Tackling Interference Induced by Data Training Loops in A/B Tests: A Weighted Training Approach

Anonymous Author(s)
Affiliation
Address
email

Abstract

The standard data-driven pipeline in contemporary recommendation systems involves a continuous cycle in which companies collect historical data, train subsequently improved machine learning models to predict user behavior, and provide improved recommendations. The user's response, which depends on the recommendation produced in this cycle, will become future training data. However, these data training-recommendation cycles can introduce interference in A/B tests, where data generated by control and treatment algorithms, potentially with different distributions, are aggregated together. To address these challenges, we introduce a novel approach called weighted training. This approach entails training a model to predict the probability of each data point appearing in either the treatment or control data and subsequently applying weighted losses during model training. We demonstrate that this approach achieves the least variance among all estimators that do not cause shifts in the training distributions. Through simulations, we demonstrate the lower bias and variance of our approach compared to other methods.

1 Introduction

Experimentation (A/B tests) has emerged as the standard method for evaluating feature and algorithmic updates in online platforms; see comprehensive guidance in Kohavi et al. [2020]. Instances of the use of A/B tests abound and are wide-ranging, from testing new pricing strategies in e-commerce, evaluating bidding strategies in online advertising, and updating and fine-tuning ranking algorithms in video-sharing platforms, just to name a few.

In such online platforms, recommendation systems are also in place to enhance user experience by displaying relevant products and engaging videos. The standard pipeline in recommendation systems operates as follows (as illustrated in Figure 1):

1) Using historical data, the system trains various machine learning (ML) models to predict users' behaviors, such as their interest in recommended items and their willingness to purchase certain products. 2) When a user request is received, the system identifies relevant items and ranks them based on the training scores generated by the machine learning models. 3) Items are recommended to users based on the ranking. 4) Users interact with the recommended items and take actions, including leaving comments below videos and making specific purchases. 5) The system records these user actions and feeds them back into the ML models, facilitating continuous model training.

This pipeline ensures that the recommendation system continuously adjusts and enhances its suggestions, taking into account user interactions and feedback. However, it also generates a feedback loop, a phenomenon discussed in both Jadidinejad et al. [2020] and Chaney et al. [2018]. As we will demonstrate later, this feedback loop causes interference in A/B tests.

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

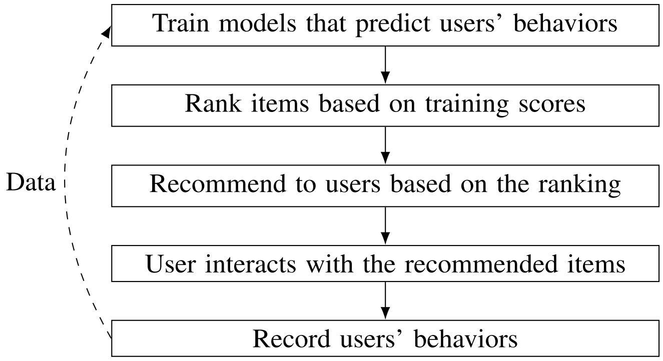

Train models that predict users' behaviors
Rank items based on training scores
Recommend to users based on the ranking
User interacts with the recommended items
Record users' behaviors
Data

Figure 1: A standard pipeline in recommendation system.

Interference, in the context of experimental design, means the violation of the Standard Unit Treatment Value Assumption (SUTVA) [Imbens and Rubin, 2015]. According to SUTVA, the outcome for a given unit should solely depend on its treatment assignment and its own characteristics, and it should remain unaffected by the treatment assignments of other units. However, when data training loops are present, prior data generated under specific treatment assignments can lead to distinct model predictions. These predictions, in turn, can influence the outcomes observed for subsequent units, thereby violating the assumptions of SUTVA.

More specifically, let's consider a user-side experiment testing two distinct ranking algorithms. In this scenario, we split the traffic in such a way that control users are subjected to control algorithms, and treatment users are subjected to treatment algorithms. Control and treatment algorithms generate data that may follow different distributions. These data sets are then combined and fed back into the ML models. This experimental procedure is represented in Figure 2.

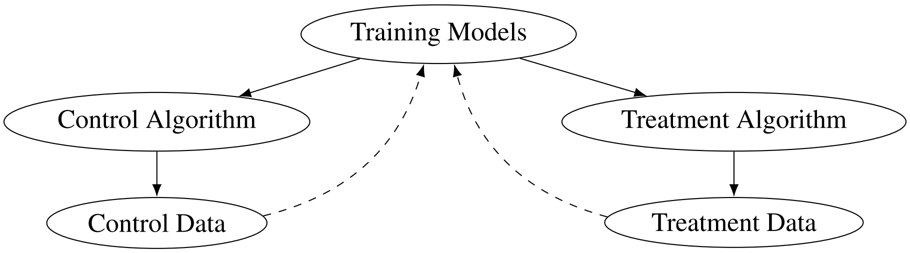


Figure 2: An A/B testing procedure

However, it's essential to recognize that this pooled distribution is distinct from both the control data and the treatment data distributions. It is widely acknowledged that variations in training distributions can lead to significantly different predictions. To further illustrate this issue, let's consider the following example.

Example 1 (Experimenting parameters of fusion formulas)**.** Imagine a video-sharing platform with two distinct ML models that predict finishing rates (FR) and stay durations (SD), respectively. The platform's ranking algorithms rank videos using a linear fusion formula: $\alpha_1\text{FR} + \alpha_2\text{SD}$. In an A/B test, we aim to compare different parameter values $\{\alpha_1, \alpha_2\}$. Let us consider a scenario where the platform hosts two types of videos: short videos, which typically have high finishing rates and low stay durations, and long videos, which exhibit the opposite characteristics. If the treatment algorithm assigns a higher α_2 to stay durations than the control algorithm, it will recommend more long videos in the treatment group. As a result, in the A/B tests, there will be a higher proportion of long videos in the pooled distribution. This can lead to different estimates of finishing rates and stay durations by the ML models, subsequently altering the recommendation outcomes produced by both the control and treatment algorithms.

This interference caused by data training loops closely relates to the concept of "symbiosis bias" recently introduced in Holtz et al. [2023]. In their paper, they discuss cluster randomized designs and data-diverted designs. Through simulations, they demonstrate that these designs can effectively reduce biases compared to the naive approach.

In this paper, we introduce a weighted training approach. The concept revolves around recognizing that a control data point may also appear in the treatment data with a different probability. To harness this insight, we create a new model that predicts the probability of each data point appearing in either the treatment or control data. Subsequently, we train the ML models using losses that are weighted based on these predicted probabilities. By doing so, we show that if the weights are accurately learned, there will be no shifts in the training distributions, while making the most efficient use of available data. Furthermore, even if that the weights are not learned perfectly, we demonstrate that our method is still better than benchmarks.

2 Related Literature

2.1 Interference in Experiments

The existence of interference is well-known in the literature. Empirical studies [Blake and Coey, 2014, Holtz et al., 2020, Fradkin, 2015] validate that the bias caused by the interference could be as large as the treatment effect itself. In the following, we review the literature on various types of interference in A/B tests that are relevant to ours.

Interference in two-sided marketplaces. In two-sided marketplaces, A/B tests are subject to interference due to competition and spillover effects. Johari et al. [2022] and Li et al. [2022] analyze biases in both user-side and supply-side experiments using stylized models. Additionally, Bright et al. [2022] consider a matching mechanism based on linear programming and propose debiased estimators via shadow prices. To mitigate bias, Johari et al. [2022] and Bajari et al. [2021] introduce two-sided randomizations, which are also known as multiple randomization designs. To measure the effectiveness of “cold start" algorithms, Ye et al. [2023] propose a similar yet different two-sided split design. Bipartite experiments are also introduced in Eckles et al. [2017], Pouget-Abadie et al. [2019], Harshaw et al. [2023], where the treatments are assigned in one group of units and the metrics are measured in another group of units. Cluster experiments can also be applied in marketplaces, as shown in Holtz et al. [2020], Holtz and Aral [2020]. Building on an equilibrium model, Wager and Xu [2021] propose a local experimentation approach capable of accurately estimating small changes in system parameters. Additionally, this idea has been extended by Munro et al. [2021], who combined it with Bernoulli experiments to estimate treatment effects of a binary intervention. For supply-side (seller-side) experiments, Ha-Thuc et al. [2020] and Nandy et al. [2021] put forth a counterfactual interleaving framework widely implemented in the industry and Wang and Ba [2023] enhance the design with a novel tie-breaking rule to guarantee consistency and monotonicity. In the context of advertising experiments, Liu et al. [2021] propose a budget-split design and Si et al. [2022] use a weighted local linear regression estimation in situations where the budget is not perfectly balanced between the treatment and control groups.

Interference induced by feedback loops. Feedback loops commonly exist in complex systems. For instance, in the context of our earlier discussion in the Introduction, data obtained from recommendations is fed back into the underlying machine learning models. In online advertising platforms, the ads shown previously can impact the subsequent ads' recommendations and bidding prices, primarily due to budget constraints. However, there is relatively limited literature that delves into experimental design dealing with interference caused by feedback loops. Goli et al. [2023] attempt to address such interference by offering a bias-correction approach that utilizes data from past A/B tests. In the context of searching ranking system, In the context of search ranking systems, Musgrave et al. [2023] suggest the use of query-randomized experiments to mitigate feature spillover effects. Additionally, for testing bandit learning algorithms, Guo et al. [2023] propose a two-stage experimental design to estimate the lower bound and upper bound of the treatment effects. Furthermore, as mentioned earlier, Holtz et al. [2023] explore similar issues to ours, which they refer to as “Symbiosis Bias."

Markovian interference. When a treatment can influence underlying states, subsequently affecting outcomes in the following periods, we refer to these experiments as being biased by Markovian interference. A classic example is experimentation with different matching or pricing algorithms in ride-sharing platforms. Farias et al. [2022] proposes a difference-in-Q estimator for simple Bernoulli experiments, and its performance is further validated through a simulation study with Douyin [Farias et al., 2023]. Moreover, leveraging Markov decision processes, optimal switchback designs have been analyzed in depth by Glynn et al. [2020] and Hu and Wager [2022]. In the specific context of queuing, Li et al. [2023] have conducted a study on switchback experiments and local perturbation experiments. They have discovered that achieving higher efficiency is possible by carefully selecting estimators based on the structural information of the model.

Temporal interference. Temporal interference arises when there are carry-over effects. Extensive investigations have been conducted on switchback experiments [Bojinov et al., 2023, Hu and Wager, 2022, Xiong et al., 2023a,b]. Besides switchback experiments, other designs [Basse et al., 2023, Xiong et al., 2019] have also been proposed and proven to be optimal in various contexts. In cases involving both spatial and temporal interference, the new designs proposed in Ni et al. [2023] combine both switchback experiments and clustering experiments.

2.2 Feedback Loops in Recommendation Systems

As modern platforms increasingly employ complex systems, issues arising from feedback loops are becoming more pronounced. Researchers such as Chaney et al. [2018], Mansoury et al. [2020], and Krauth et al. [2022] have investigated problems related to the amplification of homogeneity and popularity biases due to feedback loops. Additionally, Yang et al. [2023] and Khenissi [2022] have noted that these feedback loops can lead to fairness concerns. The concept of user feedback loops and methods for debiasing them are discussed in Pan et al. [2021], while Jadidinejad et al. [2020] consider how feedback loops affect underlying models. In our work, we specifically focus on data training feedback loops and propose valid methods to address their impact on A/B tests.

3 A Framework of A/B Tests Interfered by Data training Loops

In this section, we construct a potential outcomes model [Imbens and Rubin, 2015] for A/B tests that incorporate the training procedures. Through our model, we will demonstrate the presence of interference induced by data training loops in A/B tests.

We are focusing on user-side experiments, where users are assigned randomly to the treatment group with a probability of p and to the control group with a probability of $1-p$. Suppose there are d features associated with each user-item pair, and the system needs to predict m different types of user behaviors (e.g., finishing rates, stay durations). We represent the feature space as $\mathcal{X}$, which is a subset of $\mathbb{R}^d$, and the outcome space as $\mathcal{Y}$, which is a subset of $\mathbb{R}^m$. In modern large-scale recommendation systems, d can be as extensive as billions, and m can encompass hundreds of different behaviors. We define a model class $\mathcal{M} = \{M_\theta, \theta \in \Theta\}$, which includes various models $M_\theta : \mathcal{X} \to \mathcal{Y}$. These models are responsible for predicting user behaviors based on user-item features. In this representation, we consolidate the prediction of m distinct user behaviors into a single model, which yields a m-dimensional output for the sake of simplicity and convenience. In subsequent discussions, we will omit the subscript θ for ease of notation.

At time t, the training model M_t is trained from the previous model M_{t-1} with additional data from time $t-1$, denoted as $\mathcal{D}_{t-1}$. This training process is written as $M_t = \mathcal{F}\left(M_{t-1}, \mathcal{D}_{t-1}\right)$, where $\mathcal{F}$ denotes a training algorithm, e.g. stochastic gradient descent (SGD) or Adam [Kingma and Ba, 2014].

Further, at time t, we suppose there are n_t new users have arrived. For the i-user, $i = 1, 2, \ldots, n_t$, the system recommends an item with a feature vector $X_{i,t} = X_{i,t}\left(M_t, Z_{i,t}\right) \in \mathbb{R}^d$, where $Z_{i,t} \in \{0, 1\}$ denotes the treatment assignment. Subsequently, the potential outcome for this user is given as $Y_{i,t} = Y_{i,t}\left(X_{i,t}\right) \in \mathcal{Y}$, which represents the user's behaviors. Note that $Y_{i,t}$ is independent to $Z_{i,t}$ and M_t, given the feature vector $X_{i,t}$. This assumption is grounded in the typical behavior of recommendation systems, where the primary influence on users' behaviors stems from the modification of recommended items. Thus, $Y_{i,t}$ is not directly dependent on the treatment assignment $Z_{i,t}$ or the model state M_t once the features $X_{i,t}$ are accounted for. We remark that our approach can be readily extended to cases where the treatment variable Z directly affects the outcome Y, as we shall see in Lemma 1. Due to the data training loops, the data collected at time t is incorporated into the training dataset as follows

$$\mathcal{D}_t = \left\{\left(X_{1,t}, Y_{1,t}\right), \left(X_{2,t}, Y_{2,t}\right), \ldots, \left(X_{n_t,t}, Y_{n_t,t}\right)\right\}.$$

We plot the causal graph [Pearl, 2000] in Figure 3 to illustrate the dependence in the data training loops.

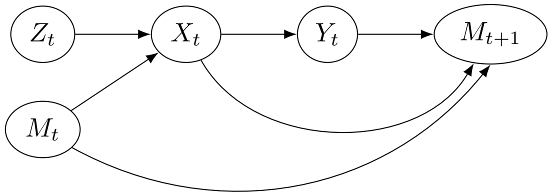


Figure 3: Dependence of different objects in the data training loops, where we omit the subscript i for simplicity

It's important to note that $\mathcal{D}_t$ consists of recommendation data, which may differ from the control and treatment data. Consequently, when applying the training algorithm $\digamma$, the model at the next time step, M_{t+1}, will differ from the model trained solely on control or treatment data. This, in turn, impacts the recommendations $X_{\cdot,t+1}$ at the subsequent period. Therefore, it becomes evident that these A/B tests are susceptible to interference caused by data training loops.

Our objective is to estimate the global treatment effect (GTE), which is defined as the difference between the metrics observed under the global treatment and the global control regimes. The global treatment regime is defined as having all $Z_{i,t}$ equal to one, while the global control regime is defined as having all $Z_{i,t}$ equal to zero. In mathematical terms, we represent this as follows: within the global treatment regime, the procedure is outlined as:

$$\begin{aligned} X_{i,t}^{\mathrm{GT}} &= X_{i,t}\left(M_t^{\mathrm{GT}},1\right), Y_{i,t}^{\mathrm{GT}} = Y_{i,t}\left(X_{i,t}^{\mathrm{GT}}\right), \\ \mathcal{D}_t^{\mathrm{GT}} &= \left\{\left(X_{1,t}^{\mathrm{GT}},Y_{1,t}^{\mathrm{GT}}\right),\left(X_{2,t}^{\mathrm{GT}},Y_{2,t}^{\mathrm{GT}}\right),\ldots,\left(X_{n_t,t}^{\mathrm{GT}},Y_{n_t,t}^{\mathrm{GT}}\right)\right\} \\ M_t^{\mathrm{GT}} &= \digamma\left(M_{t-1}^{\mathrm{GT}},\mathcal{D}_{t-1}^{\mathrm{GT}}\right), \quad \text{for } t=1,\ldots,T; \end{aligned}$$

Global control regime follows a similar procedure, where GT is replaced with GC. Here, we assume $\mathcal{D}_0^{\mathrm{GC}} = \mathcal{D}_0^{\mathrm{GT}}$ and $M_0^{\mathrm{GC}} = M_0^{\mathrm{GT}}$. The m-dimensional GTE is defined as

$$\mathrm{GTE} = \mathbb{E}\left[\frac{1}{\sum_{t=1}^{T} n_t}\sum_{t=1}^{T}\sum_{i=1}^{n_t}\left(Y_{i,t}^{\mathrm{GT}} - Y_{i,t}^{\mathrm{GC}}\right)\right].$$

In the naive A/B tests, the estimator is

$$\frac{1}{\sharp\left\{Z_{i,t}=1\right\}}\sum_{Z_{i,t}=1} Y_{i,t}\left(X_{i,t}\left(M_t,1\right)\right) - \frac{1}{\sharp\left\{Z_{i,t}=0\right\}}\sum_{Z_{i,t}=0} Y_{i,t}\left(X_{i,t}\left(M_t,0\right)\right), \tag{1}$$

where $\sharp\left\{Z_{i,t}=1\right\}$ and $\sharp\left\{Z_{i,t}=0\right\}$ are the number of users in the treatment and control, respectively. Because of the interference induced by data training loops, it is possible for the estimator to exhibit bias when estimating the Global Treatment Effect (GTE).

4 A Weighted Training Approach

Based on the potential outcome model established in Section 3, it becomes apparent that interference arises due to shifts in the training distributions. In this section, we will introduce an approach that assigns weights to the original data distributions obtained from the A/B tests. We will demonstrate that these weighted distributions have the capability to recover the data distributions for the control group and the treatment group.

In abstract terms, constructed in a probability space $(\Omega,\mathcal{F},P)$, let $D=(X,Y)$ be the random variable representing some data of $(X,Y)\in\mathcal{X}\times\mathcal{Y}$. Specifically, $D_C=(X_C,Y_C), D_T=(X_T,Y_T)$ be the random variable representing control data and treatment data, respectively. We use $\mathcal{D}_C,\mathcal{D}_T$ to denote the distributions of the control data and treatment data, respectively. Therefore, by using $\mathcal{L}(\cdot)$ to denote the law (distribution) of a random variable, we have $\mathcal{D}=\mathcal{L}(D), \mathcal{D}_C = \mathcal{L}(D_C)$ and $\mathcal{D}_T=\mathcal{L}(D_T)$. Let the treatment assignment Z also be constructed in the same probability space. Importantly, Z is independent to $\{D_C,D_T\}$, i.e., $Z\perp\{D_C,D_T\}$, which is the unconfoundedness assumption in casual inference [Rosenbaum and Rubin, 1983]. The random variable $D_E = \{X_E, Y_E\}$ represents the data obtained from the experiment and can be expressed as follows: $D_E = D_T Z + D_C (1 - Z)$, where $P(Z = 1) = p$ represents the probability of treatment assignment. Consequently, the distribution of the experimental data can be described as: $\mathcal{D}_E = p\mathcal{D}_T + (1-p)\mathcal{D}_C$, due to the independence of Z and $\{D_C, D_T\}$.

Our objective is to shift the distribution of experimental data $\mathcal{D}_E$ towards that of the control data $\mathcal{D}_C$ and the treatment data $\mathcal{D}_T$ to mitigate bias. To achieve this, we introduce a weighting function $W(\cdot) : \Omega \to \mathbb{R}_+$, with the property that $\mathbb{E}[W] = 1$. We denote the resulting weighted distribution as $W\mathcal{D}$, i.e., $W\mathcal{D}(A)$ $=\mathbb{E}\left[WI\left\{D \in A\right\}\right]$ for any measurable A in $\mathcal{X} \times \mathcal{Y}$, where we primarily focus on $\mathcal{D} = \mathcal{D}_E$ and $D = D_E$ in this paper. It is easy to check $W\mathcal{D}$ in $\mathcal{X} \times \mathcal{Y}$ is also a probability distribution as $W(\cdot)$ is non-negative and $\mathbb{E}[W] = 1$.

Our first result, presented below, demonstrates that by selecting the weight function as $\mathbb{E}\left[Z|X_E\right]/p$ or $\left(1 - \mathbb{E}\left[Z|X_E\right]\right)/(1-p)$, we can effectively recover the treatment and control data distributions, respectively.

Lemma 1. *The weighted functions*

$$W_T(X_E, Y_E, Z) = \frac{\mathbb{E}\left[Z|X_E\right]}{p} \text{ and } W_C(X_E, Y_E, Z) = \frac{1 - \mathbb{E}\left[Z|X_E\right]}{1-p}$$

satisfy $W_T\mathcal{D}_E\overset{d}{=}\mathcal{D}_T$ *and* $W_C\mathcal{D}_E\overset{d}{=}\mathcal{D}_C$, *where* $\overset{d}{=}$ *means equal in distribution.*

Remark: In cases where the treatment variable Z is able to directly affect the outcome Y_E, the adjustment can be made by substituting the conditional expectation $\mathbb{E}\left[Z|X_E\right]$ with $\mathbb{E}\left[Z|X_E, Y_E\right]$.

The proof of Lemma 1 is presented in Appendix A. Lemma 1 shows that we are able to reconstruct the treatment and control data distributions from the A/B testing data distribution, provided that we can estimate $\mathbb{E}\left[Z|X_E\right]$ with sufficient accuracy.

Since the quantity $\mathbb{E}\left[Z|X_E\right]$ is typically unknown beforehand, it becomes necessary to estimate it from the available data. To achieve this, we construct an additional machine learning model denoted as G_{θ_W}. This model is trained using the data $\{X_E, Z\}$ obtained from the experiments, treating it as a classification problem. Subsequently, the predictions generated by G_{θ_W} are utilized as weights (after proper normalization) to form weighted losses for the original machine learning models. This method is detailed in Algorithm 1.

We remark that while $\mathbb{E}\left[Z|X_E\right]$ might be complex, there is no need for precise estimation in practical applications. In fact, simple models like two-layer neural networks perform well, as demonstrated in our numerical results. Even if $\mathbb{E}\left[Z|X_E\right]$ is not estimated accurately, our method remains competitive with benchmark methods. Specifically, if $G_{\theta_W}(\cdot)$ outputs an uninformative value of p, it reduces to the naive estimator. Conversely, if $G_{\theta_W}(\cdot)$ overfits and produces extreme values close to 1 or 0, it behaves similarly to the data splitting method.

From the proof of Lemma 1, one may note that the simple weight function $\tilde{W} = Z$ also satisfy $\tilde{W}\mathcal{D}_E\overset{d}{=}\mathcal{D}_T$. Indeed, using $Z_{i,t}$ instead of training a model G_{θ_W} in Algorithm 1 results in a data splitting approach, also known as a data-diverted experiment, as discussed in Holtz et al. [2023]. In such experiments, each model is updated exclusively using data generated by users exposed to the corresponding algorithm. However, this approach lacks data efficiency, as it utilizes only a fraction of the data, namely p for the treatment model and $1 - p$ for the control model. For instance, in cases where the control data distribution is identical to the treatment data distribution, our approach can leverage all available data for training both control and treatment models. This is because $\frac{\mathbb{E}[Z|X_E]}{p} = \frac{1-\mathbb{E}[Z|X_E]}{1-p} = 1$ in this case.

Intuitively, in the finite sample regime with n samples, the variance of the estimator should be proportional to $\frac{1}{n^2}\sum_{i=1}^{n}\left(W_i/p\right)^2$. In the following, we will demonstrate that our approach can achieve this lower variance, defined in this manner, among all possible weights without causing shifts in the training distributions.

Theorem 1. $W_T(X_E, Y_E, Z) = \mathbb{E}\left[Z|X_E\right]/p$ *attains the minimum of the following optimization problem:* $\min_{W(\cdot):\Omega\to\mathbb{R}_+}\left\{\mathbb{E}\left[W^2\right] : W\mathcal{D}_E\overset{d}{=}\mathcal{D}_T\right\}$. *Similarly,* $W_C(X_E, Y_E, Z) = \left(1 - \mathbb{E}\left[Z|X_E\right]\right)/(1-p)$ *attains the minimum of the following optimization problem:*$\min_{W(\cdot):\Omega\to\mathbb{R}_+}\left\{\mathbb{E}\left[W^2\right] : W\mathcal{D}_E\overset{d}{=}\mathcal{D}_C\right\}$.

Algorithm 1 A weighted training approach for A/B tests

```
Require: The probability of treatment assignment: $p$; a model class for the weight prediction: $\mathcal{G} = \{G_{\theta_W} : \mathbb{R}^d \to [0,1], \theta_W \in \Theta_W\}$; the machine learning model class: $\mathcal{M} = \{M_\theta : \mathcal{X} \to \mathcal{Y}, \theta \in \Theta\}$; loss functions: $\ell(M(X), Y)$ (could be $m$-dimensional).
Initialize two models, the treatment model $M_{\theta_T}$ and the control model $M_{\theta_C}$, both of which are set to the current production model.
for $t \leftarrow 1$ to the end of the experiment do
    for $i \leftarrow 1$ to $n_t$ do
        User $i$ arrives. The platform randomly assigns user $i$ to the treatment with probability $p$.
        When a user is assigned to the treatment group, the platform recommends an item based on the treatment algorithm and model, and vice versa. Collect data $(X_{i,t}, Y_{i,t}, Z_{i,t})$.
    end for
    Compute weights:
        $$W_{T,i,t} = \frac{G_{\theta_W}(X_{i,t})}{p} \text{ and } W_{C,i,t} = \frac{1 - G_{\theta_W}(X_{i,t})}{1-p}, \text{ for } i = 1, 2, \ldots, n_t.$$
    Update the treatment model $M_{\theta_T}$ and the control model $M_{\theta_C}$ by minimizing the weighted losses, respectively
        $$\frac{1}{n_t}\sum_{i=1}^{n_t} W_{T,i,t}\ell(M_{\theta_T}(X_{i,t}), Y_{i,t}) \text{ and } \frac{1}{n_t}\sum_{i=1}^{n_t} W_{C,i,t}\ell(M_{\theta_C}(X_{i,t}), Y_{i,t}).$$
    Update the model $G_{\theta_W}$ using data $\{(X_{i,t}, Z_{i,t}), i = 1, \ldots, n_t\}$.
end for
    return the estimator (1).
```

Theorem 1 implies that our proposed weights, $\frac{\mathbb{E}[Z|X_E]}{p}$ and $\frac{1-\mathbb{E}[Z|X_E]}{1-p}$, achieve maximum data efficiency while adhering to the constraint of no training distributional shifts. The proof of Theorem 1 is provided in Appendix A.

5 Numerical Results

In this section, we present simulation results. In subsection 5.1, we specify the simulation setup and the implementation details. In subsection 5.2, we simulate A/B tests to demonstrate the lower bias and variance of our approach compared to other methods. Additional experiments and results on A/B tests and A/A tests can be found in Appendix C.

5.1 Simulation Setups

We conducted a simulation inspired by Example 1 in Introduction. In this simulation, we consider two types of videos: long and short, and the recommendation system relies on two metrics: finishing rates (FR) and stay durations (SD). Users arrive sequentially, and for each user, there are a total of $N = 100$ candidate videos available. These videos are divided into two equal groups, with half of them being long videos and the other half being short videos. The platform selects one video from this pool to show to each user. Furthermore, we assume that the features for user-video pairs are 10-dimensional, following independent uniform distributions in the range [0,1]. Additionally, we assume linear models that

$$\text{FR}_{\text{short}} = \text{Sigmoid}(\beta_{\text{FR,short}}^\top X - 2.5), \text{FR}_{\text{long}} = \text{Sigmoid}(\beta_{\text{FR,long}}^\top X - 2.5),$$
$$\text{SD}_{\text{short}} \sim \exp\left(\beta_{\text{SD,short}}^\top X\right), \text{SD}_{\text{long}} \sim \exp\left(\beta_{\text{SD,long}}^\top X\right),$$

where Sigmoid means the sigmoid function and $\exp(\cdot)$ means an exponential distribution and

$$\begin{aligned}\beta_{\text{FR,short}} &= 0.9 \times [0, 0.1, 0.2, \ldots, 0.9], \beta_{\text{FR,long}} = 0.6 \times [0, 0.1, 0.2, \ldots, 0.9],\\ \beta_{\text{SD,short}} &= [1, 0.9, 0.8 \ldots .0.1], \beta_{\text{SD,long}} = 1.5 \times [1, 0.9, 0.8 \ldots .0.1].\end{aligned}$$

The user's decision to finish watching a video or not follows a Bernoulli distribution with a probability equal to the finishing rate. By setting the parameters in this manner, we ensure that short videos generally have high finishing rates and short stay durations, while long videos are thee opposite.

The machine learning models employ logistic regression for predicting finishing rates and linear regression for predicting stay durations. The feature set consists of 10 user-video pair features, along with an indicator variable that specifies whether the video is long or short. It's important to note that there is a model misspecification present, as the true parameters for long and short videos are different. In our machine learning models, we assume these parameters to be equal, but we introduce an additional parameter corresponding to the video length indicator for an adjustment.

We employ Stochastic Gradient Descent (SGD) to train both machine learning models, employing a batch size of $B = n_1 = n_2 = \ldots = n_T = 128$ for all time steps. The learning rate is set to 0.1. Throughout all simulations, we maintain a fixed value of $T = 10000$. Consequently, the total number of users involved in the experiments amounts to 1,280,000.

The platform recommends the video that yields the highest value among the 100 candidate videos based on the following formula: $\alpha\widehat{\text{FR}} + \widehat{\text{SD}}$, where $\widehat{\text{FR}}$ and $\widehat{\text{SD}}$ represent the predictions generated by the machine learning models. The A/B tests are designed to assess the difference between two distinct α values. We focus on three metrics, FR, SD, and the proportion of short videos on the platform.

We compare our approach to three other methods: data pooling, snapshot, and data splitting methods.

Data pooling: This is the standard naive approach, where machine learning models are trained on the combined control and treatment data.

Snapshot: In this method, the machine learning models are never retrained during the A/B tests. Predictions are solely based on the models' initial snapshot at the beginning of the experiments.

Data splitting: Also known as data-diverted, as discussed in Holtz et al. [2023], each model is exclusively trained on the data obtained from its respective algorithm.

While Holtz et al. [2023] also explore cluster randomized experiments, it's worth noting that in our specific context, determining how to cluster users presents challenges. Consequently, we do not make direct comparisons with cluster randomized experiments. As we discussed in Section 4, the data splitting method may encounter several challenges:

High variance. Since machine learning models can only see a portion of the data, the lack of data efficiency may lead to high variance in model estimators, resulting in increased variance in the experimental metrics.

External validity. In our simulation, the data splitting method is equivalent to reducing the batch size. It is well-known that batch size plays a crucial role in machine learning, and different batch sizes can yield fundamentally different performances. Therefore, treatment effect estimates in scenarios with small batch sizes may not accurately predict treatment effects in scenarios with large batch sizes, compromising external validity.

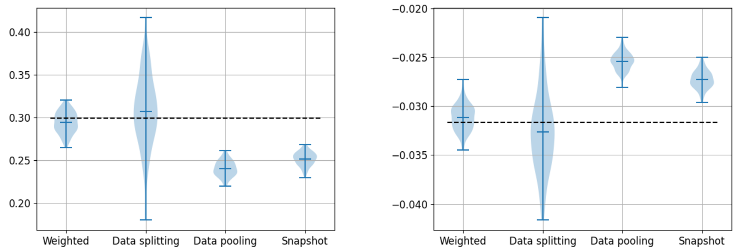
Experimentation costs. In today's platforms, thousands of experiments run each day. Consequently, experimentation costs cannot be overlooked, even though each experiment only runs for a relatively short period. Reducing the data size can compromise the performance of the machine learning model, potentially leading to suboptimal recommendations and increased experimentation costs.

In our approach, we employ a two hidden layer fully connected network with ReLU activations to train the weighting model G_{θ_W}. Each layer comprises 64 neurons, and we utilize the Adam optimizer [Kingma and Ba, 2014] with a learning rate of 0.001. Our training process for the weighting model commences after the initial 200 periods. During these initial 200 periods, the control and treatment machine learning models are trained as in the data splitting method.

Subsequently, we will conduct the A/B tests 100 times and create violin plots to visualize the estimated treatment effects.

5.2 A/B Tests

We first examine the comparison between the control parameter $\alpha_C = 10$ and the treatment parameter $\alpha_T = 9$ with a treatment assignment probability of $p = 1/2$. The logloss of our weighting model is about 0.96 (base 2). This value indicates a performance slightly above that of a purely random guess, which would yield a logloss of 1. Despite the marginal improvement over randomness in terms of logloss, it's crucial to highlight that this translates into significant gains in the accuracy of treatment effect estimation, as we shall see in Figure 4 and Table 1.



(a) treatment effects: SD

(b) treatment effects: FR

Figure 4: A/B testing results for $\alpha_C = 10$, $\alpha_T = 9$, and $p = 1/2$

In the figure, the black dotted line represents the true GTE, which have been computed through simulation. We present various estimators obtained from 100 independent A/B tests along with their respective mean, lower, and upper bounds. Specifically, we provide results for treatment effect estimators of the metrics SD and FR, respectively.

Additionally, we provide information on the bias and standard errors of treatment effect estimators obtained using various methods in Table 1. In each metric, the first column represents the bias in comparison to the true GTE. The second column displays the standard deviation calculated from the results of the 100 A/B tests. Lastly, the third column showcases the standard error estimates obtained through two-sample t-tests in a single A/B test.

Table 1: Bias, standard deviation, and standard error estimated from the experiment for the metrics in the case that $\alpha_C = 10$, $\alpha_T = 9$, and $p = 1/2$

	Stay durations			Finishing rates		
	Bias	STD	SE	Bias	STD	SE
Weighted	-0.005	0.012	0.008	0.000	0.001	0.001
Splitting	0.008	0.042	0.008	-0.001	0.004	0.001
Pooling	-0.059	0.009	0.008	0.006	0.001	0.001
Snapshot	-0.047	0.008	0.009	0.004	0.001	0.001

From Figure 4 and Table 1, it is evident that our approach consistently demonstrates the lowest bias across all metrics compared to other approaches. The data splitting method also manages to achieve relatively low biases but exhibits significantly higher variance. Furthermore, it's worth noting that the true variance of the data splitting estimator is considerably larger than the standard error estimated from a two-sample t-test. Consequently, this could lead to confidence intervals that underestimate the true level of variability.

Additional results for the experiments conducted in this section are in Appendix B.

6 Concluding Remarks

In this paper, we have introduced a weighted training approach designed to address the interference problem caused by data training loops. Our approach has demonstrated the capability to achieve low bias and reasonable variance. For future research, we have identified several intriguing directions: the first one is single model training: In our current approach, we still require training two separate models, which can be computationally expensiveIt would be interesting to explore whether it's possible to train a single model and implement adjustments to mitigate bias effectively. The second one is about variance estimation and inference. Although our approach has shown promise in reducing bias, the variance remains larger than the standard error estimated from the two-sample t-test in some cases. As a result, there is a need for more robust methods for estimating variance and developing new inference techniques that can account for the specific challenges in interference induced by data training loops in A/B tests.

References

Peter M Aronow and Cyrus Samii. Estimating average causal effects under general interference, with application to a social network experiment. 2017.

Patrick Bajari, Brian Burdick, Guido W Imbens, Lorenzo Masoero, James McQueen, Thomas Richardson, and Ido M Rosen. Multiple randomization designs. *arXiv preprint arXiv:2112.13495*, 2021.

Guillaume W Basse, Hossein Azari Soufiani, and Diane Lambert. Randomization and the pernicious effects of limited budgets on auction experiments. In *Artificial Intelligence and Statistics*, pages 1412–1420. PMLR, 2016.

Guillaume W Basse, Yi Ding, and Panos Toulis. Minimax designs for causal effects in temporal experiments with treatment habituation. *Biometrika*, 110(1):155–168, 2023.

Thomas Blake and Dominic Coey. Why marketplace experimentation is harder than it seems: The role of test-control interference. In *Proceedings of the fifteenth ACM conference on Economics and computation*, pages 567–582, 2014.

Iavor Bojinov, David Simchi-Levi, and Jinglong Zhao. Design and analysis of switchback experiments. *Management Science*, 69(7):3759–3777, 2023.

Ariel Boyarsky, Hongseok Namkoong, and Jean Pouget-Abadie. Modeling interference using experiment roll-out. *arXiv preprint arXiv:2305.10728*, 2023.

Ido Bright, Arthur Delarue, and Ilan Lobel. Reducing marketplace interference bias via shadow prices. *arXiv preprint arXiv:2205.02274*, 2022.

Ozan Candogan, Chen Chen, and Rad Niazadeh. Correlated cluster-based randomized experiments: Robust variance minimization. *Management Science*, 2023.

Allison JB Chaney, Brandon M Stewart, and Barbara E Engelhardt. How algorithmic confounding in recommendation systems increases homogeneity and decreases utility. In *Proceedings of the 12th ACM conference on recommender systems*, pages 224–232, 2018.

Shuchi Chawla, Jason Hartline, and Denis Nekipelov. A/b testing of auctions. In *Proceedings of the 2016 ACM Conference on Economics and Computation*, pages 19–20, 2016.

Thomas Cook, Alan Mishler, and Aaditya Ramdas. Semiparametric efficient inference in adaptive experiments. *arXiv preprint arXiv:2311.18274*, 2023.

Rick Durrett. *Probability: theory and examples*, volume 49. Cambridge university press, 2019.

Dean Eckles, Brian Karrer, and Johan Ugander. Design and analysis of experiments in networks: Reducing bias from interference. *Journal of Causal Inference*, 5(1), 2017.

Vivek Farias, Andrew Li, Tianyi Peng, and Andrew Zheng. Markovian interference in experiments. *Advances in Neural Information Processing Systems*, 35:535–549, 2022.

Vivek Farias, Hao Li, Tianyi Peng, Xinyuyang Ren, Huawei Zhang, and Andrew Zheng. Correcting for interference in experiments: A case study at douyin. In *Proceedings of the 17th ACM Conference on Recommender Systems*, pages 455–466, 2023.

Andrey Fradkin. Search frictions and the design of online marketplaces. In *The Third Conference on Auctions, Market Mechanisms and Their Applications*, 2015.

Peter W Glynn, Ramesh Johari, and Mohammad Rasouli. Adaptive experimental design with temporal interference: A maximum likelihood approach. *Advances in Neural Information Processing Systems*, 33:15054–15064, 2020.

Ali Goli, Anja Lambrecht, and Hema Yoganarasimhan. A bias correction approach for interference in ranking experiments. *Marketing Science*, 2023.

Huan Gui, Ya Xu, Anmol Bhasin, and Jiawei Han. Network a/b testing: From sampling to estimation. In *Proceedings of the 24th International Conference on World Wide Web*, pages 399–409, 2015.

Hongbo Guo, Ruben Naeff, Alex Nikulkov, and Zheqing Zhu. Evaluating online bandit exploration in large-scale recommender system. In *KDD-23 Workshop on Multi-Armed Bandits and Reinforcement Learning: Advancing Decision Making in E-Commerce and Beyond*, 2023.

Viet Ha-Thuc, Avishek Dutta, Ren Mao, Matthew Wood, and Yunli Liu. A counterfactual framework for seller-side a/b testing on marketplaces. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2288–2296, 2020.

Kevin Han, Shuangning Li, Jialiang Mao, and Han Wu. Detecting interference in online controlled experiments with increasing allocation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 661–672, 2023.

Christopher Harshaw, Fredrik Sävje, David Eisenstat, Vahab Mirrokni, and Jean Pouget-Abadie. Design and analysis of bipartite experiments under a linear exposure-response model. *Electronic Journal of Statistics*, 17(1):464–518, 2023.

David Holtz and Sinan Aral. Limiting bias from test-control interference in online marketplace experiments. *arXiv preprint arXiv:2004.12162*, 2020.

David Holtz, Ruben Lobel, Inessa Liskovich, and Sinan Aral. Reducing interference bias in online marketplace pricing experiments. *arXiv preprint arXiv:2004.12489*, 2020.

David Holtz, Jennifer Brennan, and Jean Pouget-Abadie. A study of" symbiosis bias" in a/b tests of recommendation algorithms. *arXiv preprint arXiv:2309.07107*, 2023.

Yuchen Hu and Stefan Wager. Switchback experiments under geometric mixing. *arXiv preprint arXiv:2209.00197*, 2022.

Michael G Hudgens and M Elizabeth Halloran. Toward causal inference with interference. *Journal of the American Statistical Association*, 103(482):832–842, 2008.

Guido W Imbens and Donald B Rubin. *Causal inference in statistics, social, and biomedical sciences*. Cambridge University Press, 2015.

Amir H Jadidinejad, Craig Macdonald, and Iadh Ounis. Using exploration to alleviate closed loop effects in recommender systems. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2025–2028, 2020.

Ramesh Johari, Hannah Li, Inessa Liskovich, and Gabriel Y Weintraub. Experimental design in two-sided platforms: An analysis of bias. *Management Science*, 2022.

Sami Khenissi. Modeling and debiasing feedback loops in collaborative filtering recommender systems. 2022.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Ron Kohavi, Diane Tang, and Ya Xu. *Trustworthy online controlled experiments: A practical guide to a/b testing*. Cambridge University Press, 2020.

Karl Krauth, Yixin Wang, and Michael I Jordan. Breaking feedback loops in recommender systems with causal inference. *arXiv preprint arXiv:2207.01616*, 2022.

Xu Kuang and Stefan Wager. Weak signal asymptotics for sequentially randomized experiments. *Management Science*, 2023.

Hannah Li, Geng Zhao, Ramesh Johari, and Gabriel Y Weintraub. Interference, bias, and variance in two-sided marketplace experimentation: Guidance for platforms. In *Proceedings of the ACM Web Conference 2022*, pages 182–192, 2022.

Shuangning Li and Stefan Wager. Random graph asymptotics for treatment effect estimation under network interference. *The Annals of Statistics*, 50(4):2334–2358, 2022.

Shuangning Li, Ramesh Johari, Stefan Wager, and Kuang Xu. Experimenting under stochastic congestion. *arXiv preprint arXiv:2302.12093*, 2023.

Luofeng Liao and Christian Kroer. Statistical inference and a/b testing for first-price pacing equilibria. *arXiv preprint arXiv:2301.02276*, 2023.

Luofeng Liao, Christian Kroer, Sergei Leonenkov, Okke Schrijvers, Liang Shi, Nicolas Stier-Moses, and Congshan Zhang. Interference among first-price pacing equilibria: A bias and variance analysis. *arXiv preprint arXiv:2402.07322*, 2024.

Min Liu, Jialiang Mao, and Kang Kang. Trustworthy and powerful online marketplace experimentation with budget-split design. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 3319–3329, 2021.

Masoud Mansoury, Himan Abdollahpouri, Mykola Pechenizkiy, Bamshad Mobasher, and Robin Burke. Feedback loop and bias amplification in recommender systems. In *Proceedings of the 29th ACM international conference on information & knowledge management*, pages 2145–2148, 2020.

Evan Munro, Stefan Wager, and Kuang Xu. Treatment effects in market equilibrium. *arXiv preprint arXiv:2109.11647*, 2021.

Paul Musgrave, Cuize Han, and Parth Gupta. Measuring service-level learning effects in search via query-randomized experiments. 2023.

Preetam Nandy, Divya Venugopalan, Chun Lo, and Shaunak Chatterjee. A/b testing for recommender systems in a two-sided marketplace. *Advances in Neural Information Processing Systems*, 34: 6466–6477, 2021.

Tu Ni, Iavor Bojinov, and Jinglong Zhao. Design of panel experiments with spatial and temporal interference. *Available at SSRN 4466598*, 2023.

Weishen Pan, Sen Cui, Hongyi Wen, Kun Chen, Changshui Zhang, and Fei Wang. Correcting the user feedback-loop bias for recommendation systems. *arXiv preprint arXiv:2109.06037*, 2021.

Judea Pearl. Causality: Models, reasoning and inference. *Cambridge, UK: CambridgeUniversityPress*, 19(2):3, 2000.

Jean Pouget-Abadie, Kevin Aydin, Warren Schudy, Kay Brodersen, and Vahab Mirrokni. Variance reduction in bipartite experiments through correlation clustering. *Advances in Neural Information Processing Systems*, 32, 2019.

Chao Qin and Daniel Russo. Adaptivity and confounding in multi-armed bandit experiments. *arXiv preprint arXiv:2202.09036*, 2022.

Paul R Rosenbaum and Donald B Rubin. The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1):41–55, 1983.

Nian Si, San Gultekin, Jose Blanchet, and Aaron Flores. Optimal bidding and experimentation for multi-layer auctions in online advertising. *Available at SSRN*, 2022. URL `https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4358914`.

David Simchi-Levi and Chonghuan Wang. Multi-armed bandit experimental design: Online decision-making and adaptive inference. *Available at SSRN 4224969*, 2022.

Johan Ugander and Hao Yin. Randomized graph cluster randomization. *Journal of Causal Inference*, 11(1):20220014, 2023.

Johan Ugander, Brian Karrer, Lars Backstrom, and Jon Kleinberg. Graph cluster randomization: Network exposure to multiple universes. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 329–337, 2013.

Stefan Wager and Kuang Xu. Experimenting in equilibrium. *Management Science*, 67(11):6694–6715, 2021.

Yan Wang and Shan Ba. Producer-side experiments based on counterfactual interleaving designs for online recommender systems. *arXiv preprint arXiv:2310.16294*, 2023.

Ruoxuan Xiong, Susan Athey, Mohsen Bayati, and Guido Imbens. Optimal experimental design for staggered rollouts. *arXiv preprint arXiv:1911.03764*, 2019.

Ruoxuan Xiong, Alex Chin, Mohsen Bayati, and Sean Taylor. Data-driven switchback design. *preprint*, 2023a. URL `https://www.ruoxuanxiong.com/data-driven-switchback-design.pdf`.

Ruoxuan Xiong, Alex Chin, and Sean Taylor. Bias-variance tradeoffs for designing simultaneous temporal experiments. In *The KDD'23 Workshop on Causal Discovery, Prediction and Decision*, pages 115–131. PMLR, 2023b.

Mengyue Yang, Jun Wang, and Jean-Francois Ton. Rectifying unfairness in recommendation feedback loop. In *Proceedings of the 46th international ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 28–37, 2023.

Zikun Ye, Dennis J Zhang, Heng Zhang, Renyu Zhang, Xin Chen, and Zhiwei Xu. Cold start to improve market thickness on online advertising platforms: Data-driven algorithms and field experiments. *Management Science*, 69(7):3838–3860, 2023.

Christina Lee Yu, Edoardo M Airoldi, Christian Borgs, and Jennifer T Chayes. Estimating the total treatment effect in randomized experiments with unknown network structure. *Proceedings of the National Academy of Sciences*, 119(44):e2208975119, 2022.

A Proofs

Proof of Lemma 1. By the causal graph (Figure 3), we have $Y_E \perp Z | X_E$, which yields

$$\mathbb{E}\left[Z|X_E\right] = \mathbb{E}\left[Z|X_E, Y_E\right] = \mathbb{E}\left[Z|D_E\right].$$

For any measurable set $A \subset \mathbb{R}^d \times \mathbb{R}^m$, we have

$$W_T \mathcal{D}_E(A) = \frac{1}{p}\mathbb{E}\left[\mathbb{E}\left[Z|D_E\right] I\left\{D_E \in A\right\}\right]$$

Due to the property of the conditional expectation [Durrett, 2019, Theorem 5.1.7], we have

$$\mathbb{E}\left[\mathbb{E}\left[Z|D_E\right] I\left\{D_E \in A\right\}\right] = \mathbb{E}\left[\mathbb{E}\left[I\left\{D_E \in A\right\} Z|D_E\right]\right] = \mathbb{E}\left[I\left\{D_E \in A\right\} Z\right]$$

Recall that $D_E = D_T Z + D_C\left(1 - Z\right)$, we have

$$\mathbb{E}\left[Z I\left\{D_E \in A\right\}\right] = \mathbb{E}\left[I\{Z=1\} I\left\{D_T \in A\right\}\right].$$

Because of the independence of Z and D_T, we have

$$\frac{1}{p}\mathbb{E}\left[I\{Z=1\} I\left\{D_E \in A\right\}\right] = \mathbb{E}\left[I\left\{D_T \in A\right\}\right].$$

. □

Proof of Theorem 1. We will focus on the treatment problem (1), as the control problem follows an identical approach. Recall that

$$W\mathcal{D}_E(A) = \mathbb{E}\left[W I\left\{D_E \in A\right\}\right] = \mathbb{E}\left[\mathbb{E}\left[W|D_E\right] I\left\{D_E \in A\right\}\right]$$

for any measurable set A in $\mathcal{X} \times \mathcal{Y}$. On the other hand, we have

$$\begin{aligned} p\mathbb{E}\left[I\left\{D_T \in A\right\}\right] &= \mathbb{E}\left[Z\right]\mathbb{E}\left[I\left\{D_T \in A\right\}\right] = \mathbb{E}\left[Z I\left\{D_T \in A\right\}\right] \\ &= \mathbb{E}\left[Z I\left\{D_E \in A\right\}\right]. \end{aligned}$$

Therefore, the constraint means that

$$\begin{aligned} \mathbb{E}\left[\mathbb{E}\left[W|D_E\right] I\left\{D_E \in A\right\}\right] &= \mathbb{E}\left[I\left\{D_T \in A\right\}\right] \\ &= \mathbb{E}\left[\left(Z/p\right) I\left\{D_E \in A\right\}\right], \end{aligned}$$

for any measurable set A in $\mathcal{X} \times \mathcal{Y}$. By the definition of the conditional expectation [Durrett, 2019, Section 5.1], we have

$$\mathbb{E}\left[W|D_E\right] = \frac{1}{p}\mathbb{E}\left[Z|D_E\right].$$

By Theorem 5.1.3 in Durrett [2019], we have

$$\mathbb{E}\left[W^2\right] = \mathbb{E}\left[\mathbb{E}\left[W^2|D_E\right]\right] \geq \mathbb{E}\left[\left(\mathbb{E}\left[W|D_E\right]\right)^2\right] = \mathbb{E}\left[\left(\mathbb{E}\left[Z|D_E\right]/p\right)^2\right].$$

We conclude the proof by noting that

$$\mathbb{E}\left[Z|D_E\right] = \mathbb{E}\left[Z|X_E\right],$$

as also shown in the proof of Lemma 1. □

B Additional Figures and Tables for the Experiments in Section 5

Table 2: Bias, standard deviation, and standard error estimated from the experiment for the metrics in the case that $\alpha_C = 10$, $\alpha_T = 9$, and $p = 1/2$

	Proportion of short videos			Stay durations			Finishing rates		
	Bias	STD	SE	Bias	STD	SE	Bias	STD	SE
Weighted	0.002	0.004	0.001	-0.005	0.012	0.008	0.000	0.001	0.001
Data splitting	-0.003	0.015	0.001	0.008	0.042	0.008	-0.001	0.004	0.001
Data pooling	0.018	0.002	0.001	-0.059	0.009	0.008	0.006	0.001	0.001
Snapshot	0.011	0.001	0.001	-0.047	0.008	0.009	0.004	0.001	0.001

We provide additional details about the experiments in Section 5. Table 2 shows the bias and standard errors of treatment effect estimators for the proportion of short videos, SD, and FR. In Figure 5, we provide results for treatment effects, global treatment, and global control regimes in the first, second, and third rows, respectively. Additionally, we report results for the proportion of short videos, SD, and FR in the first, second, and third columns, respectively.

In Table 3, we have calculated the experimentation costs. For treatment users, we computed the average treatment values based on the treatment linear fusion formula, i.e.,

$$\frac{1}{N_T}\sum_{i=1}^{N_T} \alpha_T \text{Finish}_i + \text{StayDuration}_i,$$

where N_T represents the number of users in the treatment group and Finish_i and StayDuration_i indicate whether a user finished watching a video and their duration of stay, respectively. While for control users, we averaged the control values based on the control linear fusion formula:

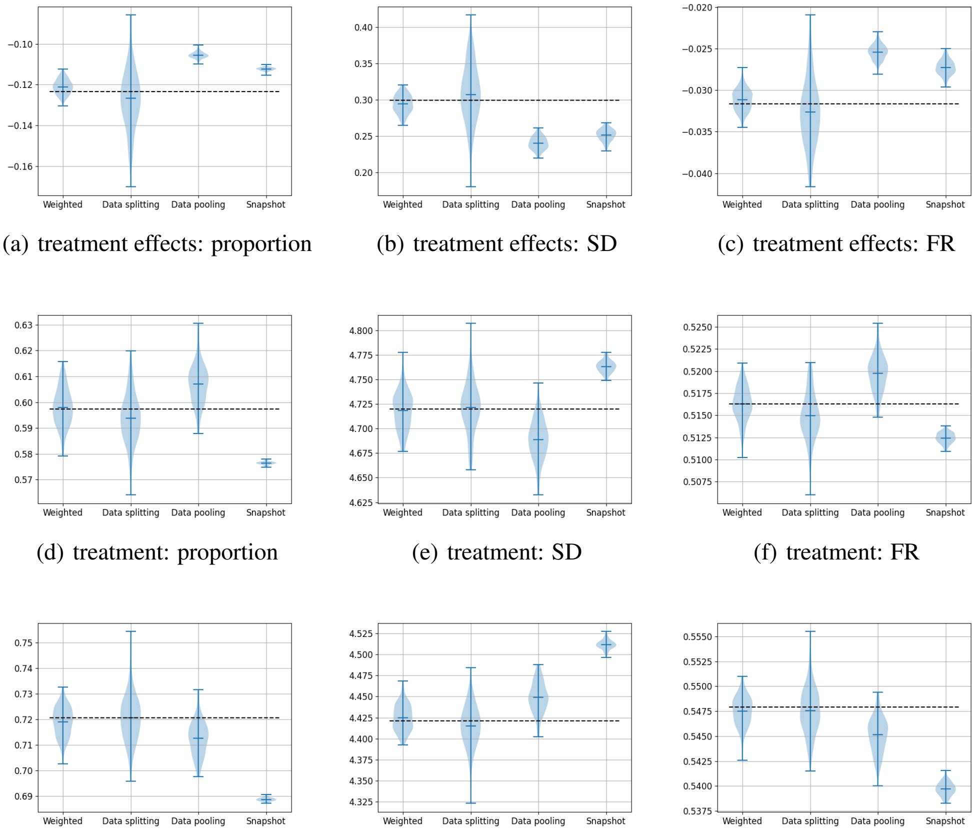
$$\frac{1}{N_C}\sum_{i=1}^{N_C} \alpha_C \text{Finish}_i + \text{StayDuration}_i.$$

It's apparent that our approach is only slightly worse than the global treatment/control regime, and the data splitting method incurs the lowest costs, indicating that it results in higher experimental expenses.

C Additional Numerical Experiments

C.1 A/B Tests

In this subsection, we present additional A/B testing simulations. Firstly, we consider $\alpha_C = 10$, $\alpha_T = 8$, and $p = 0.2$, and the results are visualized in Figure 6, while detailed bias, variance, and cost findings can be found in Tables Tables 4 and 5.



(a) treatment effects: proportion

(b) treatment effects: SD

(c) treatment effects: FR

(d) treatment: proportion

(e) treatment: SD

(f) treatment: FR

(g) control: proportion

(h) control: SD

(i) control: FR

Figure 5: A/B testing results for $\alpha_C = 10$, $\alpha_T = 9$, and $p = 1/2$

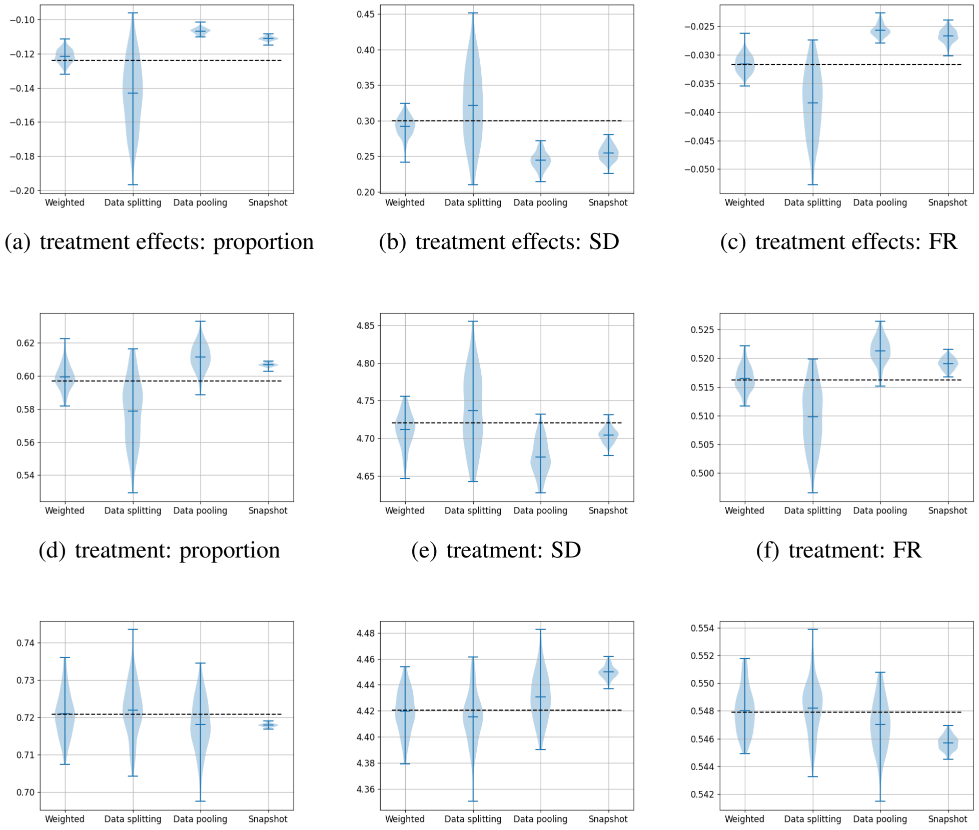
Table 3: Experimentation values in the case that $\alpha_C = 10$, $\alpha_T = 9$, and $p = 1/2$

	Treatment values	Control values
Global	9.8827 ± 0.0006	9.3523 ± 0.0006
Weighted	9.8816 ± 0.0009	9.3521 ± 0.0008
Data splitting	9.8710 ± 0.0008	9.3431 ± 0.0008
Data pooling	9.8861 ± 0.0008	9.3551 ± 0.0009
Snapshot	9.8876 ± 0.0009	9.3692 ± 0.0008

Table 4: Bias, standard deviation, and standard error estimated from the experiment for the metrics in the case that $\alpha_C = 10$, $\alpha_T = 9$, and $p = 0.2$

	Proportion of short videos			Stay durations			Finishing rates		
	Bias	STD	SE	Bias	STD	SE	Bias	STD	SE
Weighted	0.002	0.004	0.001	-0.008	0.014	0.011	0.000	0.002	0.001
Data splitting	-0.019	0.020	0.001	0.021	0.052	0.011	-0.007	0.006	0.001
Data pooling	0.017	0.002	0.001	-0.056	0.012	0.011	0.006	0.001	0.001
Snapshot	0.013	0.001	0.001	-0.046	0.011	0.011	0.005	0.001	0.001

Secondly, we consider $\alpha_C = 10$, $\alpha_T = 8$, and $p = 1/2$, and the results are visualized in Figure 7, while detailed bias, variance, and cost findings can be found in Tables 6 and 7.



(a) treatment effects: proportion (b) treatment effects: SD (c) treatment effects: FR

(d) treatment: proportion (e) treatment: SD (f) treatment: FR

(g) control: proportion (h) control: SD (i) control: FR

Figure 6: A/B testing results for $\alpha_C = 10$, $\alpha_T = 9$, and $p = 0.2$

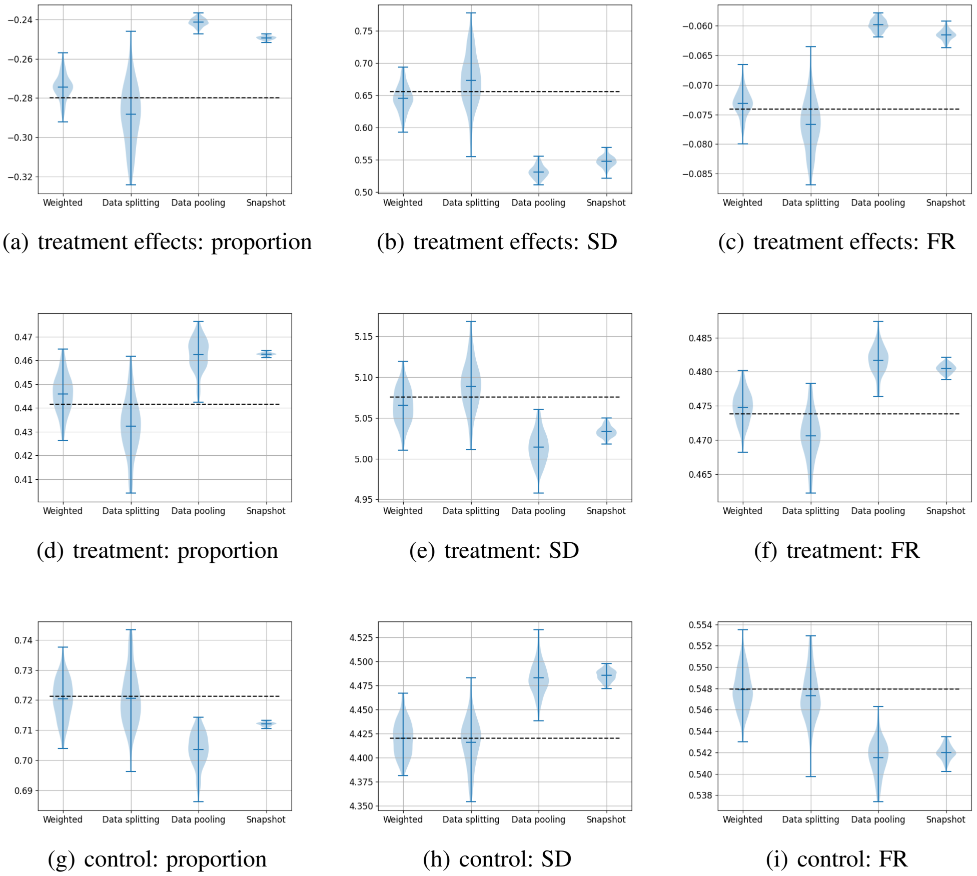
Table 5: Experimentation values in the case that $\alpha_C = 10$, $\alpha_T = 9$, and $p = 0.2$

	Treatment values	Control values
Global	9.8823 ± 0.0006	9.3515 ± 0.0005
Weighted	9.8757 ± 0.0013	9.3517 ± 0.0006
Data splitting	9.8347 ± 0.0015	9.3492 ± 0.0007
Data pooling	9.8877 ± 0.0013	9.3538 ± 0.0006
Snapshot	9.8949 ± 0.0015	9.3611 ± 0.0006

Table 6: Bias, standard deviation, and standard error estimated from the experiment for the metrics in the case that $\alpha_C = 10$, $\alpha_T = 8$, and $p = 1/2$

	Proportion of short videos			Stay durations			Finishing rates		
	Bias	STD	SE	Bias	STD	SE	Bias	STD	SE
Weighted	0.005	0.007	0.001	-0.010	0.021	0.009	0.001	0.002	0.001
Data splitting	-0.008	0.015	0.001	0.018	0.040	0.009	-0.003	0.004	0.001
Data pooling	0.039	0.002	0.001	-0.124	0.009	0.009	0.014	0.001	0.001
Snapshot	0.031	0.001	0.001	-0.107	0.009	0.009	0.013	0.001	0.001

Additionally, we explore the scenario with $\alpha_C = 10$, $\alpha_T = 9$, and $p = 0.3$, the results of which are illustrated in Figure 8, and detailed bias, variance and cost results can be found in Tables 8 and 9.



(a) treatment effects: proportion

(b) treatment effects: SD

(c) treatment effects: FR

(d) treatment: proportion

(e) treatment: SD

(f) treatment: FR

(g) control: proportion

(h) control: SD

(i) control: FR

Figure 7: A/B testing results for $\alpha_C = 10$, $\alpha_T = 8$, and $p = 1/2$

Table 7: Experimentation values in the case that $\alpha_C = 10$, $\alpha_T = 8$, and $p = 0.5$

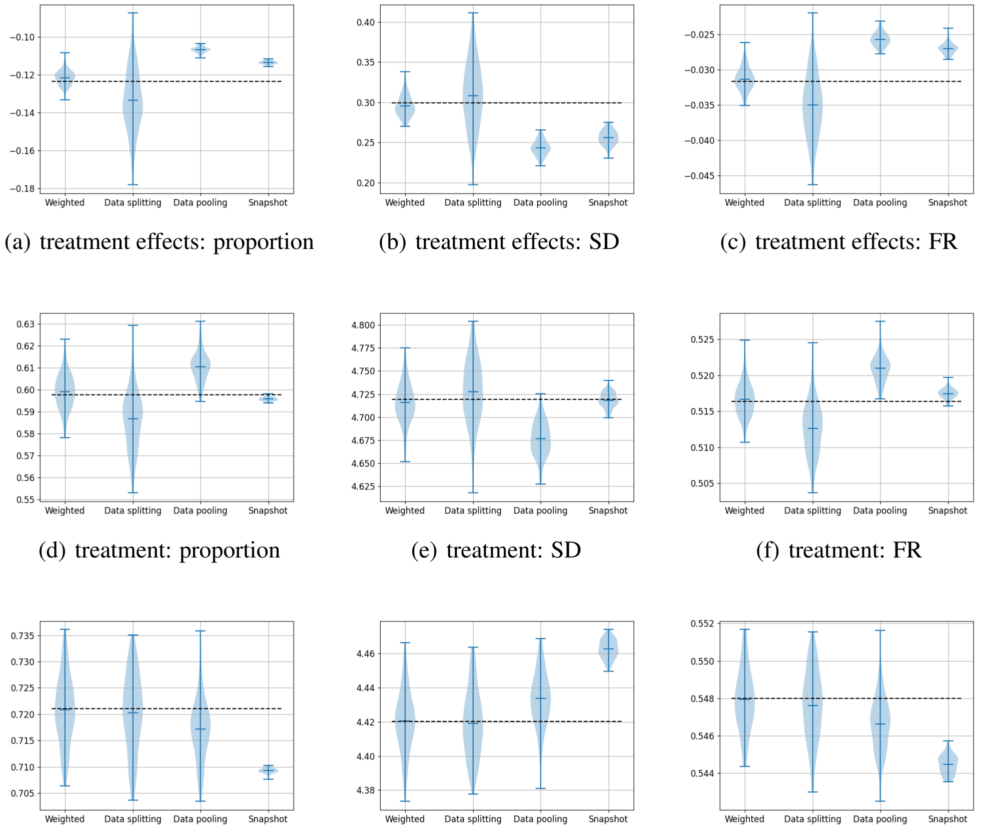
	Treatment values	Control values
Global	9.8144 ± 0.0007	8.8040 ± 0.0007
Weighted	9.8132 ± 0.0008	8.8032 ± 0.0008
Data splitting	9.7953 ± 0.0009	8.7946 ± 0.0009
Data pooling	9.8312 ± 0.0007	8.8151 ± 0.0007
Snapshot	9.8383 ± 0.0009	8.8215 ± 0.0008

Table 8: Bias, standard deviation, and standard error estimated from the experiment for the metrics in the case that $\alpha_C = 10$, $\alpha_T = 9$, and $p = 0.3$

	Proportion of short videos			Stay durations			Finishing rates		
	Bias	STD	SE	Bias	STD	SE	Bias	STD	SE
Weighted	0.002	0.004	0.001	-0.004	0.013	0.009	0.000	0.002	0.001
Data splitting	-0.010	0.016	0.001	0.009	0.042	0.009	-0.003	0.005	0.001
Data pooling	0.017	0.002	0.001	-0.056	0.009	0.009	0.006	0.001	0.001
Snapshot	0.010	0.001	0.001	-0.043	0.009	0.009	0.005	0.001	0.001

C.2 A/A Tests

In this section, we have conducted simulations for A/A tests, specifically choosing parameters such as $\alpha_C = \alpha_T = 10$ with a treatment assignment probability of $p = 1/2$. Since the treatment and



(a) treatment effects: proportion

(b) treatment effects: SD

(c) treatment effects: FR

(d) treatment: proportion

(e) treatment: SD

(f) treatment: FR

(g) control: proportion

(h) control: SD

(i) control: FR

Figure 8: A/B testing results for $\alpha_C = 10$, $\alpha_T = 9$, and $p = 0.3$

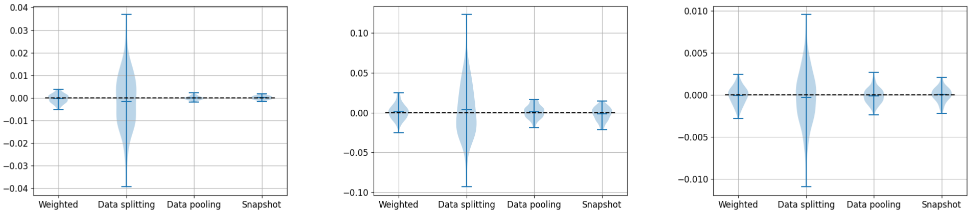
Table 9: Experimentation values in the case that $\alpha_C = 10$, $\alpha_T = 9$, and $p = 0.3$

	Treatment values	Control values
Global	9.8824 ± 0.0006	9.3521 ± 0.0007
Weighted	9.8820 ± 0.0011	9.3521 ± 0.0007
Data splitting	9.8537 ± 0.0010	9.3476 ± 0.0007
Data pooling	9.8860 ± 0.0010	9.3531 ± 0.0006
Snapshot	9.8927 ± 0.0011	9.3628 ± 0.0007

control groups share an identical parameter, the global treatment effects should ideally be zero. In Figure 9, we present visualizations of treatment effect estimations for four methods. Notably, the weighted training, data pooling, and snapshot methods exhibit similar performance. Table 10 offers details on the average estimations and type I errors obtained from various methods, gathered from 100 independent runs of the A/A tests, with a confidence level set at 0.95.

It's noteworthy that our approach exhibits a slightly larger type I error than the target of 0.05 for the metrics stay durations (SD) and finishing rates (FR), and it demonstrates a worse type I error for the metric proportion of short videos. We attribute this behavior to the sensitivity of the proportion of short videos metric to the starting period of the experiment, which may be more feedback-loop dependent.

On the contrary, the data splitting method yields much higher Type I errors, suggesting that new inference methods should be developed to address this issue.



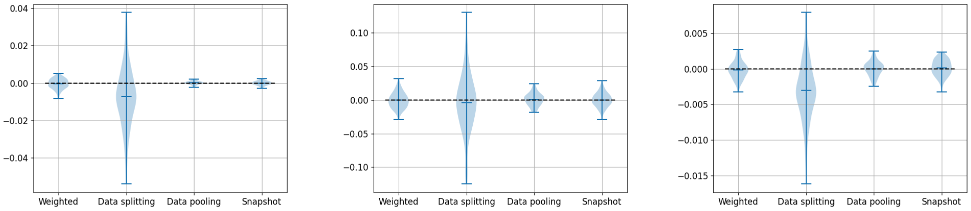
(a) treatment effects: proportion (b) treatment effects: SD (c) treatment effects: FR

Figure 9: A/A testing results for $\alpha_C = \alpha_T = 10$ and $p = 1/2$

Table 10: The average estimations and type I error for the A/A test with $\alpha_C = \alpha_T = 10$ and $p = 1/2$

	Proportion of short videos		Stay durations		Finishing rates	
	Estimation	Type I error	Estimation	Type I error	Estimation	Type I error
Weighted	-0.0003	0.45	0.0008	0.09	-0.0001	0.11
Data splitting	-0.0017	0.94	0.0039	0.65	-0.0003	0.60
Data pooling	-0.0001	0.04	0.0011	0.07	-0.0001	0.07
Snapshot	0.0001	0.06	-0.0015	0.06	0.0000	0.06

We further present additional A/A testing results with $\alpha_C = \alpha_T = 10$ and $p = 0.2$. The estimations of treatment effects are visualized in Figure 10, and Table 11 offers comprehensive details regarding the estimators and type I errors.



(a) treatment effects: proportion (b) treatment effects: SD (c) treatment effects: FR

Figure 10: A/A testing results for $\alpha_C = \alpha_T = 10$ and $p = 0.2$

Table 11: The average estimations and type I error for the A/A test with $\alpha_C = \alpha_T = 10$ and $p = 0.2$

	Proportion of short videos		Stay durations		Finishing rates	
	Estimation	Type I error	Estimation	Type I error	Estimation	Type I error
Weighted	-0.0004	0.47	-0.0005	0.07	-0.0002	0.08
Data splitting	-0.0073	0.91	-0.0036	0.56	-0.0031	0.75
Data pooling	0.0001	0.04	0.0008	0.03	0.0000	0.05
Snapshot	-0.0001	0.07	-0.0001	0.06	0.0001	0.04

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: he context and contributions of this paper are clearly and accurately stated in the abstract and the introduction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We acknowledge our current method requires two separate model training. We mentioned in Concluding remarks that improving computational efficiency is one future direction.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: All theorems and lemmas are proved in appendix.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We provide a comprehensive simulation setup.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We provide our code in the supplementary material.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: all the training and test details are provided.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We report the variance of the estimator.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [NA]

Justification: Our method can be applied using any computer resources.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: We conform to the NeurIPS Code of Ethics. Anonymity is preserved in this submission.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [NA]

Justification: This work proposes a new A/B tests. There is no direct social impact associated with this work.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: We do not use any high-risk information.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer:[NA]

    Justification: The paper does not use existing assets.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not use existing assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing and research with human subjects

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This paper does not require IRB approvals.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.