been watched (red area)?</p> 	<ul style="list-style-type: none"> <li>• 11 seconds</li> <li>• 13 seconds</li> <li>• 15 seconds</li> <li>• 9 seconds</li> <li>• Make it easier by removing 2 choices;</li> </ul> <p>New Reward = 0.5</p>	<ul style="list-style-type: none"> <li>• 11 seconds</li> <li>• 13 seconds</li> <li>• 15 seconds</li> <li>• 9 seconds</li> <li>• Make it easier by removing 2 choices;</li> </ul> <p>New Reward = 0.8</p>

*Question 5*

If the total video is 2 minutes and 39 seconds, how much of the video has been watched (red area)?

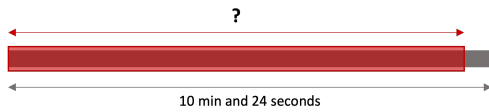


- 1 min and 39 sec
- 1 min and 50 sec
- 1 min and 56 sec
- 1 min and 46 sec
- Make it easier by removing 2 choices;  
New Reward = 0.3

- 1 min and 39 sec
- 1 min and 50 sec
- 1 min and 56 sec
- 1 min and 46 sec
- Make it easier by removing 2 choices;  
New Reward = 0.6

*Question 6*

If the total video is 10 minutes and 24 seconds, how much of the video has been watched (red area)?

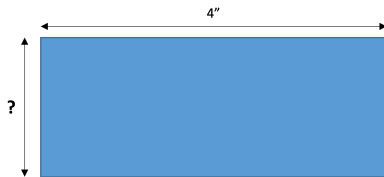


- 9 min and 49 sec
- 9 min and 30 sec
- 9 min and 4 sec
- 9 min and 15 sec
- Make it easier by removing 2 choices;  
New Reward = 0.4

- 9 min and 49 sec
- 9 min and 30 sec
- 9 min and 4 sec
- 9 min and 15 sec
- Make it easier by removing 2 choices;  
New Reward = 0.7

*Question 7*

Given that the width of the following rectangle is 4 inches, what is your estimate of its height?

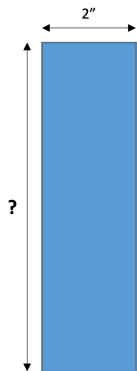


- 1.6 inches
- 1.2 inches
- 1 inch
- 1.1 inches
- Make it easier by removing 2 choices;  
New Reward = 0.5

- 1.6 inches
- 1.2 inches
- 1 inch
- 1.1 inches
- Make it easier by removing 2 choices;  
New Reward = 0.8

*Question 8*

Given that the width of the following rectangle is 2 inches, what is your estimate of its height?

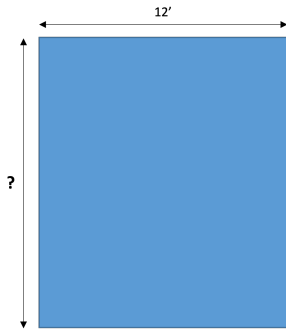


- 7 inches
- 8 inches
- 9 inches
- 6 inches
- Make it easier by removing 2 choices;  
New Reward = 0.3

- 7 inches
- 8 inches
- 9 inches
- 6 inches
- Make it easier by removing 2 choices;  
New Reward = 0.6

*Question 9*

Given that the width of the following rectangle is 12 feet, what is your estimate of its height?

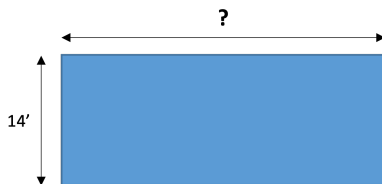


- |   |   |
|---|---|
| • 14 feet   | • 14 feet   |
| • 15 feet   | • 15 feet   |
| • 12 feet   | • 12 feet   |
| • 13 feet   | • 13 feet   |
| • Make it easier by removing 2 choices;<br>New Reward = 0.4 | • Make it easier by removing 2 choices;<br>New Reward = 0.7 |

---

*Question 10*

Given that the height of the following rectangle is 14 feet, what is your estimate of its width?

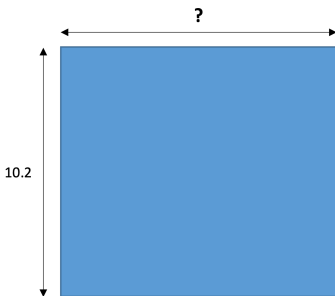


- |   |   |
|---|---|
| • 35 feet   | • 35 feet   |
| • 42 feet   | • 42 feet   |
| • 38 feet   | • 38 feet   |
| • 40.5 feet   | • 40.5 feet   |
| • Make it easier by removing 2 choices;<br>New Reward = 0.5 | • Make it easier by removing 2 choices;<br>New Reward = 0.8 |

---

*Question 11*

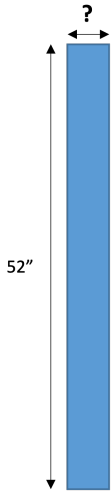
Given that the height of the following rectangle is 10.2 miles, what is your estimate of its width?



- |   |   |
|---|---|
| • 11.3 miles  | • 11.3 miles  |
| • 10.3 miles  | • 10.3 miles  |
| • 12 miles  | • 12 miles  |
| • 10.7 miles  | • 10.7 miles  |
| • Make it easier by removing 2 choices;<br>New Reward = 0.3 | • Make it easier by removing 2 choices;<br>New Reward = 0.6 |

*Question 12*

Given that the height of the following rectangle is 52 inches, what is your estimate of its width?



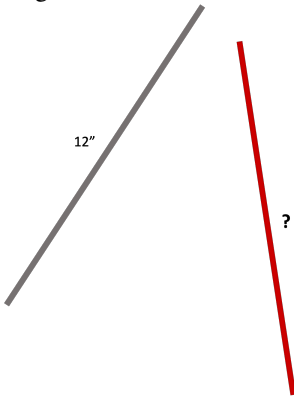
- 5 inches
- 3 inches
- 4 inches
- 2 inches
- Make it easier by removing 2 choices;  
New Reward = 0.4

- 5 inches
- 3 inches
- 4 inches
- 2 inches
- Make it easier by removing 2 choices;  
New Reward = 0.7

---

*Question 13*

Given that the grey line is 12 inches, what is the length of the red line?



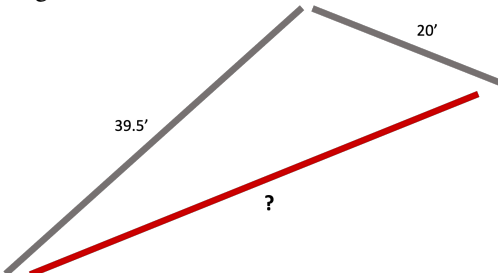
- 12 inches
- 13 inches
- 11.5 inches
- 12.5 inches
- Make it easier by removing 2 choices;  
New Reward = 0.5

- 12 inches
- 13 inches
- 11.5 inches
- 12.5 inches
- Make it easier by removing 2 choices;  
New Reward = 0.8

---

*Question 14*

Given the size of the grey lines, what is the length of the red line?

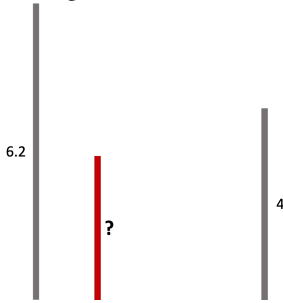


- 48 feet
- 45 feet
- 42 feet
- 39 feet
- Make it easier by removing 2 choices;  
New Reward = 0.3

- 48 feet
- 45 feet
- 42 feet
- 39 feet
- Make it easier by removing 2 choices;  
New Reward = 0.6

*Question 15*

Given the size of the grey lines in miles, what is the length of the red line?

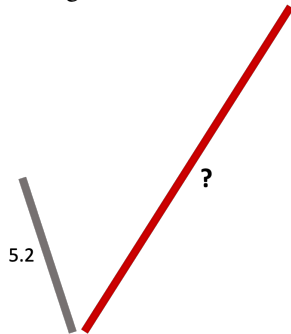


- 3 miles
- 2.6 miles
- 2.8 miles
- 3.2 miles
- Make it easier by removing 2 choices;  
New Reward = 0.4

- 3 miles
- 2.6 miles
- 2.8 miles
- 3.2 miles
- Make it easier by removing 2 choices;  
New Reward = 0.7

*Question 16*

Given that the grey line is 5.2 inches, what is the length of the red line?

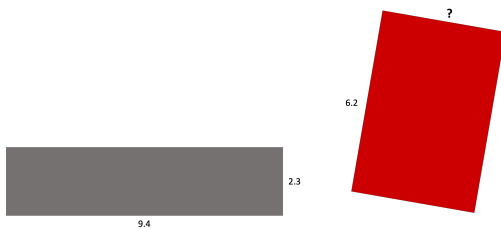


- 12.4 miles
- 13.5 miles
- 14.6 miles
- 15.7 miles
- Make it easier by removing 2 choices;  
New Reward = 0.5

- 12.4 miles
- 13.5 miles
- 14.6 miles
- 15.7 miles
- Make it easier by removing 2 choices;  
New Reward = 0.8

*Question 17*

Given the length of edges provided in inches, what is the width of the red rectangle?



- 4.2 miles
- 3.6 miles
- 4.8 miles
- 3 miles
- Make it easier by removing 2 choices;  
New Reward = 0.3

- 4.2 miles
- 3.6 miles
- 4.8 miles
- 3 miles
- Make it easier by removing 2 choices;  
New Reward = 0.6

*Question 18*

Given that the height of the grey triangle is 20 miles, what is the height of the red triangle?

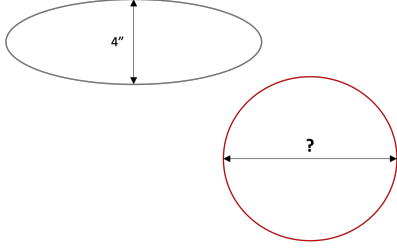


- 10.3 miles
- 9.3 miles
- 8.4 miles
- 7.8 miles
- Make it easier by removing 2 choices;  
New Reward = 0.4

- 10.3 miles
- 9.3 miles
- 8.4 miles
- 7.8 miles
- Make it easier by removing 2 choices;  
New Reward = 0.7

*Question 19*

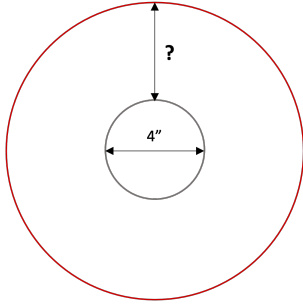
Given that the height of the grey oval is 4 inches, what is the width of the red oval?



- 8.2 inches
  - 7.6 inches
  - 7.1 inches
  - 6.8 inches
  - Make it easier by removing 2 choices;  
New Reward = 0.5
- 8.2 inches
  - 7.6 inches
  - 7.1 inches
  - 6.8 inches
  - Make it easier by removing 2 choices;  
New Reward = 0.8

*Question 20*

Given that the diameter of the small circle is 4 inches, what is distance shown in the figure below?



- 4 inches
  - 4.6 inches
  - 4.3 inches
  - 3.7 inches
  - Make it easier by removing 2 choices;  
New Reward = 0.3
- 4 inches
  - 4.6 inches
  - 4.3 inches
  - 3.7 inches
  - Make it easier by removing 2 choices;  
New Reward = 0.6

### C.1.4 Demographic Questions

At the end of our experiment, we collect some demographic information on participants. Table A4 presents these demographic questions. As shown in this table, we collect information about participants' age, education, gender, race, income, and political views.

Table A4: Demographic questions of the experiment

Question	Choices
D1: Please indicate your age.	Numeric
D2: What is your highest level of education?	<ul style="list-style-type: none"><li>• Some high-school</li><li>• High-school diploma</li><li>• Some college</li><li>• Bachelor's degree</li><li>• Master's degree or above</li></ul>
D3: What is your gender?	<ul style="list-style-type: none"><li>• Male</li><li>• Female</li><li>• Other (please specify)</li></ul>

<i>D4: What is your race/ethnicity?</i>	<ul style="list-style-type: none"> <li>• American Indian or Alaska Native</li> <li>• Asian</li> <li>• Black - non-Hispanic</li> <li>• Hispanic</li> <li>• White - non-Hispanic</li> </ul>
<i>D5: Which of these describes your personal income last year?</i>	<ul style="list-style-type: none"> <li>• \$0</li> <li>• \$1 to \$9 999</li> <li>• \$10 000 to \$24 999</li> <li>• \$25 000 to \$49 999</li> <li>• \$50 000 to \$74 999</li> <li>• \$75 000 to \$99 999</li> <li>• \$100 000 to \$149 999</li> <li>• \$150 000 and greater</li> <li>• Prefer Not to Answer</li> </ul>
<i>D6: Which of these describes your political views and ideology?</i>	<ul style="list-style-type: none"> <li>• Very Conservative</li> <li>• Moderately Conservative</li> <li>• Neutral</li> <li>• Moderately Liberal</li> <li>• Very Liberal</li> </ul>

## C.2 Theoretical Analysis of the Experiment

The main question is where the heterogeneity in treatment effects comes from. We present a simple theoretical analysis of the heterogeneity in treatment effects on the choice of the easy version. For a given question, let  $\gamma_{id}$  and  $\gamma_{ie}$  denote user  $i$ 's perceived probability of correctly answering the difficult (4-choice) and easy (2-choice) versions of the question, respectively. Further, let  $R_{id}$  and  $R_{ie}$  denote the reward associated with the difficult and easy versions of the question. We set  $R_{id} = 1$  in all instances, but  $R_{ie} = 1 - p$ , where  $p$  is essentially the price the user pays for the easy version and is randomly assigned to users from two options: high price  $p_H$  in the control condition and low price  $p_L$  in the treatment condition. For example, the high price in the control condition in Figure 6 is 0.5 (i.e., Reward = 0.5), and the low price in the treatment condition is 0.2 (i.e., Reward = 0.8).

In each condition, choosing the difficult version yields an expected return of  $\gamma_{id}R_{id} = \gamma_{id}$ , whereas choosing the easy version yields an expected return of  $\gamma_{ie}R_{ie} = \gamma_{ie}(1 - p)$ . Depending on user  $i$ 's valuation of the reward (denoted by  $\theta_i$ ) and risk aversion (denoted by  $\eta_i$ ), we can define the utility from choosing easy or hard version for each price offer  $p$ . Let  $g(p; \gamma_{ie}, \gamma_{id}, \theta_i, \eta_i)$  denote the probability that the user chooses the easy version in a question with price offer  $p$ . We can then define the CATE for that question on the outcome of choosing the easy version as follows:

$$CATE_i = g(p_L; \gamma_{ie}, \gamma_{id}, \theta_i, \eta_i) - g(p_H; \gamma_{ie}, \gamma_{id}, \theta_i, \eta_i), \quad (\text{A.35})$$



where  $p_L$  and  $p_H$  are prices offered in the treatment and control conditions. The characterization in Equation (A.35) shows that the heterogeneity in treatment effects comes from (1) perceived expertise of the users ( $\gamma_{ie}$  and  $\gamma_{id}$ ), (2) the value of reward earned ( $\theta_i$ ), and (3) risk aversion ( $\eta_i$ ). As a result, we need to collect information on these three sources as pre-treatment variables. We can further use the user's past behavior for each question to enrich the covariates used to identify heterogeneity in treatment effects. For example, for Question 10, we can use the choices and performance of the user up until that point as pre-treatment variables. This heterogeneity in treatment effects will appear in other outcomes of interest, such as whether the user answers the question correctly and the reward received for each question.

### C.3 Full Descriptive Analysis

Overall, we have the following sets of variables in our data:

- *Pre-treatment variables*: The first part of our experiment asks a set of seven pre-treatment questions, which gives us seven covariates. We name these variables  $PTQ1$  to  $PTQ7$ , as presented in Table A1. Since each variable in this part is measured using five choices, the range is from 1 to 5.
- *Instruction and demo variables*: We construct three covariates from this part of our experiment: (1) *Nudge*, which indicates whether the participant received the pro-tip as part of the instruction, (2) *DemoCorrect*, which indicates whether the participant answered the demo question correctly, and (3) *DemoNotAnswered*, which indicates whether the participant answered the demo question. If the participant does not answer the question, we do not know whether s/he has received the nudge. The variable *Nudge* can help explain a pattern of behavior where participants more carefully make choices, and *DemoCorrect* works as a proxy for the participants' expertise in this game.
- *Main game variables*: Most of our data and outcomes used come from this part of our experiment. For each question in the main game, we collect the following variables: (1) whether the participant was assigned to the treatment (the case with a lower price for the easy version) labeled as  $W1, W2, \dots, W20$  in our data, (2) whether the participant answered the easy version of the question labeled as  $E1, E2, \dots, E20$ , (3) whether the participant answered the question correctly labeled as  $C1, C2, \dots, C20$ , (4) the amount of reward received by the participant labeled as  $R1, R2, \dots, R20$ , (5) the amount of reward received by the participant through answering *difficult* versions of questions labeled as  $RD1, RD2, \dots, RD20$ , and (6) the amount of reward received by the participant through answering *easy* versions of questions labeled as  $RE1, RE2, \dots, RE20$ . It is easy to see for each question  $j$ , we have  $R_j = RD_j + RE_j$ . We have a total of  $20 \times 6 = 120$  main game variables to work with. The aggregate statistics on

these variables are presented in Table 2.

- *Demographic variables:* We use the six demographic questions to create five demographic variables: *Age*, *Education*, *Gender*, *Race*, *Income*, and *PoliticalViews*.

We now present some descriptive statistics on 16 variables that come from parts other than the game itself. Although the demographic variables are collected post-treatment, we still refer to those variables as pre-treatment because they measure some fixed characteristics of individuals. We present the summary statistics for these variables in Table A5. This is the analog of Table 2 that was defined on the game-related variables.

The first seven rows in Table A5 present the summary statistics for the pre-treatment questions. We find that all seven variables in this portion of our experiment show reasonable variation, with all the values of the range being fully covered.

When we focus on the instruction and demo questions, we find that over 32% of participants did not answer the demo question, which means that for these participants, we do not know whether they have seen the nudge. However, given their moving forward without answering the question, it is fair to assume that they did not pay much attention to the nudge (if any). Overall, 34% of users received the nudge, and around 30% of users answered the demo question correctly.

We then present summary statistics for our demographic variables. The only numerical variable in this set is the variable *Age*. The minimum value for *Age* is equal to 6, which is likely a typo by the participant when entering her age. The second smallest *Age* is equal to 20 years old, and the oldest is 79, which indicates considerable variation in that variable. For other demographic variables, we show the summary statistics for the dummy variables for each subcategory. For ordinal variables like *Education* and *Income*, we use each level as a subcategory.

#### **C.4 Randomization Checks**

We perform randomization checks to ensure that the treatments were properly randomized for each question. To do so, we run 20 regressions, where we regress the treatment status for a question on all the pre-treatment variables presented in Table A5. For each regression, we can use the F-statistics of the overall regression to see if the pre-treatment variables explain the variation in the treatment variable. If we can reject the null hypothesis for any of these regressions, that would be evidence suggesting that randomization has not been implemented properly.

We run all 20 regressions and present the F-statistics and p-values in Table A6. The first column refers to the model we run, which is an ordinary least squares (OLS) model where we regress the treatment status for a question on pre-treatment variables. The second and third columns present the F-statistic and p-value of models, respectively. As shown in this table, we fail to reject the null hypothesis for all models, confirming that randomization design in our experiment was implemented

<b>Variable</b>	<b>Mean</b>	<b>SD</b>	<b>Min</b>	<b>Median</b>	<b>Max</b>
<i>PTQ1</i>	4.0296	0.8823	1	4	5
<i>PTQ2</i>	2.9386	1.0879	1	3	5
<i>PTQ3</i>	3.2346	1.0222	1	3	5
<i>PTQ4</i>	3.4819	1.1513	1	4	5
<i>PTQ5</i>	2.9312	1.1765	1	3	5
<i>PTQ6</i>	2.7705	1.2803	1	3	5
<i>PTQ7</i>	2.4360	1.1002	1	2	5
<i>Nudge</i>	0.3442	0.4753	0	0	1
<i>DemoCorrect</i>	0.2976	0.4574	0	0	1
<i>DemoNotAnswered</i>	0.3279	0.4696	0	0	1
<i>Age</i>	41.9104	12.3242	6	39	79
<i>Education1</i>	0.0052	0.0718	0	0	1
<i>Education2</i>	0.1081	0.3106	0	0	1
<i>Education3</i>	0.2902	0.4540	0	0	1
<i>Education4</i>	0.4189	0.4936	0	0	1
<i>Education5</i>	0.1776	0.3824	0	0	1
<i>GenderMale</i>	0.5270	0.4995	0	1	1
<i>GenderFemale</i>	0.4685	0.4992	0	0	1
<i>GenderOther</i>	0.0044	0.0665	0	0	1
<i>RaceIndigenous</i>	0.0081	0.0899	0	0	1
<i>RaceAsian</i>	0.0799	0.2713	0	0	1
<i>RaceBlack</i>	0.1140	0.3179	0	0	1
<i>RaceHispanic</i>	0.0600	0.2375	0	0	1
<i>RaceWhite</i>	0.7380	0.4399	0	1	1
<i>Income1</i>	0.0081	0.0899	0	0	1
<i>Income2</i>	0.1177	0.3224	0	0	1
<i>Income3</i>	0.1429	0.3501	0	0	1
<i>Income4</i>	0.2539	0.4354	0	0	1
<i>Income5</i>	0.2073	0.4055	0	0	1
<i>Income6</i>	0.1362	0.3431	0	0	1
<i>Income7</i>	0.0674	0.2507	0	0	1
<i>Income8</i>	0.0407	0.1977	0	0	1
<i>IncomeNoAnswer</i>	0.0259	0.1589	0	0	1
<i>VeryConservative</i>	0.0844	0.2781	0	0	1
<i>Conservative</i>	0.2087	0.4066	0	0	1
<i>Neutral</i>	0.2258	0.4182	0	0	1
<i>Liberal</i>	0.3124	0.4636	0	0	1
<i>VeryLiberal</i>	0.1688	0.3747	0	0	1

Table A5: Summary statistics of pre-treatment variables.

properly.

Model	Outcome	F-statistic	p-value
1	W1	0.8204	0.7547
2	W2	0.9780	0.5038
3	W3	1.0688	0.3638
4	W4	1.2761	0.1369
5	W5	0.8753	0.6712
6	W6	0.6283	0.9509
7	W7	1.2896	0.1271
8	W8	0.8856	0.6548
9	W9	1.0353	0.4133
10	W10	1.2505	0.1570
11	W11	1.1896	0.2136
12	W12	0.9522	0.5459
13	W13	0.6905	0.9068
14	W14	1.2154	0.1881
15	W15	1.2522	0.1556
16	W16	0.5153	0.9899
17	W17	0.9356	0.5733
18	W18	0.9648	0.5254
19	W19	0.9713	0.5148
20	W20	0.9698	0.5171

Table A6: Randomization checks for treatment assignments.

## D Performance Analysis

### D.1 Ground Truth CATE Matrix

An important step in inducing adversarial missingness patterns in data is to obtain the ground truth CATE matrix. Since we have randomization for each question  $j$ , we can estimate the treatment effects estimands such as ATE and CATE for each question. For our ground truth matrix, we use Causal Forests proposed by Athey and Wager (2019). We can use any other CATE learner for this purpose. Like other CATE learners, there are three sets of inputs that we need to specify:

- *Outcome*: We use three outcomes separately: *Reward*, *Correct*, and *Easy*.
- *Treatment*: We use 20 separate treatments corresponding to the treatment status for each question. This means that for each outcome, we run 20 separate Causal Forests.
- *Covariates*: A richer set of covariates we use helps us capture heterogeneity in treatment effects. As such, we include all the pre-treatment variables as presented in Table A5. In addition, for each treatment  $j$ , we include the following six variables to our set of covariates: (1) the total number of treatments received by the user up to question  $j$ , denoted by  $TW_j$ , (2) the total number of times the user chose the easy version up to question  $j$ , denoted by  $TE_j$ , (3) the total number of times the user correctly answered  $j - 1$  questions prior to question  $j$ , denoted by  $TC_j$ , (4) the total reward received by the user up to question  $j$ , denoted by  $TR_j$ , (5) the total reward

received by choosing the easy version of the question up until question  $j$ , denoted by  $TRE_j$ , and (6) the total reward received by choosing the difficult version of the question up until question  $j$ , denoted by  $TRD_j$ . Since the treatment status in question  $j$  is independent of these variables, we can include them in our set of covariates as they help capture the heterogeneity in treatment effects. This also mimics the situation in real-world settings where platforms are able to use historical variables for targeting. For example, in similar gaming contexts, platforms assign interventions (e.g., promotion) to users based on their past performance and other historical variables.

Overall, we need to run  $3 \times 20$  Causal Forests for our 3 outcomes and 20 questions separately, which helps us build 3 ground truth CATE matrices. For each outcome, we use the corresponding ground truth CATE matrix to generate the missingness patterns, using Definition 5.

## D.2 Benchmarks

In §5.3, we focused on our proposed algorithm when XGBoost is used for CATE estimation and Double ML with Random Forest as the nuisance estimator as the benchmark. In this section, we introduce a series of alternatives for our proposed algorithm, as well as other benchmarks for conventional methods.

In all cases, the input is a missingness matrix that is generated based on the ground truth matrix and the  $\lambda$ -adversariality notion in Definition 5. The missingness matrix determines where the overlap assumption is not violated, thereby determining the feasible regions for estimating causal parameters. Given this input, we use the following versions of our proposed algorithm:

- **UXGB-SI:** We employ a U-learner with XGBoost to learn both nuisance and CATEs for each question  $j$  separately. This is the main method that we use in the main text of the paper. For each question  $j$ , we first find the set of feasible observations for which the overlap assumption is not violated according to our missingness matrix. We then use that set of observations and use XGBoost to predict  $Y_i^{(j)}$  using  $X_i^{(j)}$  as our covariates. We denote the estimated XGBoost function by  $\hat{m}_{XGB}(\cdot)$ . We then use the predicted values to construct the transformed outcome for each question  $j$  as follows:

$$U_{i,XGB}^{(j)} = \frac{Y_i^{(j)} - \hat{m}_{XGB}(X_i^{(j)})}{W_i^{(j)} - \pi(X_i^{(j)})}, \quad (\text{A.36})$$

where  $\pi$  is given and equal to  $1/2$  for the feasible points in our experiment. We can now use our covariates  $X_i^{(j)}$  to predict transformed outcomes presented in Equation (A.36) using XGBoost as follows. We denote these CATE estimates by  $\hat{\tau}_{UXGB}^{(j)}(X_i^{(j)})$ . Once we have all CATE estimates for different questions, we construct an incomplete CATE matrix  $\hat{\mathcal{T}}_{UXGB}^{\text{incomplete}}$ . Once we have the incomplete CATE matrix, we are ready to employ the matrix completion step of our algorithm.

We use softImpute as our matrix completion algorithm in this task. We use validation to tune the regularization parameter in this algorithm and then run the algorithm to complete the matrix. Our validation procedure splits the data randomly into two parts where one is used for model building using different values of the regularization parameter, and the other evaluates the performance of each model. The best-performing regularization parameter on the validation data is selected as the best parameter. Once the matrix is complete, the column-wise means would give us the ATEs for 20 different questions.

- **ULASSO-SI:** This approach is very similar to UXGB-SI, with three differences as follows: (1) we use LASSO to predict the outcomes for each question (i.e., estimating  $\hat{m}_{LASSO}(\cdot)$ ), (2) we then create transformed outcomes  $U_{i,LASSO}^{(j)}$  by replacing  $\hat{m}_{XGB}(\cdot)$  with  $\hat{m}_{LASSO}(\cdot)$  in Equation (A.36), and (3) we use LASSO to predict CATEs (i.e., estimating  $\hat{\tau}_{ULASSO}^{(j)}(X_i^{(j)})$ ) and build the incomplete matrix  $\hat{\mathcal{T}}_{ULASSO}^{incomplete}$ . Once we have the incomplete matrix, we follow the same steps as UXGB-SI in using softImpute to complete the matrix and recover ATEs.
- **GRFD-SI:** In this approach, we use Causal Forests to estimate CATEs for feasible data entries of the matrix. For each question, we use the sample of feasible data points to estimate CATEs using Causal Forests. Let  $\hat{\tau}_{GRF}^{(j)}(X_i^{(j)})$  denote the CATE estimate for observation  $i$  in question  $j$ . We plug in the point estimates for feasible observations and build the incomplete CATE matrix  $\hat{\mathcal{T}}_{GRFD}^{incomplete}$ . Once we have the incomplete matrix, we follow the same steps as UXGB-SI in using softImpute to complete the matrix and recover ATEs.
- **GRFP-SI:** This approach is very similar to GRFD-SI, with a minor difference. Instead of plugging in the point estimates for CATE, we draw from the sampling distribution of the point estimate. We can repeat this process many times and build  $K$  different incomplete CATE matrices  $\hat{\mathcal{T}}_{GRFP,1}^{incomplete}, \hat{\mathcal{T}}_{GRFP,2}^{incomplete}, \dots, \hat{\mathcal{T}}_{GRFP,K}^{incomplete}$ . We can then complete each using softImpute, similar to other methods presented above, and take the average for the final completed matrix. This helps us account for the uncertainty in the CATE estimates and reduce the impact of noisy CATE estimates that are large in magnitude.
- **SXGB-SI:** In this approach, we use an S-learner with XGBoost for learning the outcome. This approach is a form of direct outcome modeling where we use the entire data of feasible observations for each question and estimate the outcome using both covariates and the treatment variable. We use XGBoost for this prediction task and denote the resulting function by  $\hat{\mu}_{XGB}(X_i^{(j)}, W_i^{(j)})$ . We can then form an incomplete matrix of predicted outcomes with  $N$  rows and  $2J$  columns, such that  $\hat{\mathcal{T}}_{SXGB}^{incomplete} = [\hat{\mu}_{XGB}(X_i^{(j)}, 1) \mid \hat{\mu}_{XGB}(X_i^{(j)}, 0)]$ . Based on the validation procedure discussed above, we can complete this incomplete matrix of predicted outcomes using softImpute. Once we have the complete matrix, we can subtract the second  $J$  columns

from the first  $J$  columns and obtain a complete CATE matrix akin to the S-learner approach.

Recovering ATEs from the complete CATE matrix is the same as other methods.<sup>9</sup>

From the versions of our proposed algorithm presented above, we use *UXGB-SI* in the main text of our paper and present the results for the rest in this section of the Appendix. For conventional methods, we consider the following three approaches:

- **DML:** We use the Double ML method as our main benchmark in the main text of the paper. For each question  $j$ , we only focus on feasible observations and estimate ATE using Double ML regress  $Y_i^{(j)}$  as the outcome,  $W_i^{(j)}$  as the treatment, and  $X_i^{(j)}$  as controls. To do so, we need to estimate the nuisance functions for the outcome and propensity scores. We use Random Forests to estimate the outcome  $Y_i^{(j)}$  using  $X_i^{(j)}$ . In order to tune parameters, we use a two-fold cross-validation procedure. Let  $\hat{m}_{RF}(X_i^{(j)})$  denote the estimated outcomes using the Random Forests predictions. For propensity scores, we know that  $\hat{e}(X_i^{(j)}) = 1/2$  by design, so we use these values. To estimate the ATE for each question  $j$ , we regress the outcome residuals  $(Y_i^{(j)} - \hat{m}_{RF}(X_i^{(j)}))$  on propensity residuals  $(W_i^{(j)} - \hat{e}(X_i^{(j)}))$ . The estimated coefficient gives us the ATE for each question. Overall, we can obtain 20 ATEs corresponding to each question.
- **IPS:** Consistent with our discussion in the main text about the model-free approaches, we also consider Inverse Propensity Score (IPS) estimator. Let  $\mathcal{I}^{(j)}$  denote the set of feasible observations for question  $j$ . We estimate the ATE for question  $j$  using the IPS estimator as follows:

$$\hat{\tau}_{IPS}^{(j)} = \frac{1}{|\mathcal{I}^{(j)}|} \left( \sum_{i \in \mathcal{I}^{(j)}} Y_i^{(j)} \left( \frac{W_i^{(j)}}{\pi(X_i^{(j)})} - \frac{1 - W_i^{(j)}}{1 - \pi(W_i^{(j)})} \right) \right), \quad (\text{A.37})$$

where  $\pi(X_i^{(j)}) = 1/2$  by design.

- **OLS:** Finally, we also use a simple regression model where we regress the outcome  $Y_i^{(j)}$  on  $W_i^{(j)}$  for each question and estimate the coefficient.

Overall, we have 5 different versions of our proposed algorithm and 3 versions for conventional methods. We evaluate the performance of all these methods across different outcomes and missingness scenarios.

---

<sup>9</sup>Please note that we could first estimate CATEs using the S-learner approach, such that  $\hat{\tau}_{SXGB}^{(j)}(X_i^{(j)}) = \hat{\mu}_{XGB}(X_i^{(j)}, 1) - \hat{\mu}_{XGB}(X_i^{(j)}, 0)$ . However, performing the matrix completion step first is advantageous as the algorithm for completion can also incorporate the treatment structure.

## D.3 Results

### D.3.1 Full Results for All Benchmarks

In §5.3, we only compare the performance of UXGB-SI and DML for right- and left-tail missingness of CATE on *Reward*. In this section, we expand the methods and consider all 8 models described in Appendix D.2.

Table A7 presents the full results across all the benchmarks. A few patterns emerge from our results. First, we notice that all versions of our proposed algorithm perform better than the conventional methods. However, we see quite a bit of variation in performance within different versions of our proposed algorithm. In particular, we note that both XGBoost-based methods UXGB-SI and SXGB-SI perform better than other methods. This is likely because of the additional flexibility of the XGBoost model in differentiating between observations. Second, we find that the version of GRF that incorporates uncertainty (GRFP-SI) consistently performs better than the deterministic version (GRFD-SI), though the difference is not very large. Third, we find that all conventional methods perform very similarly.

### D.3.2 Full Results across Other Outcomes

So far, we have only considered *Reward* as the main outcome variable. We now extend our previous analysis to the other two outcomes collected in our experiment: (1) *Easy*, which indicates whether the user has chosen the easy version of the question, and (2) *Correct*, which indicates whether the user has correctly answered the question. We use the ground truth matrix as defined in Appendix D.1 for other outcomes to generate different missingness scenarios. We then apply all 8 methods and evaluate their performance.

We present the results for *Easy* as the outcome in Table A8. This is an important outcome since the true ATE is positive and significant for the majority of questions. Our results in Table A8 confirm our findings regarding the better performance of our proposed algorithm, specifically XGBoost-based versions. Notably, we find that our proposed algorithm performs remarkably well for cases involving left-tail missingness of CATEs. This is likely because there is more signal in the set of feasible observations, which in turn, increases the signal in the incomplete CATE matrix. This is a promising finding, as with larger samples, we can more easily overcome the low signal-to-noise problem.

Next, we focus on *Correct* as the main outcome and repeat the analysis for all 8 methods. The results are presented in Table A9. As shown in the table, we find that our proposed algorithm performs better than conventional methods. Like other cases, XGBoost-based CATE learners tend to perform better than other versions of our proposed algorithm.



Missingness		Method							
Tail	$\lambda$	UXGB-SI	ULASSO-SI	GRFD-SI	GRFP-SI	SXGB-SI	DML	IPS	OLS
Left	0.00	0.024 (0.003)	0.023 (0.004)	0.022 (0.004)	0.022 (0.004)	0.022 (0.004)	0.028 (0.006)	0.033 (0.005)	0.024 (0.004)
	0.25	0.045 (0.004)	0.069 (0.006)	0.071 (0.005)	0.067 (0.005)	0.065 (0.005)	0.078 (0.007)	0.077 (0.008)	0.075 (0.005)
	0.50	0.179 (0.005)	0.207 (0.004)	0.212 (0.004)	0.207 (0.004)	0.193 (0.004)	0.218 (0.004)	0.220 (0.005)	0.216 (0.004)
	0.75	0.267 (0.003)	0.291 (0.003)	0.299 (0.003)	0.292 (0.003)	0.264 (0.003)	0.297 (0.003)	0.298 (0.003)	0.297 (0.003)
	1.00	0.277 (0.000)	0.304 (0.001)	0.312 (0.001)	0.306 (0.000)	0.276 (0.000)	0.308 (0.002)	0.311 (0.000)	0.310 (0.000)
Right	0.00	0.023 (0.003)	0.022 (0.003)	0.022 (0.003)	0.021 (0.003)	0.022 (0.003)	0.027 (0.003)	0.037 (0.006)	0.024 (0.003)
	0.25	0.058 (0.003)	0.072 (0.004)	0.071 (0.004)	0.069 (0.004)	0.072 (0.004)	0.075 (0.005)	0.081 (0.005)	0.076 (0.004)
	0.50	0.171 (0.004)	0.209 (0.004)	0.211 (0.004)	0.206 (0.004)	0.199 (0.005)	0.216 (0.004)	0.218 (0.006)	0.217 (0.005)
	0.75	0.251 (0.002)	0.286 (0.002)	0.291 (0.002)	0.286 (0.002)	0.269 (0.002)	0.302 (0.003)	0.302 (0.003)	0.301 (0.002)
	1.00	0.260 (0.000)	0.295 (0.001)	0.302 (0.000)	0.296 (0.000)	0.280 (0.000)	0.312 (0.002)	0.311 (0.000)	0.313 (0.000)

Table A7: Performance of different methods in recovering the true ATE measured by RMSE when using *Reward* as the outcome across scenarios with different types and levels of adversariality. Numbers in parenthesis present the standard deviation of 20 replications for each model in each scenario.

### D.3.3 Robustness of the Set of Covariates

In our main analysis, we focused on the set of covariates that included all the pre-treatment variables as well as six historical variables that are based on each user’s past history with the game. Although our treatment assignment is fully randomized and therefore independent of these historical variables, one could argue that had the treatment assignments followed the adversarial cases, the historical variables would have had different values. Hence, we rerun all our models excluding historical variables. Tables A10, A11, A12 present the full results of all 8 methods across missingness scenarios with *Reward*, *Easy*, and *Correct* as the outcomes, respectively. We find that our proposed algorithm performs better than conventional methods. We also show that the XGBoost-based learners perform better than other versions.

Missingness		Method							
Tail	$\lambda$	UXGB-SI	ULASSO-SI	GRFD-SI	GRFP-SI	SXGB-SI	DML	IPS	OLS
Left	0.00	0.022 (0.003)	0.020 (0.003)	0.019 (0.003)	0.019 (0.003)	0.022 (0.003)	0.023 (0.003)	0.025 (0.003)	0.021 (0.003)
	0.25	0.023 (0.003)	0.041 (0.002)	0.040 (0.003)	0.038 (0.003)	0.023 (0.002)	0.049 (0.004)	0.049 (0.004)	0.047 (0.004)
	0.50	0.079 (0.003)	0.112 (0.003)	0.109 (0.003)	0.105 (0.003)	0.064 (0.003)	0.125 (0.003)	0.124 (0.005)	0.122 (0.004)
	0.75	0.111 (0.003)	0.142 (0.002)	0.146 (0.002)	0.140 (0.002)	0.087 (0.001)	0.165 (0.003)	0.161 (0.002)	0.159 (0.002)
	1.00	0.116 (0.000)	0.143 (0.002)	0.148 (0.000)	0.142 (0.000)	0.088 (0.000)	0.169 (0.002)	0.167 (0.000)	0.163 (0.000)
Right	0.00	0.024 (0.003)	0.020 (0.003)	0.020 (0.003)	0.020 (0.003)	0.023 (0.003)	0.024 (0.004)	0.026 (0.005)	0.023 (0.004)
	0.25	0.049 (0.003)	0.046 (0.003)	0.044 (0.003)	0.046 (0.003)	0.045 (0.003)	0.046 (0.005)	0.046 (0.005)	0.045 (0.005)
	0.50	0.099 (0.004)	0.112 (0.004)	0.120 (0.004)	0.115 (0.004)	0.111 (0.004)	0.121 (0.004)	0.120 (0.006)	0.120 (0.005)
	0.75	0.132 (0.004)	0.152 (0.004)	0.163 (0.004)	0.158 (0.004)	0.138 (0.004)	0.158 (0.004)	0.158 (0.006)	0.158 (0.005)
	1.00	0.134 (0.000)	0.136 (0.000)	0.163 (0.001)	0.159 (0.001)	0.135 (0.000)	0.162 (0.001)	0.161 (0.000)	0.162 (0.000)

Table A8: Performance of different methods in recovering the true ATE measured by RMSE when using *Easy* as the outcome across scenarios with different types and levels of adversariality.

Missingness		Method							
Tail	$\lambda$	UXGB-SI	ULASSO-SI	GRFD-SI	GRFP-SI	SXGB-SI	DML	IPS	OLS
Left	0.00	0.021 (0.003)	0.024 (0.004)	0.024 (0.004)	0.023 (0.004)	0.024 (0.004)	0.031 (0.005)	0.040 (0.007)	0.027 (0.004)
	0.25	0.053 (0.006)	0.079 (0.006)	0.081 (0.006)	0.077 (0.006)	0.075 (0.006)	0.089 (0.006)	0.089 (0.010)	0.085 (0.006)
	0.50	0.198 (0.005)	0.237 (0.005)	0.242 (0.005)	0.236 (0.005)	0.220 (0.005)	0.249 (0.006)	0.254 (0.007)	0.246 (0.005)
	0.75	0.296 (0.003)	0.331 (0.003)	0.338 (0.003)	0.334 (0.003)	0.311 (0.003)	0.347 (0.003)	0.353 (0.004)	0.347 (0.003)
	1.00	0.306 (0.000)	0.338 (0.001)	0.347 (0.001)	0.343 (0.001)	0.320 (0.000)	0.356 (0.002)	0.362 (0.000)	0.358 (0.000)
Right	0.00	0.020 (0.003)	0.021 (0.004)	0.021 (0.004)	0.020 (0.004)	0.021 (0.005)	0.027 (0.005)	0.038 (0.007)	0.023 (0.005)
	0.25	0.057 (0.004)	0.082 (0.006)	0.081 (0.006)	0.078 (0.006)	0.081 (0.005)	0.086 (0.006)	0.094 (0.008)	0.088 (0.006)
	0.50	0.197 (0.004)	0.239 (0.005)	0.244 (0.005)	0.239 (0.005)	0.228 (0.004)	0.248 (0.005)	0.252 (0.007)	0.250 (0.005)
	0.75	0.294 (0.003)	0.334 (0.002)	0.341 (0.002)	0.336 (0.002)	0.317 (0.002)	0.345 (0.003)	0.353 (0.004)	0.349 (0.002)
	1.00	0.302 (0.000)	0.341 (0.001)	0.348 (0.001)	0.343 (0.000)	0.323 (0.000)	0.352 (0.003)	0.358 (0.000)	0.356 (0.000)

Table A9: Performance of different methods in recovering the true ATE measured by RMSE when using *Correct* as the outcome across scenarios with different types and levels of adversariality.

Missingness		Method							
Tail	$\lambda$	UXGB-SI	ULASSO-SI	GRFD-SI	GRFP-SI	SXGB-SI	DML	IPS	OLS
Left	0.00	0.023	0.022	0.022	0.021	0.022	0.027	0.033	0.023
		(0.002)	(0.004)	(0.003)	(0.003)	(0.004)	(0.005)	(0.007)	(0.004)
	0.25	0.046	0.069	0.070	0.066	0.066	0.077	0.079	0.075
		(0.004)	(0.004)	(0.004)	(0.004)	(0.003)	(0.005)	(0.004)	(0.004)
	0.50	0.186	0.210	0.214	0.209	0.198	0.222	0.218	0.218
		(0.004)	(0.005)	(0.005)	(0.005)	(0.005)	(0.005)	(0.006)	(0.005)
	0.75	0.278	0.296	0.303	0.298	0.282	0.312	0.305	0.308
		(0.003)	(0.003)	(0.003)	(0.003)	(0.004)	(0.003)	(0.004)	(0.003)
	1.00	0.288	0.305	0.312	0.307	0.286	0.323	0.311	0.317
		(0.000)	(0.001)	(0.001)	(0.001)	(0.000)	(0.003)	(0.000)	(0.000)
Right	0.00	0.021	0.020	0.020	0.020	0.021	0.025	0.033	0.022
		(0.003)	(0.003)	(0.003)	(0.003)	(0.004)	(0.004)	(0.006)	(0.004)
	0.25	0.059	0.073	0.073	0.070	0.073	0.079	0.081	0.077
		(0.004)	(0.006)	(0.006)	(0.006)	(0.005)	(0.007)	(0.010)	(0.006)
	0.50	0.178	0.210	0.213	0.208	0.201	0.219	0.214	0.216
		(0.005)	(0.005)	(0.005)	(0.005)	(0.004)	(0.006)	(0.005)	(0.005)
	0.75	0.265	0.293	0.301	0.295	0.279	0.307	0.295	0.303
		(0.003)	(0.002)	(0.002)	(0.002)	(0.002)	(0.003)	(0.004)	(0.002)
	1.00	0.277	0.304	0.313	0.307	0.290	0.319	0.307	0.315
		(0.000)	(0.001)	(0.000)	(0.000)	(0.000)	(0.002)	(0.000)	(0.000)

Table A10: Performance of different methods in recovering the true ATE measured by RMSE when using *Reward* as the outcome across scenarios with different types and levels of adversariality.

Missingness		Method							
Tail	$\lambda$	UXGB-SI	ULASSO-SI	GRFD-SI	GRFP-SI	SXGB-SI	DML	IPS	OLS
Left	0.00	0.025 (0.002)	0.021 (0.003)	0.021 (0.003)	0.021 (0.003)	0.022 (0.003)	0.025 (0.004)	0.025 (0.004)	0.022 (0.004)
	0.25	0.039 (0.005)	0.060 (0.006)	0.061 (0.005)	0.057 (0.005)	0.056 (0.005)	0.068 (0.006)	0.069 (0.006)	0.066 (0.005)
	0.50	0.162 (0.004)	0.183 (0.004)	0.187 (0.004)	0.182 (0.004)	0.170 (0.003)	0.195 (0.005)	0.193 (0.005)	0.192 (0.004)
	0.75	0.231 (0.003)	0.246 (0.002)	0.253 (0.002)	0.247 (0.002)	0.220 (0.015)	0.263 (0.003)	0.259 (0.002)	0.259 (0.002)
	1.00	0.242 (0.000)	0.253 (0.000)	0.259 (0.001)	0.254 (0.001)	0.233 (0.000)	0.271 (0.002)	0.268 (0.000)	0.267 (0.000)
Right	0.00	0.024 (0.004)	0.020 (0.003)	0.019 (0.003)	0.019 (0.003)	0.020 (0.003)	0.023 (0.004)	0.026 (0.004)	0.020 (0.003)
	0.25	0.065 (0.002)	0.068 (0.003)	0.067 (0.003)	0.067 (0.003)	0.068 (0.003)	0.070 (0.005)	0.069 (0.004)	0.068 (0.004)
	0.50	0.164 (0.003)	0.187 (0.004)	0.189 (0.003)	0.185 (0.003)	0.182 (0.003)	0.194 (0.004)	0.193 (0.004)	0.193 (0.003)
	0.75	0.232 (0.002)	0.248 (0.002)	0.253 (0.002)	0.248 (0.002)	0.238 (0.002)	0.259 (0.003)	0.256 (0.003)	0.258 (0.002)
	1.00	0.239 (0.000)	0.254 (0.000)	0.259 (0.000)	0.254 (0.000)	0.242 (0.000)	0.265 (0.002)	0.263 (0.000)	0.264 (0.000)

Table A11: Performance of different methods in recovering the true ATE measured by RMSE when using *Easy* as the outcome across scenarios with different types and levels of adversariality. Numbers in parenthesis present the standard deviation of 20 replications for each model in each scenario.

Missingness		Method							
Tail	$\lambda$	UXGB-SI	ULASSO-SI	GRFD-SI	GRFP-SI	SXGB-SI	DML	IPS	OLS
Left	0.00	0.021	0.023	0.023	0.022	0.023	0.029	0.036	0.025
		(0.003)	(0.004)	(0.004)	(0.004)	(0.004)	(0.005)	(0.007)	(0.004)
	0.25	0.052	0.076	0.076	0.073	0.072	0.085	0.086	0.082
		(0.005)	(0.006)	(0.006)	(0.006)	(0.005)	(0.005)	(0.008)	(0.006)
	0.50	0.197	0.229	0.233	0.227	0.217	0.241	0.239	0.237
		(0.005)	(0.005)	(0.005)	(0.005)	(0.004)	(0.006)	(0.008)	(0.005)
	0.75	0.290	0.319	0.326	0.321	0.301	0.335	0.325	0.331
		(0.002)	(0.003)	(0.002)	(0.002)	(0.003)	(0.003)	(0.004)	(0.002)
	1.00	0.304	0.330	0.337	0.333	0.311	0.348	0.336	0.343
		(0.000)	(0.001)	(0.001)	(0.001)	(0.000)	(0.002)	(0.000)	(0.000)
Right	0.00	0.020	0.022	0.022	0.021	0.023	0.029	0.038	0.025
		(0.002)	(0.003)	(0.003)	(0.003)	(0.003)	(0.004)	(0.004)	(0.004)
	0.25	0.056	0.078	0.078	0.075	0.077	0.084	0.088	0.083
		(0.004)	(0.007)	(0.007)	(0.007)	(0.006)	(0.007)	(0.008)	(0.007)
	0.50	0.192	0.229	0.233	0.227	0.220	0.238	0.235	0.237
		(0.005)	(0.004)	(0.004)	(0.004)	(0.004)	(0.006)	(0.008)	(0.004)
	0.75	0.288	0.322	0.329	0.322	0.305	0.334	0.327	0.331
		(0.002)	(0.003)	(0.002)	(0.002)	(0.004)	(0.004)	(0.003)	(0.002)
	1.00	0.300	0.328	0.339	0.332	0.311	0.344	0.334	0.341
		(0.000)	(0.001)	(0.001)	(0.001)	(0.000)	(0.003)	(0.000)	(0.000)

Table A12: Performance of different methods in recovering the true ATE measured by RMSE when using *Correct* as the outcome across scenarios with different types and levels of adversariality.