# One-stage Deep Instrumental Variable Method for Causal Inference from Observational Data

Adi Lin*, Jie Lu*, Junyu Xuan* and Fujin Zhu*† Guangquan Zhang*
* Centre for Artificial Intelligence, FEIT, University of Technology Sydney, Australia
† School of Management and Economics, Beijing Institute of Technology, China
Email: Adi.Lin@student.uts.edu.au, Jie.Lu@uts.edu.au, Junyu.Xuan@uts.edu.au,
Fujin.Zhu@student.uts.edu.au, Guangquan.Zhang@uts.edu.au

***Abstract*—Causal inference from observational data aims to estimate causal effects when controlled experimentation is not feasible, but it faces challenges when unobserved confounders exist. The instrumental variable method resolves this problem by introducing a variable that is correlated with the treatment and affects the outcome only through the treatment. However, existing instrumental variable methods require two stages to separately estimate the conditional treatment distribution and the outcome generating function, which is not sufficiently effective. This paper presents a one-stage approach to jointly estimate the treatment distribution and the outcome generating function through a cleverly designed deep neural network structure. This study is the first to merge the two stages to leverage the outcome to the treatment distribution estimation. Further, the new deep neural network architecture is designed with two strategies (i.e., shared and separate) of learning a confounder representation account for different observational data. Such network architecture can unveil complex relationships between confounders, treatments, and outcomes. Experimental results show that our proposed method outperforms the state-of-the-art methods. It has a wide range of applications, from medical treatment design to policy making, population regulation and beyond.**

***Index Terms*—observational data, causal inference, instrumental variable, neural networks**

## I. Introduction

Causal inference is to infer the causal effect of an event A on another event B. For example, people wish to evaluate the effect of smoking on health over a long time. Though randomized controlled experiments are the gold standard to estimate causal effects, it is often either unethical, technically impossible, or too costly to implement [1]. As an example, it is immoral to force people to smoke in a controlled experimentation. Also, a double-blind assignment smoking experiment is nearly impossible, as the smokers or non-smokers will know if they smoke or not in the experiments. Even if it is feasible, it may take a considerable amount of money and several decades. An alternative practical way is to conduct causal inference from uncontrolled observational data which typically includes observed confounders, treatments, and outcomes. In the smoking example, the observed confounders denote people's characteristics, the treatment is to smoke, and the outcome is people's health. Some successful methods for causal inference from observational data include but not limited to Bayesian Additive Regression Trees [2] and multi-task Gaussian process [3]), tree-based methods [4], and neural networks [5]–[7].

Those causal inference methods assume that there are no unobserved confounders that affect both the treatment and outcome. For example, genes of people are an unobserved confounder for the smoking example because genes may influence people's smoking and impact on the people's lung cancer. One method to remove the influence of unobserved confounders is to use limited experimental data [8], but it does not work in a setting where only observational data is available. Another is instrumental variable method [9] which estimates the causal effect when there is one or more instrumental variables.

However, existing instrumental variable methods require two-stage solutions: 1) estimate the conditional treatment distribution at the first stage 2) and then fit outcomes generating function using the estimated conditional treatment distribution from the first stage. Such two-stage solutions have the following two weaknesses: 1) estimating the conditional treatment distribution cannot utilize the information from the outcome generating function (e.g., DeepIV [10]), so the final performance of causal effect estimation is restricted because outcome is one significant component of observational data; 2) the conditional treatment distribution and the target outcome generating function often have complex hidden relationships with observed confounders, many two-stage solutions lack strong representation ability to capture such relationships (e.g., 2SLS [11] and nonparametric kernel [12]).

In this paper, we propose a one-stage deep instrumental variable method to estimate causal effect. Specially, we have designed a new deep neural network architecture to jointly estimate the conditional treatment distribution and fit the outcome generating function, where former two stages can borrow the knowledge from each other in our one-stage method. An unified loss function is formulated to train the network structure using the observational data. Further, the designed deep neural networks have powerful capability of unveiling complex hidden relationships than other methods, like linear [11]. The comparative experiments with existing state-of-the-art two-stage methods show the effectiveness of our designed new one-stage deep neural network architecture. Also, the comparative experiments with classical linear-based and kernel-based methods show that our method has superior performance due to the powerful ability of deep neural network on complex function fitting.

Two contributions of this study are summarized as follow:

2374-8486/19/$31.00 ©2019 IEEE
DOI 10.1109/ICDM.2019.00052

- The one-stage deep instrumental variable method estimates the conditional treatment probability distribution and fits the target outcome generating function simultaneously. Training two quantities jointly is more effective and makes it possible to borrow the information from each other during training.

单阶段深度工具变量法估计条件处理概率分布，并同时拟合目标结果生成函数。联合训练两个量更有效，并使在训练期间相互借用信息成为可能

- A new deep neural network architecture is designed with two strategies (i.e., shared and separate) of learning a confounder representation account for different observational data. Such network architecture can unveil complex relationships between confounders, treatment, and outcomes.

设计了一种新的深度神经网络结构，采用两种策略（即共享和独立）来学习不同观测数据的混杂表示。这种网络架构可以揭示混杂因素、治疗和结果之间的复杂关系

The remainder of this paper is organized as follows. Section II discusses the related work. Section III describes the concepts of causal inference and the instrumental variable. We introduce our one-stage deep instrumental variable method in Section IV. Section V presents experiments showing performance and related analysis. Finally, Section VI concludes our study and discusses future work.

## II. Related Work

This section reviews the study on learning methods for causal inference. We organize existing works in this area into two groups: one group for general studies of causal inference; the other group focusing on instrumental variable methods.

### A. Learning methods for causal inference

Under the assumption of there is no unobserved confounders, exiting learning methods or causal inference mainly falls into three categories: methods based on treatment modeling, methods fitting outcome regressions, and doubly robust methods combining the two methods above [13].

One important school of methods for causal inference is methods based on treatment modeling. The idea is to replicate a randomized experiment that has similar covariate distributions in the treated and the control groups. Matching [14] aims to balance the covariate distributions in the treated and control groups by choosing matched subjects. That is matching methods aim to estimate a conditional treatment distribution based on observed covariates. Some examples of popular matching methods are genetic matching [15], optimal matching [16] and kernel matching [17]. Matching methods have a few key advantages: matching methods are complementary with outcome regression adjustment and perform well in combination, they can still give acceptable results when there is no sufficient overlap of covariate distributions between the treated and the control groups, and it is straightforward to obtain which performance that matching methods can assess. However, most matching methods assume fully observed covariates; it is hard for matching methods to deal with missing covariates. Additionally, selecting the suitable matching method can be unclear. The reason is different distance measure selected in matching methods may lead to different results.

Another way for treatment modeling is to apply propensity score methods. Propensity score [18] is defined as the conditional treatment distribution given observed covariates. The main propensity score methods are propensity score matching, stratification on the propensity score, inverse probability of treatment weighting (IPTW) and covariate adjustment using the propensity score. Generally, propensity score matching and IPTW gain better performance [19]. Propensity score methods work fine in simpler settings, for example, we can use a logistic regression to estimate the propensity score. Propensity score methods could balance the covariate distributions using the information of a large number of observations. But propensity score methods suffer from the high dimensionality problem, have a weak ability to deal with hidden variables, and are not designed for time-varying treatments.

The second popular school of methods falls into outcome regression adjustment [7], [20]. Bayesian methods, such as Bayesian additive regression trees (BART) [2] and multi-task Gaussian process [3], have good interpretation and gain good performance. But Bayesian methods lack flexibility in running time. Another group is ensemble methods (such as Random forests based causal tree [4] and super learner [21]). These methods are easy to understand and can often get better performance. But some of these methods need a large size of observations and most are time-consuming to run. A large number of methods belong to neural network-based methods. Balancing neural networks [5], [6] learn a "balanced" representation for covariate distributions of treated and control groups. SITE [7] preserves local similarity and balances covariate distributions simultaneously based on deep representation learning. Causal effect variational autoencoder (CEVAE) [22] uses deep neural networks to estimate the joint distribution for causal inference. Neural networks have good performance and are able to capture complex relationships among treatments, covariates and outcomes. However, neural networks need big data and often experience with overfitting.

Doubly robust [23] estimation combines propensity score methods and outcome regression to estimate the causal effect of the treatment on the outcome. When propensity score methods and outcome regression adjustment are used individually, propensity methods or outcome regression adjustment are unbiased only if the statistical models are correctly specified. The doubly robust estimators that are unbiased only need one of the two models to be correctly specified. Targeted maximum likelihood estimation (TMLE) [24] first estimates the conditional outcome distribution given the treatment and the observed covariates. Then TMLE estimates the propensity score. Next, TMLE updates the estimate of the conditional outcome distribution. Finally, TMLE generates targeted estimate of the target parameter. Double learning [25] combines the residuals of a propensity score model and the residuals of an outcome regression into a new regression to estimate average causal effect. For inference on treatment effects in the interactive model, the estimator uses the AIPTW estimator where the nuisance parameters are estimated using machine learning algorithms.

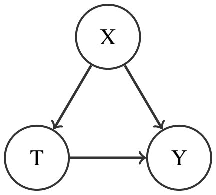


Fig. 1: Graphical model for standard causal inference. $T$ is the treatment, $X$ is the observed confounders, and $Y$ is the outcome.

### B. Instrumental variable methods

The instrumental variable method estimates the causal effect when there is unobserved confounding and one or more instrumental variables. An instrumental variable is a variable that is independent of unobserved confounders and affects the treatment but does not have a direct effect on the outcome beyond its effect on the treatment. The two-stage least squares (2SLS) [11] regression applies linear models to both two stages. Though 2SLS easily gives interpretations, it has strong assumptions of linearity and homogeneity. To relax these assumptions, nonparametric extensions of 2SLS use linear combinations of basis functions [26] or construct kernel smoothed estimators [12]. Though these methods introduce heterogeneity in low dimensional space, they need to carefully choose smoothing parameters and are not scalable with data dimensions. DeepIV [10] develops deep neural networks [27] for two stages. First, they fit the conditional distribution of the treatment given the instrument variable and the observational covariates. Then, they incorporate the fitted conditional treatment distribution to minimize the loss function. DeepIV is an exciting work that is computationally efficient and scales with high dimensional data. But the validation of DeepIV depends on the model from the first stage, and samples used for MC approximate for integral at the second stage affect the prediction accuracy of the model used at the second stage. The two-stage neural networks framework of DeepIV makes it hard to tune hyperparameters.

## III. Preliminary Knowledge

### A. Causal inference

We describe the toy smoking example in Introduction using graphical model [28] to represent relationships among the treatment, the confounders and the outcome (see Fig. 1). The nodes describe random variables, and the arrows represent causal relationships. The treatment $T$ affects the outcome $Y$, and the confounders $X$ affects both the treatment $T$ and the outcome $Y$. We also use the structural equation model [29] to represent these relationships. The relationships are described with deterministic functions. We describe the relationships as the model, $M$, using Eq. (1).

$$\begin{aligned} x &= f_x(\epsilon_x), \\ t &= f_t(x, \epsilon_t), \\ y &= f_y(t, x, \epsilon_y). \end{aligned} \tag{1}$$

The $\epsilon_x, \epsilon_t$ and $\epsilon_y$ are random disturbances representing background factors selected not to include in the analysis. The causes of a random variable should be included in the function as independent factors ($T$ and $X$ are causes of $Y$, so $T$ and $X$ are independent factors in the function $f_y$, and $Y$ is dependent on $T$ and $X$). The goal is to predict the outcome $Y$ from the variables $(X, T)$.

We assume that $Y$ is structurally determined by an unknown and potential non-linear continuous function of $X$ and $T$,

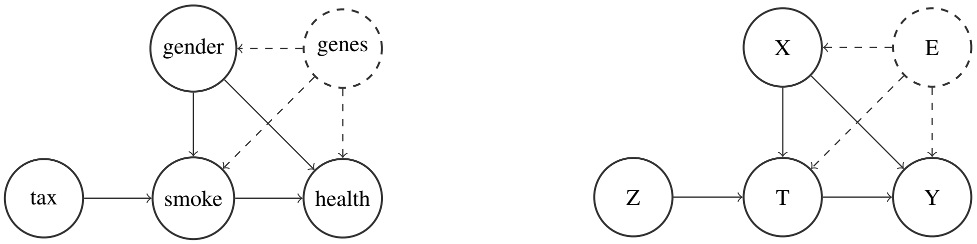
$$y = g(t, x) \tag{2}$$

Suppose a girl wants to know the effect of smoking on a her health. The effect should be $y^{'} - y^{''}$ where $y^{'}$ is the health if she smokes and $y^{''}$ is her health if she does not smoke. According to Eq. (2), $y^{'} - y^{''} = g(t^{'}, x) - g(t^{''}, x)$, where function $g$ is hidden (only god knows its form). Hence, we only need to estimate function $g(t, x)$. In the supervised learning framework, we normally approximate $g(t^{'}, x)$ and $g(t^{''}, x)$ by $\mathbb{E}[y|t^{'}, x]$ and $\mathbb{E}[y|t^{''}, x]$, respectively. Then, we have $y^{'} - y^{''} = \mathbb{E}[y|t^{'}, x] - \mathbb{E}[y|t^{''}, x] = g(t^{'}, x) - g(t^{''}, x)$. Therefore, we can obtain an unbias estimate of causal effect. However, such unbias estimate only exists when there is no unobserved confounders. Next, we will introduce how instrumental variable methods resolve this problem.

### B. Causal inference with instrumental variable

Again, we describe the toy smoking example using the causal graphical model [30] as shown in Fig. 2a. The dashed nodes represent unobserved variables. Links emanating from unobserved variables are designated by dashed arrows. Links connecting observed quantities are designated by solid arrows. This figure says: People of different genders tend to smoke at different degree (i.e., gender → smoke) and also have different health conditions (i.e., gender → health); genes that are unmeasured and decide people's gender (i.e., genes → gender), and they are also believed to affect people's health (i.e., genes → health) and may lead people to smoke (i.e., genes → smoke); and tax rate for tobacco is an instrumental variable because it affects the smoking via the price of cigarette (i.e., tax → smoke) but does not have a direct effect on the health beyond its effect on the smoking.

We generalize the above toy example to a causal graphical model shown in Fig. 2b. The observed confounders $X$ and the unobserved confounders $E$ affect both the treatment $T$ and the outcome $Y$. The unobserved confounders $E$ also affect observed confounders $X$. The treatment $T$ is also affected by an instrumental variable $Z$. The outcome $Y$ is affected by the treatment $T$ and the (observed and unobserved) confounders. To make the graphical model easy to understand, we also use a structural equation model [29] to express relationships of these variables (see Eq. (3)). The treatment $T$ is decided by the instrumental variable $Z$ and the confounders and some other noise. The outcome $Y$ is a function of $X$, $T$ and $E$. In our smoking example, the treatment $T$ is smoking, the outcome



(a) The toy example of smoking affecting general health.

(b) Causal graphical model for the instrumental variable.

Fig. 2: (Left) The causal graphical model of smoking affecting general health. Smoking is the treatment; health is the outcome; gender is the observed confounder. The unobserved variable genes decide gender, affect people's smoking, and also influences people's health. Tobacco tax is the instrumental variable, which affects health by smoking. (Right) Causal graphical model for the instrumental variable. $T$ is the treatment and $Y$ is the outcome. $X$ represents the observed confounders affecting the treatment $T$ and the outcome $Y$. $Z$ represents the instruments and $E$ represents unobserved variable influencing $X$, $T$, and $Y$.

$Y$ is health, the observed confounders $X$ is gender, and the unobserved confounders $E$ is genes.

$$\begin{aligned} z &= f_z(\epsilon_z), \\ e &= f_e(\epsilon_e), \\ x &= f_x(e, \epsilon_x), \\ t &= f_t(z, x, e, \epsilon_t), \\ y &= f_y(x, t, e) \end{aligned} \tag{3}$$

The goal is still to predict the outcome $Y$ from the variables $(X, T)$. Since we do not have the information of genes, it is not possible for us to estimate the causal effect of smoking on health just using the information of $X$, $T$ and $Y$.

First, we need to make some assumptions about the data generating process of the outcome. We use the same setting for $Y$ as in [10], that is $Y$ is structurally determined by an unknown and potential non-linear continuous function of $X$ and $T$ and an additive $E$,

$$y = g(t, x) + e \tag{4}$$

According to Eq. (4), causal effect is $y^{'} - y^{''} = (g(t^{'}, x) + e) - (g(t^{''}, x) + e) = g(t^{'}, x) - g(t^{''}, x)$ because $e$ of one person is constant. With the observational data $\{x, t, y\}$ of many people in hand, we normally use supervised learning methods to use $\mathbb{E}[y|t^{'}, x]$ and $\mathbb{E}[y|t^{'}, x]$ as estimation of $g(t^{'}, x)$ and $g(t^{''}, x)$, and then we have $y^{'} - y^{''} = \mathbb{E}[y|t^{'}, x] - \mathbb{E}[y|t^{''}, x]) = g(t^{'}, x) - g(t^{''}, x) + \mathbb{E}[e|t^{'}, x] - \mathbb{E}[e|t^{''}, x]$. Since $\mathbb{E}[e|t^{'}, x] - \mathbb{E}[e|t^{''}, x]$ is not definitely zero, there will exist a bias if we use this method.

We need additional information in order to get an unbiased estimate of the causal effect $y^{'} - y^{''}$, i.e., estimate of $g(t, x)$. Instrumental variable methods introduce a variable that is independent of unobserved confounders and affects the treatment but does not have a direct effect on the outcome beyond its effect on the treatment. They satisfy the following three assumptions

1) **Relevance** $T \not\perp\!\!\!\perp Z|X$, which means $Z$ is correlated with the treatment conditioning on the observed confounders.
2) **Exclusion** $Z \perp\!\!\!\perp Y|(X, T, E)$, $Z$ does not affect the outcome directly except affecting the outcome through affecting the treatment.
3) **Mean Independence** $\mathbb{E}[e|x, z] = 0$ for any $(x, z)$, the expectation of $E$ conditioning on $X$ and $Z$ is zero for any $(X, Z)$.

We show the variable $Z$ in Fig. 2b is an instrumental variable if $\mathbb{E}[e|x, z] = 0$. The arrow between the treatment $T$ and the variable $Z$ represents assumption **Relevance** which means the variable $Z$ is correlated with the treatment $T$ conditional on the observed confounders $X$. The lack of an arrow between $Z$ and $Y$ represents assumption **Exclusion**. And we have $\mathbb{E}[e|x, z] = 0$, so assumption **Mean Independence** holds.

The idea behind instrumental variable methods is to use the conditional expectation of $Y$ on $Z$ and $X$ to estimate $Y$ instead of using the conditional expectation of $Y$ on $T$ and $X$. Taking conditional expectation on both sides of Eq. (4) and $\mathbb{E}[y|x, z]$ could be described as the following form by using the assumptions above, that is

$$\mathbb{E}[y|x, z] = \int g(t, x) dF(t|x, z) \tag{5}$$

where $F(t|x, z)$ is the conditional treatment distribution. We can see that there is no unobserved confoudners in the Eq. (5), so there is no influence from the unobserved confoudners. Then the problem becomes to estimate $g(t, x)$ from the empirical conditional distribution function of $F(t|x, z)$ and the empirical conditional expectation of $\mathbb{E}[y|x, z]$. Most instrumental variable methods apply two-stage solutions. A two-stage solution firstly regresses the treatment $T$ on the variables $X$ and $Z$, and then estimates the effect of predicted $T$ on $Y$.

Usually, two-stage methods has a disadvantage that estimating the conditional treatment distribution cannot utilize the information from the outcome generating function (e.g., DeepIV [10]), so the final performance of causal effect estimation is restricted because outcome is one significant component of observational data. Apparently, merging two stages into one stage could make the two stages mutually learn from each other, but it is not trivial.

## IV. One-Stage Instrumental Variable Method

We notice that if the treatment is categorical (assume $C$ categories), then Eq. (5) has a nice form as the following equation

$$\mathbb{E}[y|x,z] = \sum_{c=1}^{C} g(t_c,x)F(t_c|x,z) \tag{6}$$

Then it is possible to estimate the conditional treatment distribution $F(t_c|x,z)$ and the unknown function $g(t,x)$ at one stage. Intuitively, the problem should be treated as an optimization problem to find an estimated function $g$ to minimize the mean squared error of $\sum_i (y_i - \mathbb{E}[y|x_i,z_i])^2 = \sum_i (y_i - \sum_{c=1}^{C} g(t_c,x)F(t_c|x,t_c))^2$. But, $F(t|x,z)$ is still unknown in the equation above. Our idea is to add the information of data likelihood of the empirical conditional treatment distribution. Thus, the overall idea is to combine the mean squared error $\sum_i (y_i - \mathbb{E}[y|x_i,z_i])^2$ and the likelihood of the empirical conditional treatment distribution in the loss function.

We transform the Eq. (6) to the following optimization problem,

$$\min_{\hat{g}\in\mathcal{G}} \sum_i \left[ y_i - \sum_{c=1}^{C} \hat{g}(t_c,x_i)F(t_c|x_i,z_i) \right]^2 \tag{7}$$

Such optimization problem is to find a best-fitted $\hat{g}$ from functional space $\mathcal{G}$ to minimize the error $\sum_i (y_i - \mathbb{E}[y|x_i,z_i])^2$. Since there is an unknown $F(t|x,z)$ in the objective function, we still need to estimate the conditional treatment distribution. The basic idea is to add the likelihood of the empirical conditional treatment distribution as the evaluation of distribution estimation. For this goal, we design a loss function which is a simple linear combination of $\sum_i \left[ y_i - \sum_{c=1}^{C} \hat{g}(t_c,x_i)\hat{F}(t_c|x_i,z_i) \right]^2$ and negative logarithm likelihood of $\hat{F}(t_i|x_i,z_i)$. Finally, we have the loss function as follow

$$\min_{\theta,\phi} w_1 \sum_i \left[ y_i - \sum_{c=1}^{C} \hat{g}_\theta(t_c,x_i)\hat{F}_\phi(t_c|x_i,z_i) \right]^2 + w_2 \sum_i -\log \hat{F}_\phi(t_i|x_i,z_i) \tag{8}$$

We use a deep neural network architecture for our one-stage instrumental variable method (1SIV) to estimate these two quantities jointly. Assume the labels of the treatment are $0,1,\ldots,C-1$. First, we describe how to estimate the conditional treatment distribution. The output of the fitted conditional treatment probability mass function is the probability of the treatment belonging to each category. That is, the output is a vector of $(\Pr(t=0|x,z), \Pr(t=1|x,z), \ldots, \Pr(t=C-1|x,z))$ for every input of $(x,z)$. The output could be realized simply by a softmax activation function. Similarly, in order to obtain an estimate of the outcome generating function, we estimate a vector of $(g(t=0,x), g(t=1,x), \ldots, g(t=C-1,x))$ for every input $x$. It is easy to use a straightly linear activation function here.

The observed confounders $X$ may share, or not share, a common part with the treatment and the outcome. In our experience, we notice that the observed confounders $X$ often have different relationships with the treatment $T$ and the outcome $Y$. That is treatments and outcomes usually have different complex relations with observed confounders. For example, the function $t = f_t(z,x,e,\epsilon_t)$ from Eq. (3) may be a linear model with $x$, and the function $y = f_y(x,t,e)$ probably has a non-linear relationship with $x$. Some two-stage instrumental variable methods lack the ability to capture the complex relationships, for example, 2SLS [11] is restrictive with its linear assumptions.

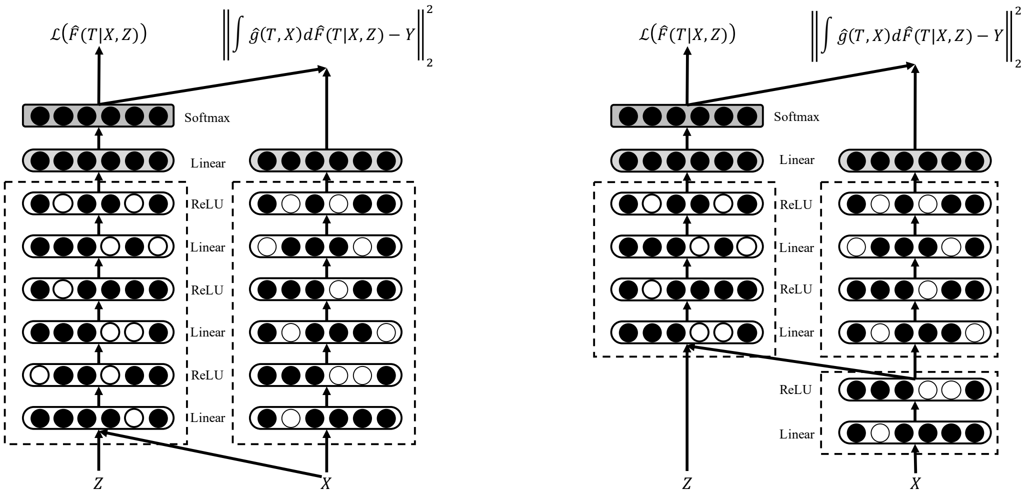
To capture these two different relationships, we design two separate latent representations for the observed confounders $X$ in our neural network. We present our one-stage instrumental variable method framework in Fig. 3a. The output for the conditional treatment probability mass function $\Pr(t|x,z)$ uses a representation from the instrument variable $z$ and the observed confounders $x$. And the output of the target outcome generating function $\hat{g}(t,x)$ uses another representation of the observed confounders $x$. The multi-output neural network fits the conditional treatment probability mass function $\Pr(t|x,z)$ and estimates the target outcome generating function $\hat{g}(t,x)$ simultaneously. The proposed loss function uses the information of the fitted conditional treatment distribution and integration of the target function over the conditional treatment distribution.

If the confounders have a common relationship with the treatment and the outcome, we need to show that our one-stage neural network architecture has the ability to capture the common complex relationships. We design a neural network (we call the neural work as 1SIV-S) with the same architecture as our proposed network above but taking a shared latent representation of observed confounders $X$ as input (see Fig. 3b) instead of taking directly observed confounders $X$ as input. 1SIV-S aims to obtain strong representation ability to capture the common relationship of observed confounders with treatments and outcomes.

Now we describe the procedure for out-of-sample validation. Validation of methods for causal inference usually suffers from no samples of counterfactual outcome [6]. Fortunately, our method uses only observational data. A simple procedure to validate is to evaluate the loss of validation samples. We use the simple linear combination proposed above for our method. An alternative validation method is to use new combinations of the square error $\sum_i (y_i - \mathbb{E}[y|x_i,z_i])^2$ and negative log likelihood of $\hat{F}(t_i|x_i,z_i)$.

## V. Experiments

To evaluate the effectiveness of the proposed method, we will design a series of experiments in this section. We first introduce the data simulation method and then explain the basic experimental setting. We then move to the evaluation of different perspectives of the proposed model, including hyperparameter sensitivity, one stage, and complex relationship learning using deep neural networks.



$\mathcal{L}(\hat{F}(T|X,Z))$
$\left\|\int \hat{g}(T,X)d\hat{F}(T|X,Z)-Y\right\|_2^2$
Softmax
Linear
ReLU
Linear
ReLU
Linear
ReLU
Linear
Z
X
$\mathcal{L}(\hat{F}(T|X,Z))$
$\left\|\int \hat{g}(T,X)d\hat{F}(T|X,Z)-Y\right\|_2^2$
Softmax
Linear
ReLU
Linear
ReLU
Linear
ReLU
Linear
Z
X

(a) One-stage neural network architecture without a shared latent representation of the observed confounders $X$ for causal inference.

(b) One-stage neural network architecture with a shared latent representation of the observed confounders $X$ for causal inference.

Fig. 3: $\mathcal{L}$ represents the loss function. The multi-output neural network applies latent representations of the observed confounders $X$. The estimate of the conditional treatment distribution also uses information from the instrumental variable $Z$.

### A. Data simulation

Since real non-experimental observational data has no ground truth, we conduct simulation experiments to validate the proposed method align with the studies in this literature. We simulate data according to the causal graphical model in Fig. 2b. We simulate a binary treatment $t$, a continuous outcome $y$, the observed confounders $x$, the unobserved confounders $e$ and the instrumental variable $z$. We use a non-linear function $\psi(x) = \exp(x-0.5)+(x-1)^2-x/2$ to describe the complex relationship of $x$ and $y$. The unobserved variable $e$ has an zero expectation. The overall process is as follow,

$$\begin{aligned} z &\sim \mathcal{N}(0,1), \\ e &\sim \mathcal{N}(0,2^2), \\ x &\sim \mathcal{N}(0.3+0.2e,2^2) \\ t &\sim Bern(\text{expit}(-0.3+0.5x+0.2z+e)) \\ \psi(x) &= \exp(x-0.5)+(x-1)^2-x/2, \\ y &= 100+(10+t)\psi(x)-20t+e \end{aligned} \tag{9}$$

where ‘expit’ is the logistic function. The number of training samples is five thousand in our simulation. The observational data is $\{z,x,t,y\}$.

In causal inference, we normally use a different data simulation process for test data comparing the process for training data. The new data simulation process for test data is based on randomized controlled trials [6]. We generate test samples by following: Firstly, we generate another five thousand samples using Eq. (9). Next, we redraw the treatment $t$ to make the treatment variable independent from the observed confounders $x$, the unobserved confounders $e$, the instrumental variable $z$, and other noise. Finally, we use the treatment $t$ and the observed confounders $x$ to predict the model. Such test method above is also used in DeepIV [10].

### B. Basic experimental setting

With the observational data $\{z,x,t,y\}$ in hand, our target is to estimate the outcome generating function $g(t,x)=100+(10+t)*\psi(x)-20t$. When we have a new observed data $\{x,t\}$, we can predict the causal effect of $t$ on $x$. We use the mean squared prediction error (RMSE) between estimated outcome generating function $\hat{g}$ and the ground truth $g$ as the evaluation metric for prediction.

For our proposed method, we use multi-layer perceptrons for our neural network and the network has three hidden layers with 128, 64, and 32 units, respectively. The network uses the ReLU activation function and dropout regularization. We optimize the models using Adam [31] with learning rate = 0.01, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 1e-8$. We train our model for 20 epochs with a batch size of 128 and a dropout rate of 0.5. We set $w_1 = 0.0017$ and $w_2 = 1$ in the loss function.

### C. Evaluation on hyperparameter sensitivity

We conduct experiments to analyze the sensitivity of the hyperparameters for our method and observe the method behavior under changing hyperparameters within a selected range. The two hyperparameters are: dropout rate and batch size. Each value of hyperparameters within the range is independently tested ten times and statistics of ten root of mean squared errors are reported. When testing one hyperparameter, we keep other hyperparameters constant.

As for the dropout rate, the range is from 0.5 to 0.9 and the step is 0.1. The results is shown in Fig. 4. We see from the

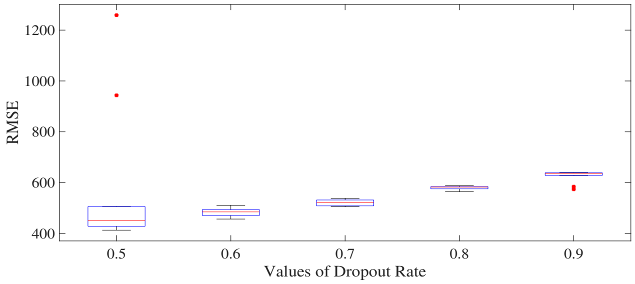


Fig. 4: Sensitivity analysis for different dropout rates. We change the dropout rate from 0.5 to 0.9 in steps of 0.1. The RMSE is used as the evaluation metric.

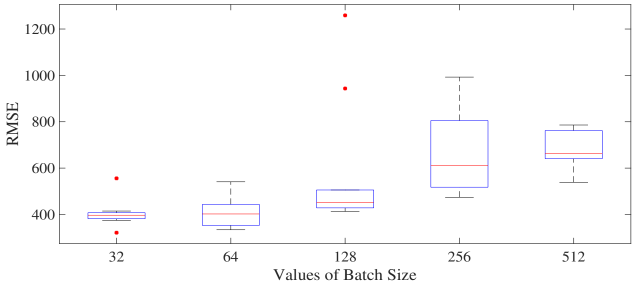


Fig. 5: Experimental analysis for the influence of different batch size values. The values of the batch size tested are 32, 64, 128, 256 and 512. We use the RMSE as the evaluation metric.

figure that the performance of our method gradually decreases with the increasing of dropout rate, and the variance of the prediction also decreases with the increasing of dropout rate. The reason may be that the deep neural networks in our method are used for regression task, the high dropout rate may hurt the performance although the high dropout rate makes increase the model converge rate (the variance is lower).

As for the batch size, we increase its value from 32 to 512 in rates of 2. The result is shown in Fig. 5. The performance of our approach stays stable when the batch size is set to 32, 64 or 128. However, the performance drops significantly and the variance increases as well after the batch size is set to 256 or greater. Our proposed neural network prefers batch sizes below 128.

### D. Evaluation on propensity scores

Causal inference often faces difficulties when propensity scores are low (low rates of objects tend to receive the treatment). As for propensity scores (i.e., the conditional probability of treatment assignment given the observed confounders), we simulate data using Eq. (9) with only the treatment assignment changed as $t \sim Bern(\text{expit}(\alpha_t - 1.5x + 1.7z + e))$. We change $\alpha_t$ from -4.7 to 0.3 in steps of 1 to in order to change the propensity scores. Generally, data with low values of $\alpha_t$ have low propensity scores. We simulate ten datasets for each value of $\alpha_t$. We test our proposed method in these datasets. All the data is tested using the same hyperparameters. The result is shown in Figure 6. The performance gradually

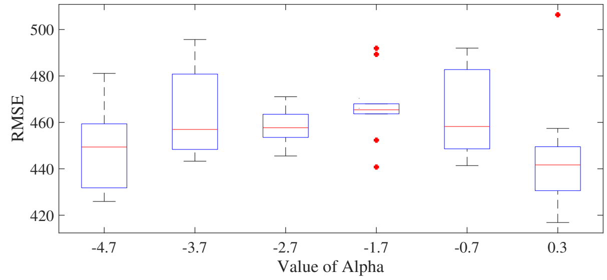


Fig. 6: Experimental analysis for the influence of different propensity scores. As the values of $\alpha_t$ increased, the mean propensity scores increased. The RMSE is used as the evaluation metric.

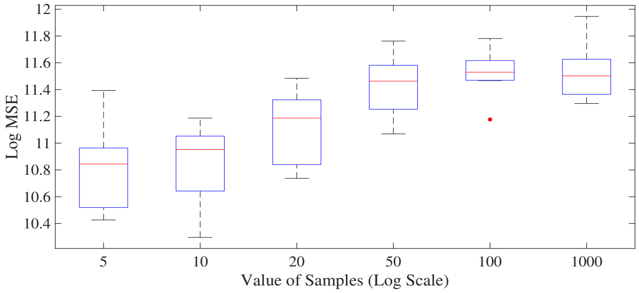


Fig. 7: The performance (logarithm mean squared error) of DeepIV using different values of samples drawn from the associated probability distribution. We present the values of the samples in a logarithm scale on the horizontal axis.

decrease from -4.7 to -1.7 and achieves the worst at -1.7. After that, the performance keeps increasing. The reason is related to the functional form of propensity scores and its influence to the outcome function.

### E. Evaluation on effectiveness of one stage

One of the main contributions of this study is the one-stage instrumental variable method which effectiveness will be verified here.

Before we compare its performance with two-stage methods. We first analyze a state-of-the-art two-stage method: DeepIV [10]. DeepIV fits the conditional treatment distribution at the first stage and uses MC approximate integral in Eq. (5) via sampling from the estimated distribution. In order to show the impact from sampling to final performance, we test the performance of different samples (5, 10, 20, 50, 100 and 1000 samples) drawn from the associated probability distribution. We run ten times for each value of samples. The influence of different values of samples is presented in Fig. 7. It is clear that more samples did not tend to achieve better performance. The reason is that the estimated probability distribution from the first stage has a bias with the true distribution, and it brings other bias for MC approximate for integral in the second stage using the samples drawn from the estimated distribution. The more accurate in the first stage, the more bias will bring to the second stage, like ‘overfitting’. Hence, it could be hard

TABLE I: The performance (RMSE) of estimating the outcome generating function from simulation data. The treatments and outcomes have less common relationships with observed confounders. Our proposed methods are 1SIV and 1SIV-S. DeepIV is a two-stage instrumental variable method.

| Dataset | 1SIV | DeepIV | 1SIV-S |
|---|---|---|---|
| sd-1 | **358.97** | 642.67 | 456.65 |
| sd-2 | **446.09** | 640.81 | 449.80 |
| sd-3 | 514.58 | 639.11 | **464.29** |
| sd-4 | 533.72 | 641.22 | 475.19 |
| sd-5 | 477.23 | 644.50 | **453.72** |
| sd-6 | **495.79** | 644.24 | 544.43 |
| sd-7 | **455.62** | 642.51 | 473.47 |
| sd-8 | **396.04** | 643.27 | 469.50 |
| sd-9 | **426.23** | 643.01 | 449.95 |
| sd-10 | 379.63 | 645.52 | 389.47 |

to tune the neural networks in the two stages to get a good performance.

Next, we compare the performance of the proposed method and DeepIV. We use the same hyperparameters (i.e., dropout rate, batch size, and layer number). The result is given in Table I where the performance for each dataset is described in one row. From the table, we can see that our proposed one-stage method is better than two-stage method DeepIV. One may concern that our proposed method performs better than DeepIV is due to the selected hyperparameters. To clear that concern, we test different hyperparameters using above generated data $sd-1$. We change one hyperparameter and keep other hyperparameters constant during our test. The values of dropout are 0.6 to 0.9, the values of batch size are 32, 64, 256, and 512. Additionally, we set epochs as 50, 100, to 300. The results are summarized in Table II. We see that 1) our method, including 1SIV and 1SIV-S, performs better than DeepIV on all settings except one with batch size 512; 2) 1SIV is better than 1SIV-S on most settings. Hence, we can safely claim that 1) our one-stage method can achieve better performance than two-stage methods in this experiment; 2) separate representations for observed confounders is better than shared representations in this experiment, but it is not always right because it depends on the hidden data generating process.

We design a new experiment to show that shared representation for observed confounders is better for some data. The new experiment is the same with above but only has a different treatment generating process with Eq. (9). We generate the treatment by $t \sim Bern(\text{expit}(-0.3 + \psi(x)(0.2 - 2.4z) + e))$, where $\psi(x)$ is a common part of the treatment generating process and the outcome generating process. We implement ten simulations for the generating process. The result is given in Table III. We can see that 1) DeepIV achieves a similar performance as its performance in the former simulation data. It seems the shared relationships of $X$ with $T$ and $Y$ does not affect DeepIV much; 2) As our expectation, 1SIV-S obtains better performance than 1SIV and DeepIV. The better performance of 1SIV-S is mostly because its outputs take a shared latent representation of the observed confounders $X$.

TABLE II: The experimental analysis for different hyperparameters applied to neural networks. The evaluation metric is RMSE. Three different methods are tested here, 1SIV and 1SIV-S are our proposed methods, and DeepIV is a traditional two-stage instrumental method with neural networks.

| Hyperparameter | Value | 1SIV | DeepIV | 1SIV-S |
|---|---|---|---|---|
| dropout | 0.6 | **461.92** | 642.46 | 473.34 |
| | 0.7 | **505.81** | 641.91 | 523.98 |
| | 0.8 | 575.73 | 644.16 | **564.65** |
| | 0.9 | 633.33 | 645.25 | **617.10** |
| batch size | 32 | **350.58** | 641.99 | 420.19 |
| | 64 | 385.28 | 643.95 | **368.21** |
| | 256 | 713.12 | 636.72 | **511.55** |
| | 512 | 651.24 | **636.38** | 670.48 |
| epochs | 50 | **151.90** | 646.39 | 329.89 |
| | 100 | **113.25** | 641.06 | 292.86 |
| | 300 | **113.99** | 635.73 | 132.91 |

TABLE III: The experimental results (RMSE) of simulation data that treatments and outcomes share some relationships with observed confounders. 1SIV and 1SIV-S are our proposed approaches. DeepIV uses a framework of two-stage neural networks.

| Method | 1SIV | DeepIV | 1SIV-S |
|---|---|---|---|
| sd2-1 | **412.95** | 643.00 | **411.93** |
| sd2-2 | **348.69** | 642.31 | **361.34** |
| sd2-3 | 524.14 | 639.45 | **359.40** |
| sd2-4 | **419.14** | 642.11 | 508.10 |
| sd2-5 | **425.49** | 645.09 | **452.31** |
| sd2-6 | **431.38** | 645.78 | **461.22** |
| sd2-7 | 523.03 | 645.48 | **482.83** |
| sd2-8 | **446.28** | 643.56 | **452.58** |
| sd2-9 | **379.33** | 643.77 | **357.49** |
| sd2-10 | **411.43** | 644.75 | **411.43** |

TABLE IV: The experimental performance (RMSE) of neural networks applied with different hyperparameters. The treatments and outcomes share a common function with the observed confounders. Our proposed approaches are 1SIV and 1SIV-S. DeepIV applies two-stage neural networks.

| Parameter | Value | 1SIV | DeepIV | 1SIV-S |
|---|---|---|---|---|
| dropout | 0.6 | 493.66 | 642.26 | **450.95** |
| | 0.7 | **506.28** | 641.08 | 509.05 |
| | 0.8 | 581.07 | 643.34 | **563.11** |
| | 0.9 | 638.59 | 644.61 | **617.19** |
| batch size | 32 | **374.46** | 640.11 | 407.12 |
| | 64 | 409.93 | 643.20 | **394.12** |
| | 256 | 474.00 | 636.32 | **442.03** |
| | 512 | 538.62 | 635.52 | **523.42** |
| epochs | 50 | 299.87 | 645.05 | **203.79** |
| | 100 | 298.74 | 641.91 | **165.47** |
| | 300 | 157.18 | 640.83 | **147.97** |

TABLE V: The performance (RMSE) of estimating the outcome generating function from simulation data. Our proposed 1SIV and 1SIV-S are one-stage methods. 2SLS and NonPar are two-stage instrumental variable methods.

| Dataset | 1SIV | 2SLS | NonPar | 1SIV-S |
|---|---|---|---|---|
| sd-1 | **358.97** | 649.25 | 427.18 | 456.65 |
| sd-2 | **446.09** | 635.93 | 489.89 | 449.80 |
| sd-3 | 514.58 | 637.15 | 496.11 | **464.29** |
| sd-4 | 533.72 | 636.25 | **56.78** | 475.19 |
| sd-5 | 477.23 | 637.19 | 508.34 | **453.72** |
| sd-6 | **495.79** | 642.89 | 538.79 | 544.43 |
| sd-7 | **455.62** | 638.82 | 520.89 | 473.47 |
| sd-8 | **396.04** | 743.45 | 502.92 | 469.50 |
| sd-9 | **426.23** | 644.80 | 540.78 | 449.95 |
| sd-10 | 379.63 | 671.76 | **90.58** | 389.47 |

We notice that 1SIV has a comparative result with 1SIV-S. The performance of 1SIV does drop but not much compared with the performance in the former simulation. We also analyze the influence of different hyperparameters to the final performance. The experiment setting is the same with above. We see that 1) 1SIV-S has the strongest ability to capture the common part of the treatment generating process and the outcome generating process; 2) The performance of 1SIV is comparable with 1SIV-S.

### F. Evaluation on complex relationship modeling

To verify the effectiveness of deep neural network architecture, we compare the proposed method with a linear and kernel-based methods:

- two-stage least squares regression (2SLS) [11] first estimates the treatment using a linear regression $t = v + \alpha_1 x + \alpha_2 z$, then regresses $y$ on $\hat{t}$ and $x$ by $y = e + \beta_1 x + \beta_2 t$. We use python package linearmodels [32] to conduct the 2SLS estimation.
- NonPar [12] uses an estimation procedure for $g(t, x)$ based on Tikhonov regularization. We use R package np [33] to complement the nonparametric instrumental variable estimation.

We use the generated datasets (i.e., sd-1 to sd-10) in Table I and summarize the results of these methods in Table V. The performance of these datasets is present as sd-1, sd-2,... in each row. We can see that our proposed approach 1SIV outperforms the other methods in most datasets. The reason for the poor performance of 2SLS is due to its linearity and homogeneity assumptions. NonPar has very good performance in two datasets, but it is worthy of notice that such method is extremely slow in practice. Thus, it is not suitable for large datasets. Therefore, we can draw the conclusion that our designed deep neural network is stronger than other methods on complex relationship modeling.

## VI. Conclusion and Future Work

We have proposed a one-stage deep instrumental variable method for causal inference with unobserved confounders. A new deep neural network architecture is specially designed for the one-stage method, which contains two strategies to handle representation of observed confounders: separate and shared. The experiments have verified that one-stage method is generally better than two-stage method on causal effect estimation in our experiments, and the comparative advantages of deep neural network on complex hidden relationship modeling has also been demonstrated. Besides, our method has good performance with data of low propensity scores.

The limitations of our approach are the lack of ability to estimate uncertainty over predictions, and it is restrictive to categorical treatments. In our follow-up study, we concentrate on combining Bayesian methods and deep learning which has the potential to explicitly estimate the uncertainty in predictions and extend to more general treatment situations.

## Acknowledgments

This work is supported by the Australian Research Council under Discovery Grant DP170101632 and DP190101733.

## References

[1] P. Spirtes and K. Zhang, “Causal discovery and inference: concepts and recent methodological advances,” in *Applied informatics*, vol. 3, no. 1. SpringerOpen, 2016, p. 3.

[2] J. L. Hill, “Bayesian nonparametric modeling for causal inference,” *Journal of Computational and Graphical Statistics*, vol. 20, no. 1, pp. 217–240, 2011.

[3] A. M. Alaa and M. van der Schaar, “Bayesian inference of individualized treatment effects using multi-task gaussian processes,” in *Advances in Neural Information Processing Systems 30*. Curran Associates, Inc., 2017, pp. 3424–3432.

[4] S. Athey and G. Imbens, “Recursive partitioning for heterogeneous causal effects,” *Proceedings of the National Academy of Sciences*, vol. 113, no. 27, pp. 7353–7360, 2016.

[5] F. Johansson, U. Shalit, and D. Sontag, “Learning representations for counterfactual inference,” in *Proceedings of The 33rd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 48. PMLR, 2016, pp. 3020–3029.

[6] U. Shalit, F. D. Johansson, and D. Sontag, “Estimating individual treatment effect: generalization bounds and algorithms,” in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 70. PMLR, 2017, pp. 3076–3085.

[7] Y. L. M. H. J. G. A. Z. Liuyi Yao, Sheng Li, “Representation learning for treatment effect estimation from observational data,” in *Neural Information Processing Systems*, 2018.

[8] N. Kallus, A. M. Puli, and U. Shalit, “Removing hidden confounding by experimental grounding,” in *Advances in Neural Information Processing Systems 31*. Curran Associates, Inc., 2018, pp. 10 911–10 920.

[9] J. D. Angrist, G. W. Imbens, and D. B. Rubin, “Identification of causal effects using instrumental variables,” *Journal of the American statistical Association*, vol. 91, no. 434, pp. 444–455, 1996.

[10] J. Hartford, G. Lewis, K. Leyton-Brown, and M. Taddy, “Deep IV: A flexible approach for counterfactual prediction,” in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 70. PMLR, 2017, pp. 1414–1423.

[11] J. Kmenta, “Mostly harmless econometrics: An empiricist’s companion,” 2010.

[12] S. Darolles, Y. Fan, J.-P. Florens, and E. Renault, “Nonparametric instrumental regression,” *Econometrica*, vol. 79, no. 5, pp. 1541–1565, 2011.

[13] N. Kreif and K. DiazOrdaz, “Machine learning in policy evaluation: new tools for causal inference,” *arXiv preprint arXiv:1903.00402*, 2019.

[14] E. A. Stuart, “Matching methods for causal inference: A review and a look forward,” *Statistical science: a review journal of the Institute of Mathematical Statistics*, vol. 25, no. 1, p. 1, 2010.

[15] A. Diamond and J. S. Sekhon, “Genetic matching for estimating causal effects: A general multivariate matching method for achieving balance in observational studies,” *Review of Economics and Statistics*, vol. 95, no. 3, pp. 932–945, 2013.

[16] N. Kallus, “A framework for optimal matching for causal inference,” in *Artificial Intelligence and Statistics*, 2017, pp. 372–381.

[17] E. Leuven and B. Sianesi, “Psmatch2: Stata module to perform full mahalanobis and propensity score matching, common support graphing, and covariate imbalance testing,” 2018.

[18] P. R. Rosenbaum and D. B. Rubin, “The central role of the propensity score in observational studies for causal effects,” *Biometrika*, vol. 70, no. 1, pp. 41–55, 1983.

[19] P. C. Austin, “The performance of different propensity score methods for estimating marginal hazard ratios,” *Statistics in medicine*, vol. 32, no. 16, pp. 2837–2849, 2013.

[20] W. Luo, Y. Zhu, and D. Ghosh, “On estimating regression-based causal effects using sufficient dimension reduction,” *Biometrika*, vol. 104, no. 1, pp. 51–65, 2017.

[21] M. J. Van der Laan, E. C. Polley, and A. E. Hubbard, “Super learner,” *Statistical applications in genetics and molecular biology*, vol. 6, no. 1, 2007.

[22] C. Louizos, U. Shalit, J. M. Mooij, D. Sontag, R. Zemel, and M. Welling, “Causal effect inference with deep latent-variable models,” pp. 6446–6456, 2017.

[23] H. Bang and J. M. Robins, “Doubly robust estimation in missing data and causal inference models,” *Biometrics*, vol. 61, no. 4, pp. 962–973, 2005.

[24] M. J. Van Der Laan and D. Rubin, “Targeted maximum likelihood learning,” *The International Journal of Biostatistics*, vol. 2, no. 1, 2006.

[25] V. Chernozhukov, D. Chetverikov, M. Demirer, E. Duflo, C. Hansen, and W. Newey, “Double/debiased/neyman machine learning of treatment effects,” *American Economic Review*, vol. 107, no. 5, pp. 261–65, 2017.

[26] W. K. Newey and J. L. Powell, “Instrumental variable estimation of nonparametric models,” *Econometrica*, vol. 71, no. 5, pp. 1565–1578, 2003.

[27] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *nature*, vol. 521, no. 7553, p. 436, 2015.

[28] J. Pearl, “Bayesian networks: A model of self-activated memory for evidential reasoning,” in *Proceedings of the 7th Conference of the Cognitive Science Society, 1985*, 1985, pp. 329–334.

[29] J.Pearl, *Causality: models, reasoning and inference*. Cambridge university press, 2009.

[30] J. Pearl, “Causal diagrams for empirical research,” *Biometrika*, vol. 82, no. 4, pp. 669–669, 1995.

[31] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

[32] K. Sheppard, “linearmodels: Instrumental variable and linear panel models for Python,” 2017–, [Online; accessed today].

[33] T. Hayfield and J. S. Racine, “Nonparametric econometrics: The np package,” *Journal of Statistical Software*, vol. 27, no. 5, 2008.