Networked Instrumental Variable for Treatment Effect Estimation With Unobserved Confounders

Ziyu Zhao, Anpeng Wu, Kun Kuang, Ruoxuan Xiong, Bo Li, Zhihua Wang, and Fei Wu, *Senior Member, IEEE*

***Abstract*—Treatment effect estimation from observational data is a fundamental problem in causal inference, and its critical challenge is to address the confounding bias arising from the confounders. The effectiveness of the conventional methods proposed to solve this problem depends on the unconfoundedness assumption. In practice, however, the unconfoundedness assumption is frequently violated since we cannot guarantee that all the confounders are measured. To this end, recent studies suggest using auxiliary network architectures to mine information about unmeasured confounders in the data to relax this assumption. However, these methods cannot address the confounding bias from unmeasured confounders unrelated to the network information. Inspired by the insight that some neighboring features that influence one's treatment choice (e.g., which movie to watch) but do not affect the outcome (e.g., assessment of the movie) can be treated as instrumental variables (IVs), we propose a novel Network Instrumental Variable Regression (NetIV) framework exploits IV information from neighborhoods to perform a two-stage regression for treatment effect estimation. Extensive experiments demonstrate that our NetIV method outperforms the state-of-the-art methods for treatment effect estimation in the presence of unmeasured confounders.**

***Index Terms*—Treatment effect estimation, unmeasured confounders, instrumental variable, networked observational data.**

I. Introduction

TREATMENT effect estimation is one of the fundamental problems in causal inference [1], [2], [3] and is crucial for explanatory analysis [4], [5] and decision-making [6], [7]. The golden standard approach for treatment effect estimation is to conduct randomized controlled trials (RCT), where the treatments are randomly assigned to units. However, fully randomized controlled trials are always expensive, time-consuming, and even unethical [8], [9]. Therefore, more and more attention has been focused on estimating the treatment effect from the observational data.

In observational studies, a significant challenge in estimating treatment effects is the presence of confounders that influence both the treatment and the outcome. This overlap can lead to confounding bias, where the estimated treatment effect is skewed, reflecting a mixture of the true effect of the treatment and the influence of the confounders. Such bias can result in incorrect conclusions about the relationship between the treatment and the outcome, potentially causing misestimations of the treatment's true effect. For example, consider a study examining the efficacy of a new drug intended to reduce the incidence of heart disease. If the group receiving the drug consists predominantly of younger, more health-conscious individuals compared to those not receiving the drug, then factors like age and health consciousness act as confounders. Younger individuals naturally exhibit a lower risk of heart disease, and health-conscious behaviors (e.g., regular exercise, and healthy eating) further decrease risk. Consequently, any observed decline in heart disease within the drug-treated group could be misleadingly attributed to the drug, when it may be largely or entirely due to these confounders. This type of bias can lead researchers to overestimate the drug's effectiveness. Therefore, addressing confounding bias is essential to ensure that the estimated causal effects accurately reflect the influence of the treatment, rather than the influence of confounders [10], [11].

Traditional approaches for addressing confounding bias employ propensity scores through unit matching, stratification, and sample reweighting to reduce confounding bias in observational data [12], [13]. Recently, machine learning methods with deep neural networks have been applied for treatment effect estimation [14], [15], [16]. Although these methods are gaining ground in practical application, their validity relies on the *unconfoundedness assumption* that *all confounders are measurable and observed in the data*, i.e., no unobserved confounders. However, this is unverifiable and likely to be unrealistic in practice. Therefore, estimating the treatment effect in the presence of *unobserved confounders* is inevitable [17].

To address this problem, recently, many methods [18], [19], [20] have been proposed to find a proxy for the unobserved confounders by leveraging the *networked observational data*. Network information is ubiquitous in various observational data [21], [22], such as social networks, and can be used to identify patterns of unobserved confounders. [18] proposed a network deconfounder using graph neural networks to learn

Received 18 November 2022; revised 26 June 2024; accepted 19 October 2024. Date of publication 11 December 2024; date of current version 12 January 2025. This work was supported in part by the National Natural Science Foundation of China under Grant 62376243 and under Grant 62441605, in part by the Starry Night Science Fund at Shanghai Institute for Advanced Study (Zhejiang University), and in part by the Ant Group. Recommended for acceptance by Z. Wang. *(Corresponding author: Kun Kuang.)*

Ziyu Zhao, Anpeng Wu, Kun Kuang, Zhihua Wang, and Fei Wu are with the College of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China (e-mail: benzhao.styx@gmail.com; anpwu@zju.edu.cn; kunkuang@zju.edu.cn; zhihua.wang@zju.edu.cn; wufei@cs.zju.edu.cn).

Ruoxuan Xiong is with the Department of Quantitative Theory & Methods, Emory University, Atlanta, GA 30322 USA (e-mail: ruoxuan.xiong@emory.edu).

Bo Li is with the School of Economics and Management, Tsinghua University, Beijing 100190, China (e-mail: libo@sem.tsinghua.edu.cn).

Digital Object Identifier 10.1109/TKDE.2024.3491776

representations that approximate unobserved confounders and estimate treatment effects through counterfactual regression. [19] further considered the imbalance in the network structure and proposed a Graph Infomax Adversarial Learning model to capture the information from imbalanced networked data. [20] proposed CNE using an unsupervised embedding model to learn a proxy for unobserved confounders. The validity of these methods requires that the learned representation or embedding can block all backdoor paths of unobserved confounders, which is not verifiable and unrealistic in real-world applications since numerous unmeasured confounders cannot be approximated and represented from network information. For example, personal attributes unrelated to the network generation mechanism may not be observed but can also act as confounders that hinder treatment effect estimation. Hence, accurate estimation of treatment effects with unobserved confounders from networked observational data remains an open problem.

The Instrumental Variable (IV) regression method is a classical approach that addresses the problem of unobserved confounding through a two-stage procedure. IVs [23] are exogenous variables correlated to the treatment but do not directly affect the outcome, and it is not influenced by unobserved confounders. In the first step of IV regression, the instrumental variable is applied in a regression to estimate the treatment. Since the IV is independent with the unobserved confounders, the estimated values from this first-stage regression remain independent of these confounders. Subsequently, these estimated values are used to assess the impact of the treatment on the outcome. This two-stage approach effectively sidesteps the distortion caused by unobserved confounders, facilitating a more accurate elucidation of the causal relationship. Two-stage least squares (2SLS) [24] is a conventional method for conducting IV regression. In the first stage, this approach performs a linear regression of the treatment on the instrumental variable. In the second stage, it predicts the outcome by regressing on the conditional expectation of the treatment given the instrumental variable. Many nonlinear IV regression variants [25], [26], [27] have recently been proposed to generalize previous IV-based methods on high-dimensional and nonlinear settings. We extend this line of work to the network setting.

Due to the prevalence of social networks, information exchanged among neighbors can influence an individual's decisions, and this information might not directly impact the individual's outcomes. This aspect of network information can serve as an IV, offering a unique opportunity to address unobserved confounders in observational network data. For instance, the preferences and activities of social media connections might shape an individual's online behavior and consumption choices, yet not necessarily affect their personal opinions or satisfaction with those choices [28]. Similarly, the educational background of a person's peers can sway their decision to pursue further education without directly influencing their own academic achievements [29]. Moreover, the political views of community members can drive an individual's political engagement and voting behavior without directly altering their satisfaction with the election outcomes [30]. These scenarios

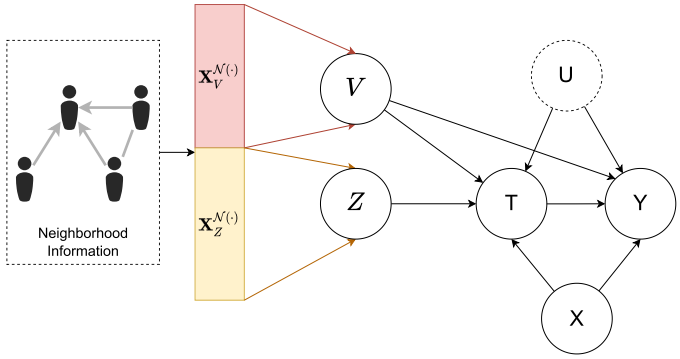


Fig. 1. This causal diagram illustrates the networked data structure. Variables are as follows: X represents the observed confounder, T denotes the treatment, Y signifies the outcome and U indicates the unobserved confounder for each node. Neighboring information or features for each node are divided into two categories: (1) Latent instrumental variables ($\mathbf{X}_Z^{N(\cdot)}$), affecting only the treatment of the current node, summarized as Z; (2) Latent confounders ($\mathbf{X}_V^{N(\cdot)}$), influencing both the treatment and the outcome, summarized as V.

demonstrate that neighboring information can often be viewed as IVs to estimate treatment effects, thereby offering a chance to eliminate unobserved confounding in networked observational data.

Inspired by this insight, we propose a networked instrumental regression (NetIV) framework by exploiting network patterns to leverage the IVs for counterfactual regression. Fig. 1 shows the causal diagram of the proposed NetIV, where T, Y, X, and U denote treatment, outcome, observed confounders, and unobserved confounders, respectively. Here we assume that the neighborhood information contains twofold information: latent confounders (denoted by V) and latent IVs (denoted by Z). We leverage the IV information to remove the confounding bias conducted by both observed and unobserved confounders (V and X are observed, U is unobserved). Specifically, we propose a novel NetIV framework that performs a two-stage regression for treatment effect estimation. In the first stage, using a Graph Neural Network [31], we perform a regression from $\{X, V, Z\}$ to T (i.e., $P(T|X, V, Z)$) [1] to remove the confounding from unobserved confounders; in the second stage, we regress the outcome Y on the re-sampled treatment $\hat{T} \sim P(T|X, V, Z)$, self-feature X, and the neighborhood information $\{V, Z\}$. Furthermore, we incorporate two additional modules, the confounder balancing module, and the information bottleneck module, into the second stage regression to remove the confounding bias from the observed confounders and reduce the variance. We validate the superiority of the proposed algorithm on four networked observational datasets. It is worth noting that our framework is the first to propose combining instrumental variable mining with graph data, providing methods for unbiased treatment effect estimation and identifiability conditions. Our experimental and theoretical results have opened up new directions in this field by considering neighbor information as partially existing IVs, an

[1] It is worth noting that we do not explicitly separate V and Z, but rather learn a joint representation through graph neural networks.

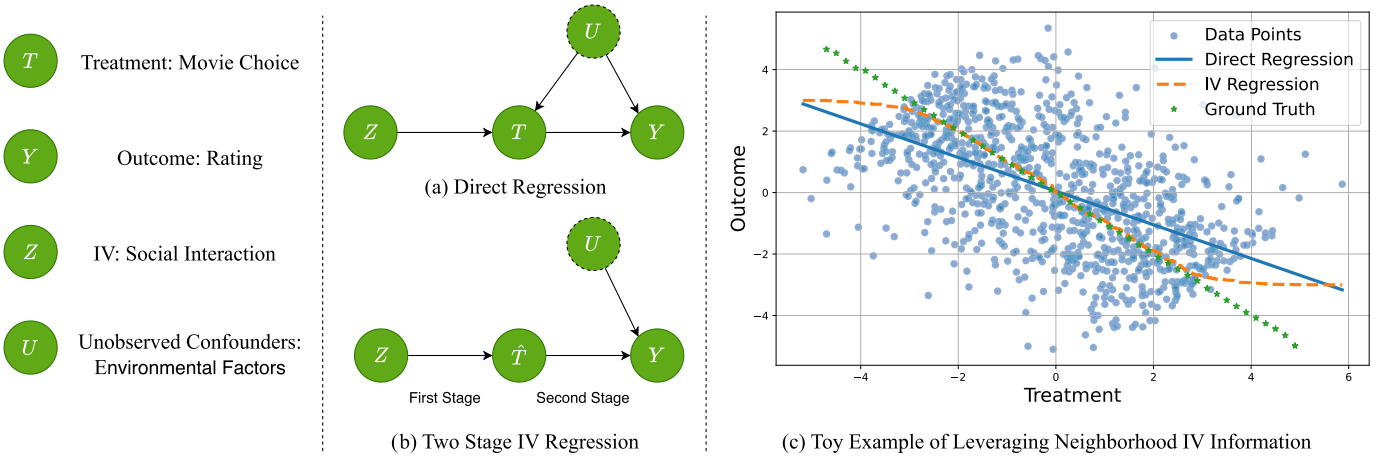


Fig. 2. Toy example on the networked IV regression. The treatment T denotes the movie choice, the outcome Y denotes the movie rating, the instrumental variable (IV) Z denotes social interaction, and the unobserved confounders U denote environmental factors. Figure (a) shows the causal graph of direct regression, where the treatment effect is confounded by the unobserved confounder U. Figure (b) illustrates the two-stage IV regression, which leverages the first stage regressed $\hat{T}$ to estimate the treatment effect, making it unbiased and agnostic to the unobserved confounder. Figure (c) presents the regression results comparing direct regression and IV regression, demonstrating the effectiveness of using networked IV information to resolve unobserved confounders.

aspect overlooked by traditional methods, which often results in biased treatment effect estimation in the presence of unobserved confounds.

To summarize, the main contributions of this paper are:

- We study the problem of estimating treatment effect from networked observational data in the presence of unobserved confounders, where the unobserved confounders cannot be sufficiently recovered from the network information.
- We propose a novel networked instrumental variable regression (NetIV) framework by exploiting neighborhood information and network structure to eliminate the bias from both observed and unobserved confounders.
- We perform extensive experiments to demonstrate the superiority of the proposed algorithm across four semi-synthetic datasets based on real-world social networks.

II. Preliminaries

A. Toy Example on Networked IV for Resolving Unobserved Confounders

Before formally introducing our setup, we will use a simplified toy example to present the background knowledge and explain our motivation. Here we have simplified the causal diagram, where we focus on the effect of treatment (movie choice) on the outcome (rating). Additionally, there are some unobserved confounders (environmental influences) that affect both the treatment and the outcome. In this case, interactions in the social network will influence movie choices, serving as instrumental variables (IV). We constructed a toy dataset according to the following data generation mechanism.

The instrumental variable Z is generated as $Z \sim \text{Unif}(-3, 3)$, and the unobserved confounder U is generated from $U \sim N(0, 1)$. The treatment variable T is generated from $T = Z + U + \epsilon$, where $\epsilon \sim N(0, 0.1)$ represents the random noise. The outcome variable Y is generated from $Y = -T + 2U + \epsilon'$, where $\epsilon' \sim N(0, 0.1)$ is the random noise in the outcome. The direct method is to regress Y on T as shown in Fig. 2(a). However, this approach is vulnerable to the influence of unobserved confounders, which can bias the estimation of the causal effect of T on Y. Fig. 2(b) shows the causal graph of conducting two-stage regression using IV information. In the first stage, T is regressed on Z to obtain $\hat{T}$, which is independent of U, thereby eliminating the influence of unobserved confounders. Consequently, in the second stage, the effect of T on Y can be accurately estimated by regressing Y on $\hat{T}$. Fig. 2(c) shows a comparison of these two regression methods. It can be seen that direct regression leads to biased estimation, which fails to correctly capture the relationship between Y and T. However, using IV information for two-stage regression eliminates the influence of unobserved confounders, resulting in a more accurate regression outcome.

However, in practical scenarios, finding such instrumental variables is very challenging. Therefore, in this paper, we aim to extract instrumental variable information from network data, enabling us to eliminate the influence of unobserved confounders when estimating causal effects.

B. Problem Setup

In this paper, we focus on estimating the Average Treatment Effect (ATE) from networked observational data in the presence of unobserved confounders, which cannot be fully recovered or approximated by the network structure and neighborhood information. In the networked observational data, let $\mathbf{G}(\mathbf{V}, \mathbf{E})$ be an undirected graph, where $\mathbf{V}$ represents n nodes in $\mathbf{G}$ and $\mathbf{E}$ represents a set of edges between nodes. We use $(v_i, v_j) \in \mathbf{E}$ to indicate that there is an edge between node v_i and node v_j. A is the adjacency matrix of $\mathbf{G}$ which satisfies that if $(v_i, v_j) \in \mathbf{E}$, $A_{i,j} = 1$, otherwise $A_{i,j} = 0$. Then the networked observational data can be denoted as $\mathbb{D} = (\{X^i, T^i, Y^i\}_{i=1}^n, A)$. For each unit i, we observe a binary treatment $T^i \in \{0, 1\}$, confounders $X^i \in \mathbb{R}^{m_X}$ and the outcome $Y^i \in \mathbb{R}$. In addition, there are

some unobserved confounders $U^i \in \mathbb{R}^{m_U}$, which are irrelevant to the network structure and cannot be approximated by the neighborhood information.

The main goal of this paper is to estimate the Average Treatment Effect (ATE) with the following definition:

$$ATE = \mathbb{E}\left[Y|do(T=1)\right] - \mathbb{E}\left[Y|do(T=0)\right]. \quad (1)$$

To precisely estimate the ATE, one has to address the biases from both observed and unobserved confounders. With the networked observational data, in this paper, we propose to utilize the network structure and neighborhood information of each unit (i.e., each node in graph $\mathbf{G}$) to estimate the treatment effect. We argue that such information can serve as instrumental variables or latent confounders for treatment effect estimation.

In the networked observational data, for each unit i, we use $N(i)$ to denote its 1-hop neighbors and $\mathbf{X}^{N(i)}$ to denote the observed covariates of its neighboring nodes. Following [32], [33], we assume that each node is only influenced by its 1-hop neighbors. In this paper, we further assume that the neighborhood information can be decomposed into two parts, i.e., $\mathbf{X}^{N(i)} = \{\mathbf{X}_V^{N(i)}, \mathbf{X}_Z^{N(i)}\}$, where $\mathbf{X}_V^{N(i)}$, refers to the neighborhood information that would affect both the treatment and the outcome of unit i, while $\mathbf{X}_Z^{N(i)}$ only affects the treatment of unit i and would not directly affect its outcome. Suppose there is a summary function $\phi(\cdot)$, for each node i, we have:

$$V^i = \phi\left(\mathbf{X}_V^{N(i)}\right), \quad (2)$$

$$Z^i = \phi\left(\mathbf{X}_Z^{N(i)}\right), \quad (3)$$

where V^i acts as a latent confounder and Z^i is a valid IV defined as follows.

Definition 1: An Instrumental Variable Z is an exogenous variable that only affects the outcome through its strong association with the treatment. Besides, a valid instrument variable should satisfy the following three assumptions:

Relevance: Z is a cause of T, i.e., $P(T|Z) \neq P(T)$.

Exclusion: Z has no direct effect on Y, i.e., $P(Y|X,V,Z,U) = P(Y|X,V,U)$.

Unconfounded: Z is independent of all confounders, i.e., $Z \perp X, V, U$.

Similar to [25], we assume that the outcome Y is generated by the treatment T, the observed covariates X, the unobserved confounders U and the neighborhood information V as:

$$Y = f\left(T, X, V\right) + U, \quad (4)$$

where $f(\cdot)$ is an unknown non-linear continuous function. The unobserved confounders U are *additive noise* [34], [35] that satisfies $\mathbb{E}U = 0$ but might be correlated with X and T: $\mathbb{E}[U|T,X,V] \neq 0$.

Notice that a direct regression from $\{T, X, V\}$ to Y will be affected by U and cause a bias since:

$$\begin{aligned}\mathbb{E}[Y|T,X,V] &= \mathbb{E}[f(T,X,V) + U|T,X,V] \\ &= f(T,X,V) + \mathbb{E}[U|T,X,V] \end{aligned} \quad (5)$$

where the problematic term $\mathbb{E}[U|T,X,V]$ varies with T, which impedes identifying the treatment effect.

TABLE I
SYMBOLS AND DEFINITIONS

Symbol	Definition
$\mathbf{G}$	graph
$A, \mathbf{E}, \mathbf{V}$	adjacency matrix, edges and nodes
n	Sample size
X^i, T^i, Y^i	covariates, treatment and outcome of unit i
$\mathbf{X}, \mathbf{T}, \mathbf{Y}$	covariates, treatments and outcomes of all units
U^i	unobserved confounders of unit i
m_X, m_U	the dimension of X and U
$N(i)$	the neighboring nodes of unit i
$\mathbf{X}^{N(i)}$	neighboring nodes' covariates of unit i
$\mathbf{X}_Z^{N(i)}$	neighbors' covariates that contains information of IV
$\mathbf{X}_V^{N(i)}$	neighbors' covariates that contains information of confounder
$\phi(\cdot)$	summary function
Z^i, V^i	summarized IV and latent confounder for unit i
$\hbar$	counterfactual prediction function
r	learned representation
g_θ	graph neural network with parameter θ
f_μ	fully connected layers with parameter μ
π_μ	logistic regression network with parameter μ
$\mathcal{L}$	loss function

One can eliminate the bias from unobserved confounders with the counterfactual prediction function [25], [27], which is defined as:

$$\hbar(T,X,V) \equiv f(T,X,V) + \mathbb{E}[U|X,V], \quad (6)$$

where $\hbar(T,X,V)$ is the expectation of the outcome given $\{T,X,V\}$, and the distribution of U is *constant* as T is changed.

If the counterfactual prediction function $\hbar(\cdot)$ can be precisely modeled, one can identify the treatment effect by:

$$\begin{aligned} ATE &= \mathbb{E}[\hbar(T=1,X,V) - \hbar(T=0,X,V)] \quad (7)\\ &= \mathbb{E}\left[f(T=1,X,V) + \mathbb{E}[U|X,V]\right] \\ &\quad -\mathbb{E}\left[f(T=0,X,V) + \mathbb{E}[U|X,V]\right] \\ &= \mathbb{E}[f(T=1,X,V) - f(T=0,X,V)] \\ &= \mathbb{E}[f(T=1,X,V)] - \mathbb{E}[f(T=0,X,V)]. \quad (8)\end{aligned}$$

Equation (8) holds since the term $\mathbb{E}[U|X,V]$ is a constant as T changes.

To fully resolve the bias from unobserved confounders, in this paper, we propose to leverage the neighborhood IV information and employ a graph version of two-stage regression to precisely model the counterfactual prediction function. Table I summarizes the main notations and their definitions.

III. METHODOLOGY

A. Networked Instrumental Variable Regression Framework

Inspired by the idea that neighborhood information can act as valid IVs for a unit, we propose a novel networked instrumental variable regression (NetIV) framework in this section.

Taking the expectation of (4) conditional on $\{X,V,Z\}$, where $\{V,Z\}$ are obtained by aggregating the neighborhood information, we further establish the relationship [25], [27], [34]:

$$\begin{aligned}\mathbb{E}[Y|X,V,Z] &= \mathbb{E}[f(T,X,V) + U|X,V,Z] \\ &= \mathbb{E}[f(T,X,V)|X,V,Z] + \mathbb{E}[U|X,V]\end{aligned}$$

$$= \int (f(T,X,V) + \mathbb{E}[U|X,V])dP(T=t|X,V,Z)$$
$$= \int \hbar(T=t,X,V)dP(T=t|X,V,Z), \quad (9)$$

where $P(T|X,V,Z)$ is the conditional treatment distribution and $\hbar(T,X,V)$ is the counterfactual prediction function. Equation (9) defines an inverse problem of $\hbar(T,X,V)$ in terms of $\mathbb{E}[Y|X,V,Z]$ and $P(T|X,V,Z)$ [25], [27], [34], and leads to a two-stage approach to estimate the prediction function $\hbar(T,X,V)$, as follows.

Treatment Regression Stage: By plugging the proxies $\{V,Z\}$ into two-stage regression (9), one can directly use $\{V,Z\}$ from the neighbors and self-features X to obtain the treatment distribution $\hat{P}(T|X,V,Z)$.

Outcome Regression Stage: With the re-sampled treatment $\hat{T} \sim \hat{P}(T|X,V,Z)$, we regress the expectation of outcome $\mathbb{E}[Y|\hat{T},X,V]$ to obtain the counterfactual prediction function $\hbar(\hat{T},X,V)$.

B. Algorithm and Optimization

This section elaborates on the proposed Networked Instrumental Variable regression framework. We first introduce the graph representation learning module in Section III-B1, which is used in both treatment and outcome regression stages. Then we discuss the treatment regression stage in Section III-B2 and the outcome regression stage in Section III-B3. The entire framework is shown in Fig. 1.

1) Representation Learning: We formulate the representation learning function as $\mathbf{X} \times A \to R$, $\mathbf{X} \in \mathbb{R}^{n \times m_X}$ and $R \in \mathbb{R}^d$. Graph Convolutional Networks (GCN) [31], [36], [37] is used to capture the neighborhood information in networks. Each layer of the GCN model can be parameterized with the following layer-wise propagation rule:

$$h_{l+1} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}h_l W_l\right) \quad (10)$$

where $\tilde{A} = A + I_n$ is the adjacency matrix of the graph $\mathbf{G}$, $\tilde{D}$ denotes the degree matrix of the graph, W_l is a trainable layer-specific weight matrix and $\sigma(\cdot)$ is the activation function. By stacking such layers, we can approximate the function g: $\mathbf{X} \times A \to R$.

We first learn a joint representation r using a representation learning function g: $\mathbf{X} \times A \to R$, $R \in \mathbb{R}^d$ to aggregate the information of Z and V from the neighboring nodes' covariates $\mathbf{X}^{N(\cdot)}$ and the self-feature X of current node:

$$r_\theta \equiv g_\theta(A, \mathbf{X}), \quad (11)$$

where θ is a learnable parameter of GCN g_θ, we use r_θ to denote the representation containing the information of $\{X,V,Z\}$. It is worth noting that we do not explicitly distinguish the information from V and Z in the representation, but learn a representation r_θ that aggregates all the information through a graph neural network.

2) Treatment Regression Network: In the treatment regression stage, we plug the neighborhood information $\{V,Z\}$ to estimate the treatment distribution $P(T|X,V,Z)$.

With the learned representation r_{θ_0} obtained from the representation learning module, we learn the treatment distribution $\hat{P}(T|X,V,Z)$ conditional on $\{X,V,Z\}$ by a logistic regression network π_{μ_0}, as shown in Fig. 3(a):

$$\mathcal{L}_t = CrossEntropy\left(\pi_{\mu_0}\left(r_{\theta_0}\right), T\right) \quad (12)$$

where $\pi_{\mu_0}(r_{\theta_0}) = \hat{P}(T=1|X,V,Z)$, μ_0 and θ_0 are learnable parameters.

3) Outcome Regression Network: In the outcome regression stage, with the learned treatment distribution $\hat{P}(T|X,V,Z)$, and the learned representation r_{θ_1}, we propose to regress the outcomes with a deep neural network. The outcome regression network contains mainly three components.

Regressing the outcome Y: With the learned representation r_{θ_1}, which contains both information of self-feature and the neighborhood information, we regress the outcome by an outcome regression network f_{μ_1} that maps from the representation r_{θ_1} and the re-sampled treatment $\hat{T} \sim \hat{P}(T|X,V,Z)$ into the outcome Y, as follows:

$$\mathcal{L}_y = \frac{1}{n}\sum_{i=1}^{n}\left(Y^i - \sum_{t\in\{0,1\}} f_{\mu_1}\left(t^i, r_{\theta_1}^i\right)\hat{P}\left(t^i|X^i,V^i,Z^i\right)\right)^2 \quad (13)$$

where f_{μ_1} is a neural network parameterized with μ_1. Note that f_{μ_1} is a two-head neural network that estimates $Y(T=1,r_{\theta_1})$ and $Y(T=0,r_{\theta_1})$ separately.

Information Bottleneck: It is worth noting that by means of a graph neural network, the learned representation r_{θ_1} does not implicitly distinguish between IVs Z and latent confounders V. Due to the exclusion assumption of IV, we have $P(Y|X,V,Z,U) = P(Y|X,V,U)$, the IVs Z carry no extra information for outcome prediction. Therefore, introducing IVs Z into the outcome regression stage does not lead to bias, but increases the variance and degrades the performance of the treatment effect estimation [38]. To solve this problem, we introduce the information bottleneck [39], [40] to compress the information provided by r_{θ_1} for Y.

The original version of the graph information bottleneck can be formulated as:

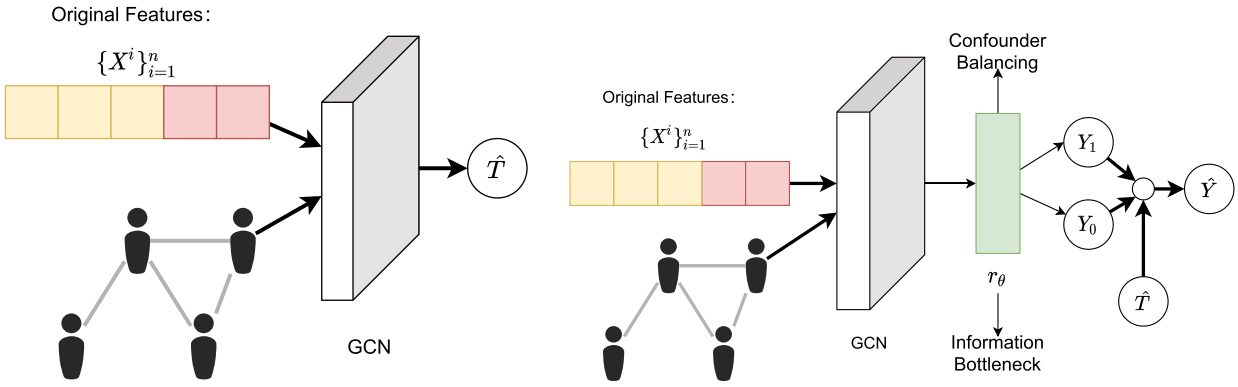
$$\min_{\mathbb{P}(r_{\theta_1}|\mathbf{X},A)} -I\left(Y;r_{\theta_1}\right) + \beta I\left(X,\mathbf{X}^{N(\cdot)};r_{\theta_1}\right) \quad (14)$$

the first term ensures that the learned representation r_{θ_1} contain sufficient information to predict the outcome Y, which can be ensured by (13), and the second term constrains the mutual information between the representation and the input data, which compresses the information in the representation r_{θ_1} to resolve the influence of unnecessary variables Z for predicting Y.

We leverage the Hilbert-Schmidt independence criterion (HSIC) to replace the mutual information term for the concern of time complexity [39]. The HSIC is defined as:

$$\mathrm{HSIC}(\mu,\nu) = (N-1)^{-2}\,\mathrm{tr}\left(\mathrm{K}^{\mu}\mathrm{H}\mathrm{K}^{\nu}\mathrm{H}\right) \quad (15)$$

where K^{μ} and K^{ν} are the kernel matrices of μ and ν, H is the centering matrix $\mathrm{H} = \mathbf{I}_m - \frac{1}{m}\mathbf{1}_m\mathbf{1}_m^T$. In the practical implementation, we restrict the HSIC between representation



(a) Treatment Regression model. (b) Outcome Regression Model.

Fig. 3. The overall architecture of our model. In the first stage, the model estimates the probability of receiving treatment given $\mathbf{X}$ and the adjacency matrix A. In the second stage, if we have some prior knowledge to separate the neighboring IV information, the model estimates the expectation of outcome given V, X and the estimated $\hat{T}$ in the first stage; If there is no prior knowledge about IV, we can estimate the outcome directly from all the information of neighboring nodes.

r_{θ_1} and the original features $\{X, \mathbf{X}^{N(\cdot)}\}$. The constraint can be formulated as follows:

$$\mathcal{L}_{IB} = \text{HSIC}\left(r_{\theta_1}, \left\{X, \mathbf{X}^{N(\cdot)}\right\}\right) \tag{16}$$

Representation Balancing: Traditional IV regression methods do not account for confounder balance, however, a recent work [27] found that the observed confounders can still lead to bias in the second stage regression. Here we propose to learn a balanced representation r_θ by minimizing the discrepancy between the distributions of the treated and control groups. Following previous works on representation learning [14], [18], [27], we leverage the Wasserstein distance as the distribution distance metric to measure the discrepancy. The Wasserstein distance is defined as follows:

$$Wass_p(\mu, \nu) \overset{\text{def}}{=} \left(\inf_{\pi \in \Pi(\mu,\nu)} \int_{\Omega^2} D(x,y)^p d\pi(x,y)\right)^{1/p} \tag{17}$$

where Ω is an arbitrary space, D is a metric on that space, $\Pi(\mu, \nu)$ is the set of all probability measures on Π^2 that have marginals μ and ν. We adopt [41] for an efficient approximation algorithm to compute the Wasserstein distance. The constraint is formulated as follows:

$$\mathcal{L}_{CB} = Wass\left(\left\{r^i_{\theta_1}\right\}_{i:T^i=0}, \left\{r^i_{\theta_1}\right\}_{i:T^i=1}\right). \tag{18}$$

The overall objective function of the second stage is as follows:

$$\mathcal{L} = \mathcal{L}_y + \alpha\mathcal{L}_{CB} + \beta\mathcal{L}_{IB} + \gamma\left(\|\theta_1\| + \|\mu_1\|\right) \tag{19}$$

where α, β and γ are the trade-off hyper-parameters. Then the treatment effect can be estimated by:

$$\widehat{ATE} = \mathbb{E}\left[f_{\mu_1}\left(t=1, r_{\theta_1}\right) - f_{\mu_1}\left(t=0, r_{\theta_1}\right)\right] \tag{20}$$

The pseudo-code is summarized in Algorithm 1.

Algorithm 1: NetIV Algorithm.

Input: X: Observed variable matrix; T: Treatment variable; Y: Outcome; A: Adjacency matrix; $\alpha > 0$ and $\beta > 0$: tradeoff parameters;
Output: $\hat{Y}_0 = f_{\mu_1}(T=0, r_{\theta_1})$, $\hat{Y}_1 = f_{\mu_1}(T=1, r_{\theta_1})$;
Treatment Regression Stage:
repeat
  $r_{\theta_0} = g_{\theta_0}(A, X)$
  $\pi_{\mu_0}(r_{\theta_0}) \rightarrow \hat{P}(T|X, V, Z)$
  $\mathcal{L}_t = CrossEntropy(\pi_{\mu_0}(r_{\theta_0}, T))$
  update $\mu_0, \theta_0 \leftarrow SGD\{\mathcal{L}_t\}$
until (Max Epoch Reach)
Outcome Regression Stage:
repeat
  $r_{\theta_1} = g_{\theta_1}(A, X)$
  $\mathcal{L}_y = \frac{1}{n}\sum_{i=1}^n (Y^i - \sum_{t\in\{0,1\}} f_{\mu_1}(T^i = t, r^i_{\theta_1})\hat{P}(T^i = t|X^i, A))^2$
  $\mathcal{L}_b = Wass(\{r^i_{\theta_1}\}_{i:t_i=0}, \{r^i_{\theta_1}\}_{i:t_i=1})$
  $\mathcal{L} = \mathcal{L}_y + \alpha\mathcal{L}_{CB} + \beta\mathcal{L}_{IB} + \gamma(\|\mu_1\| + \|\theta_1\|)$
  update $\mu_1, \theta_1 \leftarrow Adam\{\mathcal{L}\}$
until (Max Epoch Reach)

IV. Experiments

A. Baselines

We compared our method (NetIV) with two groups of methods. The first group of methods does not consider utilizing the networked data and is valid only when the strong ignorability assumption holds:

Counterfactual Regression (CFR): CFR [14] learns a balanced distribution between treatment and control groups by restricting the Wasserstein distance between the representation of the two groups. It relies on the strong ignorability assumption and can not handle the unobserved confounders.

Treatment-agnostic Representation Network (TARNet): TARNet [14] is an ablation version of CFR without the representation balancing regulation term.

Causal Effect Variational Autoencoder (CEVAE): CEVAE [42] uses a neural network latent variable model to learn a joint distribution. However, its models often fail to accurately recover the joint distribution, resulting in poor performance in estimating the average treatment effect.

The second group of methods utilizes networked data:

Network Deconfounder (NetDeconf): NetDeconf [18] utilizes the GCN to learn the representation of latent confounders and minimize the integral probability metric between treatment and control groups to obtain a balanced representation. However, the method lacks the ability to handle the unobserved confounders.

Graph Infomax Adversarial Learning (GIAL): GIAL [19] further considers the imbalanced network structure based on [18]. However, it has the same problem as NetDeconf, GAIL cannot handle the problem of unobserved confounding.

Causal Network Embeddings (CNE): CNE [20] views the network structure as a proxy of unobserved confounders and proposes a semi-supervised method to partially solve the problem of unmeasured confounding. It assumes that the learned embedding blocks all backdoor paths of the confounders, which is impossible and not verifiable in real-world situations.

B. Dataset

Due to the counterfactual problem, we can only observe one of the two potential outcomes in the observational data, which makes it impossible to obtain the ground-truth treatment effect. Following previous works on networked data [18], [19], [43], in this section, we conduct experiments on four semi-synthetic networked datasets $\mathbf{G}$, including: two real-world social network *Flickr* & *BlogCatalog*[2], and two network structure of Facebook pages *Public-Figure* & *Sport*[3].

- *BlogCatalog [44]:* BlogCatalog is a graph dataset for a network of social relationships of bloggers listed on the BlogCatalog website.
- *Flickr [44]:* Flickr is an online social network where users share images and videos. The dataset is built by forming links between images sharing common metadata from Flickr. Edges are formed between images from the same location, submitted to the same gallery, group, or set, images sharing common tags, images taken by friends, etc.
- *Public-Figure* and *Sport* in Facebook-pages *[45]:* The data is collected from Facebook pages and represents blue verified Facebook page networks of different categories. The nodes represent the pages and edges are mutual likes among them. In the experiments, we choose two categories Public-Figure, and Sport.

More information about the used social network structures is shown in Table II.

[2]We use the preprocessed datasets of BlogCatalog and Flickr in https://github.com/rguo12/network-deconfounder-wsdm20

[3][Online]. Available: https://networkrepository.com/

TABLE II
REAL WORLD SOCIAL NETWORKS

Dataset Name	Instance	Edges
BlogCatalog	5,196	173,468
Flickr	7,575	239,738
Fb-pages-public-figure	11,565	67,113
Fb-pages-sport	13,866	86,857

Similar to [18], [27], we use the four network structures as information transmission relationships between nodes, and then generate the semi-synthetic datasets as follows:

Confounders X and unobserved confounders U: For each unit $k = 1, \ldots, n$, the confounders $X^{(k)} = (x_1, \ldots, x_{m_X})$ and unobserved confounders $U^{(k)} = (u_1, \ldots, u_{m_U})$ was generated:

$$X^{(k)}, U^{(k)} \sim N\left(0, \Sigma_{m_X+m_U}\right) \tag{21}$$

where m_X and m_U are the dimensions of observed and unobserved confounders. $\Sigma_{m_X+m_U} = I_{m_X+m_U} \times 0.95 + \mathbb{1}_{m_X+m_U} \times 0.05$ denotes that all elements except diagonal are 0.05 in the covariance matrix.

Latent instrumental variables Z and latent confounders V: The instrumental variables $Z^{(k)} = (z_1, \ldots, z_{m_Z})$ and latent confounders $V^{(k)} = (v_1, \ldots, v_{m_V})$ are generated in the following way:

$$z_i^{(k)} = \frac{\sum_{j \in N(k)} x_i^{(j)}}{|N(k)|}, v_i^{(k)} = \frac{\sum_{j \in N(k)} x_{i+m_Z}^{(j)}}{|N(k)|} \tag{22}$$

where m_Z and m_V are the dimensions of IV and latent confounders, $m_Z + m_V = m_X$. $N(k)$ denotes the neighboring nodes of the k-th unit.

The treatment variables T: Since the compared baselines can only be applied to binary treatment, here we only consider the case of binary treatment.

$$P\left(T|Z, X, V, U\right) = \frac{1}{1 + exp\left(\sum_{i=1}^{m_Z} z_i x_i + \sum_{i=1}^{m_X} x_i + \sum_{i=1}^{m_V} v_i + \sum_{i=1}^{m_U} u_i\right)} \tag{23}$$

And T is generated from $T \sim Bernoulli(P(T|Z, X, V, U))$.

The outcome variables Y:

$$Y\left(T, X, U, V\right) = \frac{T}{m_X + m_V + m_U}\left(\sum_{i=1}^{m_X} x_i + \sum_{i=1}^{m_V} v_i + \sum_{i=1}^{m_U} u_i\right) \tag{24}$$

$$+ \frac{1 - T}{m_X + m_V + m_U}\left(\sum_{i=1}^{m_X} x_i^2 + \sum_{i=1}^{m_V} v_i^2 + \sum_{i=1}^{m_U} u_i^2\right) \tag{25}$$

Fig. 4 elaborates on the experimental setting of this paper. In the social network of three people, each person i has three features index by x_1^i, x_2^i, x_3^i. There also exist unobserved confounders U^i for each unit i. The original features can be divided into two parts, a part that acts as IV for the neighboring nodes X_Z, and a part that acts as confounders for the neighboring nodes X_V. Taking the first individual as an example, the right half of the figure shows how the data was generated. V is the

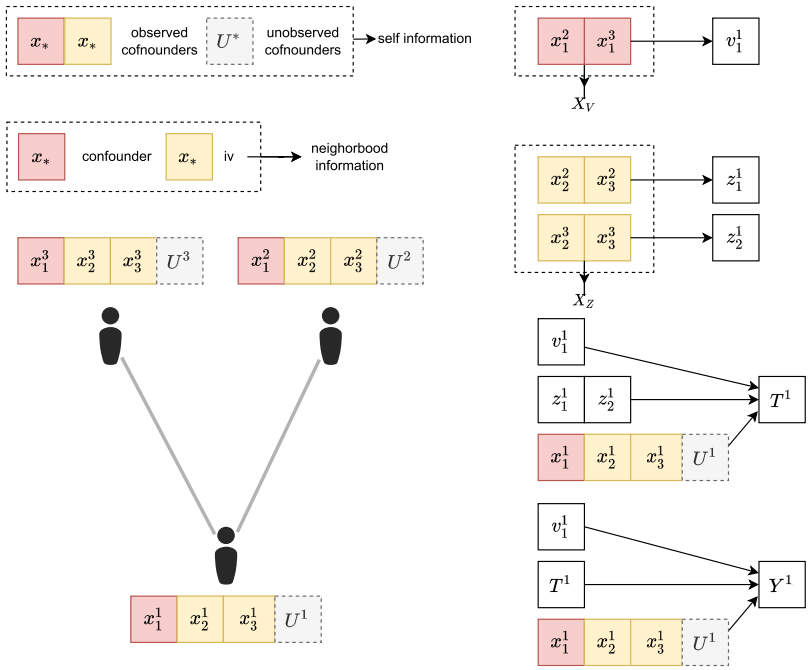


Fig. 4. An example of a network of three units to elaborate the experimental setting.

aggregation of the neighboring X_V and Z is the aggregation of the neighboring X_Z. T is generated by $[X, Z, V, U]$ and Y is generated by $[X, V, T, U]$.

We generate datasets of different dimensions and verify the effectiveness of our method in the next section.

C. Results

In this section, we verify the performance of the proposed method under different settings. We use **G**-m_Z-m_X-m_U to denote the used real-world social network **G**, the dimension of IV m_Z, the dimension of confounder m_X, and the dimension of unobserved confounder m_U. Besides, the dimension of latent confounders V is $m_V = m_X - m_Z$. We conduct the experiments with ten repetitions independently and evaluate the *Bias* and the *standard deviations (std)*. Here, we evaluate the performance of various methods using both within-sample and out-of-sample assessments. The within-sample error is calculated from the training sets, and the out-of-sample error is derived from the test set.

1) Average Treatment Effect Estimation: We report the results of the average treatment effect estimation in Table III. Comparing NetIV with two lines of baselines, we have the following observations:

- Without utilizing the networked information, the methods (CEVAE, CFR, and TARNet) do not perform well in estimating treatment effect;
- Although the network methods (NetDeconf, GIAL, and CNE) use the auxiliary networked data in the presence of unobserved confounders, they still fail to reduce the unmeasured confounding bias and even the performance worse than that of TARNet;
- Our NetIV significantly improves over baselines in estimating treatment effect because we control the confounding bias induced by both observed and unobserved confounders.

2) Varying the Dimension of Unobserved Confounders: We examine the performance of all methods in more complex settings and report in Fig. 5 the bias of the ATE estimates as the dimension of the unobserved confounder changes. From the results, we have the following observations:

- When there is no unobserved confounder, our method performs similarly to the previous methods on networked observational data.
- Our method can address the bias caused by unobserved confounders, while other methods perform poorly in this case.
- Other methods have significant performance degradation as the dimension of unobserved confounders increases, while our methods show stability in prediction for ATE in this case.

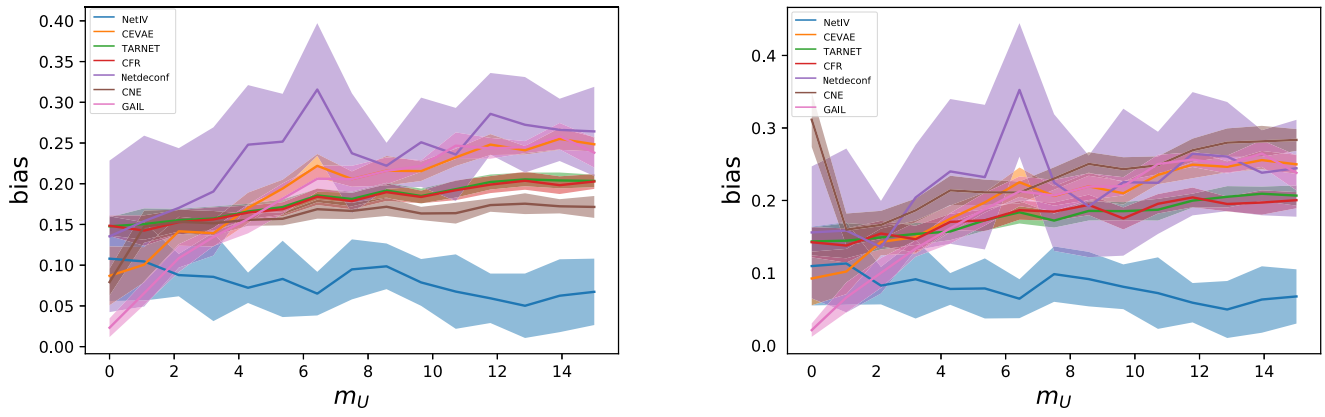
3) Varying the Dimension of IVs: We also evaluate the performance of treatment effect estimation as the IV dimension changes in Fig. 6. We can see that the performance of NetIV continues to improve as the IV information of neighboring nodes increases. The performance of the other methods remains mostly unchanged. In the absence of IV in neighbor information, our method degenerates and performs similarly to previous methods (Netdeconf, GAIL, CNE) utilizing network information.

4) Ablation Studies: In the ablation experiment, we need to answer two questions. The first question is what is the advantage of two-stage regression over direct regression, and the second is whether introducing two additional modules in the outcome regression stage improves the model.

TABLE III
THE BIAS (MEAN ± STD) OF ATE ESTIMATION ON DIFFERENT SEMI-SYNTHETIC DATASETS (**G**-m_Z-m_X-m_U, $m_V = M_x - m_Z$)

	Within-sample			**Out-of-sample**		
Method	Flickr-6-10-10	Flickr-8-15-10	Flickr-15-20-15	Flickr-6-10-10	Flickr-8-15-10	Flickr-15-20-15
CEVAE	0.294 ± 0.111	0.333 ± 0.147	0.153 ± 0.090	0.292 ± 0.103	0.311 ± 0.140	0.133 ± 0.115
CFR	0.166 ± 0.060	0.177 ± 0.057	0.161 ± 0.033	0.172 ± 0.070	0.185 ± 0.055	0.155 ± 0.036
TARNet	0.170 ± 0.058	0.281 ± 0.163	0.147 ± 0.061	0.176 ± 0.041	0.280 ± 0.174	0.145 ± 0.067
netdeconf	0.113 ± 0.016	0.126 ± 0.015	0.122 ± 0.025	0.112 ± 0.026	0.128 ± 0.027	0.127 ± 0.025
GAIL	0.130 ± 0.026	0.155 ± 0.028	0.139 ± 0.030	0.131 ± 0.033	0.151 ± 0.032	0.133 ± 0.035
CNE	0.170 ± 0.078	0.155 ± 0.045	0.156 ± 0.029	0.251 ± 0.071	0.177 ± 0.055	0.164 ± 0.030
NetIV	**0.071 ± 0.087**	**0.076 ± 0.056**	**0.042 ± 0.057**	**0.064 ± 0.078**	**0.075 ± 0.062**	**0.044 ± 0.057**
	Within-sample			**Out-of-sample**		
Method	BlogCatalog-6-10-10	BlogCatalog-8-15-10	BlogCatalog-15-20-15	BlogCatalog-6-10-10	BlogCatalog-8-15-10	BlogCatalog-15-20-15
CEVAE	0.443 ± 0.088	0.068 ± 0.101	0.242 ± 0.087	0.472 ± 0.161	0.120 ± 0.081	0.229 ± 0.130
CFR	0.162 ± 0.009	0.157 ± 0.018	0.146 ± 0.014	0.162 ± 0.024	0.162 ± 0.020	0.143 ± 0.024
TARNet	0.165 ± 0.054	0.152 ± 0.027	0.127 ± 0.032	0.150 ± 0.057	0.155 ± 0.022	0.131 ± 0.040
netdeconf	0.115 ± 0.015	0.102 ± 0.027	0.090 ± 0.025	0.119 ± 0.032	0.104 ± 0.034	0.093 ± 0.030
GAIL	0.185 ± 0.014	0.125 ± 0.040	0.118 ± 0.020	0.190 ± 0.018	0.129 ± 0.043	0.102 ± 0.032
CNE	0.087 ± 0.023	0.136 ± 0.017	0.123 ± 0.015	0.283 ± 0.030	0.143 ± 0.034	0.134 ± 0.023
NetIV	**0.040 ± 0.053**	**0.022 ± 0.026**	**0.041 ± 0.041**	**0.039 ± 0.053**	**0.018 ± 0.022**	**0.045 ± 0.045**
	Within-sample			**Out-of-sample**		
Method	figure-6-10-10	figure-8-15-10	figure-15-20-15	figure-6-10-10	figure-8-15-10	figure-15-20-15
CEVAE	0.370 ± 0.137	0.394 ± 0.057	0.360 ± 0.064	0.360 ± 0.181	0.396 ± 0.096	0.367 ± 0.141
CFR	0.156 ± 0.011	0.161 ± 0.056	0.163 ± 0.011	0.150 ± 0.013	0.162 ± 0.058	0.168 ± 0.017
TARNet	0.155 ± 0.010	0.166 ± 0.010	0.114 ± 0.012	0.158 ± 0.016	0.163 ± 0.013	0.115 ± 0.015
netdeconf	0.177 ± 0.016	0.111 ± 0.009	0.118 ± 0.012	0.167 ± 0.012	0.108 ± 0.014	0.115 ± 0.015
GAIL	0.171 ± 0.016	0.127 ± 0.015	0.145 ± 0.016	0.173 ± 0.020	0.131 ± 0.015	0.146 ± 0.014
CNE	0.133 ± 0.013	0.148 ± 0.013	0.149 ± 0.015	0.154 ± 0.022	0.160 ± 0.020	0.155 ± 0.012
NetIV	**0.072 ± 0.090**	**0.052 ± 0.058**	**0.052 ± 0.058**	**0.075 ± 0.092**	**0.054 ± 0.059**	**0.055 ± 0.061**
	Within-sample			**Out-of-sample**		
Method	sport-6-10-10	sport-8-15-10	sport-15-20-15	sport-6-10-10	sport-8-15-10	sport-15-20-15
CEVAE	0.342 ± 0.105	0.309 ± 0.070	0.360 ± 0.039	0.359 ± 0.108	0.340 ± 0.083	0.357 ± 0.065
CFR	0.193 ± 0.005	0.132 ± 0.005	0.175 ± 0.008	0.196 ± 0.015	0.131 ± 0.008	0.177 ± 0.008
TARNet	0.149 ± 0.010	0.117 ± 0.026	0.176 ± 0.013	0.155 ± 0.010	0.121 ± 0.023	0.175 ± 0.014
netdeconf	0.207 ± 0.010	0.081 ± 0.009	0.121 ± 0.016	0.210 ± 0.014	0.084 ± 0.012	0.123 ± 0.023
GAIL	0.148 ± 0.017	0.128 ± 0.017	0.134 ± 0.033	0.142 ± 0.018	0.128 ± 0.017	0.135 ± 0.035
CNE	0.168 ± 0.015	0.105 ± 0.009	0.157 ± 0.016	0.169 ± 0.018	0.119 ± 0.010	0.159 ± 0.018
NetIV	**0.051 ± 0.055**	**0.050 ± 0.057**	**0.042 ± 0.044**	**0.053 ± 0.055**	**0.047 ± 0.055**	**0.045 ± 0.046**

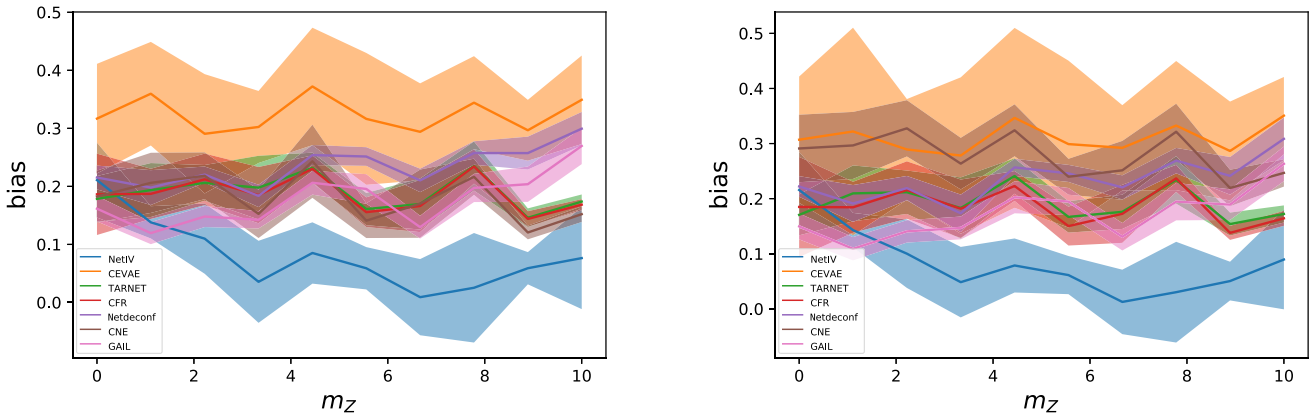
We emphasize the best-performing method in each setting by using bold text and underlining the second-best method.



(a) Within-sample. (b) Out-of-sample.

Fig. 5. Performance of different methods varying the unobserved confounders dimension.

To answer the first question, we focus specifically on two baseline models, Netdeconf and CFR. Netdeconf can be viewed as an ablation version of NetIV without two-stage regression. Its essence is to regress $\mathbb{E}[Y|T, V, X]$. CFR can be viewed as an ablation version of Netdeconf without considering the network information. Its essence is to regress $\mathbb{E}[Y|T, X]$. The proposed NetIV is equivalent to regression $\mathbb{E}[Y|T, X, V]$. These methods all use Wasserstein distance to constrain the representation to eliminate confounding bias. From Fig. 7, we have the following observations. As the dimension of unobserved confounders increases, the performance of Netdeconf and CFR continuously decreases, while NetIV consistently performs better. Note that when the dimension of unobserved confounders is zero, which means that there are no unobserved confounders, Netdeconf shows a significant improvement over CFR since it considers the confounder proxy in the network structure information. At



(a) Within-sample. (b) Out-of-sample.

Fig. 6. Performance of different methods varying the IV dimension.

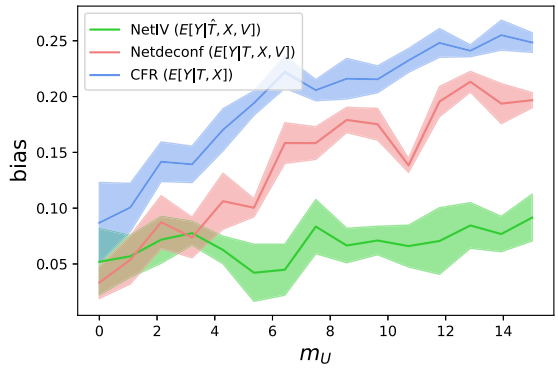


Fig. 7. Ablation studies with m_U change.

the same time, our model still outperforms in such cases. The performance of Netdeconf rapidly deteriorates as the dimension of U increases, eventually degenerating to the same level as CFR.

We perform two ablation studies of the proposed NetIV algorithm on the four benchmarks to answer the second question. The first is NetIV (w/o CB), where the confounder balancing module in (18) is removed. We directly regress the outcome without considering the confounding bias induced by observed confounders. The second ablation study is NetIV (w/o IB) where the information bottleneck module in (14) is removed. In this way, redundant information is not eliminated in the second stage of regression.

The performance of ATE estimation is reported in Table IV, where we notice the performance drop after removing the confounder balancing module and the information bottleneck module, which implies the effectiveness of these modules. Although NetIV (w/o IB) performs better a modest fraction of the time, adding the information bottleneck leads to more stable performance and a more robust model.

5) Time Complexity: Given that directly analyzing the time complexity of neural networks is notably challenging, we assessed their computational efficiency by comparing inference times (in seconds) on the validation set across different methods.

TABLE IV
ABLATION STUDIES ON DIFFERENT SEMI-SYNTHETIC DATASETS
(m_Z-m_X-m_U, $m_V = M_x - m_Z$)

	6-10-10	8-15-10	15-20-15
Flickr			
NetIV	**0.064 ± 0.078**	0.075 ± 0.062	**0.044 ± 0.057**
NetIV (w/o CB)	0.109 ± 0.162	0.106 ± 0.126	0.070 ± 0.078
NetIV (w/o IB)	0.082 ± 0.093	**0.067 ± 0.072**	0.049 ± 0.046
BlogCatalog			
NetIV	**0.039 ± 0.053**	**0.018 ± 0.022**	**0.045 ± 0.045**
NetIV (w/o CB)	0.228 ± 0.071	0.066 ± 0.086	0.090 ± 0.059
NetIV (w/o IB)	0.051 ± 0.063	0.184 ± 0.083	0.200 ± 0.062
figure			
NetIV	**0.075 ± 0.092**	0.054 ± 0.059	**0.055 ± 0.061**
NetIV (w/o CB)	0.081 ± 0.072	0.080 ± 0.061	0.061 ± 0.063
NetIV (w/o IB)	0.081 ± 0.085	**0.049 ± 0.057**	0.087 ± 0.078
sport			
NetIV	**0.053 ± 0.055**	**0.047 ± 0.055**	0.045 ± 0.046
NetIV (w/o CB)	0.108 ± 0.106	0.063 ± 0.090	0.078 ± 0.083
NetIV (w/o IB)	0.088 ± 0.109	0.051 ± 0.056	**0.045 ± 0.037**

The findings are illustrated in Fig. 8. CFR, which does not account for network structure information and employs the simplest MLP architecture for regression, demonstrated the highest computational efficiency. In contrast, Netdeconf, leveraging a GCN to harness network data, exhibited marginally slower computation times than CFR. NetIV, implementing a two-stage regression to address the effects of unobserved confounding biases, showed reduced computational efficiency relative to both CFR and Netdeconf. Nevertheless, the computation speed of NetIV is less than double that of these methods (a constant factor) since it involves a two-stage regression process. The results show that NetIV's time complexity is acceptable and scalable.

6) Hyper-Parameter Analysis: In our experiments, we have three hyper-parameters α, β, and γ. We vary the choice of α, β and γ in the scope $\{0, 1e-4, 1e-3, 1e-2, 1e-1, 1\}$. The results in the dataset $Flickr - 6 - 10 - 10$ are shown in Fig. 9. In our implementation, we make the hyper-parameters of each line of experiments satisfy $\alpha = 1e-3$, $\beta = 1e-2$ and $\gamma = 1e-2$.

7) Implement Details: For treatment regression, we use one graph convolution layer to extract the feature and three layers

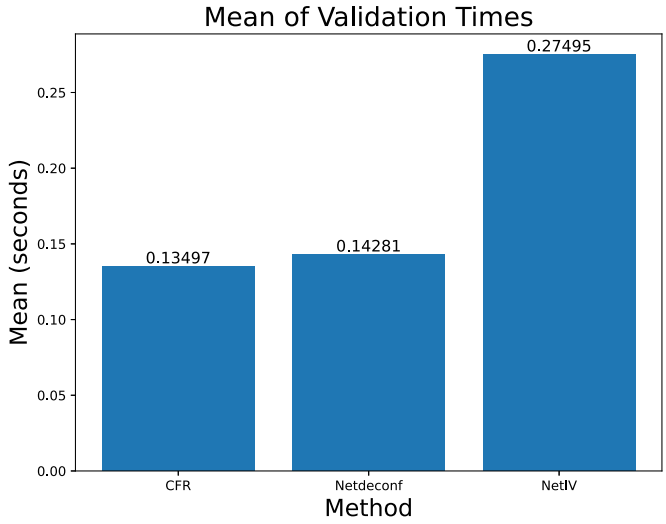


Fig. 8. Comparing time complexity of NetIV with baseline methods.

of multi-layer perceptrons with ReLU activation function as our logistic regression network π. We optimize the loss function $\mathcal{L}_t$ with stochastic gradient descent [46].

For the outcome regression network, we use one graph convolution layer to extract the feature, then with two heads of 3 layers of multi-layer perceptrons each for separately estimate $\hat{Y}_0$ and $\hat{Y}_1$. We use Adam [47] to optimize the loss function in (19).

All the experiments are conducted under the following environment. Hardware used: Ubuntu 16.04.5 LTS operating system with 2 * Intel Xeon E5-2678 v3 CPU, and 4 * GeForce GTX 1080Ti GPU with 12GB of VRAM. Software used: Python with Pytorch 1.1.0, NumPy 1.18.5, and MatplotLib 3.3.4.

V. Related Work

A. Treatment Effect Estimation

The field of deep learning has flourished during the past decade. Various methods are proposed to estimate the treatment effect with deep neural networks. CFR [14] tries to learn a balanced representation between treated and control groups. To calculate the dissimilarity of distributions from different treatment arms, they apply Maximum Mean Discrepancy and Wasserstein distance and fit a balanced representation by minimizing the discrepancy. [48] proposed Dragonnet that adopts neural networks to fit the expected outcome and the probability of treatment. Then they plug the fitted models A-IPTW estimator [49] to estimate the treatment effect. In addition to this method, several generative models have been used to estimate the treatment effect. [42] propose to use the variational autoencoder to learn the joint distribution in order to estimate the treatment effect. [50] propose to infer treatment effect based on the Generative Adversarial Nets framework.

These methods depend on the strong ignorability assumption and cannot resolve unobserved confounders.

B. Networked Observational Data

There are mainly two lines of work that considers to utilize the networked observational data. The first line tries to leverage the network structure to capture the patterns of hidden confounders. [18] first propose network deconfounder to resolve the confounding bias from hidden confounders. They assume that the network structure is a sufficient proxy for hidden confounders. GIAL [19] extends the network deconfounder [18] and further considers the imbalanced network structure. They propose a Graph Infomax Adversarial Learning method (GIAL) to estimate the treatment effects from imbalanced networked observational data. In fact, these methods rely on an assumption similar to strong ignorability, i.e., it is possible to fully estimate the treatment assignment mechanism using information from neighboring nodes. If there exist some unobserved confounders that cannot be covered by the learned representation (U in Fig. 1), these methods fail to identify the treatment effect. CNE [20] proposes to use semi-supervised prediction to learn a node embedding of each unit to replace the unobserved confounders. However, such a black-box embedding model only works when the assumption that the learned embedding covers all the information of the confounders holds, which is a strong assumption and is non-verifiable in real-world applications. To summarize, previous works on utilizing network structure to resolve the bias from unmeasured confounders heavily rely on the assumption that the learned representation or embedding covers all the information of the confounders. In this paper, we allow the existence of unmeasured confounding that is not related to the network structure, which makes the previous methods invalid in estimating treatment effects with such a setting.

There is another line of work that studies the interference [43], [51], [52] or spillover effect [53], [54], where the outcome of a unit can be influenced by the neighboring nodes' treatments or outcomes. These works do not consider solving the bias from unmeasured confounders. In this paper, we focus on exploiting the network structure to eliminate the bias from the unobserved confounders.

C. Instrumental Variable Regression

Instrumental variable regression is a popular method for estimating causal effects from observational data in the presence of unobserved confounders. Instrumental variable regression has a long history in economics, epidemiology, and sociology [24], [55], [56].

Two-stage least squares regression [24], [57] is a classical IV method that applies linear models to estimate the expectation of outcome given treatment and instrumental variable. Kernel IV [58] relaxes the linear assumption by mapping the feature to a reproducing kernel Hilbert space (RKHS) and performing kernel ridge regression. AGMM [59] and DeepGMM [60] learn the struct function by optimizing the sample averages of moment conditions.

Here we follow the framework of IV methods in [25], [26], [27] that utilize the deep neural network to do two-stage regression. In the first stage, these methods learn a mapping from instrumental variables and observed confounders to the treatment. In the second stage, they use the expectation of treatment conditional on instrumental variables and observed confounders to regress the outcome. Especially, [27] adds an extra confounder balancing part in the second stage to eliminate

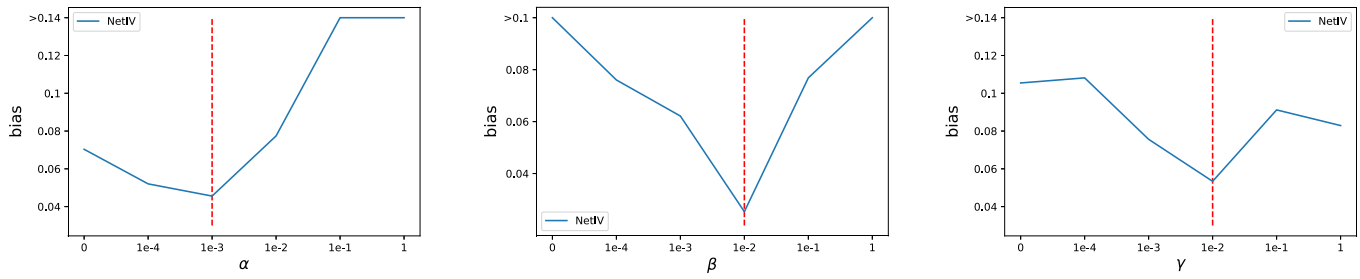


Fig. 9. Hyper-parameter sensitivity analysis on dataset Flickr-6-10-10.

the confounding bias conducted by the observed confounders. We propose a two-stage regression framework similar to these approaches.

D. Causal Inference With Graph-Structure Treatments

There are several works that studies to estimate conditional average treatment (CATE) effects using graph-structure treatments [61], [62]. They study the treatments like drugs, which are associated with graphs representing their molecular structures. And these methods learn representations of graph interventions for CATE estimation. The setting is different from our paper since we study the problem of doing causal inference with an additional network structure of units.

VI. Conclusion

In this paper, we study the problem of estimating treatment effects in the presence of unobserved confounders from networked observational data. We take an alternative perspective to exploit the IV information in the network structure and propose a networked instrumental variable regression framework (NetIV), a two-stage procedure, to eliminate the confounding bias caused by unobserved confounders. Moreover, to eliminate the confounding bias caused by observed confounders and reduce the variance, we introduce the confounding balancing module and the information bottleneck module into the second regression stage. Further experiments show that the proposed NetIV algorithm outperforms the state-of-art methods.

Here we also present some fascinating directions for future work. We believe that a more in-depth analysis of networked observational data is necessary. Since the neighborhood information of the network structure can serve as a proxy, it is possible to address the unmeasured confounding by some negative control methods [63]. We also believe that the proposed method can be used in recommendation systems to address the problem of poor evaluation of utility functions due to unobserved confounders [64].

Acknowledgment

All opinions in this paper are those of the authors and do not necessarily reflect the views of the funding agencies.

References

[1] A. Wu et al., "Learning decomposed representations for treatment effect estimation," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 5, pp. 4989–5001, May 2023.
[2] K. Kuang et al., "Stable prediction with leveraging seed variable," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 6, pp. 6392–6404, Jun. 2023.
[3] K. Kuang et al., "Data-driven variable decomposition for treatment effect estimation," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 5, pp. 2120–2134, May 2022.
[4] K. Kuang et al., "Causal inference," *Engineering*, vol. 6, no. 3, pp. 253–263, 2020.
[5] H. Wang et al., "Out-of-distribution generalization with causal feature separation," *IEEE Trans. Knowl. Data Eng.*, vol. 36, no. 4, pp. 1758–1772, Apr. 2024.
[6] Z. Ziyu et al., "Differentiated matching for individual and average treatment effect estimation," *Data Mining Knowl. Discov.*, vol. 37, no. 1, pp. 205–227, 2023.
[7] J. Li, S. Ma, T. Le, L. Liu, and J. Liu, "Causal decision trees," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 2, pp. 257–271, Feb. 2017.
[8] N. Kallus and A. Zhou, "Confounding-robust policy improvement," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2018, pp. 9289–9299.
[9] R. Guo, L. Cheng, J. Li, P. R. Hahn, and H. Liu, "A survey of learning causality with data: Problems and methods," *ACM Comput. Surveys*, vol. 53, no. 4, pp. 1–37, 2020.
[10] N. Hassanpour and R. Greiner, "Counterfactual regression with importance sampling weights," in *Proc. Int. Joint Conf. Artif. Intell.*, 2019, pp. 5880–5887.
[11] S. Assaad et al., "Counterfactual representation learning with balancing weights," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2021, pp. 1972–1980.
[12] P. C. Austin, "An introduction to propensity score methods for reducing the effects of confounding in observational studies," *Multivariate Behav. Res.*, vol. 46, no. 3, pp. 399–424, 2011.
[13] A. Abadie and G. W. Imbens, "Matching on the estimated propensity score," *Econometrica*, vol. 84, no. 2, pp. 781–807, 2016.
[14] U. Shalit, F. D. Johansson, and D. Sontag, "Estimating individual treatment effect: Generalization bounds and algorithms," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 3076–3085.
[15] F. Johansson, U. Shalit, and D. Sontag, "Learning representations for counterfactual inference," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 3020–3029.
[16] L. Yao, S. Li, Y. Li, M. Huai, J. Gao, and A. Zhang, "Representation learning for treatment effect estimation from observational data," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2018, pp. 2638–2648.
[17] M. Baiocchi, J. Cheng, and D. S. Small, "Instrumental variable methods for causal inference," *Statist. Med.*, vol. 33, no. 13, pp. 2297–2340, 2014.
[18] R. Guo, J. Li, and H. Liu, "Learning individual causal effects from networked observational data," in *Proc. 13th Int. Conf. Web Search Data Mining*, 2020, pp. 232–240.
[19] Z. Chu, S. L. Rathbun, and S. Li, "Graph infomax adversarial learning for treatment effect estimation with networked observational data," 2021, *arXiv:2106.02881*.
[20] V. Veitch, Y. Wang, and D. Blei, "Using embeddings to correct for unobserved confounding in networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2019, pp. 13792–13802.
[21] H. Wang et al., "Medication combination prediction using temporal attention mechanism and simple graph convolution," *IEEE J. Biomed. Health Informat.*, vol. 25, no. 10, pp. 3995–4004, Oct. 2021.

[22] C. Gao, S. Yin, H. Wang, Z. Wang, Z. Du, and X. Li, "Medical-knowledge-based graph neural network for medication combination prediction," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 10, pp. 13246–13257, Oct. 2024.
[23] G. W. Imbens and D. B. Rubin, *Causal Inference in Statistics, Social, and Biomedical Sciences*. Cambridge, U.K.: Cambridge Univ. Press, 2015.
[24] J. D. Angrist and A. B. Krueger, "Instrumental variables and the search for identification: From supply and demand to natural experiments," *J. Econ. Perspectives*, vol. 15, no. 4, pp. 69–85, 2001.
[25] J. Hartford, G. Lewis, K. Leyton-Brown, and M. Taddy, "Deep IV: A flexible approach for counterfactual prediction," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 1414–1423.
[26] L. Xu, Y. Chen, S. Srinivasan, N. de Freitas, A. Doucet, and A. Gretton, "Learning deep features in instrumental variable regression," 2020, *arXiv: 2010.07154*.
[27] A. Wu, K. Kuang, B. Li, and F. Wu, "Instrumental variable regression with confounder balancing," in *Proc. Int. Conf. Mach. Learn.*, 2022, pp. 24056–24075.
[28] S. J. Taylor and D. Eckles, "Randomized experiments to detect and estimate social influence in networks," in *Complex Spreading Phenomena in Social Systems: Influence and Contagion in Real-World Social Networks*. Berlin, Germany: Springer, 2018, pp. 289–322.
[29] L. Wang and E. T. Tchetgen, "Bounded, efficient and multiply robust estimation of average treatment effects using instrumental variables," *J. Roy. Stat. Soc. Ser. B: Statist. Methodol.*, vol. 80, no. 3, pp. 531–550, 2018.
[30] P. M. Aronow and A. Carnegie, "Beyond late: Estimation of the average treatment effect with an instrumental variable," *Political Anal.*, vol. 21, no. 4, pp. 492–506, 2013.
[31] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*.
[32] S. Jiang and Y. Sun, "Estimating causal effects on networked observational data via representation learning," in *Proc. 31st ACM Int. Conf. Inf. Knowl. Manage.*, 2022, pp. 852–861.
[33] D. Arbour, D. Garant, and D. Jensen, "Inferring network effects from observational data," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2016, pp. 715–724.
[34] W. K. Newey and J. L. Powell, "Instrumental variable estimation of nonparametric models," *Econometrica*, vol. 71, no. 5, pp. 1565–1578, 2003.
[35] F. P. Hartwig, L. Wang, G. D. Smith, and N. M. Davies, "Average causal effect estimation via instrumental variables: The no simultaneous heterogeneity assumption," 2020, *arXiv: 2010.10017*.
[36] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2016, pp. 3844–3852.
[37] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 1025–1035.
[38] L. Yao, S. Li, Y. Li, H. Xue, J. Gao, and A. Zhang, "On the estimation of treatment effect with text covariates," in *Proc. Int. Joint Conf. Artif. Intell.*, 2019, pp. 4106–4113.
[39] W.-D. K. Ma, J. Lewis, and W. B. Kleijn, "The HSIC bottleneck: Deep learning without back-propagation," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 5085–5092.
[40] T. Wu, H. Ren, P. Li, and J. Leskovec, "Graph information bottleneck," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2020, pp. 20437–20448.
[41] M. Cuturi and A. Doucet, "Fast computation of Wasserstein barycenters," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 685–693.
[42] C. Louizos, U. Shalit, J. Mooij, D. Sontag, R. Zemel, and M. Welling, "Causal effect inference with deep latent-variable models," 2017, *arXiv: 1705.08821*.
[43] J. Ma, M. Wan, L. Yang, J. Li, B. Hecht, and J. Teevan, "Learning causal effects on hypergraphs," in *Proc. 28th ACM SIGKDD Conf. Knowl. Discov. Data Mining*, 2022, pp. 1202–1212.
[44] J. Leskovec and R. Sosič, "SNAP: A general-purpose network analysis and graph-mining library," *ACM Trans. Intell. Syst. Technol.*, vol. 8, no. 1, 2016, Art. no. 1.
[45] R. A. Rossi and N. K. Ahmed, "An interactive data repository with visual analytics," *ACM SIGKDD Explorations Newslett.*, vol. 17, no. 2, pp. 37–41, 2016.
[46] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *J. Mach. Learn. Res.*, vol. 12, no. 7, pp. 2121–2159, 2011.
[47] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
[48] C. Shi, D. M. Blei, and V. Veitch, "Adapting neural networks for the estimation of treatment effects," 2019, *arXiv: 1906.02120*.
[49] J. M. Robins, "Robust estimation in sequentially ignorable missing data and causal inference models," in *Proc. Amer. Statist. Assoc.*, Indianapolis, IN, 2000, pp. 6–10.
[50] J. Y. Yoon, J. Jordon, J. Mihaela vand Jordon, and M. Vand Der Schaar, "GANITE: Estimation of individualized treatment effects using generative adversarial nets," in *Proc. Int. Conf. Learn. Representations*, 2018.
[51] Y. Ma and V. Tresp, "Causal inference under networked interference and intervention policy enhancement," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2021, pp. 3700–3708.
[52] Z. Zhao, K. Kuang, R. Xiong, and F. Wu, "Learning individual treatment effects under heterogeneous interference in networks," 2022, *arXiv:2210.14080*.
[53] L. Forastiere, E. M. Airoldi, and F. Mealli, "Identification and estimation of treatment and interference effects in observational studies on networks," *J. Amer. Statist. Assoc.*, vol. 116, no. 534, pp. 901–918, 2021.
[54] E. J. Tchetgen Tchetgen, I. R. Fulcher, and I. Shpitser, "Auto-g-computation of causal effects on a network," *J. Amer. Statist. Assoc.*, vol. 116, no. 534, pp. 833–844, 2021.
[55] E. J. T. Tchetgen, S. Walter, S. Vansteelandt, T. Martinussen, and M. Glymour, "Instrumental variable estimation in a survival context," *Epidemiology*, vol. 26, no. 3, 2015, Art. no. 402.
[56] J. G. Altonji, T. E. Elder, and C. R. Taber, "An evaluation of instrumental variable strategies for estimating the effects of catholic schooling," *J. Hum. Resour.*, vol. 40, no. 4, pp. 791–821, 2005.
[57] G. W. Imbens and J. D. Angrist, "Identification and estimation of local average treatment effects," *Econometrica*, vol. 62, no. 2, pp. 467–475, 1994.
[58] R. Singh, M. Sahani, and A. Gretton, "Kernel instrumental variable regression," 2019, *arXiv: 1906.00232*.
[59] G. Lewis and V. Syrgkanis, "Adversarial generalized method of moments," 2018, *arXiv: 1803.07164*.
[60] A. Bennett, N. Kallus, and T. Schnabel, "Deep generalized method of moments for instrumental variable analysis," 2019, *arXiv: 1905.12495*.
[61] J. Kaddour, Y. Zhu, Q. Liu, M. J. Kusner, and R. Silva, "Causal effect inference for structured treatments," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2021, pp. 24841–24854.
[62] S. Harada and H. Kashima, "GraphITE: Estimating individual effects of graph-structured treatments," in *Proc. 30th ACM Int. Conf. Inf. Knowl. Manage.*, 2021, pp. 659–668.
[63] N. Kallus, X. Mao, and M. Uehara, "Causal inference under unmeasured confounding with negative controls: A minimax learning approach," 2021, *arXiv:2103.14029*.
[64] T. Joachims, B. London, Y. Su, A. Swaminathan, and L. Wang, "Recommendations as treatments," *AI Mag.*, vol. 42, no. 3, pp. 19–30, 2021.

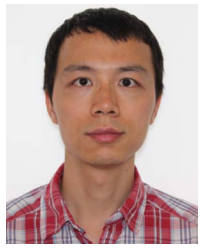
Ziyu Zhao received the BS degree from the Department of Computer Science and Technology, Zhejiang University, in 2021. He is currently working toward the PhD degree with the Department of Computer Science and Technology, Zhejiang University. His main research interests include causal inference, machine learning, and data mining.

Anpeng Wu received the BS degree from the College of Science, Zhejiang University of Technology, in 2020. He is currently working toward the PhD degree with the Department of Computer Science and Technology, Zhejiang University. His main research interests include causal inference, representation learning, and reinforcement learning.

Kun Kuang received the PhD degree from the Department of Computer Science and Technology, Tsinghua University, in 2019. He is an associate professor with the College of Computer Science and Technology, Zhejiang University. He was a visiting scholar with Prof. Susan Athey's Group at Stanford University. His main research interests include causal inference, causality inspired machine learning, and smart justice. He has published more than 100 papers in prestigious conferences and journals in data mining and machine learning, including the *Cell Patterns*, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *IEEE Transactions on Knowledge and Data Engineering*, *Engineering*, ICML, NeurIPS, KDD, ICDE, WWW, SIGIR, ACM Multimedia, etc. He received the ACM SIGAI China Rising Star Award in 2022.

Ruoxuan Xiong received the PhD degree in management science and engineering from Stanford University, in 2020. She is an assistant professor with the Department of Quantitative Theory and Methods, Emory University. She was a postdoctoral fellow with the Stanford Graduate School of Business from 2020 to 2021. Her research is at the intersection of econometrics and operations research, focusing on causal inference, experimental design and factor modeling, and with applications in finance and healthcare. Her work was awarded the Honorable Mention in the 2019 INFORMS George Nicholson Student Paper Competition, and was among the finalists of the 2020 MSOM Student Paper Competition.

Bo Li received the bachelor's degree in mathematics from Peking University, and the PhD degree in statistics from the University of California, Berkeley. He is an associate professor with the School of Economics and Management, Tsinghua University. His research interests include business analytics and risk-sensitive artificial intelligence. He has published widely in academic journals and conferences across a range of fields including statistics, computer science, management science, and economics.



Zhihua Wang received the PhD degree from Imperial College London. He is the director of the machine learning and education innovation center of Shanghai Institute for Advanced Study, Zhejiang University. Before that, he worked as the head of product and project in startup companies and investment banking industries for eight years. His main interests include machine learning, federated learning, system architecture, and financial technology.

Fei Wu (Senior Member, IEEE) received the PhD degree from the College of Computer Science, Zhejiang University. Now, he is the professor and dean with the College of Computer Science, Zhejiang University. His main research interests include multimedia information analysis and retrieval, and digital library.