Data Mining and Knowledge Discovery
https://doi.org/10.1007/s10618-021-00797-x

Continuous treatment effect estimation via generative adversarial de-confounding

Kun Kuang[1] · Yunzhe Li[1] · Bo Li[2] · Peng Cui[2] · Hongxia Yang[3] · Jianrong Tao[4] · Fei Wu[1]

Received: 27 November 2020 / Accepted: 4 September 2021

Abstract
One fundamental problem in causal inference is the treatment effect estimation in observational studies, and its key challenge is to handle the confounding bias induced by the associations between covariates and treatment variable. In this paper, we study the problem of effect estimation on continuous treatment from observational data, going beyond previous work on binary treatments. Previous work on binary treatment focuses on de-confounding by balancing the distribution of covariates between the treated and control groups with either propensity score or confounder balancing techniques. In the continuous setting, those methods would fail as we can hardly evaluate the distribution of covariates under each treatment status. To tackle the case of continuous treatments, we propose a novel Generative Adversarial De-confounding (GAD) algorithm to eliminate the associations between covariates and treatment variable with two main steps: (1) generating an "calibration" distribution without associations between covariates and treatment by randomly perturbation on treatment variable; (2) learning sample weights that transfer the distribution of observed data to the "calibration" distribution for de-confounding with a Generative Adversarial Network. We show, both theoretically and with empirical experiments, that our GAD algorithm can remove the associations between covariates and treatment, hence, precisely estimating the causal effect of continuous treatment. Extensive experiments on both synthetic and real-world datasets demonstrate that our algorithm outperforms the state-of-the-art methods for effect estimation of continuous treatment with observational data.

Keywords Treatment effect · Continuous treatment · Generative adversarial de-confounding · Causal inference

Responsible editor: Sriraam Natarajan.

✉ Kun Kuang
kunkuang@zju.edu.cn

Extended author information available on the last page of the article

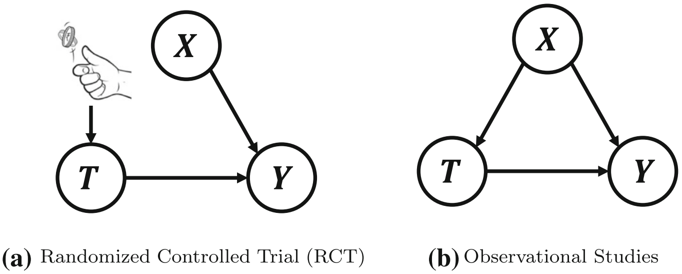
Published online: 22 September 2021

1 Introduction

Causal inference (Holland 1986), which refers to the process of drawing a conclusion about a causal connection based on the conditions of the occurrence of an effect, is a powerful statistical modeling tool for explanatory analysis. Treatment effect estimation is one fundamental problem in causal inference and gains an essential role for explainable decision making with answering the counterfactual questions (Rubin 1974; Pearl 2009). For example, how many doses of a medication will cause better outcomes for patients. Pearl (2009) demonstrate that the gold standard approach for treatment effect estimation is to run a Randomized Controlled Trial (RCT), such as A/B testing, where the treatments are randomly assigned to units[1] and independent of the covariates as shown in Fig. 1a. A related class of techniques called uplift modeling, which also strives to get the best possible estimator with properly randomized data (Soltys et al. 2014; Rudas and Jaroszewicz 2018; Olaya et al. 2020). In many real applications, however, fully randomized experiments are always expensive, unethical, or even infeasible (Kohavi and Longbotham 2011). In this paper, hence, we focus on approximately estimate the treatment effect from off-line data collected from observational studies. In such datasets, the assignment of treatment depends on the covariates as we shown in Fig. 1b, leading to confounding bias between treatment and covariates, i.e., $P(T|\mathbf{X}) \neq P(T)$. Therefore, confounding bias removing is the key challenge for treatment effect estimation in observational studies.

In literature, many methods have been proposed for effect estimation with binary treatment (treated or control), including matching methods (Kallus 2019; Liu et al. 2019), propensity score based methods (Rosenbaum and Rubin 1983; Bang and Robins 2005; Austin 2011), and confounder balancing techniques (Hainmueller 2012; Kuang et al. 2017; Athey et al. 2018). The motivation of these methods is to remove the association between treatment and covariates for de-confounding. Matching methods (Liu et al. 2019) proposed to match units with almost the same covariates but different treatment. Inverse of propensity weighting (IPW) (Austin 2011) attempted to re-weight samples for removing confounding bias between treatment and covariates. Confounder balancing methods (Kuang et al. 2017, 2019) proposed to balance the distribution of covariates between treated and control groups. These methods achieved good performance in real applications for treatment effect estimation, and can be used in related research field (e.g. discrimination measuring in Žliobaitė (2017)). However, all of them focus on the binary treatment and cannot be applied for estimating the causal effect of continuous treatment. Some researchers also consider the heterogeneous causal effect problem (Wager and Athey 2015; Athey and Imbens 2016; Künzel et al. 2019), which the same treatment may affect individuals differently. Recently, Zou et al. (2020) proposed a method to learn continuous treatment policy by decomposing treatment effect functions into different factors under heterogeneous causal effect setting, which demonstrates feasibility relating continuous to effect definition under binary treatment setting.

[1] Units represent the objects of treatment. For example, in medical experiments, the units refer to the patients who take a particular medication.



(a) Randomized Controlled Trial (RCT) **(b)** Observational Studies

Fig. 1 Casual structure for RCT and observational studies, where $\mathbf{X}$ denotes the observed covariates, T refers to the treatment variable, and Y is the outcome. In RCT, the treatment is independent with covariates, while in observational studies, the treatment is affected by the covariates

The classical methods for estimating continuous treatment effect are based on regression models, including Y-model (Imbens 2004; Hill 2011) to regress outcome Y on the covariates and treatment, T-model (Hirano and Imbens 2004; Imai and Van Dyk 2004; Galvao and Wang 2015; Zhu et al. 2015; Galagate 2016) to regress the treatment T on the covariates, and doubly robust methods (Robins and Rotnitzky 2001) by combining both Y-model and T-model. The performance of these methods entirely relies on the correct specification of their models. Recently, a non-parametric covariate balancing generalized propensity score (Fong et al. 2018) was proposed to minimize the association between the covariates and treatment for de-confounding, and achieved great performance in real applications. However, it is limited by its liner assumption on T-model. Galagate (2016) extended IPW for continuous treatment with considering second moments of covariates, but it assumes linear correlation between Y and T. Overall, if one has NO prior knowledge on the grounded models, existing methods for continuous treatment cannot fully remove the confounding bias in observational studies, leading to imprecise estimation of continuous treatment effect.

To better remove the confounding bias in observational studies, we propose a non-parametric data-driven method, named Generative Adversarial De-confounding (GAD) algorithm by sample re-weighting techniques. Specifically, there are two main components in our GAD algorithm, including "calibration" distribution generation and approximation. Firstly, we generate an "calibration" distribution by randomly shuffle the covariates across units, such that the covariates would become independent with the treatment, which fully removes the confounding bias. Then, we propose a sample weight learning schema on the observed data for approximating the the "calibration" distribution with a Generative Adversarial Network (GAN), achieving de-confounding between continuous treatment and covariates. Using both empirical experiments and theoretical analysis, we demonstrate that our algorithm can remove the confounding bias in observed data and precisely estimate the casual effect of continuous treatment. We validate our GAD algorithm with extensive experiments on both synthetic and real datasets. The experimental results clearly show that our algorithm outperforms the state-of-the- art methods on continuous treatment effect estimation in observational studies.

The main contributions of this paper are summarized as follows:

- We investigate the problem of causal effect estimation with continuous treatment from observational data, going beyond previous work on binary treatments.
- We propose a novel Generative Adversarial De-confounding (GAD) algorithm to learn a sample weight for removing the associations between treatment and covariates, and estimating the causal effect of continuous treatment.
- We give theoretical analysis on our proposed algorithm and prove that our algorithm can remove the confounding between treatment and covariates, hence, precisely estimating the effect of continuous treatment.
- Extensive experiments on both synthetic and real world datasets demonstrate the superior performance of our proposed algorithms on the problem of continuous treatment effect estimation with observational data.

The rest of this paper is organized as follows. Section 2 reviews the related work. Section 3 gives the notations and formulates our problem. The details of our proposed algorithm for continuous treatment effect estimation are introduced in Sect. 4. Experimental results and analyses are reported in Sect. 5. Finally, Sect. 6 concludes the paper.

2 Related work

2.1 Treatment effect estimation in causal inference

Previous work on treatment effect estimation in observational studies can be categorized by the type of treatment variable as causal effect estimation on binary treatment and continuous treatment.

On binary treatment The classical method for causal effect estimation on binary treatment is propensity score based methods (Rosenbaum and Rubin 1983; Bang and Robins 2005; Chan et al. 2010; Austin 2011; Kuang et al. 2020a). The propensity score was first proposed by Rosenbaum and Rubin (1983), where it was estimated via a logistic regression. Then many other machine learning algorithms (e.g., lasso by Chernozhukov et al. 2016, boosting regression by McCaffrey et al. 2004, bagged CART and neural network by Westreich et al. 2010) are employed for estimating propensity score. Various methods have been proposed based on propensity score, such as propensity score matching, inverse propensity weighting, double robust estimators (Bang and Robins 2005; Chan et al. 2010; Austin 2011). However, these estimators require correct model specification on treatment assignment or precise estimation of the propensity score, which may not be the case in many applications. Moreover, these methods focus on the causal effect estimation on binary treatment.

Bypassing propensity score estimation, recently, researchers proposed directly confounder balancing via sample weight learning (Egel et al. 2008; Tan 2010; Imai and Ratkovic 2014; Zubizarreta 2015; Chan et al. 2016; Kuang et al. 2017; Athey et al. 2018; Kuang et al. 2019). Imai and Ratkovic (2014) introduced covariate balancing propensity score, which models treatment assignment while optimizing covariates balancing. Based on covariate balancing propensity score, several improvements were also proposed either by introducing doubly robust property (Fan et al. 2016), or by introducing a more general framework with tailored loss function (Zhao 2016). Besides covariate balancing propensity score based methods, some researchers find other ways to perform confounder balancing. Zubizarreta (2015) proposed stable balancing weights, which finds the weights of minimum variance that balance covariates between control and treated group up to levels prespecified. Chan et al. (2016) proposed empirical balancing calibration weighting, which finds the weights that minimize the aggregate distance between final weights and uniform ones while satisfying moment conditions. Athey et al. (2018) proposed approximate residual balancing algorithm, which combines outcome modeling using the LASSO with balancing weights constructed to approximately balance covariates between treatment and control groups. Kuang et al. (2017) proposed a differentiated variable balancing algorithm by jointly optimizing sample weights and variable weights. These methods achieved good performance in many real applications for treatment effect estimation, but all of these methods still focus on the problem of binary treatment and cannot be directly applied for continuous treatment.

On continuous treatment In practice, the most common approach for estimating continuous treatment effect is regression model based, including Y-model (Imbens 2004; Hill 2011) and T-model (Hirano and Imbens 2004; Imai and Van Dyk 2004; Galvao and Wang 2015; Zhu et al. 2015). Y-model method refers to the regression modeling of how the outcome Y relates to covariates and treatment variable. T-model methods mainly adapted propensity score based approaches to model how the treatment T relates to the covariates, namely modeling treatment assignment mechanism. However, the performance of these methods relies entirely on the correct specification of either the outcome model or the treatment model. By combining Y-model and T-model, many doubly robust estimators (Robins and Rotnitzky 2001) are proposed and achieved consistent estimation of effects of continuous treatment as long as one of two models is correctly specified and modeled well enough.

Recently, many non-parametric methods (Neugebauer and van der Laan 2007; Kennedy et al. 2017; Fong et al. 2018; Kallus and Santacatterina 2019) have been proposed to reduce the model dependency for continuous treatment effect estimation. Neugebauer and van der Laan (2007) extended traditional parametric marginal structural model to an nonparametric one and does not require correct specification of a parametric model but instead relies on a working model for precise prediction. Kennedy et al. (2017) developed a kernel smoothing based non-parametric method for doubly robust estimation of continuous treatment effect, allowing for misapplication of either the treatment model or outcome model. Fong et al. (2018) proposed a non-parametric covariate balancing generalized propensity score to minimize the association between the covariates and treatment, however, it only focus on the linear association and would fail if the true T-model is non-linear. Kallus and Santacatterina (2019) proposed a convex optimization-based method which finds weights that minimize the worst-case penalized functional covariance between the continuous treatment and the confounders, with relatively higher computational cost.

Table 1 Symbols and definitions

Symbol	Definition
n	Sample size
p	Dimension of observed variables
$T \in \mathbb{R}^{n\times 1}$	Treatment
$T' \in \mathbb{R}^{n\times 1}$	Treatment after randomly shuffle
$Y \in \mathbb{R}^{n\times 1}$	Outcome
$\mathbf{X} \in \mathbb{R}^{n\times p}$	Observed variables
$\mathbf{w} \in \mathbb{R}^{n\times 1}$	Sample weight

2.2 Causal inference and explainable AI

Owing to the big data and computing power, many machine learning algorithms, especially deep learning methods, have been proposed and shown high accuracy in many real applications (Rong et al. 2020), but lacking of explainability. Recently, many researches focus on how to enhance the explainability of AI algorithm, including interpreting the knowledge of machine leaning (Li et al. 2020b) and designing interpretable models (Rudin 2019; Li et al. 2020a). The recent research on adversarial learning (Tian et al. 2021; Ren et al. 2020), artificial general intelligent (Lu and Wang 2020), distributional robustness optimization (Duchi and Namkoong 2018) et al. also try to improve the explainability of AI from different aspects. Causal inference (Kuang et al. 2020b) (including treatment effect estimation and causal discovery) is also one of the ways to explainable AI, where treatment effect estimation can help to identify the causations from spurious correlation in observational data, and causal discovery can be applied for identify causal features of outcome variable or identify the causal relationships among variables for a explainable prediction. By marrying causal inference and machine learning, many causal learning methods have been proposed, including stable learning (Kuang et al. 2018), causal transfer learning (Rojas-Carulla et al. 2018), causal representation (Schölkopf et al. 2021), to enhance the explainability of AI.

3 Problem and assumptions

In this paper, we focus on continuous treatment effect estimation based on potential outcome framework (Imbens and Rubin 2015) as shown in Fig. 1b. With the framework, we define a treatment as a random variable T and a potential outcome as $Y(t)$ which corresponds to a specific treatment $T = t$. The continuous treatment of interest can take values in $t \in \mathcal{T}$, where $\mathcal{T}$ is an interval $[t_0, t_1]$. Then, for each unit indexed by $i = 1, 2, \ldots, n$, we observe a treatment T_i, an outcome Y_i^{obs} and a vector of observed variables $X_i \in \mathbb{R}^{p\times 1}$, where the observed outcome Y_i^{obs} of unit i is corresponding to its treatment and denotes as $Y_i^{obs} = Y(T_i)$. The numbers of units are equal to n and the dimension of all observed variables is p. Table 1 summarized the symbol and definition. In our paper, for any column vector $\mathbf{v} = (v_1, v_2, \ldots, v_m)^T$, let $\|\mathbf{v}\|_\infty = \max(|v_1|, \ldots, |v_m|)$, $\|\mathbf{v}\|_2^2 = \sum_{i=1}^m v_i^2$, and $\|\mathbf{v}\|_1 = \sum_{i=1}^m |v_i|$.

The important goal of causal inference in observational studies is to evaluate the casual effect of treatment T on outcome Y. In the setting with continuous treatment, the causal effect of treatment can be captured by the *Average Dose Response Function* (ADRF) and *Marginal Treatment Effect Function* (MTEF) (Kreif et al. 2015). The ADRF refers to the expectation of potential outcome $Y(t)$ on each treatment status t over all units, which could be further used to demonstrate average treatment effect caused by change of treatment level in continuous treatment setting. Formally, the ADRF on treatment t is defined as:

$$ADRF(t) = \mathbb{E}[Y_i(t)]. \tag{1}$$

The MTEF represents the effect of increasing the level of treatment on the expected potential outcome over all units, which demonstrates average treatment effect caused by change at each level of treatment. Formally, the MTEF is defined as:

$$MTEF = \frac{\mathbb{E}[Y_i(t)] - \mathbb{E}[Y_i(t - \triangle t)]}{\triangle t}, \tag{2}$$

where $Y_i(t)$ represents the potential outcome of units i with treatment status $T = t$ and $\mathbb{E}(\cdot)$ refers to the expectation function. $\triangle t$ denotes the increasing the level of treatment, for example, with $\triangle t = 1$, MTEF captures the incremental change in the potential outcome, for a unit change in the level of treatment.

The Eqs. (1) and (2) are infeasible because of the counterfactual problem (Chan et al. 2010). For each unit i with treatment status $T = t$, we can only observe one of the potential outcomes $Y_i(t)$, and the other potential outcomes $Y_i(t'), t' \in \mathcal{T} \setminus t$ are unobserved or counterfactual. One can address this counterfactual problem by approximate the unobserved potential outcome. The simplest approach is to directly estimate the ARDF $\mathbb{E}[Y_i(t)]$ on treatment level $T = t$ only over the units with that treatment. However, in observational studies, the treatment is not randomly assigned to units as we shown in Fig. 1b, which leads to the confounding bias between treatment and covariates (Chan et al. 2010), and the distribution of covariates would be different over the units with different treatment level.

To address the counterfactual problem and confounding bias issue, throughout this paper, we assume following standard assumptions (Rosenbaum and Rubin 1983) are satisfied.

Assumption 1: Stable Unit Treatment Value Given the observed covariates, the distribution of potential outcome for one unit is assumed to be unaffected by the particular treatment assignment of another unit.

Assumption 2: Unconfoundedness Given the observed covariates, the distribution of treatment is independent of potential outcome. Formally we have, $T \perp Y(t) | \mathbf{X}, \forall t \in \mathcal{T}$.

Assumption 3: Overlap Every unit has a nonzero probability to receive either treatment status when given the observed covariates. Formally we have, $P(r(T = t, \mathbf{X} = x) > 0) = 1$, where $r(T = t, \mathbf{X} = x) = f_{T|\mathbf{X}}(t|x)$ denotes the conditional density of treatment given covariates.

Under these assumptions, we propose a sample re-weighting technique for removing the confounding bias between treatment T and covariates $\mathbf{X}$. The re-weighting method forms the surrogates of the unobserved potential outcome $Y_i(t)$ over all units by re-weighting units with sample weights $\mathbf{w} \in \mathbb{R}^{n\times 1}$ to make the treatment T become independent with the covariates $\mathbf{X}$. Then, the unobserved potential outcome $Y_i(t)$ over all units can be approximated by the observed outcome $Y_i(t)$ over the units with treatment $T = t$. Finally, with the learned sample weights $\mathbf{w}$, we can approximately estimate the ADRF on each treatment level t by:

$$\widehat{ADRF} = \sum_{i:T_i=t} w_i \cdot Y_i(t). \tag{3}$$

Similarity, we can also approximately estimate the MTEF as:

$$\widehat{MTEF} = \frac{\sum_{i:T_i=t} w_i \cdot Y_i(t) - \sum_{i:T_i=t-\triangle t} w_i \cdot Y_i(t)}{\triangle t}. \tag{4}$$

4 Method

In this section, we give the details of our proposed Generative Adversarial Deconfounding (GAD) algorithm for continuous treatment effect estimation in observational studies.

4.1 Generative adversarial de-confounding algorithm

To fully remove the confounding bias induced by the dependency between treatment T and covariates $\mathbf{X}$ in observational studies as shown in Fig. 1b, we propose to make treatment T become independent with the covariates $\mathbf{X}$ by sample re-weighting, that is our Generative Adversarial De-confounding (GAD) algorithm. In our GAD algorithm, there are two key components: (i) "calibration" distribution generation: Based on the observed data $\mathbf{D}_{obs} = \{T, \mathbf{X}\}$, we generate an "calibration" data $\mathbf{D}_{cal} = \{T, \mathbf{X}'\}$ by change the distribution of covariates such that $P(T|\mathbf{X}') = P(T)$, namely $T \perp \mathbf{X}'$. (ii) "calibration" distribution approximation: We develop a Generative Adversarial Network to learn a sample weight $\mathbf{w}$ on the observed data $\mathbf{D}_{obs}$ such that the distribution of weighted observed data would be similar even identical with the "calibration" data $\mathbf{D}_{cal}$, formally $\mathbf{w}P(T, \mathbf{X}) = P(T, \mathbf{X}')$. Finally, the learned sample weight $\mathbf{w}$ can guarantee precise estimation on the causal effect of continuous treatment, since it ensures the treatment is dependent of the covariates on the weighted observed data, achieving de-confounding between treatment and covariates.

4.1.1 "Calibration" distribution generation

In this component, our goal is to generate an "calibration" distribution $\mathbf{D}_{cal} = \{T, \mathbf{X}'\}$, where the treatment T is independent of the covariates $\mathbf{X}'$, ensuring there is no confounding between treatment and covariates.

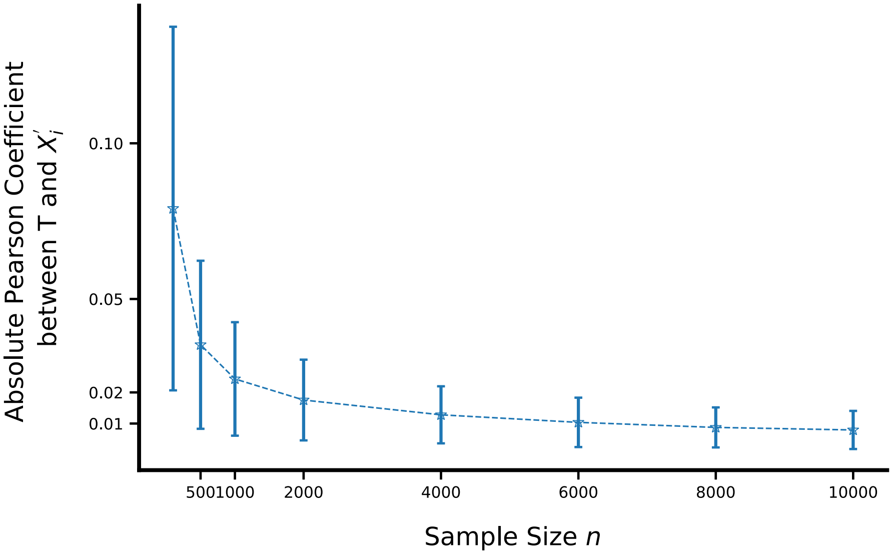


Fig. 2 Pearson correlation after shuffle versus sample size, where $T = \mathbf{X}$. The Pearson correlation between shuffled variable $\mathbf{X}'$ and T would decrease to $zero$ as the sample size $n \to \infty$

Proposition 1 *By randomly shuffle the covariates* $\mathbf{X}$ *over all samples in observed data* $\mathbf{D}_{obs} = \{T, \mathbf{X}\}$, *the shuffled covariates* $\mathbf{X}^{'}$ *would become independent with the treatment* T *if sample size* $n \to \infty$.

The randomly shuffle processing refers to random permutation of the unit index of observed covariates $\mathbf{X} \in \mathbb{R}^{n \times p}$. If $n \to \infty$, the shuffled covariates, denoted as $\mathbf{X}'$, should be independently random variables.[2] Hence, the treatment variable T would be independent with the shuffled covariates $\mathbf{X}'$. An empirical evidence for Proposition 1 is given in Fig. 2, where we employ the Pearson coefficient between variables to approximate their dependency.

Therefore, we can obtain an "calibration" data $\mathbf{D}_{cal} = \{T, \mathbf{X}'\}$ under Proposition 1, where the confounding bias between the treatment T and covariates $\mathbf{X}'$ are removed.

Need to note that the "calibration" data is meaningless except for its non-confounding or independence property between its treatment and covariates. Many other methods can also be employed for generating an "calibration" data, we leave it in future work.

4.1.2 "Calibration" distribution approximation

In this component, we aim to adjust the distribution of observed data $\mathbf{D}_{obs} = \{T, \mathbf{X}\}$ by sample weighting such that with the identical distribution of the "calibration" data $\mathbf{D}_{cal} = \{T, \mathbf{X}'\}$, resulting in the treatment becoming independent of the covariates in the adjusted observed data.

[2] $\mathbf{X}'$ should have the identical marginal distribution with the observed covariates, that is $P(\mathbf{X}') = P(\mathbf{X})$.

Inspired by the immense success of Generative Adversarial Network (GAN) (Goodfellow et al. 2014) in producing simulated data that highly resembles the distribution of real-world samples, we propose a novel framework that leverages the objective of GAN to the task of generating weights to ensure that the distribution of adjusted observed data has the identical distribution of the "calibration" one.

To be self-contained, we briefly revisit the key idea of GAN (Goodfellow et al. 2014). The goal of GAN is to learn a generative model $g(\cdot)$ of an unknown distribution $\mathcal{D}_{data}$ using a class of discriminators $d(\cdot)$ to gauge the similarity between data distributions. The GAN framework can be described as a game between the generator $g(\cdot)$ and the discriminator $d(\cdot)$, where the generator $g(\cdot)$ simulates data $g(z)$ with an input random variable z from a predefined distribution $\mathcal{D}_z$, then the discriminator $d(\cdot)$ attempters to bridge the distribution between the simulated data $g(z)$ and real samples s in $\mathcal{D}_{data}$ by minimizing the expected classification error in the real and simulated samples as:

$$L(g, d) = \mathbb{E}_{s\sim\mathcal{D}_{data}}[l(d(s), 1)] + \mathbb{E}_{z\sim\mathcal{D}_z}[l(d(g(z), 0)], \tag{5}$$

where $l(\cdot)$ is the loss function. Given the discriminator model $d(\cdot)$, the generator $g(\cdot)$ attempts to maximize the expected error with following objective function to find:

$$g^\star = \arg\max_g \big(\min_d L(g, d)\big). \tag{6}$$

In our problem, we employ the generator $g(\cdot)$ to optimize a sample weight vector $\mathbf{w} = (w_1, w_2, \ldots, w_n)$ to adjust the distribution of observed data $\mathbf{D}_{obs} = \{T, \mathbf{X}\}$, such that the discriminator $d(\cdot)$ cannot distinguish the adjusted observed distribution and the "calibration" distribution by minimizing the expected classification error in the adjusted observed and "calibration" samples as:

$$\begin{aligned} L(\mathbf{w}, d) = \; & \mathbb{E}_{(t,x)\sim\mathbf{D}_{cal}}[l(d(t, x), 1)] \\ & +\mathbb{E}_{(t,x)\sim\mathbf{D}_{obs}}[w_{(t,x)} \cdot l(d(t, x), 0)], \\ & s.t. \mathbb{E}_{(t,x)\sim\mathbf{D}_{obs}}[w_{(t,x)}] = 1, \mathbf{w} \succeq 0 \end{aligned} \tag{7}$$

where $w_{(t,x)}$ refers to the sample weight related to the sample (t, x) in the observed data, and $l(\cdot)$ is the loss function. The term $\mathbb{E}_{(t,x)\sim\mathbf{D}_{obs}}[w_{(t,x)}] = 1$ avoids all sample weights to be $zero$, and $\mathbf{w} \succeq 0$ constrains each sample weight to be non-negative. Given the discriminator model $d(\cdot)$, the generator $g(\cdot)$ attempts to maximize the expected error with following objective function to find:

$$\hat{\mathbf{w}} = \arg\max_{\mathbf{w}} \big(\min_d L(\mathbf{w}, d)\big). \tag{8}$$

Following the objective function in Eq. (7), we know only the term $\mathbb{E}_{(t,x)\sim\mathbf{D}_{obs}}[w_{(t,x)} \cdot l(d(t, x), 0)]$ is related to the parameter $\mathbf{w}$. Then to optimize $\mathbf{w}$ with discriminator $d(\cdot)$ fixed, we could either maximize $\mathbb{E}_{(t,x)\sim\mathbf{D}_{obs}}[w_{(t,x)} \cdot l(d(t, x), 0)]$ with gradient ascending methods, or instead choose to minimize $\mathbb{E}_{(t,x)\sim\mathbf{D}_{obs}}[-w_{(t,x)} \cdot l(d(t, x), 0)]$ or $\mathbb{E}_{(t,x)\sim\mathbf{D}_{obs}}[w_{(t,x)} \cdot l(d(t, x), 1)]$ with gradient descending methods as mentioned in

Algorithm 1 Generative Adversarial De-confounding

```
Input: Observed Data $\mathbf{D}_{obs} = \{T, \mathbf{X}\}$, stopping criterion $h(\mathbf{D}_{obs}, \mathbf{D}_{target}, \mathbf{w})$, optimizer for discriminator, $SGD(\theta, L_d(\mathbf{w}, d))$, and optimizer for $\mathbf{w}$, $Ranger(\mathbf{w}, L_w(\mathbf{w}, d))$
Output: sample weight $\mathbf{w}$
1: Generating shuffled covariate $\mathbf{X}'$ by randomly permuting unit indices of $\mathbf{X}$
2: Generate target data $\mathbf{D}_{cal} = \{T, \mathbf{X}'\}$
3: Initialize sample weight $\mathbf{w}^0 = [1, 1, \dots, 1]$
4: Initialize discriminator $d(\cdot)$ with parameter $\theta^0$
5: Initialize the iteration variable $t \leftarrow 0$
6: repeat
7:     $t \leftarrow t + 1$
8:     Update $\theta^t \leftarrow SGD(\theta^{t-1}, L_d(\mathbf{w}^{t-1}, d))$
9:     Update $\mathbf{w}^t \leftarrow Ranger(\mathbf{w}^{t-1}, L_w(\mathbf{w}^{\mathbf{t-1}}, d))$
10:    $\mathbf{w}_i^t \leftarrow n\mathbf{w}_i^t / \sum_{i=1}^n \mathbf{w}_i^t, \quad i = 1, 2, \dots, n$
11: until $h(\mathbf{D}_{obs}, \mathbf{D}_{cal}, \mathbf{w}^t)$ satisfied or max iteration is reached
12: return sample weight $\mathbf{w}$
```

Goodfellow et al. (2014). In practice, we switch 0/1 labels for two data distributions, resulting the following loss functions for both $\mathbf{w}$ and discriminator $d(\cdot)$ to minimize alternately:

$$\begin{aligned} L_d(\mathbf{w}, d) &= L(\mathbf{w}, d) \\ L_w(\mathbf{w}, d) &= \mathbb{E}_{(t,x)\sim\mathbf{D}_{obs}}[w_{(t,x)} \cdot l(d(t, x), 1)], \\ &\qquad s.t.\mathbb{E}_{(t,x)\sim\mathbf{D}_{obs}}[w_{(t,x)}] = 1, \mathbf{w} \succeq 0 \end{aligned} \tag{9}$$

Besides the original GAN objective, other variants of GAN, e.g. WGAN (Arjovsky et al. 2017), could also be applied to our problem. The details of our GAD algorithm is summarized in Algorithm 1, where steps 1–4 is for generating the "calibration" distribution $\mathbf{D}_{cal} = \{T, \mathbf{X}'\}$, and steps 5–12 is for approximating the "calibration" distribution by learning sample weight.

Finally, with the optimized sample weights $\mathbf{w}$ by our GAD algorithm, we can estimate the ADRF with Eq. (3) and MTEF with Eq. (4).

4.2 Theoretical analysis

In this section, we give theoretical analysis about our algorithm, and prove it can fully remove the confounding bias between treatment and covariates by making them become independent via sample weight learning. A key requirement for the method to work is following assumption.

Assumption 1 Each unit in "calibration" data is also included in the observed data. Formally, $P_{(t,x)\sim\mathbf{D}_{cal}}(t, x) > 0 \Longrightarrow P_{(t,x)\sim\mathbf{D}_{obs}}(t, x) > 0$.

Then, we have following theorem.

Theorem 1 *Under Assumption 1 and Proposition 1, there exists a sample weights* $\mathbf{w}^\star$ *such that*

$$T^{\mathbf{w}^\star} \perp \mathbf{X}^{\mathbf{w}^\star} \tag{10}$$

in the weighted observed data $\mathbf{D}_{obs}^{\mathbf{w}^\star} = \{T^{\mathbf{w}^\star}, \mathbf{X}^{\mathbf{w}^\star}\}$ *with probability 1. In particular, one solution to such* $\mathbf{w}^\star$ *that satisfies Eq. (10) is* $\mathbf{w}^\star = \frac{P_{(t,x)\sim\mathbf{D}_{cal}}(t,x)}{P_{(t,x)\sim\mathbf{D}_{obs}}(t,x)}$

Proof Firstly, we prove that the distribution of weighted observed data would similar even identical with the "calibration" distribution. For any function $f(\cdot)$, with the sample weight $\mathbf{w}^\star$, we have

$$\begin{aligned}
&\mathbb{E}_{(t,x)\sim\mathbf{D}_{obs}}[\mathbf{w}^\star \cdot f(t,x)] \\
&= \mathbb{E}_{(t,x)\sim\mathbf{D}_{obs}}\left[\frac{P_{(t,x)\sim\mathbf{D}_{cal}}(t,x)}{P_{(t,x)\sim\mathbf{D}_{obs}}(t,x)} \cdot f(t,x)\right] \\
&= \int_t\int_x \left(\frac{P_{(t,x)\sim\mathbf{D}_{cal}}(t,x)}{P_{(t,x)\sim\mathbf{D}_{obs}}(t,x)} \cdot f(t,x)\right) \cdot P_{(t,x)\sim\mathbf{D}_{obs}}(t,x)\, dtdx \\
&= \int_t\int_x P_{(t,x)\sim\mathbf{D}_{cal}}(t,x) \cdot f(t,x)\, dtdx \\
&= \mathbb{E}_{(t,x)\sim\mathbf{D}_{cal}}[f(t,x)]
\end{aligned} \tag{11}$$

From the property of Moments,[3] we know that a distribution of variables is uniquely determined by the collection of all the moments (of all orders, from 0 to ∞). Here, we can adopt different functions $f(\cdot)$ to represent the moments of distribution of weighted observed data $\mathbf{D}_{obs}^{\mathbf{w}^\star}$ and "calibration" data $\mathbf{D}_{cal} = \{T, \mathbf{X}'\}$, and obtain that $\mathbf{D}_{obs}^{\mathbf{w}^\star}$ and $\mathbf{D}_{cal}$ are identical distribution since they have the identical moment on all orders as proved in Eq. (11).

From Proposition 1, we know T is independent with the $\mathbf{X}'$ in the "calibration" data $\mathbf{D}_{cal} = \{T, \mathbf{X}'\}$. Hence, one can infer that the weighted treatment $T^{\mathbf{w}^\star}$ would also be independent with $X^{\mathbf{w}^\star}$ in the weighted observed data $\mathbf{D}_{obs}^{\mathbf{w}}$ with the sample weight $\mathbf{w}^\star$, namely $T^{\mathbf{w}^\star} \perp X^{\mathbf{w}^\star}$. □

With Proposition 1 and Theorem 1, we can derive the following property of the sample weight $\hat{\mathbf{w}}$ optimized by our algorithm in Eq. (8).

Property 1. *Under Assumption 1, the confounding bias between treatment and covariates in observed data would be removed or de-confounded by sample weighting with* $\hat{\mathbf{w}}$.

5 Experiment

In this section, we evaluate the effectiveness of our proposed method on both synthetic and real-world datasets.

5.1 Baseline methods

We implement or use the following baseline methods for comparison.

[3] https://en.wikipedia.org/wiki/Moment_(mathematics).

- *Inverse Probability Weighting (IPW)* (Robins et al. 2000): *IPW* is a classic, well-researched method in continuous treatment setting. This method estimates conditional probability $P(T_i|\mathbf{X_i})$ by regressing treatment T on covariates $\mathbf{X}$, then uses it to generate sample weights. Both unstablized ($IPW_{unstable} = \frac{1}{P(T_i|\mathbf{X}_i)}$) and stablized ($IPW_{stable} = \frac{P(T_i)}{P(T_i|\mathbf{X}_i)}$) versions are evaluated. Performance of *IPW* largely relies on estimation of $P(T_i|\mathbf{X}_i)$. Thus, it's not attractive in most real-world applications.
- *Inverse Second-Moment Weighting (ISMW)* (Galagate 2016): This method is an extension of *IPW* with second-moment. Under linear assumption of Y-T relation, *ISMW* generates sample weights matrix in closed form as $\mathbb{E}(B_i B_i^T|\mathbf{X}_i)^{-1}$, where $B_i = [1, t_i]^T$. Thus, *ISMW* could perform well under limited assumptions. However, if Y-T relation is more complex, *ISMW* might be less attractive due to its restriction to the means of higher-order terms.
- *Generalized Propensity Score by Boosting Modeling (GBM)* (Zhu et al. 2015): *GBM* is an extension of *IPW*, which better improves generalized propensity score estimation with more flexible modeling capability. This method generates weights in the same way as IPW_{stable}, except that it uses boosting to model generalized propensity score. To determine the best parameter for number of trees, GBM calculates average correlation coefficient (e.g. Pearson Correlation, Spearman Correlation) for each value in searching grid, then chooses the best one. With the boosting algorithm and parameter tuning procedure, GBM provides better conditional density estimation. However the main problem of IPW on difficulty in estimation of conditional probability still remains unsolved in GBM.
- *Covariate-Balancing Generalized Propensity Score (CBGPS) and non-parametric version (npCBGPS)* (Fong et al. 2018): *CBGPS* is a recent well-performed method based on generalized propensity score. This method adapts covariate balancing condition for continuous treatment that $\mathbb{E}(P(T_i|\mathbf{X}_i)T_i\mathbf{X}_i) = \mathbb{E}(T_i)\mathbb{E}(\mathbf{X}_i) = 0$, where $\mathbf{X}$ and T are centralized and orthogonalized in preprocessing.

5.2 Evaluation metrics

In synthetic experiments, we evaluate the performance based on three metrics:

- Bias(MTEF): mean absolute error of MTEF estimation over all samples

$$MTEF_{Bias} = \frac{1}{n}\sum_{i=1}^{n}|MTEF(T_i) - \widehat{MTEF}(T_i)|$$

- RMSE(MTEF): rooted mean squared error of MTEF estimation over all samples

$$MTEF_{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}[MTEF(T_i) - \widehat{MTEF}(T_i)]^2}$$

– RMSE(ADRF): rooted mean squared error of ADRF estimation over all samples

$$ADRF_{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}[ADRF(T_i) - \widehat{ADRF}(T_i)]^2}$$

All metrics above measures performance of causal effect estimation, with respect to specific choice of causal effect under continuous treatment setting based on commonly-used definitions we introduced previously, ADRF and MTEF. Normally, MTEF-based metrics are more important than ADRF-based in synthetic experiments, as it eliminates effect of intercept which involves means of covariates and noise.

5.3 Experiments on synthetic data

In this section, we introduce data generation process for synthetic datasets, and demonstrate the effectiveness of our proposed weighting method with extensive experiments.

5.3.1 Dataset

The process of generating synthetic datasets basically follows Fong et al. (2018). As the dimension of observed variables is fixed in the original procedure, we carry out data generation with slight modification for further experiments with varying sample size and dimensions of observed variables, where we consider three sample sizes $n = \{2000; 5000; 8000\}$ and also vary the dimension of observed variables $p = \{10; 30; 50\}$. We first generate covariates $\mathbf{X} = (x_1, x_2, \ldots, x_p)$ independently with **Standard Normal** distribution as:

$$x_1, x_2, \ldots, x_p \overset{i.i.d}{\sim} N(0, 1)$$

Then we generate treatment T and outcome Y generally as:

$$T = f(\mathbf{X}) + \epsilon_t; \quad Y = g(\mathbf{X}) + \mu(T) + \epsilon_y$$

where

$$f(\mathbf{X}) = \sum_{j=1}^{p} \alpha_{mod(j,10)} \cdot x_j,$$

$$g(\mathbf{X}) = \sum_{j=1}^{p} \beta_{mod(j,10)} \cdot x_j,$$

$\alpha = [1, 1, 0.2, 0.2, 0.2, 0, 0, 0, 0, 0]$ and $\epsilon_t \sim N(0, 2)$. Function $mod(a, b)$ returns the modulus after division of a by b. β, $\mu(T)$ and ϵ_y varies under different settings with considering the relation (linear and non-linear) between Y and T, and between Y and $\mathbf{X}$:

YT-linear:

$$\mu(T) = T \text{ and } \epsilon_y \sim N(0, 5)$$

YT-nonlinear:

$$\mu(T) = T^2 + T, \epsilon_y \sim N(0, 9) \text{ and } g(\mathbf{X}) = 2g(\mathbf{X})$$

YX-linear:

$$\beta = [0, 1, 0, 0.1, 0.1, 0.1, 0, 0, 0, 0]$$

YX-nonlinear:

$$\beta = [0, 2, 0, 0.5, 0.5, 0.5, 0, 0, 0, 0] \text{ and for } x_j \text{ in } g(\mathbf{X}),$$
$$x_j = I(mod(j, 10) = 1)(x_j + 0.5)^2 + I(mod(j, 10) \neq 1)x_j$$

By combining different YX relations and YT relations, we could evaluate all methods under 4 different settings which cover a large variety of common cases. As treatment assignment mechanism doesn't always satisfy linear assumption made by some methods, to demonstrate performance of all methods when misspecification of treatment assignment occurs, we introduce settings under YX-linear relation, with sample size $n = 2000$ and dimension of covariates $p = 10$. Similar to YX-nonlinear setting, we add nonlinear term in treatment assignment function by modifying x_j in $f(\mathbf{X})$ as,

$$x_j = I(mod(j, 10) = 1)(x_j + 0.5)^2 + I(mod(j, 10) \neq 1)x_j$$

In simulation, we know the ground-truth ADRF and MTEF as:

YT-linear:

$$ADRF(T) = T + \mathbb{E}(g(X)) \text{ and } MTEF = 1$$

YT-nonlinear:

$$ADRF(T) = T^2 + T + 2\mathbb{E}(g(X))$$
$$\text{and } MTEF = 2T + 1$$

Then, we evaluate the ADRF and MTEF with our algorithm, comparing with baselines.

5.3.2 Implementation details

We implement both versions of *IPW*, and *ISMW* as baseline methods. For *GBM*, we use implementation provided in Zhu et al. (2015). As for both versions of *CBGPS*, the

R package 'CBPS' is used to carry out experiments on both synthetic and real-world datasets. Implementation details for experiments on *TWINS* dataset are the same.

The core part of *IPW* and *ISMW* is to estimate $P(T_i)$, $P(T_i|\mathbf{X}_i)$ and $\mathbb{E}(B_i B_i^T|\mathbf{X}_i)$. As the main estimation part involves only regression, we follows similar procedure for generalized propensity score estimation in Zhu et al. (2015). We use ordinary least square method to perform regression without penalty term, thus no hyper-parameters are needed to be tuned.

For *GBM*, we follows the whole procedure described in Zhu et al. (2015). The original algorithm includes tuning procedure for the most important hyper-parameter, the number of trees in boosting. We use the same procedure with the same searching range for tuning the number of trees in all experiments.

Parameters of *CBGPS* and *npCBGPS* also are chosen according to the original description in Fong et al. (2018) and documentation in R package. The only hyper-parameter needs to be set manually is prior correlation in *npCBGPS*, we follow the description in documentation to use $.1/n$ as prior correlation, as the solution is likely to exist while the balance is fine enough under such choice.

As hyper-parameters in our method are mostly normal ones for neural network training, including keep probability for dropout layer, learning rates and internal steps for optimizers. We use a hold-out dataset generated in the same procedure to tune these hyper-parameters, based on the criterion that the ones achieved minimum Pearson correlation coefficient are chosen. The final choices are as follows. Dropout layer with keep probability $= 0.5$ is applied to last hidden layer. We use SGD with learning rate $lr = 1e^{-3}$ as optimizer of discriminator, Ranger (a combination of RAdam and Look-Ahead) with learing rate $lr = 3e^{-4}$, betas $= (0.0, 0.9)$, internal step $k = 5$ as optimizer of sample weights.

5.3.3 Results and analyses

To evaluate the performance of our proposed algorithm on continuous treatment effect estimation, we carry out experiments for 10 times independently for each setting. Based on the estimated ADRF and MTEF, we report Bias(MTEF), RMSE(MTEF) and RMSE(ADRF), and their standard error (SD) over 10 times experiments in Tables 2, 3, 4 and 5. In Tables 2 and 4 we varied the sample size with fixed covariate dimensions, and in Table 3 we varied the dimension of covariates. In Table 5 we demonstrate the performance when possible misspecification of treatment assignment model occurs. From these results, we have following observations and analyses:

- Model based regression method, OLS, cannot precisely estimate the causal effect of continuous treatment even the model is correctly specified, since it ignores the confounding bias between treatment and covariates.
- With constraints on the variance of weights, IPW_{stable} achieves better performance than $IPW_{unstable}$ across most settings. Moreover, with considering the second moments, $ISMW$ obtains the best performance among IPW based methods under setting with YT-linear. However, in the setting with YT-nonlinear, the performance of $ISMW$ is very poor and even worse than OLS, since it entirely relies on the linear assumption between T and Y. When treatment assignment

Table 2 Results on synthetic datasets with varying the sample size n in different settings

Setting	Method	$n = 2000$, $p = 10$			$n = 5000$, $p = 10$			$n = 8000$, $p = 10$		
		BIAS_{MTEF}	RMSE_{MTEF}	$RMSE_{ADRF}$	BIAS_{MTEF}	RMSE_{MTEF}	$RMSE_{ADRF}$	BIAS_{MTEF}	RMSE_{MTEF}	$RMSE_{ADRF}$
linear Y-X, linear Y-T	OLS	0.153 (0.044)	0.153 (0.044)	0.392 (0.102)	0.171 (0.036)	0.171 (0.036)	0.429 (0.090)	0.174 (0.024)	0.174 (0.024)	0.432 (0.059)
	$\text{IPW}_{unstable}$	0.141 (0.080)	0.141 (0.080)	0.463 (0.182)	0.222 (0.179)	0.222 (0.179)	0.729 (0.466)	0.199 (0.179)	0.199 (0.179)	0.719 (0.396)
	IPW_{stable}	0.071 (0.079)	0.071 (0.079)	0.235 (0.193)	0.042 (0.030)	0.042 (0.030)	0.134 (0.090)	0.044 (0.028)	0.044 (0.028)	0.142 (0.064)
	ISMW	0.049 (0.026)	0.049 (0.026)	**0.159 (0.072)**	0.035 (0.027)	0.035 (0.027)	0.135 (0.047)	**0.027 (0.019)**	**0.027 (0.019)**	**0.115 (0.053)**
	GBM	0.056 (0.060)	0.056 (0.060)	0.226 (0.162)	0.042 (0.025)	0.042 (0.025)	0.129 (0.049)	0.039 (0.045)	0.039 (0.045)	0.154 (0.089)
	CBGPS	0.063 (0.069)	0.063 (0.069)	0.223 (0.171)	0.041 (0.029)	0.041 (0.029)	0.130 (0.092)	0.039 (0.024)	0.039 (0.024)	0.156 (0.057)
	npCBGPS	0.146 (0.100)	0.146 (0.100)	0.440 (0.283)	0.069 (0.100)	0.069 (0.100)	0.303 (0.250)	0.134 (0.083)	0.134 (0.083)	0.470 (0.258)
	Ours	**0.037 (0.036)**	**0.037 (0.036)**	0.167 (0.084)	**0.023 (0.018)**	**0.023 (0.018)**	**0.092 (0.055)**	0.035 (0.033)	0.035 (0.033)	0.131 (0.069)
linear Y-X, nonlinear Y-T	OLS	0.310 (0.078)	0.332 (0.079)	0.816 (0.181)	0.343 (0.065)	0.347 (0.065)	0.861 (0.163)	0.347 (0.043)	0.350 (0.045)	0.867 (0.109)
	$\text{IPW}_{unstable}$	0.286 (0.124)	0.337 (0.148)	0.875 (0.413)	0.337 (0.198)	0.404 (0.232)	1.039 (0.535)	0.348 (0.209)	0.407 (0.226)	1.006 (0.553)
	IPW_{stable}	0.211 (0.138)	0.252 (0.160)	0.583 (0.364)	0.120 (0.081)	0.145 (0.103)	0.327 (0.227)	0.140 (0.061)	0.166 (0.082)	0.375 (0.142)
	ISMW	1.026 (0.527)	1.053 (0.527)	2.577 (1.323)	0.926 (0.316)	0.930 (0.312)	2.300 (0.774)	0.811 (0.209)	0.816 (0.208)	2.025 (0.527)
	GBM	0.196 (0.112)	0.243 (0.137)	0.573 (0.309)	0.119 (0.049)	0.141 (0.064)	0.309 (0.126)	0.147 (0.076)	0.175 (0.096)	0.395 (0.186)
	CBGPS	0.195 (0.126)	0.237 (0.152)	0.558 (0.334)	0.118 (0.083)	0.143 (0.105)	0.320 (0.234)	0.138 (0.060)	0.165 (0.081)	0.373 (0.141)
	npCBGPS	0.383 (0.233)	0.465 (0.293)	1.062 (0.669)	0.273 (0.206)	0.339 (0.255)	0.783 (0.565)	0.381 (0.291)	0.455 (0.368)	1.071 (0.746)
	Ours	**0.151 (0.097)**	**0.184 (0.124)**	**0.452 (0.192)**	**0.083 (0.034)**	**0.099 (0.043)**	**0.230 (0.082)**	**0.067 (0.043)**	**0.077 (0.046)**	**0.199 (0.095)**

Table 2 continued

Setting	Method	$n = 2000, p = 10$			$n = 5000, p = 10$			$n = 8000, p = 10$		
		BIAS$_{MTEF}$	RMSE$_{MTEF}$	$RMSE_{ADRF}$	BIAS$_{MTEF}$	RMSE$_{MTEF}$	$RMSE_{ADRF}$	BIAS$_{MTEF}$	RMSE$_{MTEF}$	$RMSE_{ADRF}$
nonlinear Y-X, linear Y-T	OLS	0.353 (0.066)	0.353 (0.066)	0.882 (0.159)	0.369 (0.048)	0.369 (0.048)	0.916 (0.125)	0.372 (0.035)	0.372 (0.035)	0.922 (0.087)
	IPW$_{unstable}$	0.184 (0.087)	0.184 (0.087)	0.580 (0.206)	0.291 (0.239)	0.291 (0.239)	0.898 (0.669)	0.239 (0.244)	0.239 (0.244)	0.804 (0.607)
	IPW$_{stable}$	0.101 (0.062)	0.101 (0.062)	0.299 (0.139)	0.057 (0.031)	0.057 (0.031)	0.186 (0.076)	0.055 (0.044)	0.055 (0.044)	0.193 (0.104)
	ISMW	0.062 (0.051)	0.062 (0.051)	**0.234 (0.092)**	0.040 (0.041)	0.040 (0.041)	**0.177 (0.091)**	**0.033 (0.027)**	**0.033 (0.027)**	0.148 (0.071)
	GBM	0.086 (0.053)	0.086 (0.053)	0.314 (0.102)	0.055 (0.032)	0.055 (0.032)	0.204 (0.073)	0.062 (0.055)	0.062 (0.055)	0.232 (0.121)
	CBGPS	0.096 (0.067)	0.096 (0.067)	0.294 (0.138)	0.055 (0.030)	0.055 (0.030)	0.180 (0.071)	0.054 (0.043)	0.054 (0.043)	0.193 (0.098)
	npCBGPS	0.179 (0.079)	0.179 (0.079)	0.519 (0.221)	0.093 (0.086)	0.093 (0.086)	0.353 (0.188)	0.097 (0.091)	0.097 (0.091)	0.407 (0.256)
	Ours	**0.047 (0.028)**	**0.047 (0.028)**	0.237 (0.103)	**0.034 (0.022)**	**0.034 (0.022)**	0.212 (0.106)	0.036 (0.029)	0.036 (0.029)	**0.134 (0.075)**
nonlinear Y-X, nonlinear Y-T	OLS	0.753 (0.120)	0.871 (0.145)	1.982 (0.320)	0.783 (0.080)	0.918 (0.092)	2.069 (0.231)	0.792 (0.068)	0.931 (0.082)	2.086 (0.183)
	IPW$_{unstable}$	0.364 (0.127)	0.420 (0.141)	1.045 (0.333)	0.435 (0.285)	0.509 (0.342)	1.307 (0.845)	0.416 (0.315)	0.479 (0.355)	1.222 (0.875)
	IPW$_{stable}$	0.283 (0.106)	0.339 (0.130)	0.746 (0.275)	0.160 (0.077)	0.194 (0.099)	0.451 (0.203)	0.156 (0.067)	0.186 (0.080)	**0.458 (0.191)**
	ISMW	1.716 (0.952)	1.800 (0.930)	4.410 (2.328)	1.771 (0.424)	1.808 (0.416)	4.562 (0.987)	1.664 (0.261)	1.704 (0.259)	4.296 (0.626)
	GBM	0.245 (0.124)	0.296 (0.157)	0.734 (0.268)	0.168 (0.069)	0.202 (0.086)	0.495 (0.165)	0.154 (0.087)	0.186 (0.099)	0.502 (0.226)
	CBGPS	0.267 (0.106)	0.317 (0.127)	**0.714 (0.271)**	0.163 (0.075)	0.199 (0.095)	**0.449 (0.195)**	0.158 (0.065)	0.189 (0.078)	0.460 (0.182)
	npCBGPS	0.542 (0.254)	0.654 (0.333)	1.419 (0.657)	0.394 (0.278)	0.485 (0.354)	1.076 (0.592)	0.436 (0.364)	0.540 (0.458)	1.223 (0.797)
	Ours	**0.232 (0.125)**	**0.269 (0.145)**	0.941 (0.387)	**0.127 (0.062)**	**0.154 (0.075)**	0.638 (0.295)	**0.115 (0.076)**	**0.136 (0.090)**	0.621 (0.198)

The value in bracket refers to corresponding standard deviations of 10 times experiments. The smaller of these metrics, the better. Bold number refers to the best result in means for each metric

Table 3 Results on synthetic datasets with varying the dimension p in different settings

Setting	Method	$n = 2000$, $p = 10$			$n = 2000$, $p = 30$			$n = 2000$, $p = 50$		
		BIAS$_{MTEF}$	RMSE$_{MTEF}$	$RMSE_{ADRF}$	BIAS$_{MTEF}$	RMSE$_{MTEF}$	$RMSE_{ADRF}$	BIAS$_{MTEF}$	RMSE$_{MTEF}$	$RMSE_{ADRF}$
linear Y-X, linear Y-T	OLS	0.153 (0.044)	0.153 (0.044)	0.392 (0.102)	0.314 (0.037)	0.314 (0.037)	1.021 (0.125)	0.360 (0.039)	0.360 (0.039)	1.386 (0.153)
	IPW$_{unstable}$	0.141 (0.080)	0.141 (0.080)	0.463 (0.182)	0.196 (0.071)	0.196 (0.071)	0.723 (0.211)	0.221 (0.103)	0.221 (0.103)	1.009 (0.362)
	IPW$_{stable}$	0.071 (0.079)	0.071 (0.079)	0.235 (0.193)	0.111 (0.062)	0.111 (0.062)	0.431 (0.222)	0.186 (0.122)	0.186 (0.122)	0.830 (0.430)
	ISMW	0.049 (0.026)	0.049 (0.026)	**0.159 (0.072)**	0.052 (0.030)	0.052 (0.030)	**0.212 (0.078)**	**0.049 (0.023)**	**0.049 (0.023)**	**0.225 (0.058)**
	GBM	0.056 (0.060)	0.056 (0.060)	0.226 (0.162)	0.117 (0.051)	0.117 (0.051)	0.470 (0.144)	0.230 (0.050)	0.230 (0.050)	0.901 (0.184)
	CBGPS	0.063 (0.069)	0.063 (0.069)	0.223 (0.171)	0.206 (0.157)	0.206 (0.157)	0.791 (0.484)	0.306 (0.159)	0.306 (0.159)	1.608 (0.587)
	npCBGPS	0.146 (0.100)	0.146 (0.100)	0.440 (0.283)	0.129 (0.066)	0.129 (0.066)	0.548 (0.291)	0.158 (0.129)	0.158 (0.129)	0.893 (0.511)
	Ours	**0.037 (0.036)**	**0.037 (0.036)**	0.167 (0.084)	**0.042 (0.028)**	**0.042 (0.028)**	0.283 (0.167)	0.084 (0.058)	0.084 (0.058)	0.443 (0.224)
linear Y-X, nonlinear Y-T	OLS	0.310 (0.078)	0.332 (0.079)	0.816 (0.181)	0.625 (0.068)	0.631 (0.070)	2.041 (0.235)	0.720 (0.073)	0.723 (0.073)	2.777 (0.287)
	IPW$_{unstable}$	0.286 (0.124)	0.337 (0.148)	0.875 (0.413)	0.352 (0.109)	0.395 (0.136)	1.325 (0.313)	0.451 (0.163)	0.491 (0.164)	1.977 (0.577)
	IPW$_{stable}$	0.211 (0.138)	0.252 (0.160)	0.583 (0.364)	0.239 (0.088)	0.268 (0.104)	0.890 (0.393)	0.406 (0.168)	0.459 (0.177)	1.743 (0.720)
	ISMW	1.026 (0.527)	1.053 (0.527)	2.577 (1.323)	1.425 (1.128)	1.778 (1.411)	5.034 (3.900)	0.631 (0.069)	0.655 (0.075)	2.485 (0.275)
	GBM	0.196 (0.112)	0.243 (0.137)	0.573 (0.309)	0.248 (0.080)	0.273 (0.072)	0.954 (0.214)	0.459 (0.089)	0.480 (0.081)	1.839 (0.310)
	CBGPS	0.195 (0.126)	0.237 (0.152)	0.558 (0.334)	0.522 (0.337)	0.586 (0.387)	1.840 (1.115)	0.857 (0.574)	1.028 (0.741)	3.606 (2.030)
	npCBGPS	0.383 (0.233)	0.465 (0.293)	1.062 (0.669)	0.337 (0.157)	0.403 (0.194)	1.270 (0.588)	0.441 (0.189)	0.513 (0.216)	1.963 (0.745)
	Ours	**0.151 (0.097)**	**0.184 (0.124)**	**0.452 (0.192)**	**0.180 (0.133)**	**0.219 (0.168)**	**0.756 (0.432)**	**0.256 (0.111)**	**0.307 (0.146)**	**1.175 (0.375)**
nonlinear Y-X, linear Y-T	OLS	0.353 (0.066)	0.353 (0.066)	0.882 (0.159)	0.620 (0.063)	0.620 (0.063)	2.015 (0.226)	0.749 (0.073)	0.749 (0.073)	2.884 (0.287)

Table 3 continued

Setting	Method	$n = 2000,\ p = 10$			$n = 2000,\ p = 30$			$n = 2000,\ p = 50$		
		BIAS_{MTEF}	RMSE_{MTEF}	$RMSE_{ADRF}$	BIAS_{MTEF}	RMSE_{MTEF}	$RMSE_{ADRF}$	BIAS_{MTEF}	RMSE_{MTEF}	$RMSE_{ADRF}$
	$\text{IPW}_{unstable}$	0.184 (0.087)	0.184 (0.087)	0.580 (0.206)	0.318 (0.140)	0.318 (0.140)	1.211 (0.360)	0.490 (0.175)	0.490 (0.175)	2.101 (0.776)
	IPW_{stable}	0.101 (0.062)	0.101 (0.062)	0.299 (0.139)	0.180 (0.089)	0.180 (0.089)	0.776 (0.237)	0.356 (0.192)	0.356 (0.192)	1.600 (0.782)
	ISMW	0.062 (0.051)	0.062 (0.051)	**0.234 (0.092)**	0.089 (0.059)	0.089 (0.059)	**0.342 (0.168)**	**0.079 (0.043)**	**0.079 (0.043)**	**0.373 (0.181)**
	GBM	0.086 (0.053)	0.086 (0.053)	0.314 (0.102)	0.230 (0.099)	0.230 (0.099)	0.910 (0.313)	0.528 (0.081)	0.528 (0.081)	2.072 (0.293)
	CBGPS	0.096 (0.067)	0.096 (0.067)	0.294 (0.138)	0.336 (0.330)	0.336 (0.330)	1.447 (0.988)	0.585 (0.270)	0.585 (0.270)	4.611 (5.679)
	npCBGPS	0.179 (0.079)	0.179 (0.079)	0.519 (0.221)	0.212 (0.085)	0.212 (0.085)	1.045 (0.380)	0.153 (0.183)	0.153 (0.183)	1.102 (0.777)
	Ours	**0.047 (0.028)**	**0.047 (0.028)**	0.237 (0.103)	**0.066 (0.044)**	**0.066 (0.044)**	0.782 (0.315)	0.193 (0.092)	0.193 (0.092)	1.115 (0.360)
nonlinear Y-X, non-linear Y-T	OLS	0.753 (0.120)	0.871 (0.145)	1.982 (0.320)	1.280 (0.132)	1.452 (0.163)	4.362 (0.497)	1.504 (0.135)	1.665 (0.149)	6.115 (0.580)
	$\text{IPW}_{unstable}$	0.364 (0.127)	0.420 (0.141)	1.045 (0.333)	0.620 (0.278)	0.702 (0.329)	2.440 (0.681)	1.036 (0.302)	1.139 (0.295)	4.461 (1.184)
	IPW_{stable}	0.283 (0.106)	0.339 (0.130)	0.746 (0.275)	0.412 (0.117)	0.452 (0.106)	1.627 (0.405)	0.726 (0.348)	0.801 (0.383)	3.331 (1.452)
	ISMW	1.716 (0.952)	1.800 (0.930)	4.410 (2.328)	4.379 (4.156)	5.458 (5.231)	15.304 (14.382)	1.386 (0.184)	1.614 (0.265)	5.710 (0.828)
	GBM	0.245 (0.124)	0.296 (0.157)	0.734 (0.268)	0.496 (0.190)	0.551 (0.204)	1.907 (0.613)	1.133 (0.129)	1.268 (0.173)	4.615 (0.510)
	CBGPS	0.267 (0.106)	0.317 (0.127)	**0.714 (0.271)**	0.934 (0.637)	1.078 (0.745)	3.515 (2.248)	2.400 (2.693)	2.864 (3.432)	9.912 (10.317)
	npCBGPS	0.542 (0.254)	0.654 (0.333)	1.419 (0.657)	0.608 (0.273)	0.727 (0.346)	2.298 (0.836)	0.514 (0.418)	0.623 (0.525)	2.682 (1.471)
	Ours	**0.232 (0.125)**	**0.269 (0.145)**	0.941 (0.387)	**0.264 (0.181)**	**0.313 (0.216)**	**1.537 (0.646)**	**0.406 (0.179)**	**0.463 (0.210)**	**2.598 (0.699)**

The value in bracket refers to corresponding standard deviations of 10 times experiments. The smaller of these metrics, the better. Bold number refers to the best result in means for each metric

Table 4 Results on synthetic datasets with varing sample size n with fixed high dimension $p = 50$ in different settings

Setting	Method	$n = 2000,\ p = 50$			$n = 5000,\ p = 50$			$n = 8000,\ p = 50$		
		BIAS$_{MTEF}$	RMSE$_{MTEF}$	$RMSE_{ADRF}$	BIAS$_{MTEF}$	RMSE$_{MTEF}$	$RMSE_{ADRF}$	BIAS$_{MTEF}$	RMSE$_{MTEF}$	$RMSE_{ADRF}$
linear Y-X, linear Y-T	OLS	0.360 (0.039)	0.360 (0.039)	1.386 (0.153)	0.369 (0.020)	0.369 (0.020)	1.413 (0.089)	0.367 (0.013)	0.367 (0.013)	1.404 (0.061)
	IPW$_{unstable}$	0.221 (0.103)	0.221 (0.103)	1.009 (0.362)	0.243 (0.124)	0.243 (0.124)	1.053 (0.550)	0.229 (0.148)	0.229 (0.148)	0.966 (0.592)
	IPW$_{stable}$	0.186 (0.122)	0.186 (0.122)	0.830 (0.430)	0.196 (0.148)	0.196 (0.148)	0.871 (0.633)	0.162 (0.128)	0.162 (0.128)	0.673 (0.506)
	ISMW	**0.049 (0.023)**	**0.049 (0.023)**	**0.225 (0.058)**	**0.035 (0.026)**	**0.035 (0.026)**	**0.167 (0.099)**	**0.026 (0.020)**	**0.026 (0.020)**	**0.137 (0.069)**
	GBM	0.230 (0.050)	0.230 (0.050)	0.901 (0.184)	0.237 (0.050)	0.237 (0.050)	0.924 (0.162)	0.219 (0.040)	0.219 (0.040)	0.850 (0.153)
	CBGPS	0.306 (0.159)	0.306 (0.159)	1.608 (0.587)	0.266 (0.248)	0.266 (0.248)	1.599 (1.357)	0.176 (0.215)	0.176 (0.215)	1.026 (0.845)
	npCBGPS	0.158 (0.129)	0.158 (0.129)	0.893 (0.511)	0.197 (0.198)	0.197 (0.198)	1.186 (0.882)	0.099 (0.073)	0.099 (0.073)	0.878 (0.565)
	Ours	0.083 (0.058)	0.083 (0.058)	0.435 (0.229)	0.083 (0.050)	0.083 (0.050)	0.372 (0.199)	0.080 (0.039)	0.080 (0.039)	0.377 (0.126)
linear Y-X, nonlinear Y-T	OLS	0.720 (0.073)	0.723 (0.073)	2.777 (0.287)	0.735 (0.037)	0.738 (0.039)	2.823 (0.169)	0.732 (0.025)	0.733 (0.025)	2.803 (0.114)
	IPW$_{unstable}$	0.451 (0.163)	0.491 (0.164)	1.977 (0.577)	0.441 (0.172)	0.471 (0.205)	1.935 (0.849)	0.436 (0.220)	0.468 (0.241)	1.791 (0.876)
	IPW$_{stable}$	0.406 (0.168)	0.459 (0.177)	1.743 (0.720)	0.399 (0.244)	0.440 (0.244)	1.753 (1.060)	0.342 (0.211)	0.383 (0.217)	1.427 (0.848)
	ISMW	0.631 (0.069)	0.655 (0.075)	2.485 (0.275)	0.647 (0.043)	0.663 (0.047)	2.515 (0.197)	0.631 (0.029)	0.637 (0.029)	2.426 (0.127)
	GBM	0.459 (0.089)	0.480 (0.081)	1.839 (0.310)	0.480 (0.108)	0.490 (0.104)	1.883 (0.366)	0.462 (0.094)	0.473 (0.092)	1.805 (0.350)
	CBGPS	0.857 (0.574)	1.028 (0.741)	3.606 (2.030)	0.557 (0.458)	0.646 (0.495)	2.769 (2.075)	0.531 (0.349)	0.630 (0.368)	2.348 (1.368)
	npCBGPS	0.441 (0.189)	0.513 (0.216)	1.963 (0.745)	0.631 (0.345)	0.732 (0.384)	2.723 (1.507)	0.411 (0.261)	0.508 (0.329)	2.039 (1.085)

Table 4 continued

Setting	Method	$n = 2000$, $p = 50$			$n = 5000$, $p = 50$			$n = 8000$, $p = 50$		
		BIAS$_{MTEF}$	RMSE$_{MTEF}$	$RMSE_{ADRF}$	BIAS$_{MTEF}$	RMSE$_{MTEF}$	$RMSE_{ADRF}$	BIAS$_{MTEF}$	RMSE$_{MTEF}$	$RMSE_{ADRF}$
	Ours	**0.255 (0.115)**	**0.305 (0.151)**	**1.181 (0.375)**	**0.193 (0.089)**	**0.219 (0.092)**	**0.786 (0.334)**	**0.182 (0.076)**	**0.206 (0.088)**	**0.842 (0.202)**
nonlinear Y-X, linear Y-T	OLS	0.749 (0.073)	0.749 (0.073)	2.884 (0.287)	0.773 (0.040)	0.773 (0.040)	2.962 (0.165)	0.773 (0.022)	0.773 (0.022)	2.956 (0.083)
	IPW$_{unstable}$	0.490 (0.175)	0.490 (0.175)	2.101 (0.776)	0.407 (0.176)	0.407 (0.176)	1.953 (0.741)	0.358 (0.222)	0.358 (0.222)	1.696 (0.759)
	IPW$_{stable}$	0.356 (0.192)	0.356 (0.192)	1.600 (0.782)	0.312 (0.189)	0.312 (0.189)	1.628 (0.627)	0.302 (0.142)	0.302 (0.142)	1.432 (0.481)
	ISMW	**0.079 (0.043)**	**0.079 (0.043)**	**0.373 (0.181)**	**0.058 (0.051)**	**0.058 (0.051)**	**0.319 (0.162)**	**0.063 (0.045)**	**0.063 (0.045)**	**0.294 (0.145)**
	GBM	0.528 (0.081)	0.528 (0.081)	2.072 (0.293)	0.501 (0.100)	0.501 (0.100)	1.976 (0.336)	0.493 (0.064)	0.493 (0.064)	1.923 (0.241)
	CBGPS	0.585 (0.270)	0.585 (0.270)	4.611 (5.679)	0.436 (0.295)	0.436 (0.295)	2.683 (1.980)	0.413 (0.184)	0.413 (0.184)	2.472 (1.160)
	npCBGPS	0.153 (0.183)	0.153 (0.183)	1.102 (0.777)	0.226 (0.189)	0.226 (0.189)	1.556 (0.976)	0.258 (0.197)	0.258 (0.197)	2.070 (1.155)
	Ours	0.193 (0.092)	0.193 (0.092)	1.115 (0.360)	0.138 (0.079)	0.138 (0.079)	1.021 (0.376)	0.122 (0.082)	0.122 (0.082)	0.864 (0.410)
nonlinear Y-X, nonlinear Y-T	OLS	1.504 (0.135)	1.665 (0.149)	6.115 (0.580)	1.548 (0.077)	1.711 (0.097)	6.263 (0.356)	1.549 (0.041)	1.706 (0.050)	6.232 (0.172)
	IPW$_{unstable}$	1.036 (0.302)	1.139 (0.295)	4.461 (1.184)	0.876 (0.223)	0.955 (0.200)	3.990 (0.981)	0.803 (0.310)	0.864 (0.317)	3.736 (1.018)
	IPW$_{stable}$	0.726 (0.348)	0.801 (0.383)	3.331 (1.452)	0.677 (0.306)	0.748 (0.313)	3.391 (1.077)	0.674 (0.297)	0.756 (0.344)	3.088 (1.043)
	ISMW	1.386 (0.184)	1.614 (0.265)	5.710 (0.828)	1.442 (0.143)	1.687 (0.215)	5.922 (0.652)	1.405 (0.070)	1.632 (0.111)	5.745 (0.318)
	GBM	1.133 (0.129)	1.268 (0.173)	4.615 (0.510)	1.053 (0.205)	1.182 (0.243)	4.349 (0.745)	1.047 (0.153)	1.176 (0.194)	4.265 (0.602)
	CBGPS	2.400 (2.693)	2.864 (3.432)	9.912 (10.317)	1.265 (0.854)	1.520 (1.068)	6.029 (3.777)	1.268 (0.513)	1.543 (0.665)	5.603 (2.303)
	npCBGPS	0.514 (0.418)	0.623 (0.525)	2.682 (1.471)	0.909 (0.887)	1.093 (1.121)	3.830 (2.996)	0.942 (0.466)	1.125 (0.607)	4.599 (2.100)
	Ours	**0.405 (0.169)**	**0.461 (0.198)**	**2.549 (0.716)**	**0.336 (0.121)**	**0.373 (0.126)**	**2.154 (0.574)**	**0.254 (0.164)**	**0.287 (0.189)**	**1.824 (0.702)**

The value in bracket refers to corresponding standard deviations of 10 times experiments. The smaller of these metrics, the better. Bold number refers to the best result in means for each metric

Table 5 Results on synthetic datasets with sample size $n = 2000$, dimension $p = 10$, when treatment assignment model is possibly misspecified due to nonlinearity

Setting	Method	$n = 2000, d = 10$		
		BIAS_{MTEF}	RMSE_{MTEF}	RMSE_{ADRF}
linear Y-X, linear Y-T	OLS	0.115(0.035)	0.115(0.035)	0.341(0.102)
	$\text{IPW}_{unstable}$	0.761(0.491)	0.761(0.491)	8.680(7.341)
	IPW_{stable}	0.352(0.236)	0.352(0.236)	1.281(0.585)
	ISMW	0.197(0.035)	0.197(0.035)	1.169(0.232)
	GBM	0.064(0.061)	0.064(0.061)	0.264(0.184)
	CBGPS	0.128(0.096)	0.128(0.096)	0.485(0.321)
	npCBGPS	0.056(0.039)	0.056(0.039)	0.233(0.186)
	Ours	**0.045(0.036)**	**0.045(0.036)**	**0.212(0.112)**
linear Y-X, nonlinear Y-T	OLS	0.211(0.061)	0.245(0.055)	0.780(0.176)
	$\text{IPW}_{unstable}$	1.820(1.244)	1.984(1.305)	6.196(4.170)
	IPW_{stable}	0.615(0.374)	0.741(0.426)	2.214(0.994)
	ISMW	2.135(2.573)	2.210(2.563)	10.367(10.992)
	GBM	0.190(0.087)	0.190(0.087)	0.729(0.264)
	CBGPS	0.387(0.264)	0.486(0.329)	1.452(0.859)
	npCBGPS	0.225(0.139)	0.225(0.139)	0.734(0.472)
	Ours	**0.150(0.063)**	**0.173(0.078)**	**0.572(0.184)**

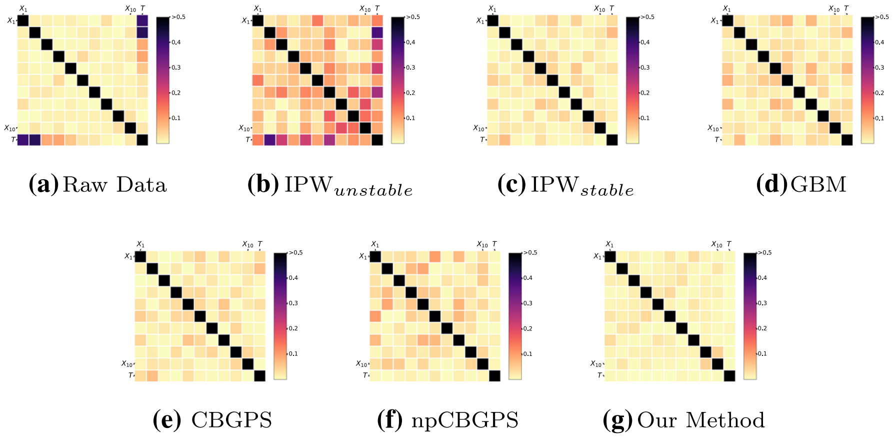
The value in bracket refers to corresponding standard deviations of 10 times experiments. The smaller of these metrics, the better. Bold number refers to the best result in means for each metric

model is misspecified due to nonlinearity, all three methods perform even worse than OLS, as terms in weight estimation are inaccurate.

- GBM uses more flexible modeling method to estimate generalized propensity score. However when the linear relation assumption of confounder and treatment is satisfied, it could only outperform IPW_{stable} in a few settings when dimension of observed variables is relatively low. When misspecification of treatment assignment model occurs, GBM shows better performance due to modelling flexibility of boosting algorithm.
- By directly minimizing the association between treatment and covariates, CBGPS and npCBGPS obtain good performances across all settings. When the dimension of observed variables is relatively low (i.e. $p = 10$), npCBGPS performs worse than CBGPS. When the dimension is relatively high (i.e. $p = 50$), vice versa, as npCBGPS adopts non-parametric solution which could better handle the difficulty of finding feasible weights with increasing dimension. Misspecification of treatment assignment problem also has impact on performance of both methods, as CBGPS and npCBGPS also makes linear assumption of covariates-treatment relation in covariate balancing condition. However with covariate balancing condition, both methods don't heavily rely on accuracy of treatment assignment modelling. Thus CBGPS and npCBGPS largely outperform IPW methods.
- Our algorithm, by directly making treatment become independent with covariates, achieves better performance over the baselines in different settings, especially on MTEF-based metrics. Under setting YT-linear, our algorithm can achieve comparable results with the best baseline, ISMW, and is better than other baselines in means of metrics. Under setting YT-nonlinear where the assumptions in ISMW are violated, our GAD algorithm, a non-parametric method, almost obtains the best performance. As we also adopt flexible modeling of treatment assignment problem by introducing weights obtained under independence measurement, our method also achieves best performance when linear treatment assignment assumption is violated.

5.3.4 Discussion

To explain the improvements we obtain when compared to previous work, we demonstrate the Pearson correlation coefficients (PCC) between treatment T and covariates $\mathbf{X}$ on raw data with confounding bias and weighted data from baselines and our algorithm in Fig. 3, where we do not compare with ISMW, since its weights are matrix form and cannot be applied for PCC calculation. Ideally, T should be independent of $\mathbf{X}$ and their PCC should be $zero$. From Fig. 3, we can find that in the raw data, the treatment T is highly correlated with $\mathbf{X}$, which clearly demonstrate the confounding bias in the observational data. The confounding bias become more serious on the weighted data with $IPW_{unstable}$ since it reweights samples by the inverse of propensity score with high variance. With constraints on the variance of sample weight, IPW_{stable} achieve a better performance on confounding bias removing with smaller value of PCC than $IPW_{unstable}$. By directly minimizing the associations between treatment and covariates, CBGPS and npCBGPS achieve better performances than other baselines and can approximately remove the confounding bias. Aiming to make treatment become



(a) Raw Data **(b)** IPW$_{unstable}$ **(c)** IPW$_{stable}$ **(d)** GBM

(e) CBGPS **(f)** npCBGPS **(g)** Our Method

Fig. 3 Visualization of Absolute Pearson Correlation Coefficient among variables with setting $n = 5000$, $p = 10$, YX-linear and YT-nonlinear. **a** On raw data; **b** on data weighted by $IPW_{unstable}$; **c** on data weighted by IPW_{stable}; **d** on data weighted by GBM; **e** on data weighted by CBGPS; **f** on data weighted by npCBGPS; **g** on data weighted by our GAD algorithm. The higher correlation between treatment T and covariates $\mathbf{X}$ approximately refers to more confounding bias in data

independent of covariates, our GAD algorithm obtains the best performance. This is the main reason that our algorithm can achieve accurate estimation of causal effect on continuous treatment.

5.4 Real-world data: *TWINS*

Considering that few real-world datasets with continuous treatment contain ground-truth of causal effect. As in most cases the major problem is to tackle relation between confounder and treatment, an alternative evaluation method is to carry out semi-simulation with covariates and treatment from real-world datasets while outcome is generated, like previous work on continuous treatment (Kallus and Zhou 2018). We perform a semi-simulation on *TWINS*, a dataset previously used in binary or categorical treatment research for evaluation.

5.4.1 Dataset

TWINS is a dataset commonly used in binary treatment research (e.g. Flores and Flores-Lagunes 2009; Louizos et al. 2017; Liu et al. 2018), which contains data of over 70,000 twins in total. The treatment of this dataset is to be the light one or not when born. Originally, the treatment is generated from a continuous variable, *born weight*. The dataset also includes 50 covariates recording information of parents, which are almost the same for a pair of twins.

To conduct semi-simulation on *TWINS* dataset, we first filter dataset by limiting weight under 2 kilogram. Data of 4,821 pairs of twins are left for further experiments. We set the difference between born weight with 2 kilogram as treatment T in our experiment. To ensuring the ground-truth, we propose to semi-simulate the outcome

variable Y from treatment and covariates to represent the risk of death after born. We reorganize a few columns of covariates according to twins identity, such as birth order. Also, we concatenate original binary treatment to covariates. From observation of dataset, as weight difference increases, death rate over dataset population also increases. Thus, we can generate outcome as follows with different settings of Y-T relations:
YT-linear:

$$Y = 4 \cdot T - 40 + \mathbf{X}\gamma + \epsilon$$

YT-nonlinear:

$$Y = 0.15 \cdot T^2 + T - 20 + \mathbf{X}\gamma + \epsilon$$

where $\gamma \in \mathbb{R}^{p \times 1}$ and $\gamma_i \sim N(0, 0.25)$, $\epsilon \sim N(0, 2.25)$. Then we can get the ground-truth ADRF and MTEF as
YT-linear:

$$ADRF(T) = 4T - 40 + \mathbb{E}(\mathbf{X}_{i,\cdot}\gamma) \text{ and } MTEF = 4$$

YT-nonlinear:

$$ADRF(T) = 0.15 \cdot T^2 + T - 20 + \mathbb{E}(\mathbf{X}_{i,\cdot}\gamma)$$
$$\text{and } MTEF = 0.3 \cdot T + 1$$

5.4.2 Results and analyses

We report the results in Table 6. Though we can only carry out semi-simulation on real dataset, the hidden T-X relation is still a major challenge to tackle for methods based on generalized propensity score or other methods requiring a T-model. Thus, $IPW_{unstable}$ and IPW_{stable} perform worse on causal effect estimation on continuous treatment due to possible misspecified T-model and inaccuracy estimation on generalized propensity score as demonstrated in Table 6. Contrast to the results on synthetic data, under the setting with YT-linear, *ISMW* doesn't achieve the best performance among baselines due to possible misspecification of treatment assignment model, even though its assumption on linear YT relation is satisfied. Benefitting from modelling flexibility of boosting, *GBM* shows slight better performance than methods mentioned above. However, it still relies on accuracy of treatment assignment modelling. With further constraints on covariate balancing, performances of *CBGPS* and *npCBGPS* show great capability of handling complex treatment assignment model. By directly making treatment independent of covariates, our method achieves better result than other methods, since our method is non-parametric and can theoretically guarantee the de-confounding between treatment and covariates.

Table 6 Results on TWINS dataset

Setting	Method	*TWINS*		
		BIAS$_{MTEF}$	RMSE$_{MTEF}$	RMSE$_{ADRF}$
linear Y-X, linear Y-T	OLS	0.125 (0.082)	0.125 (0.082)	0.569 (0.371)
	IPW$_{unstable}$	0.112 (0.097)	0.112 (0.097)	0.575 (0.411)
	IPW$_{stable}$	0.140 (0.130)	0.140 (0.130)	0.725 (0.527)
	ISMW	0.166 (0.142)	0.166 (0.142)	0.811 (0.686)
	GBM	0.116 (0.083)	0.116 (0.083)	0.553 (0.353)
	CBGPS	0.043 (0.040)	0.043 (0.040)	0.620 (0.378)
	npCBGPS	0.041 (0.025)	0.041 (0.025)	0.265 (0.133)
	Ours	**0.022 (0.017)**	**0.022 (0.017)**	**0.149 (0.068)**
linear Y-X, nonlinear Y-T	OLS	0.208 (0.079)	0.236 (0.089)	0.686 (0.350)
	IPW$_{unstable}$	0.227 (0.127)	0.258 (0.144)	0.794 (0.480)
	IPW$_{stable}$	0.232 (0.129)	0.264 (0.146)	0.821 (0.489)
	ISMW	0.295 (0.143)	0.336 (0.162)	1.180 (0.509)
	GBM	0.207 (0.149)	0.231 (0.169)	0.737 (0.371)
	CBGPS	0.187 (0.137)	0.216 (0.158)	0.683 (0.380)
	npCBGPS	0.079 (0.032)	0.095 (0.040)	0.350 (0.140)
	Ours	**0.065 (0.039)**	**0.076 (0.048)**	**0.248 (0.120)**

Bold number refers to the best result in means for each metric

6 Conclusion

In this paper, we focus on the problem of causal effect estimation on continuous treatment in observational studies. We argue that traditional methods for continuous treatment effect estimation are basically regression model based, hence, their performances entirely rely on correctly specified models or some impractical assumptions. Hence, we propose a non-parametric method, Generative Adversarial De-confoudning (GAD) algorithm to remove the confounding bias between treatment and covariates for precisely estimation on continuous treatment effect. In our GAD algorithm, we propose a Generative Adversarial Network based de-confounding algorithm to generate sample weights for making treatment and covariates independent of each other. We prove that the learned sample weight from our GAD algorithm can fully remove the confounding bias with both theoretical analysis and empirical experiments. The experimental results on both synthetic and real world datasets show that our GAD algorithm outperforms the baselines for causal effect estimation on continuous treatment in observational studies.

Acknowledgements This work was supported in part by National Natural Science Foundation of China (Nos. 61625107, 62006207), National Key Research and Development Program of China (Nos. 2018AAA0101900, 2020YFC0832500), the Fundamental Research Funds for the Central Universities and Zhejiang Province Natural Science Foundation (No. LQ21F020020).

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

Arjovsky M, Chintala S, Bottou L (2017) Wasserstein generative adversarial networks. In: Proceedings of the 34th international conference on machine learning, PMLR, proceedings of machine learning research, vol 70, pp 214–223

Athey S, Imbens G (2016) Recursive partitioning for heterogeneous causal effects. Proc Natl Acad Sci 113:7353–7360

Athey S, Imbens GW, Wager S (2018) Approximate residual balancing: debiased inference of average treatment effects in high dimensions. J R Stat Soc: Ser B (Stat Methodol) 80(4):597–623

Austin PC (2011) An introduction to propensity score methods for reducing the effects of confounding in observational studies. Multivar Behav Res 46(3):399–424

Bang H, Robins JM (2005) Doubly robust estimation in missing data and causal inference models. Biometrics 61(4):962–973

Chan D, Ge R, Gershony O, Hesterberg T, Lambert D (2010) Evaluating online ad campaigns in a pipeline: causal models at scale. In: Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining, pp 7–16

Chan KG, Yam SC, Zhang Z (2016) Globally efficient non-parametric inference of average treatment effects by empirical balancing calibration weighting. J R Stat Soc Ser B Stat Methodol 78(3):673–700

Chernozhukov V, Chetverikov D, Demirer M, Duflo E, Hansen C et al (2016) Double machine learning for treatment and causal parameters. arXiv preprint arXiv:1608.00060

Duchi J, Namkoong H (2018) Learning models with uniform performance via distributionally robust optimization. arXiv preprint arXiv:1810.08750

Egel D, Graham BS, de Xavier Pinto CC (2008) Inverse probability tilting for moment condition models with missing data. Single equation models eJournal, Econometrics

Fan J, Imai K, Liu H, Ning Y, Yang X (2016) Improving covariate balancing propensity score: a doubly robust and efficient approach. Technical report

Flores CA, Flores-Lagunes A (2009) Identification and estimation of causal mechanisms and net effects of a treatment under unconfoundedness. IZA Institute of Labor Economics Discussion Paper Series

Fong C, Hazlett C, Imai K et al (2018) Covariate balancing propensity score for a continuous treatment: application to the efficacy of political advertisements. Ann Appl Stat 12(1):156–177

Galagate D (2016) Causal inference with a continuous treatment and outcome: alternative estimators for parametric dose-response functions with applications. Ph.D. thesis

Galvao AF, Wang L (2015) Uniformly semiparametric efficient estimation of treatment effects with a continuous treatment. J Am Stat Assoc 110(512):1528–1542

Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014) Generative adversarial nets. In: Advances in neural information processing systems, pp 2672–2680

Hainmueller J (2012) Entropy balancing for causal effects: a multivariate reweighting method to produce balanced samples in observational studies. Polit Anal 20(1):25–46

Hill JL (2011) Bayesian nonparametric modeling for causal inference. J Comput Graph Stat 20(1):217–240

Hirano K, Imbens GW (2004) The propensity score with continuous treatments. Applied Bayesian modeling and causal inference from incomplete-data perspectives 226164:73–84

Holland PW (1986) Statistics and causal inference. J Am Stat Assoc 81(396):945–960

Imai K, Ratkovic M (2014) Covariate balancing propensity score. J R Stat Soc: Ser B (Stat Methodol) 76(1):243–263

Imai K, Van Dyk DA (2004) Causal inference with general treatment regimes: generalizing the propensity score. J Am Stat Assoc 99(467):854–866

Imbens GW (2004) Nonparametric estimation of average treatment effects under exogeneity: a review. Rev Econ Stat 86(1):4–29

Imbens GW, Rubin DB (2015) Causal inference in statistics, social, and biomedical sciences. Cambridge University Press, Cambridge

Kallus N (2019) Generalized optimal matching methods for causal inference. J Mach Learn Res (forthcoming)

Kallus N, Santacatterina M (2019) Kernel optimal orthogonality weighting: a balancing approach to estimating effects of continuous treatments. arXiv, Methodology

Kallus N, Zhou A (2018) Policy evaluation and optimization with continuous treatments. In: International conference on artificial intelligence and statistics, pp 1243–1251

Kennedy EH, Ma Z, McHugh MD, Small DS (2017) Non-parametric methods for doubly robust estimation of continuous treatment effects. J R Stat Soc: Ser B (Stat Methodol) 79(4):1229–1245

Kohavi R, Longbotham R (2011) Unexpected results in online controlled experiments. ACM SIGKDD Explor Newsl 12(2):31–35

Kreif N, Grieve R, Díaz I, Harrison D (2015) Evaluation of the effect of a continuous treatment: a machine learning approach with an application to treatment for traumatic brain injury. Health Econ 24(9):1213–1228

Kuang K, Cui P, Li B, Jiang M, Yang S (2017) Estimating treatment effect in the wild via differentiated confounder balancing. In: Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining, pp 265–274. ACM

Kuang K, Cui P, Athey S, Xiong R, Li B (2018) Stable prediction across unknown environments. In: Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining, pp 1617–1626

Kuang K, Cui P, Li B, Jiang M, Wang Y, Wu F, Yang S (2019) Treatment effect estimation via differentiated confounder balancing and regression. ACM Trans Knowl Discov Data (TKDD) 14(1):1–25

Kuang K, Cui P, Zou H, Li B, Tao J, Wu F, Yang S (2020) Data-driven variable decomposition for treatment effect estimation. IEEE Trans Knowl Data Eng. https://doi.org/10.1109/TKDE.2020.3006898

Kuang K, Li L, Geng Z, Xu L, Zhang K, Liao B, Huang H, Ding P, Miao W, Jiang Z (2020b) Causal inference. Engineering 6(3):253–263

Künzel SR, Sekhon JS, Bickel PJ, Yu B (2019) Metalearners for estimating heterogeneous treatment effects using machine learning. Proc Natl Acad Sci 116(10):4156–4165

Li F, Li L, Yin J, Zhang Y, Zhou Q, Kuang K (2020a) How to interpret machine knowledge. Engineering 6(3):218–220

Li M, Kuang K, Zhu Q, Chen X, Guo Q, Wu F (2020b) IB-M: a flexible framework to align an interpretable model and a black-box model. In: 2020 IEEE international conference on bioinformatics and biomedicine (BIBM), pp 643–649. IEEE

Liu J, Ma Y, Wang L (2018) An alternative robust estimator of average treatment effect in causal inference. Biometrics 74(3):910–923

Liu Y, Dieng A, Roy S, Rudin C, Volfovsky A (2019) Interpretable almost matching exactly for causal inference. AISTATS

Louizos C, Shalit U, Mooij J, Sontag D, Zemel R, Welling M (2017) Causal effect inference with deep latent-variable models. In: Proceedings of the 31st annual conference on neural information processing systems

Lu C, Wang S (2020) The general-purpose intelligent agent. Engineering 6(3):221–226

McCaffrey DF, Ridgeway G, Morral AR (2004) Propensity score estimation with boosted regression for evaluating causal effects in observational studies. Psychol Methods 9(4):403

Neugebauer R, van der Laan M (2007) Nonparametric causal effects based on marginal structural models. J Stat Plan Inference 137(2):419–434

Olaya D, Coussement K, Verbeke W (2020) A survey and benchmarking study of multitreatment uplift modeling. Data Min Knowl Disc 34:273–308

Pearl J (2009) Causality. Cambridge University Press, Cambridge

Ren K, Zheng T, Qin Z, Liu X (2020) Adversarial attacks and defenses in deep learning. Engineering 6(3):346–360

Robins J, Rotnitzky A (2001) Comment on inference for semiparametric models: some questions and an answer, by P.J. Bickel and J. Kwon. Stat Sin 11:920–936

Robins JM, Hernan MA, Brumback B (2000) Marginal structural models and causal inference in epidemiology

Rojas-Carulla M, Schölkopf B, Turner R, Peters J (2018) Invariant models for causal transfer learning. J Mach Learn Res 19(1):1309–1342

Rong G, Mendez A, Assi EB, Zhao B, Sawan M (2020) Artificial intelligence in healthcare: review and prediction case studies. Engineering 6(3):291–301

Rosenbaum PR, Rubin DB (1983) The central role of the propensity score in observational studies for causal effects. Biometrika 70(1):41–55

Rubin DB (1974) Estimating causal effects of treatments in randomized and nonrandomized studies. J Educ Psychol 66(5):688

Rudas K, Jaroszewicz S (2018) Linear regression for uplift modeling. Data Min Knowl Disc 32:1275–1305

Rudin C (2019) Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nat Mach Intell 1(5):206–215

Schölkopf B, Locatello F, Bauer S, Ke NR, Kalchbrenner N, Goyal A, Bengio Y (2021) Toward causal representation learning. Proc IEEE 109(5):612–634

Soltys M, Jaroszewicz S, Rzepakowski P (2014) Ensemble methods for uplift modeling. Data Min Knowl Disc 29:1531–1559

Tan Z (2010) Bounded, efficient and doubly robust estimation with inverse weighting. Biometrika 97:661–682

Tian Q, Kuang K, Jiang K, Wu F, Wang Y (2021) Analysis and applications of class-wise robustness in adversarial training. arXiv preprint arXiv:2105.14240

Wager S, Athey S (2015) Estimation and inference of heterogeneous treatment effects using random forests. J Am Stat Assoc 113:1228–1242

Westreich D, Lessler J, Funk MJ (2010) Propensity score estimation: neural networks, support vector machines, decision trees (cart), and meta-classifiers as alternatives to logistic regression. J Clin Epidemiol 63(8):826–833

Zhao Q (2016) Covariate balancing propensity score by tailored loss functions. arXiv, Methodology

Zhu Y, Coffman D, Ghosh D (2015) A boosting algorithm for estimating generalized propensity scores with continuous treatments. J Causal Inference 3:25–40

Zou WY, Shyam S, Mui M, Wang M, Pedersen J, Ghahramani Z (2020) Learning continuous treatment policy and bipartite embeddings for matching with heterogeneous causal effects. arXiv:2004.09703

Zubizarreta J (2015) Stable weights that balance covariates for estimation with incomplete outcome data. J Am Stat Assoc 110:910–922

Žliobaitė I (2017) Measuring discrimination in algorithmic decision making. Data Min Knowl Disc 31:1060–1089

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

Kun Kuang[1] · Yunzhe Li[1] · Bo Li[2] · Peng Cui[2] · Hongxia Yang[3] · Jianrong Tao[4] · Fei Wu[1]

✉ Kun Kuang
kunkuang@zju.edu.cn

Yunzhe Li
illusivemanlyz@gmail.com

Bo Li
libo@sem.tsinghua.edu.cn

Peng Cui
cuip@tsinghua.edu.cn

Hongxia Yang
yang.yhx@alibaba-inc.com

Jianrong Tao
hztaojianrong@corp.netease.com

Fei Wu
wufei@cs.zju.edu.cn

[1] Department of Computer Science and Technology, Zhejiang University, Hangzhou, Zhejiang Province, China

[2] Tsinghua University, Beijing, China

[3] Alibaba Group, Hangzhou, Zhejiang Province, China

[4] NetEase Fuxi AI Lab, Hangzhou, Zhejiang Province, China