Causal Effect Estimation combining Importance Sampling Weights and Variational Auto-Encoder

by

Zhizhong Lin

(Under the Direction of Nicole Lazar)

Abstract

Causal Inference has been increasingly drawing attention during the past decades, especially in this pandemic when people care more about effective protection approaches against COVID-19. The potential outcome framework is the primary theoretical framework for researchers to illustrate and estimate causal effects. However, this framework includes a controversial assumption called *Ignorability Assumption*, which is usually difficult to justify in real-world scenarios. Besides, the data size has grown tremendously during the past decades, bringing the "curse of dimensionality" to many traditional causal inference models. In this thesis, we propose a model named Importance-sampling Causal Effect with Disentangled Variational Auto-Encoder (ICEDVAE), which combines Causal Bayesian Network and Variational Auto-Encoder to estimate causal effects by relaxing the Ignorability Assumption and overcome the curse of dimensionality. Numerical studies show that our model is comparable with other State-Of-The-Art models and has the best performance when the treatment effect is homogeneous over different subjects.

Index words: [causal inference, observational data, causal Bayesian network, variational auto-encoder, directed acyclic graph, deep learning]

Causal Effect Estimation combining Importance Sampling Weights and Variational Auto-Encoder

by

Zhizhong Lin

B.S., University of Science and Technology of China, China, 2009
M.S., University of Science and Technology of China, China, 2016

A [Dissertation] Submitted to the Graduate Faculty of the University of Georgia in Partial Fulfillment of the Requirements for the Degree

doctor of philosophy

Athens, Georgia

2022

Causal Effect Estimation combining Importance Sampling Weights and Variational Auto-Encoder

by

Zhizhong Lin

Major Professor: Nicole Lazar

Committee: Jeongyoun Ahn
Sheng Li
Cheolwoo Park

Electronic Version Approved:

Ron Walcott
Vice Provost for Graduate Education and Dean of the Graduate School
The University of Georgia
August 2022

Contents

List of Figures

List of Tables

Chapter 1

Introduction

People seem to care more about causal relation than correlation between two events, such as "If my country or state had implemented a mask mandate three months ago, how many lives would have been saved from the COVID-19?" In daily life, however, the concepts of causality and correlation are usually misused. For example, some media states that wearing glasses can lower the chance of getting COVID-19 since they observe a lower prevalence of COVID-19 in the glasses-wearing population. They believe that glasses act like goggles and protect eyes from contacting the virus. However, it is possible that people wearing glasses read a lot and thus have more knowledge about how to protect themselves; or they are just couch potatoes who watch TV a lot at home and thus have less chance of being exposed to the virus.

The most rigorous way to derive causal effect is to conduct a Randomized Control Trial (RCT) by randomly assigning subjects into control and treatment groups. In the glasses-wearing example, we need to hire enough people to randomly assign each of them to a room full of COVID-19 virus to perform an RCT. Then we tell subjects in one room to wear glasses (control group) and not wear glasses (treatment group) in the other room. However, RCT is expensive, maybe unethical, or even infeasible (Pearl et al., 2009), and these problems are pretty obvious in the glasses-wearing example. First, hiring people and setting up the experimental environment is costly. Second, it is inhumane to force subjects to walk into a room filled with COVID-19.

To avoid those issues, drawing causal effects from an observational study (Rubin, 2007) provides a more attractive solution. Observational data are collected by researchers who observe subjects without interfering, which means there is no random assignment mechanism during the procedure, and researchers

record data based on objective observations. This procedure is much cheaper since data acquisition is convenient in this big-data era. Because researchers do not interfere with subjects, observational data perfectly bypass ethical problems. However, observational data also has limitations (Schwab et al., 2020). First, we cannot observe the "counterfactual" outcome, which means we can only observe the outcome corresponding to that treatment (the factual outcome) and do not know the outcome of the subject if he/she were assigned to other treatment groups. Although not observed in the RCT setting either, counterfactual outcomes can be easily estimated due to the random assignment mechanism. Second, subjects are not assigned randomly, which often exhibits selection bias (G. W. Imbens and Rubin, 2015). For example, at the early stage of the COVID vaccination, the doctors with higher infection risk were the first to get vaccinated (treatment group). Suppose we use these data to calculate the infection ratio and compare it with the one from the non-vaccinated groups to estimate the vaccine effectiveness (treatment effect). We may underestimate the vaccination effect since people in the treatment group are more likely to get infected. We can even obtain the exact opposite conclusion, which is a typical scenario of Simpson's Paradox (Blyth, 1972; Good and Mittal, 1987; Pearl et al., 2000). More specifically, suppose we have the observation data shown in Table 1.1, where the percentage reflects the infected rate. If we only check the number in the **Total** column, we will conclude that getting vaccinated will increase the chance of infection. However, if we take a deeper look at each column, we can find that the vaccinated group has a lower infection rate than the non-vaccinated group. The paradox happens because of the existence of the *confounding variable* "Doctor," which relates to both "vaccinated or not" (treatment) and the "chance of infection" (outcome). Since doctors are more likely to get vaccinated, the percentage in **Total** is closer to the larger subgroup (Doctor) in the vaccinated group. To be more specific, suppose Y is a binary random variable indicating whether a person is infected, X represents if a person is a doctor, and T is a binary variable with 1 as being vaccinated and 0 otherwise. Then $E[Y = 1|T = 1]$ is the infection proportion of the vaccinated group, which equals

$$\begin{aligned} E[Y=1|T=1] &= E[Y=1|X=1,T=1]P(X=1|T=1) \\ &\quad +E[Y=1|X=0,T=1]P(X=0|T=1) \\ &= 20\% \times \frac{500}{50+500} + 10\% \times \frac{50}{50+500} \\ &= 19\% \end{aligned}$$

by conditioning on X. Notice that 19% is closer to 20% since the proportion of doctors in the vaccinated group, $P(X = 1|T = 1)$, is much larger than $P(X = 0|T = 1)$. Similarly, 16% is closer to 15% since **Other** is the larger subgroup.

Table 1.1: Example of Simpson's Paradox

	$X = 0$ (Other)	$X = 1$ (Doctor)	Total
$T = 0$ (not vaccinated)	15% (210/1400)	30% (30/100)	16% (240/1500)
$T = 1$ (vaccinated)	10% (5/50)	20% (100/500)	19% (105/550)

Researchers have developed various frameworks to perform causal inference from observational data to overcome the challenges mentioned above. In this thesis, we will focus on the potential outcome framework (Rubin, 1974), which is also known as Neyman–Rubin Potential Outcomes or Rubin Causal Model. This framework assumes that each subject has several potential outcomes corresponding to different treatments. More specifically, for each subject-treatment pair, the outcome corresponding to that treatment, $Y(i)$, is the potential outcome of that subject (G. W. Imbens and Rubin, 2015), where i corresponds to the treatment. From now on, we assume the treatment variable is binary for simplicity, and thus there are two potential outcomes, $Y(0)$ and $Y(1)$, for each subject. Average Treatment Effect (ATE) and Conditional Average Treatment Effect (CATE) are two frequently used estimands for causal effect estimation. While ATE reflects the general treatment effect for the whole population, CATE focuses on the treatment effect at the subgroups level. Under the Potential outcome framework, ATE and CATE can be expressed as

$$\begin{aligned} ATE &= E[Y(1) - Y(0)] \\ CATE &= E[Y(1) - Y(0)|\mathbf{X} = \mathbf{x}] \end{aligned}$$

where $\mathbf{X}$ denotes the covariates representing subjects' pre-treatment attributes.

The potential outcome framework can formally express the causal effect clearly. In the previous vaccine example, doctors would have a particular infection rate if they were vaccinated $E[Y(1)|X = 1]$, where X here only includes the occupation information ("Doctor" or "Others"). $Y(i)$ is binary with value 1 as being infected and 0 otherwise. In contrast, doctors would have a different infection rate, $E[Y(0)|X = 1]$, if they were not vaccinated. To measure the causal effect of vaccination on doctors, we need to calculate the difference in the infection rates of the same group, i.e., $E[Y(1) - Y(0)|X = 1]$. However, the potential outcome framework does not solve the non-observable issue

of the counterfactual outcome. To overcome this critical obstacle, researchers propose several assumptions, which are **Stable Unit Treatment Value Assumption (SUTVA)**, **Consistency**, **Positivity** and **Ignorability** assumption (also known as **Unconfoundedness** assumption). These assumptions make it possible to estimate the counterfactual outcome even if we cannot observe it from the data. We will further explain these assumptions and emphasize their roles when estimating causal effect (such as ATE) in Section 2.1.

The potential outcome framework also clarifies why RCT is the golden standard for estimating the causal effect. In an RCT, subjects are randomly selected from the whole population, making the sampled potential outcome distributions representative of the whole population. Moreover, the potential outcome distributions are similar in both the control and treatment groups due to the random assignment mechanism, which makes $\{Y(0), Y(1)\} \perp T$. As a result, we have

$$E[Y(1)|T=0] = E[Y(1)|T=1]$$

We then can replace non-observed $E[Y(1)|T=0]$ with observed $E[Y(1)|T=1]$ and obtain an estimable estimand for ATE, i.e.

$$\begin{aligned} ATE &= E[Y(1)-Y(0)] = E[Y(1)-Y(0)|T=0] \\ &= E[Y(1)|T=0] - E[Y(0)|T=0] \\ &= E[Y(1)|T=1] - E[Y(0)|T=0] \quad (estimable) \end{aligned}$$

Both $E[Y(1)|T=1]$ and $E[Y(0)|T=0]$ can be easily estimated using $\frac{1}{n_1}\sum_{i:T_i=1} Y_i$ and $\frac{1}{n_0}\sum_{i:T_i=0} Y_i$ respectively, where $n_t, (t=0,1)$ means the number of subjects in the group with treatment $T=t$. A direct estimator for ATE in the RCT setting is then

$$\hat{ATE} = \frac{1}{n_1}\sum_{i:T_i=1} Y_i - \frac{1}{n_0}\sum_{i:T_i=0} Y_i \tag{1.1}$$

This simple calculation makes RCT the golden standard approach for estimating ATE. However, the confounding variable makes covariate distribution in the treatment group different from the ones in the control group, which leads to the change in potential outcome distribution and thus increases the difficulty of estimating the counterfactual outcome and causal effects. To sum up, missing counterfactuals is the major challenge in the potential outcome framework. The confounding variable worsens the situation and becomes the essential ob-

stacle to making causal inferences with observational data.

To solve the biased estimate of the causal effect brought by confounders in Table 1.1, we can condition on the confounding variable (Doctor) to eliminate its effect on the outcome (infection rate), which makes each subgroup like an RCT. The effect on the outcome then comes from the treatment (vaccination) only. After estimating the causal effect in each subgroup, we can conduct a weighted average of the treatment effect over the subgroup's distribution to obtain the treatment effect for the whole population. In Table 1.1, our goal is to estimate the effect of vaccination on the infection rate over the whole population, i.e. $E[Y(1) - Y(0)] = E[Y(1)] - E[Y(0)]$. Since $E[Y(1)] = E_X E[Y(1)|X]$ holds due to the **Positivity Assumption**, and

$$\begin{aligned} E[Y(1)|X] &= E[Y(1)|T=1,X] \quad (Ignorability) \\ &= E[Y|T=1,X] \quad (Consistency) \end{aligned}$$

we have

$$\begin{aligned} E[Y(1)] &= E[Y(1)|X=0]P(X=0) + E[Y(1)|X=1]P(X=1) \\ &= E[Y|X=0,T=1]P(X=0) + E[Y|X=1,T=1]P(X=1) \\ &= 10\% \times \frac{1400+50}{1400+50+100+500} + 20\% \times \frac{100+500}{1400+50+100+500} \\ &\approx 12.9\% \end{aligned}$$

Similarly, the infection rate for the non-vaccinated subgroup is

$$\begin{aligned} E[Y(0)] &= E_X[E[Y(0)|X]] \\ &= E[Y|X=0,T=0]P(X=0) + E[Y|X=1,T=0]P(X=1) \\ & 15\% \times \frac{1400+50}{1400+50+100+500} + 30\% \times \frac{100+500}{1400+50+100+500} \\ &\approx 19.4\% \end{aligned}$$

The result is much more reasonable than the one derived by directly checking the percentage in the **Total** column since now we have the infection rate of the vaccinated group lower than the non-vaccinated group. This "conditioning on" trick partially resolves the Simpson's Paradox and is a specific example of a more general method called *stratification*. It is also called as *sub-classification* or *blocking* (G. W. Imbens and Rubin, 2015), which is a typical method to adjust the effect of confounders. From the above application, we can see that the goal of stratification is to adjust the bias caused by the confounders, and the

basic idea is to split the entire group into subgroups where confounders have less effect on the outcome. More theoretically, stratification tries to construct a scenario similar to RCT in each subgroup and make the covariate distributions from different treatment groups as similar as possible. After that, the weighted average based on each subgroup's sample size is applied to estimate the ATE for the whole population.

Stratification is like a "global" method to balance the covariate distribution since it does not check the specific values of each covariate, which could lead to insufficient overlap of covariate distribution in some subgroups and cause high estimation variance. Compared with this "global" method, *matching* methods (Abadie et al., 2004; Iacus et al., 2012; G. W. Imbens, 2004; Rubin, 1973) are more "local" in a sense that it matches each treatment subject with the most "similar" one in the control group with some similarity metric. Different similarity metrics can induce different matching methods; examples are Euclidean distance (Rubin, 1973), Mahalanobis distance (Rubin and Thomas, 2000), and propensity score distance (Austin, 2011; Stuart, 2010), etc.. A classic matching procedure includes the following steps:

1. Randomly order the list of treated and control subjects, respectively.
2. Start with a non-paired treated subject and match it with the control subject with the smallest distance.
3. Remove the matched control subject from the control group
4. Move on to the next treated subject, match to the control with the smallest distance
5. Repeat step 2 through step 4

Researchers also propose different variations besides the one-on-one matching. For example, one-on-K nearest neighbor matching (Austin, 2011) matches K most similar subjects in the control group with the unpaired treatment subject. Key covariates matching (Stuart, 2010) performs matching based on some critical covariates instead of using all of them. The most significant advantage of matching methods is the algorithm speed, and there are many available R packages, such as *MatchIt* (Stuart et al., 2011), with good implementation. In sum, matching-based approaches reduce the estimation bias brought by confounders by pairing most similar control-treatment individuals and provide a way to estimate causal effect at the individual level.

Although matching methods are quite intuitive and can be very fast, they usually suffer from the local-optimum problem since they use the greedy strategy in the algorithm and ignore the global paired-distance minimization. To address this local-optimum issue, several revised matching methods were proposed to minimize the global distance measure and have been implemented in R packages such as *optmatch* (Hansen, 2007) and *rcbalance* (Pimentel, 2016). However, these global-optimum methods are, without doubt, computationally demanding since they apply a brute-force way to find the best pairs. Another trade-off is to determine the number of neighbors for matching. While large numbers may make the neighbors less qualified and result in estimation bias, fewer neighbors could decrease the bias but lead to high estimation variance due to low data utilization efficiency. Figure 1.1 depicts a simple example to reflect the low efficiency problem. In the figure, X is the only covariate taking the value $X = x$, and there is only one subject in the treatment group and nine subjects in the control group. If we apply one-on-one matching to the single treated subject, we will waste the remaining eight control subjects.

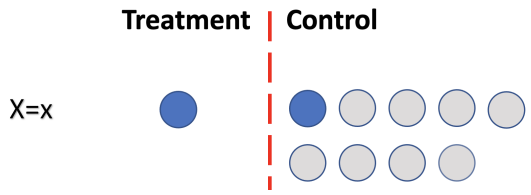

Figure 1.1: Imbalanced distribution for matching

Another issue for matching is that obtaining the distances among subjects will bring the "curse of dimensionality" (Richard, 1961) when the dimension of X increases. One way to avoid this problem is to choose the most "useful" covariates and exclude redundant covariates to reduce the covariate dimension (Pearl, 2012; Rosenbaum and Rubin, 1984; Wooldridge, 2009). Tree-based methods, such as Classification And Regression Tree (CART), become a good choice for this goal since each node is split based on the "best" choice of covariates during tree construction. Furthermore, we can also view tree-based models as a process of partitioning the covariate space, and a simple prediction model is fitted for the partitioned space (Loh, 2011), which is similar to the idea of stratification. For example, we can use the treatment and covariates as predictors and the outcome as the response to train a CART. Then for a specific subject, we input his/her covariate value and set the treatment to 0 and 1 respectively to obtain an estimated $E[Y(0)|X = x]$ and $E[Y(1)|X = x]$. However, CARTs are designed for classification and regression problems and can only reveal association rather than causation. To reliably unveil causal relations, some refined

models (Athey and Imbens, 2016; H. Chipman et al., 2006; H. A. Chipman et al., 2010; Wager and Athey, 2018; Wang et al., 2015) have been developed. We will further elaborate one of the state-of-the-art tree-based methods, named *X-learner* (Künzel et al., 2019), in Chapter 4.

Another way to overcome the curse of dimensionality is to directly balance the covariate distributions of different treatment groups without matching. As we briefly mentioned before, the effect of confounders can also be viewed as selection bias, i.e., the covariate distributions of different treatment groups are different from each other, and they are all not representative of the covariate distribution of the whole population. Both stratification and matching methods are directly based on these biased distributions and can thus produce biased causal inference. Instead of directly jumping into original biased distributions and performing causal inference, *re-weighting* methods assign each subject an appropriate weight, creating a pseudo-population where the distributions of the treatment and control group are similar. The key concept in re-weighting methods is the *balancing score* $b(\mathbf{X})$, a function of covariates $\mathbf{X}$. Balancing score has a special property: if subjects have the same balancing score, then their covariate is independent of the treatment, i.e., $X \perp T|b(\mathbf{X})$. Among all the designs of balancing scores, the most typical and frequently used one is the *propensity score*, which is the probability of a subject assigned to the treatment given the observed covariate $\mathbf{x}$,

$$e(\mathbf{x}) = P(T = 1|\mathbf{X} = \mathbf{x}) \tag{1.2}$$

Although the propensity score itself can already be used to reduce the selection bias by matching (Caliendo and Kopeinig, 2008) due to its conditional independence property, the most common approach is to construct the weights by incorporating the propensity scores. Inverse propensity weighting (IPW) (Rosenbaum, 1987; Rosenbaum and Rubin, 1983) assigns a weight w_i to each subject

$$\hat{w}_i = \frac{T_i}{\hat{e}(\mathbf{x}_i)} + \frac{1 - T_i}{1 - \hat{e}(\mathbf{x}_i)}$$

where $T_i = 1$ denotes the i^{th} subject being assigned to the treated group, and $\hat{e}(\mathbf{x}_i)$ is the estimated propensity score commonly obtained with logistic regression. To further explain how IPW balances the distributions, we go back to the example in Figure 1.1. Instead of performing matching on two imbalanced distributions, we assign IPW to each subject of different treatment groups to generate balanced pseudo-populations. Intuitively speaking, the i^{th} subject is replicated w_i times in the pseudo-population. Figure 1.2 illustrates the replica-

tion process with more details. As we can see, the blue points on top of Figure 1.2 are the original subjects from two treatment groups. Since there's only one subject assigned to the treatment group, the corresponding propensity score $\hat{e}(\mathbf{x}) = 0.1$ and thus the IPW is $\frac{1}{\hat{e}(\mathbf{x})} = 10$. This IPW replicates the subject in the treatment group 10 times and generates the pseudo-population on the bottom left of Figure 1.2. Similarly, each subject of the control group is replicated $\frac{10}{9}$ times, and the generated pseudo-population is shown at the bottom right.

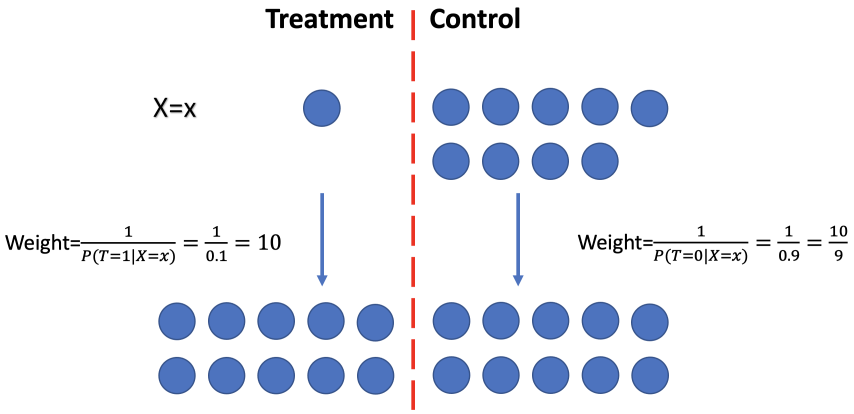

Figure 1.2: Pseudo-population created by IPW

After weighting, the IPW version of ATE based on the pseudo-population is changed from a direct estimator in Equation 1.1 to

$$\hat{ATE}_{ipw} = \frac{1}{n}\sum_{i=1}^{n}\frac{T_i Y_i}{\hat{e}(\mathbf{x}_i)} - \frac{1}{n}\sum_{i=1}^{n}\frac{(1-T_i)Y_i}{1-\hat{e}(\mathbf{x}_i)} \tag{1.3}$$

where n is the total number of subjects. There is also a normalized version of Equation 1.3

$$\sum_{i=1}^{n}\frac{T_i Y_i}{\hat{e}(\mathbf{x}_i)} \Big/ \sum_{i=1}^{n}\frac{T_i}{\hat{e}(\mathbf{x}_i)} - \sum_{i=1}^{n}\frac{(1-T_i)Y_i}{1-\hat{e}(\mathbf{x}_i)} \Big/ \sum_{i=1}^{n}\frac{1-T_i}{1-\hat{e}(\mathbf{x}_i)}$$

which is preferred when $\hat{e}(\mathbf{x}_i)$ is obtained by estimation (G. W. Imbens, 2004). With IPW, traditional statistical models such as linear regression can also be used to estimate the causal effect, and this combination is called Marginal Structure Model (Robins et al., 2000). For example, a univariate linear regression model cannot be directly used to estimate ATE due to confounding variables. After assigning the IPW to each subject, we balance the covariate distribution in the pseudo-population and reduce the confounder effect; thus, the simple

regression model can better estimate ATE with the weighted data.

In practice, however, the accuracy of the IPW estimator highly relies on the performance of the propensity score estimation, and a slight propensity score mis-specification could lead to drastic estimation error (Imai and Ratkovic, 2014). To solve this issue, we can either augment the original estimator to make it robust even when the propensity score is incorrect (Robins et al., 1994) or directly improve the balancing performance of the propensity score (Fong et al., 2018; Imai and Ratkovic, 2014).

Another drawback of the original IPW estimator is that it could be unstable if the propensity score is close to 0 or 1. This phenomenon is not surprising since the Positivity Assumption is violated in this extreme case. A straightforward solution is to eliminate subjects whose propensity scores are beyond a pre-defined interval, which is the basic idea of Trimming (Lee et al., 2011) and is routinely applied as a regularization strategy in this extreme situation. However, this trimming approach is highly sensitive to the subjective interval (Ma and Wang, 2020). Another alternative strategy to overcome the instability is to redesign the weight such that it is naturally bounded or with limited variance. Zubizarreta, 2015 optimize the weights to simultaneously balance covariates within a pre-specified threshold and minimize the variance of weights. Li et al., 2018 propose the overlapping weight, which is proportional to the probability of a subject being assigned to the opposite treatment group. They also prove that the overlapping weight has the minimum asymptotic variance among all balancing weights. However, the overlapping weight creates a pseudo-population where subjects are assigned to different treatment groups with equal probability, making the estimator less representative of the whole population. Zhao et al., 2019 introduce a more general framework to adjust the weight and loss function according to the sub-population we are interested in.

The weighting methods could achieve the covariate balance assuming that each covariate is considered a confounder, which might be a strong assumption in actual cases; e.g., if a covariate is only related to outcome, there is no need to balance this covariate. More specifically, we can further divide covariates into four categories: adjustment variables, which are only related to the outcome; instrumental variables (Baiocchi et al., 2014), which are only related to the treatment received; confounding variables, which are related to both outcome and the treatment assigned; and some other irrelevant variables (Kuang et al., 2017). Accommodating these variables in different ways can lead to dif-

ferent estimation performance, such as decreasing the estimation variance by adjusting on adjustment variables using Lasso (Bloniarz et al., 2016; Sauer et al., 2013), causing over-fitting and leading to inefficient and inaccurate estimators by including irrelevant variables (Abadie and Imbens, 2006; Häggström, 2018; Hahn, 1998). Based on this covariate categorization idea, Kuang et al., 2017 propose Data-Driven Variable Decomposition (D^2VD) method to distinguish different variables. In this thesis, we propose a model inspired by this assumption, and more details will be discussed in Section 3.2.

As the data size has been growing tremendously and more covariates are collected with modern technologies, data usually lie in a high-dimensional space which makes the curse of dimensionality even stronger: distance-based matching methods gradually lose efficiency and become computationally demanding; data will be sparse in each stratified subgroup if there are too many confounders; the propensity score will be hard to estimate accurately with high-dimensional $\mathbf{X}$, to name a few. To overcome this issue, researchers borrow the idea of the *representation learning* (Bengio, Courville, et al., 2013) from the machine learning community and develop customized models for causal inference. Representation learning focuses on transforming the input data and mapping them into a lower-dimensional representation space while keeping as much original data information as possible. Heavily relying on deep learning architecture, representation learning models consist of multiple non-linear transformations $\Phi(\cdot)$ yielding more complex and useful representations (Bengio, Courville, et al., 2013). With the lower-dimensional representation covariate vectors $\Phi(\mathbf{X})$, some traditional methods, such as matching (Chang and Dy, 2017; Chu et al., 2020), can be re-used in the representation space. However, the most powerful part of representation learning is to learn the representations catering to our needs.

Before discussing the application of representation learning in causal inference, let us first take a different perspective on selection bias. As we mentioned before, selection bias problems brought by confounders can be viewed as the covariate distributions of control and treatment groups straying apart, thus introducing the confounders' effect on the outcome. For example, suppose we want to estimate the causal effect in the control group; what we will do is use the fitted model trained on data from the treatment group (source data) to predict the counterfactual outcome in the control group (target data). However, this model may not be able to detect the treatment effect due to the discrepancy between the source data and target data. This data discrepancy problem is referred

to as *domain adaptation* problem in the machine learning literature.

Representation learning plays an important role in domain adaptation. Its goal is to map the source data and the target data into another space where their distributions are similar, such that the trained algorithm/model is more applicable to the target data. Following the same idea and applying representation learning in causal inference, researchers have tried to minimize the distribution discrepancy of different treatment groups in the representation space and thus eliminate the effect of confounding variables. Once obtained, the mapped data $\Phi(\mathbf{X})$ are directly concatenated with the scalar treatment variable T as the input of the neural network for outcome prediction (F. Johansson et al., 2016). The disadvantage of this direct concatenation, however, is that the treatment effect tends to be diminished, referred to as the "zero bias" phenomenon if the dimension of $\Phi(\mathbf{X})$ is comparably much higher than the scalar treatment variable (Shalit et al., 2017). To overcome this issue, Shalit et al., 2017 propose to use separate branches to estimate the outcome for different treatment groups, which is also a common strategy in tree-based methods (Athey and Imbens, 2016). This multi-branch architecture can also be extended to any number of treatments (Schwab et al., 2018). Figure 1.3 gives a brief summary of this design, where $L(h_t(\Phi), Y_t)$ is the loss function for the outcome prediction error in the $T = t$ group, and $disc(p_\Phi^{T=0}, p_\Phi^{T=1})$ measures the discrepancy of representation distributions from different treatment groups. Another advantage of the design in Figure 1.3 is that it uses one set of network parameters to learn $\Phi(\mathbf{X})$ for treatment and control covariates sharing the statistical power in the joint representation layers.

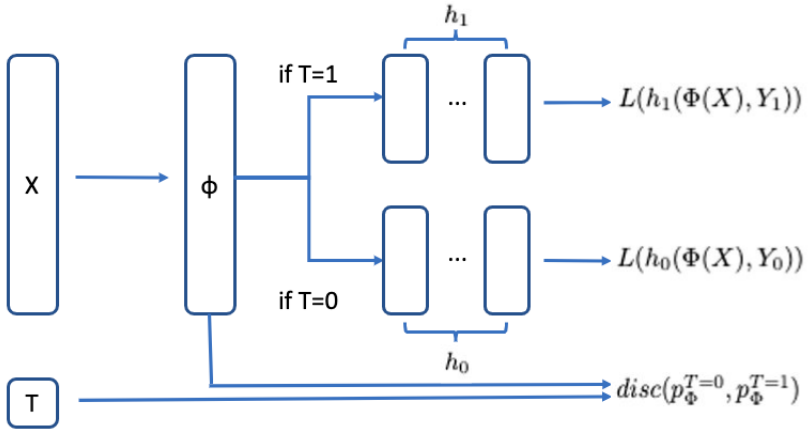

Figure 1.3: Representation learning with two-head architecture

Researchers have also proposed several "cocktails" following the idea in Figure 1.3. For instance, F. D. Johansson et al., 2018 present re-weighting methods on top of shift-invariant representation learning. In this thesis, we propose a model inspired by the work of Hassanpour and Greiner, 2019 in which the representation learning is combined with a context-aware weighting constructed based on importance sampling.

As stratification to the matching method, minimizing the discrepancy between distributions of different treatment groups in the representation space is also a "global" method that could omit local similarity information contributing to improving the causal effect estimation. Yao et al., 2018, 2019 develop a more "local" method named Similarity preserved Individual Treatment Effect (SITE), of which the objective function mainly contains three parts: distribution discrepancy, local similarity, and outcome prediction accuracy. We will further discuss this method in Section 4.2.

With the power of deep learning, researchers have been trying to relax some basic assumptions of the potential outcome framework. The most controversial one among them is the *ignorability* assumption, which states that all confounding variables have been observed during the data collection process. Identifying and collecting all confounders are unrealistic, and thus this assumption is hard to satisfy in real-world scenarios. Even in this Big-Data era, some variables, such as socioeconomic status, genetic, and environmental factors (Shalit et al., 2017), are still challenging to measure and thus become unobserved confounders.

A typical approach to relax the ignorability assumption is to introduce latent variables and build models describing the relationships between latent and observed ones. As we can imagine, the relationships will look like a network, with nodes as variables and edges as relations between two nodes. The Bayesian Network (BN) (Heckerman et al., 1995) is an appropriate model to describe this scenario. Furthermore, researchers assume that causal relations exist between treatment and the outcome and among latent and observed variables. The most widely used paradigm to learn these causal relationships is the Causal Bayesian network (CBN) (Pearl et al., 2000). A CBN adds a causal interpretation to the directed edges of a BN by imposing some assumptions, which will be discussed in detail in Section 2.2. Several deep learning models have been proposed and combined with CBN to learn causal effects (Ellis and Wong, 2008; Guo et al., 2020). In this thesis, we apply Variational AutoEncoder (VAE) (Kingma and Welling, 2013; Rezende et al., 2014), a deep learning architecture to learn latent

models, following the pioneering work of Louizos et al., 2017 where VAE is used to infer the joint distribution of latent confounders and observed variables in a CBN. Although there is little current available theory justifying that VAE can identify the true model, VAE has the significant advantage of making weaker assumptions about the data generating process and the structure of the hidden confounders. Furthermore, VAE has been showing its remarkable effectiveness in capturing latent structures from a wide range of fields such as modeling images (Gregor et al., 2015), time series (Chung et al., 2015), and fairness (Louizos et al., 2015). We will introduce VAE with further details in Section 2.3.

In summary, causal inference is a trendy topic, especially among statisticians and scholars from the machine learning community. Due to the natural property of missing counterfactual records in the data, we choose the potential outcome framework to better describe the ideas behind all models. The golden standard to estimate the causal effect is the RCT, which sometimes can be expensive, infeasible, and even unethical. As a result, drawing causal effects from the observational data becomes an attractive solution since they are collected without interfering with subjects and avoid ethical issues. Due to the Big-Data technologies, acquiring data becomes less expensive, bringing the great advantage of observational studies over RCT. However, observational studies have their problems, and the most noteworthy one is the selection bias caused by confounding variables. Researchers have developed plenty of traditional statistical methods such as matching, stratification, re-weighting, and weighted regression, trying to solve this issue and estimate causal effect more accurately. Popular tree-based machine learning methods are also customized from pure classification and regression to causal effect estimation. However, Big-Data Era not only increases the size of the data but also collects more covariates, which brings the curse of dimensionality to traditional methods. Representation learning, a dimension reduction technique with deep learning architectures, is introduced to overcome the challenges. Traditional techniques such as matching and distribution balancing can then be performed in the representation space. With the help of deep learning, we take a step further and try to relax the most controversial assumption, i.e., ignorability assumption, from the potential outcome framework to model more complex situations such as latent variables and causal network structure. In this thesis, we use CBN as the theoretical foundation and assume the existence of different kinds of latent variables generating observed data. We apply VAE to learn the latent variable distributions and importance sampling weights to improve the accuracy of causal effect estimation.

The rest of the thesis is organized as follows. In Chapter 2, we introduce the necessary preliminaries for the work of this thesis; specifically, we focus on the topics of CBN and VAE. In Chapter 3, we propose our model based on VAE and use CBN to identify latent variables. To enhance the performance of outcome prediction, we also apply the importance sampling weight techniques to related terms in the objective function. Chapter 4 evaluates our model on simulated, semi-simulated, and real-world datasets by comparing it with other state-of-the-art methods. Chapter 5 concludes our results and looks to future research work. We will also discuss how to improve causal inference better from both micro and macro perspectives and introduce this topic to a broader audience.

Chapter 2

Preliminaries

This chapter introduces some essential concepts and related assumptions that serve as our model's foundation. First, Section 2.1 introduces the necessary definitions and assumptions of the potential outcome framework. Next, Section 2.2 introduces the Causal Bayesian Network and connects information flow with causality. Finally, Section 2.3 introduces the Variational Auto-Encoder, a famous deep learning architecture to train latent models.

2.1 Potential Outcome Framework

The potential outcome framework (Rubin, 1974) assumes that each subject had a potential outcome if he/she were assigned to a specific treatment. Definition 2.1.1 defines these outcomes more formally.

Definition 2.1.1. The i^{th} subject has the potential outcome $Y_i(0)$ if he/she were given the treatment 0; similarly, $Y_i(1)$ is the potential outcome of the i^{th} subject if he/she were given treatment 1.

2.1.1 Four Basic Assumptions

It is impossible to estimate potential outcomes $Y_i(0), Y_i(1)$ without any assumptions since we cannot observe them simultaneously. The key assumptions are:

1. **Stable Unit Treatment Value Assumption (SUTVA) (Rubin, 2005)**. This assumption consists of two statements:

    - Subjects do not interfere with each other, i.e., the treatment assignment of one subject does not affect the outcome of another subject.

- There is only one version of the treatment, i.e., no matter how subject i received treatment T, the potential outcome will always be $Y_i(T)$.

As the independent identically distributed assumption to statistical inference, SUTVA is a reasonable assumption for causal inference in most cases. However, there are also real-world scenarios when subjects naturally interact with each other, and thus SUTVA does not hold. For instance, subjects can interact with each other via the message network on the social media platform. For such network data, Graph Convolutional Network (Kipf and Welling, 2016) is usually a popular choice for analysis (Guo et al., 2020; Shalizi and Thomas, 2011). We will not discuss the details of this situation and assume that SUTVA holds for our model.

The second part of the SUTVA assumption states that there is only one version of the treatment. If there are multiple versions of treatments, as might arise for surgery treatment when different surgeons perform the surgery, the potential outcome is not well-defined since different surgeons will generate different $Y_i(T)$ for each subject.

The one-version statement part of the SUVTA assumption is hard to verify, since even in the aforementioned surgery example, even the same surgeon may have different mental and physical conditions which affect his/her surgery skills.

2. **Consistency:** The potential outcome of treatment T,$Y(T)$, is equal to the observed outcome Y if the actual treatment received is T.

   This assumption is impossible to verify since we cannot observe the actual potential outcome. What we can observe is the factual outcome, which we believe is the only realization of the potential outcome corresponding to the treatment.

3. **Positivity:** For any value of the subject covariate $\mathbf{X}$, treatment assignment is not deterministic, i.e. $P(T = t|\mathbf{X} = \mathbf{x}) > 0$, for all t and $\mathbf{x}$.

   This assumption assumes that any subject is possible to be assigned to each treatment and thus guarantees the possibility that we can estimate the causal effect of any sub-populations. To make this assumption valid in real-world applications, Caliendo and Kopeinig, 2008 apply *trimming*

to find the common support of positive propensity scores for each treatment, i.e.

$$\{\mathbf{X} : P(T = 1|\mathbf{X}) > 0 \quad and \quad P(T = 0|\mathbf{X}) > 0\}$$

4. **Ignorability:** Given the covariates $\mathbf{X}$, treatment assignment T is independent of the potential outcomes, i.e. $\{Y(1), Y(0)\} \perp T|\mathbf{X}$. In other words, among people with the same values of $\mathbf{X}$, we can think of treatment as being randomly assigned to subjects like in RCT.

   This assumption is the most controversial one among all assumptions mentioned above since we need to collect enough covariates not to miss any confounders. This assumption has become more and more realistic due to modern technologies. However, there are still cases where this assumption cannot be satisfied, such as the existence of unobserved confounders, e.g., social-economic status, genetic and environmental factors (Shalit et al., 2017). Even though these can be measured, it is still possible that they do not exist in the original data for a specific study.

2.1.2 Revisiting RCT

With four assumptions mentioned in Section 2.1, we can build the link between the causal effect and observed data Y. As an example, we will prove that our intuitive estimator in the RCT

$$\frac{1}{N_1}\sum_{i=1}^{N} I(T_i = 1)Y_i - \frac{1}{N_0}\sum_{i=1}^{N} I(T_i = 0)Y_i \tag{2.1}$$

is an unbiased estimator of ATE, where N_1 (N_0) is the number units in the treatment (control) group.

With the SUTVA, the potential outcome of any subject with $T = 1$ can be written as $Y(1)$, a function of the subject's own treatment without other individuals' treatment. As a result, ATE can be expressed as

$$\begin{aligned} ATE &= E_Y[Y(1)] - E_Y[Y(0)] \\ &= E_{\mathbf{X}}[E_{Y|\mathbf{X}}[Y(1)|\mathbf{X}]] - E_{\mathbf{X}}[E_{Y|\mathbf{X}}[Y(0)|\mathbf{X}]] \end{aligned} \tag{2.2}$$

Next we will introduce the treatment T behind the conditioning bar, which is executable because the positive value of $p(T, \mathbf{X}) = p(T|\mathbf{X})p(\mathbf{X})$ due to the

Positivity Assumption, which guarantees the positive value of $p(T|\mathbf{X})$. With the Ignorability Assumption, Equation 2.2 equals

$$E_{\mathbf{X}}[E_{Y|\mathbf{X}_1}[Y(1)|T=1,\mathbf{X}] - E_{Y|\mathbf{X}_0}[Y(0)|T=0,\mathbf{X}]] \tag{2.3}$$

where $\mathbf{X}_t$ denotes the joint distribution $p(\mathbf{X}=\mathbf{x}, T=t), t=0,1$. By the Consistency Assumption, we can rewrite Equation 2.3 by replacing $Y(T)$ with Y,

$$E_{\mathbf{X}}[E_{Y|\mathbf{X}_1}[Y|T=1,\mathbf{X}] - E_{Y|\mathbf{X}_0}[Y|T=0,\mathbf{X}]] \tag{2.4}$$

Since $p(\mathbf{X}=\mathbf{x}|T=t)$ is the same as $p(\mathbf{X}=\mathbf{x})$ in the RCT setting, we have $p(Y=y|T=t,\mathbf{x}) = \frac{p(y,T=t,\mathbf{x})}{p(\mathbf{x}|T=t)p(t)} = \frac{p(y,T=t,\mathbf{x})}{p(\mathbf{x})p(T=t)}$, we have

$$\begin{aligned}
E_{\mathbf{X}}[E_{Y|\mathbf{X}_t}[Y|T=t,\mathbf{X}]] &= \int \left(\int y \cdot \frac{p(y,T=t,\mathbf{x})}{p(\mathbf{x})p(T=t)} dy \right) p(\mathbf{x}) d\mathbf{x} \\
&= \int \left(\int y \cdot \frac{p(y,T=t,\mathbf{x})}{p(T=t)} d\mathbf{x} \right) dy \\
&= \int y \cdot \frac{p(y,T=t)}{p(T=t)} dy \\
&= E_Y[Y|T=t]
\end{aligned}$$

Thus, we can simplify Equation 2.4 as

$$E_Y[Y|T=1] - E_Y[Y|T=0] \tag{2.5}$$

The direct estimator for the estimand in Equation 2.5 is Equation 2.1, which proves our initial statement.

2.2 Causal Bayesian Network

A Causal Bayesian Network is built on a special type of graph called Directed Acyclic Graphs (DAG) incorporated with probability and definitions of causality. Each node in the graph represents a random variable, and the edge between two nodes describes the relations between two random variables. To derive causality from a DAG, we start by introducing a probability distribution into the graph. Recall that by the chain rule, we can factorize any probability density

p as

$$\begin{aligned} p(x_1, \cdots, x_n) &= p(x_1)p(x_2|x_1)\cdots p(x_n|x_{n-1}, \cdots, x_1) \\ &= \prod_{i=1}^{n} p(x_i|x_{i-1}, \cdots, x_1) \end{aligned} \tag{2.6}$$

This factorization can potentially introduce too many parameters for statistical models since there are $(i-1)$ random variables on the right hand side of the bar in each $p(x_i|x_{i-1}, \cdots, x_1)$. To make Equation 2.6 more compact, we first need a concept called *parents*:

Definition 2.2.1. Suppose X and Y are two nodes in a DAG. If there is a directed edge from X to Y, then X is a *parent* of Y, Y is a *child* of X.

Next, we introduce the *Local Markov Assumption* (Lauritzen et al., 1990), which can be applied directly to simplify Equation 2.6.

Assumption 2.2.1. *(Local Markov Assumption) Given its parents in a DAG, a node is independent of its non-descendants.*

With the Local Markov Assumption, Equation (2.6) can be rewritten as

$$p(x_1, \cdots, x_n) = \prod_{i=1}^{n} p(x_i|x_{A_i})$$

where x_{A_i} are parents of x_i. To avoid ambiguity when constructing a Bayesian network based on the factorization, we also have the following assumption,

Assumption 2.2.2. *Adjacent nodes in the DAG are dependent.*

Here is a simple example demonstrating the importance of Assumption 2.2.2. Suppose we have a simple graph with two nodes x and y where a directed edge points to y from x, as shown in Figure 2.1. With the Local Markov assumption, we can factorize the joint probability density as $p(x, y) = p(x|y)p(y) = p(x)p(y)$ since X doesn't have a parent. However, this factorization also implies X and Y are independent. With Assumption 2.2.2, we cannot factorized $p(x, y)$ as $p(x|y)p(y)$ and the only correct way is $p(x, y) = p(y|x)p(x)$.

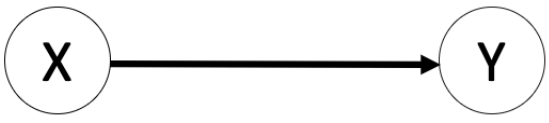

Figure 2.1: A simple graph with ambiguous factorization

Next, we introduce causality relations into directed graphs using Definition 2.2.2 and Assumption 2.2.3.

Definition 2.2.2. A variable X is said to be a *cause* of a variable Y if Y can change in response to changes in X.

Assumption 2.2.3. *(Causal edges assumption) In a directed graph, every parent is a direct cause of all its children.*

A direct application of Definition 2.2.2 and Assumption 2.2.3 is to differentiate non-causal associations and causal associations between two nodes in a directed graph, which is crucial since we need to remove non-causal associations to estimate the causal effect.

2.2.1 Building Blocks of Causal Graphs and Information Flow

With the Local Markov Assumption, we can decompose a joint probability distribution into smaller, local conditional probability products concerning the graph structure. Moreover, this factorization procedure also introduces certain independence among some random variables. These independent relationships can be elegantly discovered and described from the directed graph by checking three types of graph structures.

Suppose a DAG G has three nodes A, B, and C. This simple G essentially has only three possible structures: *Chain*, *Fork*, and *Collider*, each of which leads to different independence result:

- The *Chain* structure (Figure 2.2). $A \not\perp C$ if B is not conditioned on, but $A \perp C|B$ since

$$\begin{aligned} p(A, C|B) &= \frac{p(A, B, C)}{p(B)} = \frac{p(C|B)p(B|A)p(A)}{p(B)} \\ &= p(C|B)p(A|B) \end{aligned}$$

  Here, the intuition is that B holds all the information that determines the outcome of C; thus, C does not change whatever value A takes once B is conditioned on. The Chain structure presents a typical case of causal information flow from A to C.

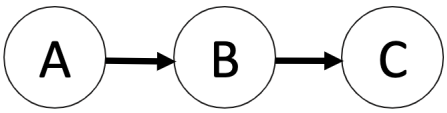

Figure 2.2: The Chain structure

- The *Fork* structure (Figure 2.3). $A \not\perp C$ if B is not conditioned on, but $A \perp C|B$ since

$$\begin{aligned} p(A, C|B) &= \frac{p(A, B, C)}{p(B)} = \frac{p(A|B)p(C|B)p(B)}{p(B)} \\ &= p(A|B)p(C|B) \end{aligned}$$

  The intuition here is the same as the *Chain* structure, and there exists a non-causal information flow between A and C if B is not conditioned on.

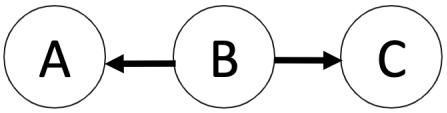

Figure 2.3: The Fork structure

- The *Collider (Immortality)* structure (Figure 2.4). $A \perp C$ if B or any descendent of B is NOT conditioned on since

$$\begin{aligned} p(A, C) &= \sum_B p(A, B, C) = \sum_B p(A)p(B|A, C)p(C) \\ &= p(A)p(C) \sum_B p(B|A, C) \\ &= p(A)p(C) \end{aligned}$$

  However, A is dependent with C if B or any descendent of B is conditioned on. In other words, there is a non-causal information flow between A and C once B is conditioned on.

The fact that conditioning on a collider creates dependence might not be intuitive. Here is a whimsical example from Jordan, 2004 that makes this idea more acceptable. Suppose a person's friend is late for a meeting. There are two explanations: he was kidnapped by aliens, or that person forgot to set the watch

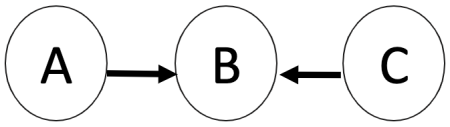

Figure 2.4: The Collider structure

ahead one hour for daylight savings time (Figure 2.5). **Aliens** and **Watch** are blocked by a collider **Late** which implies that they are marginally independent. This independence seems reasonable since we expect these two variables to be independent before knowing whether the friend is late or not. We also expect that $P(Aliens = yes|Late = yes) > P(Aliens = yes)$ since learning that the friend is late certainly increases the chance that he was kidnapped. However, when we know that person forgot to set the watch properly, the chance that the friend was kidnapped certainly decreases. Hence, $P(Aliens = yes|Late = yes) \neq P(Aliens = yes|Late = yes, Watch = no)$. Thus, **Aliens** and **Watch** are dependent given **Late**.

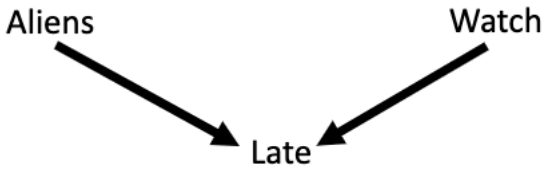

Figure 2.5: Jordan's alien example

From the three basic graphical structures mentioned above, we can find that dependence relations can be "blocked" by conditioning. Definition 2.2.3 formally introduces the concept of *blocking* using conditioning.

Definition 2.2.3. A path between nodes X and Y is blocked by a conditioning set Z (could be empty) if either of the following is true:

1. Along the path, there's a chain $\cdots \rightarrow W \rightarrow \cdots$ or a fork $\cdots \leftarrow W \rightarrow \cdots$ where W ($W \in Z$) is conditioned.

2. There's a collider W on the path that's not conditioned ($W \notin Z$) and none of its descendants, denoted as $de(W)$, are conditioned on, i.e. $de(W) \nsubseteq Z$.

2.2.2 Do-operator

The Do-operator concerning a random variable T is written as

$$do(T = t)$$

which means we *intervene* T by *setting* it to a specific value t without changing all other variables. The idea of "intervention" is closely related to "conditioning" but has a significant difference. We will further elaborate this difference in Section 2.2.3. For now we will use an example to illustrate the basic idea. Suppose we have a CBN in Figure 2.6 and the joint distribution has the form $f(y,t,z) = f(z)f(t|z)f(y|t,z)$. We can use a pseudo-code to generate the

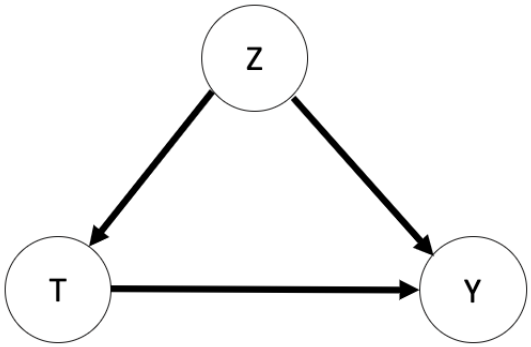

Figure 2.6: Conditioning versus intervening

data:

$$\begin{aligned} For \quad i &= 1, \cdots, n: \\ z_i &\leftarrow f(z) \\ t_i &\leftarrow f(t|z_i) \\ y_i &\leftarrow f(y|t_i, z_i) \end{aligned}$$

Repeating this code many times yields data $(y_1, t_1, z_1), \cdots, (y_n, t_n, z_n)$. If we try to find the distribution of Y conditioning on $T = t$, we need to do the following computation:

$$\begin{aligned} P(Y = y|T = t) &= \frac{P(Y = y, T = t)}{P(T = t)} = \frac{f(y,t)}{f(t)} \\ &= \frac{\sum_z f(y,t,z)}{f(t)} = \frac{\sum_z f(z)f(t|z)f(y|t,z)}{f(t)} \\ &= \sum_z f(y|t,z)f(z|t) \end{aligned} \tag{2.7}$$

Now suppose we *intervene* T by changing the code and *setting* $T = t$. The code now looks like this:

$$\begin{aligned} set \quad T &= t \\ For \quad i &= 1, \cdots, n: \\ z_i &\leftarrow f(z) \\ y_i &\leftarrow f(y|t, z_i) \end{aligned}$$

If we again want to find the distribution of Y with the intervened value $do(T = t)$, we need the new joint distribution $f^\star(y, z)$ and marginalize it over z:

$$P(Y = y|do(T = t)) = f^\star(y) = \sum_z f^\star(y, z) = \sum_z f(y|t, z)f(z) \quad (2.8)$$

Compared with Equation 2.7, Equation 2.8 is clearly a new distribution, which reflects the difference between "intervention" and "conditioning".

Definition 2.2.4 formally defines the distribution of a random variable with intervention (do-operator) and connects the do-operator to the potential outcome framework.

Definition 2.2.4. The interventional distribution of the do-operator is defined as

$$P(Y = y|do(T = t)) \triangleq P(Y(t) = y)$$

$P(Y = y|do(T = t))$ is usually written as $P(y|do(t))$ for short.

We can rewrite many causal estimands using the *do-operator* with definition 2.2.4. For example, we can rewrite ATE as

$$ATE = E[Y(1) - Y(0)] = E[Y|do(T = 1)] - E[Y|do(T = 0)]$$

Another important causal estimand is CATE, or Individual Treatment Effect (ITE), which is the causal effect concerning the subgroup or individual level, i.e., the treatment effect for a specific subject or subgroups with the covariate value $\mathbf{X} = \mathbf{x}_i$. Under the potential outcome framework, CATE can be expressed as

$$CATE(\mathbf{x}_i) = \tau(\mathbf{x}_i) = E[Y(1) - Y(0)|\mathbf{X} = \mathbf{x}_i] \quad (2.9)$$

Using the do-operator, we can rewrite Equation 2.9 as

$$\begin{aligned}\tau(\mathbf{x}_i) &= E[Y(1) - Y(0)|\mathbf{X} = \mathbf{x}_i] \\ &= E[Y|do(T = 1), \mathbf{X} = \mathbf{x}_i] \\ &\quad -E[Y|do(T = 0), \mathbf{X} = \mathbf{x}_i] \end{aligned} \tag{2.10}$$

2.2.3 Assumptions for Identifiability

Now we have two different distributions: the statistical distribution $P(Y, T)$ and the interventional distribution $P(Y|do(T = t))$. The former is for statistical estimand such as $E[Y|T = 1]$, and the latter is for causal estimand such as $E[Y|do(T = 1)]$. To estimate the causal effect, we need to replace the causal estimand with the corresponding statistical estimand. Sometimes we can perform this replacement without any expense, for instance, in the RCT setting since we already prove that

$$\begin{aligned} ATE &= E_Y[Y(1)] - E_Y[Y(0)] \\ &= E_Y[Y|do(T = 1)] - E_Y[Y|do(T = 0)] \\ &= E_Y[Y|T = 1] - E_Y[Y|T = 0] \end{aligned}$$

by Equation (2.2)-(2.5). In this situation, we say the causal estimand is *identifiable*. However, causal estimands are usually not identifiable in most cases due to confounding variables, especially with the observational data. To replace it with the corresponding statistical estimand, we need to build CBNs and make another assumption called *Modularity Assumption*.

Assumption 2.2.4. *(Modularity Assumption) If we intervene on a set of nodes $S \subseteq [n]$, i.e. setting them to specific values, then for all i, we have the following:*

1. *If $i \notin S$, then $P(x_i|x_{A_i})$ remains unchanged, where x_{A_i} are the parent nodes of node x_i.*
2. *If $i \in S$, then $P(x_i|x_{A_i}) = 1$ if x_i is consistent with the intervention (i.e., X_i is set to by the intervention); otherwise, $P(x_i|x_{A_i}) = 0$*

Note that with the *Modularity Assumption*, we can write the intervened joint distribution as follows,

$$P(x_1, \cdots, x_n|do(S = s)) = \prod_{i \notin S} P(x_i|x_{A_i}) \tag{2.11}$$

The Modularity Assumption provides a guide to manipulating the graph once some variables are intervened. More specifically, when a variable is intervened,

only that variable and its parents are affected, and incoming edges of that variable are removed from the graph. All other relationships in the graph remain unchanged.

The Modularity Assumption essentially distinguishes $P(Y|do(T = t))$ and $P(Y|T = t)$. Let us go back to the pseudo-code example shown in Figure 2.6. In that example, we implicitly assume the "setting" action doesn't influence the distribution $f(z)$ and $f(y|t, z_i)$, which is actually the Modularity Assumption. Hence, Equation 2.8 can be directly derived by applying Equation 2.11. Comparing Equation 2.7 and Equation 2.8, we can find that $P(Y|do(t)) = P(Y|t)$ as long as $f(z|t) = f(z)$, which implies that if $f(z|t) = f(z)$ in some setting, such as RCT, the non-observable $P(Y|do(t))$ can be estimated with data.

2.2.4 Backdoor Adjustment

As seen in the Fork and Collider structure, non-causal information flow can exist in a CBN. Although Modularity Assumption can help us identify causal effects by simplifying the distribution into the form in Equation 2.11, the simplified equation can still be complex if the CBN includes many nodes and edges. To further simplify Equation 2.11, researchers have developed adjustment methods using the Modularity Assumption as the starting point. Here we only introduced the *backdoor adjustment*, which will be used in Section 3.2.

We need two concepts named as *backdoor path* and *backdoor criteria* to further elaborate the backdoor adjustment.

Definition 2.2.5. For a node T in a graph, a path is a *backdoor path* of T if it has incoming edges into T.

Definition 2.2.6. A set of variables W satisfies the *backdoor criterion* relative to T and Y if the following are true:

1. W blocks all backdoor paths from T to Y;
2. W does not contain any descendants of T.

Then we introduce Theorem 2.2.1 based on the backdoor criterion. Theorem 2.2.1 is also called *Rule* 2 in the do-calculus (Pearl and Mackenzie, 2018), which is the summary of manipulations aiming at transforming expressions containing the do-operator into expressions that do not.

Theorem 2.2.1. *(Rule 2)* $P(Y|do(t), Z) = P(Y|t, Z)$ *if* Z *satisfy the backdoor criterion.*

We will use an example to justify this rule using the Modularity Assumption. Suppose we have a CBN shown in Figure 2.7 and we condition on W_1. Apparently, W_1 satisfies the backdoor criterion of the node T. By applying the Modularity Assumption conditioning on W_1, we have

$$P(Y, W_2|do(t), W_1) = P(Y|W_2, t, W_1)P(W_2|W_1) \tag{2.12}$$

Since $W_2 \perp T$ if Y is not conditioned due to the Collider structure, we can replace $P(W_2|W_1)$ with $P(W_2|W_1, T)$ and rewrite Equation 2.12 as

$$\begin{aligned} P(Y, W_2|do(t), W_1) &= P(Y|W_2, , t, W_1)P(W_2|t, W_1) \\ &= P(Y, W_2|t, W_1) \end{aligned} \tag{2.13}$$

Marginalizing Equation 2.13 by W_2, we have

$$P(Y|do(t), W_1) = P(Y|t, W_1)$$

which is the required result. The intuition behind Rule 2 is that all non-causal information flow from T to Y is blocked, and all information flow starting from T (causal effect) is kept.

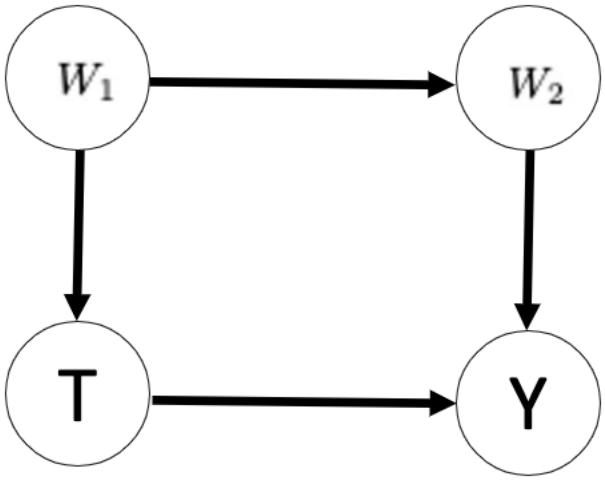

Figure 2.7: Example for Rule 2 of do-calculus

The backdoor adjustment is the result of the direct application of Theorem 2.2.1.

Theorem 2.2.2. *Given W that satisfies the backdoor criterion, we can identify the causal effect of T on Y as $P(y|do(t)) = \sum_W P(y|t, W)P(W)$*

Proof.

$$\begin{aligned} P(y|do(t)) &= \sum_W P(y|do(t), W)P(W|do(t)) \\ &= \sum_W P(y|t, W)P(W|do(t)) \qquad (2.14) \\ &= \sum_W P(y|t, W)P(W) \qquad (2.15) \end{aligned}$$

□

Equation 2.14 holds by applying Theorem 2.2.1. To justify Equation 2.15, let us think about how the information can flow from T to W: all incoming edges of T are removed according to the graph intuition of intervention, so information cannot flow from T to W via incoming edges of T; W is not a descendent of T due to the backdoor criterion, so the only possible paths must include Colliders, thus T must be independent with W and we can replace $P(W|do(t))$ with $P(W)$.

Theorem 2.2.2 is useful since it not only simplify Equation 2.11, but includes factorization terms easier to be estimated from the observational data. For example, if Y is binary, we can use any kind of machine learning classification models to estimate $P(y|t, W)$ and perform a weighted average with respect to $P(W)$ to estimate $P(y|do(t))$.

2.3 Variational Auto-Encoder

As mentioned in Chapter 1, latent models are popular choices to relax the Ignorability Assumption in the potential outcome framework. However, not every latent model is as simple as the pseudo-code example in Section 2.2.2 where there is only one latent variable z and causal relations are sequential. The number of latent variables and their relations with $\{Y, X, T\}$ can vary and produce a complex CBN structure. Deep learning architectures, also called neural networks, thus become natural choices for researchers due to their supreme ability to learn complex data characteristics. Deep learning models usually include multiple layers of nodes. Each node has its own parameters to learn the characteristics from nodes in previous layers if these two nodes are connected. Figure 2.8 presents an example of two layers of nodes with equal sizes of four. These

two layers are also called *Fully Connected (FC)* since each node of the second layer is connected with each node of the previous layer. We can imagine that the architecture becomes complex once we increase the number of nodes and layers, making deep learning models powerful in learning complex structures and information from data. In this section, we introduce a class of latent models named *Auto-Encoder* (AE), which are specifically designed deep learning architectures to learn latent models.

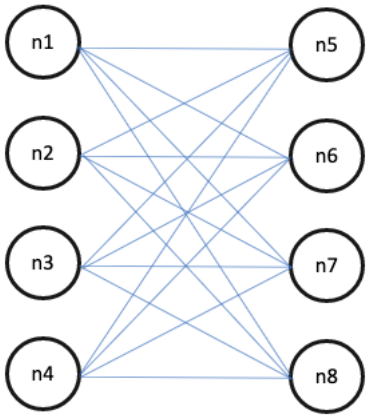

Figure 2.8: An example of Fully Connected layers

A typical AE consists of three parts: *Encoder*, *Bottleneck*, and *Decoder*. The Encoder is a module consisting of multiple FC layers with decreasing number of nodes, which compresses the input data into a low-dimensional space. The Bottleneck usually refers to the last layer of the Encoder, which contains the compressed representations and is, therefore, several orders of magnitude smaller than the input data. The Decoder usually has the symmetric architecture of the Encoder starting from the Bottleneck with an increasing number of nodes in each layer that reconstruct the compressed data from the Bottleneck back to the original data. Figure 2.9 (**wiki:AE**) shows a sketch of an AE, where the length of vertical lines of each quadrilateral represents the size of the layer, and the dashed line means the adjacent layers are fully connected. We use the trapezoid to represent the decreasing/increasing number of nodes in each layer from the Encoder/Decoder. The objective function of an AE is called the *reconstruction loss*, which measures the difference between the reconstructed output and original input data.

The idea of AEs has been popular for decades, and its first application can date back to last centuries (Hinton and Zemel, 1993; Kramer, 1991). Compared

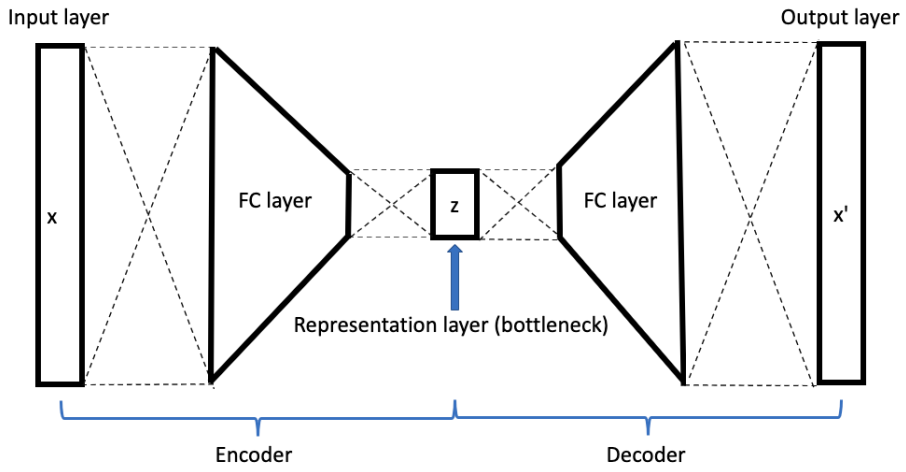

Figure 2.9: The sketch of a standard AE

with traditional methods such as Principal Component Analysis (PCA), an AE was initially used as a more complex dimensional reduction technique since it can utilize non-linear transformation of input data to capture more complex information. In the machine learning community, the compressed data are called *representations* and the corresponding low-dimensional space is called *representation space*. Now this idea has been generalized and several variations of AEs have been proposed for different uses or better encoding performances, such as Denoising Auto-Encoder (Bengio, Yao, et al., 2013) for filtering noise from images, Contractive Auto-Encoders (Rifai et al., 2011) for preserving local similarities in the representation space, and Variational Auto-Encoders (VAE) (Kingma and Welling, 2013) for learning the distribution of representations to detect outliers (An and Cho, 2015) or generate fake images (Liu et al., 2017; Yan et al., 2016).

Among all the variants of AEs, VAE is the most special one in that it learns the representations, which can be viewed as latent vectors in latent models, in a non-deterministic way. Figure 2.10 (EugenioTL, 2021) presents a standard structure of a VAE. The only difference with the structure shown in Figure 2.9 is the bottleneck part: while standard AEs produce deterministic vectors from the Bottleneck, VAEs generates random vectors and thus have corresponding distributions. This feature gives VAE more flexibility and enables it to perform various tasks, such as detecting outliers or generating fake images, using latent variable distributions. However, precisely estimating the latent distri-

butions is computationally intractable most of the time (Kingma and Welling, 2013). Besides, learning non-deterministic components is challenging for neural networks. To train this particular AE, Bishop and Nasrabadi, 2006 propose Coordinate Ascent Variational Inference (CAVI), also named as Variational Expectation-Maximization Algorithm, to solve the optimization problem that appeared in the training process. We will talk about Variational Inference (VI) with details in Section 2.3.2 since understanding VI is critical to designing our proposed model.

As a starting point, we first briefly introduce the Expectation-Maximization (EM) algorithm in the context of VAE in Section 2.3.1 and point out why *posterior distributions* play a central role when training a latent model. Then, in Section 2.3.2, we introduce VI, which applies a particular class of distribution family to solving the intractability issue. We point out how VI directs us to construct the Encoder for VAEs and provide an insight into the training algorithm in Section 2.3.3. At last, we introduce a technique named *reparameterization trick* in Section 2.3.4, which helps us bypass the random component issue when training VAEs.

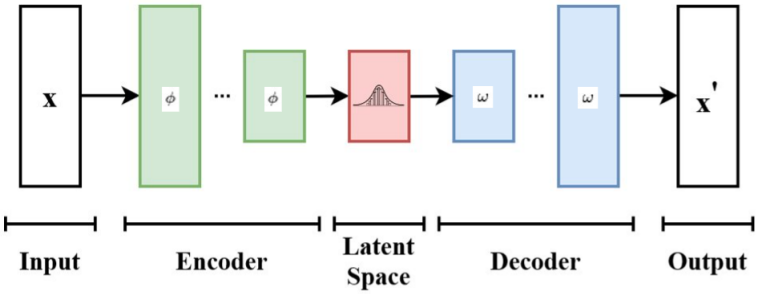

Figure 2.10: VAE framework

2.3.1 EM-algorithm in the Context of VAE

Suppose we have marginal log likelihood $\sum_{i=1}^{N} \log p(x_i; \omega)$ with some parameters ω and there are unobserved latent variables z_i in this model. Our goal is to maximize this log-likelihood with these latent variables,

$$\max_{\omega} \sum_{i=1}^{N} \log p(x_i; \omega) = \max_{\omega} \sum_{i=1}^{N} \log \int p(x_i, z_i; \omega) dz_i \tag{2.16}$$

Notice that we intentionally name the parameters as ω, which is the same as the ω in the blue box in Figure 2.10. The goal of EM algorithm in the context

of VAE is to find and maximize the lower bound of this log-likelihood, and thus the original log-likelihood will also be maximized. Since we can rewrite Equation (2.16) as

$$\begin{aligned}\sum_{i=1}^{N}\log\int p(x_i,z_i;\omega)dz_i &= \sum_{i=1}^{N}\log\int q(z_i)\frac{p(x_i,z_i;\omega)}{q(z_i)}dz_i \\ &= \sum_{i=1}^{N}\log E_{q(z_i)}\left[\frac{p(x_i,z_i;\omega)}{q(z_i)}\right]\end{aligned}$$

With Jensen's Inequality, we can derive a family of lower bounds, named the Evidence of Lower Bound (ELBO), concerning fixed parameters ω and the latent variable distribution $q(z_i)$.

$$\begin{aligned}\sum_{i=1}^{N}\log E_{q(z_i)}\left[\frac{p(x_i,z_i;\omega)}{q(z_i)}\right] &\geq \sum_{i=1}^{N}E_{q(z_i)}\left[\log\frac{p(x_i,z_i;\omega)}{q(z_i)}\right] \\ &\triangleq L_{ELBO}(\omega,q) \qquad (2.17)\end{aligned}$$

To maximize the lower bound, we need to update q and ω alternatively, which are called **E-step** and **M-step**, respectively. More specifically, during the **E-step** when ω is fixed, maximizing the lower bound is equivalent to minimize the gap between the original marginal log-likelihood and the lower bound,

$$\begin{aligned}gap &= \sum_{i=1}^{N}\log p(x_i;\omega) - \sum_{i=1}^{N}E_{q(z_i)}\left[\log\frac{p(x_i,z_i;\omega)}{q(z_i)}\right] \\ &= \sum_{i=1}^{N}E_{q(z_i)}[\log p(x_i;\omega)] - \sum_{i=1}^{N}E_{q(z_i)}\left[\log\frac{p(x_i,z_i;\omega)}{q(z_i)}\right] \\ &= \sum_{i=1}^{N}\int E_{q(z_i)}\left[\log\frac{p(x_i;\omega)q(z_i)}{p(x_i,z_i;\omega)}\right] = \sum_{i=1}^{N}E_{q(z_i)}\left[\log\frac{q(z_i)}{p(z_i|x_i;\omega)}\right] \\ &= \sum_{i=1}^{N}KL[q(z_i)||p(z_i|x_i;\omega)] \qquad (2.18)\end{aligned}$$

where KL stands for the Kullback-Leibler divergence (Kullback, 1997; Kullback and Leibler, 1951), a measure of how one probability distribution is different from another reference probability. Since the KL divergence has the following

properties,

$$\begin{aligned} KL(P||Q) &\geq 0 \\ KL(P||Q) &= 0 \iff P = Q \end{aligned}$$

minimizing Equation 2.18 is equivalent to setting $q(z_i) = p(z_i|x_i;\omega)$. This result shows that the best distribution candidate for latent variables z_i the posterior distribution with respect to observed data x_i and fixed parameters ω.

M-step: After setting $q(z_i) = p(z_i|x_i;\omega)$ in the E-step, we fix $q(z_i)$ and then update ω to maximize the lower bound $L_{ELBO}(\omega, q)$ in Equation 2.17. We iterate E-step and M-step until convergence.

2.3.2 Variational Inference

As we discussed in Section 2.3.1, estimating the posterior distribution of latent variables is one of the core problems for training latent models. However, posterior distributions are usually difficult to learn unless they have conjugate priors or we apply some approximation methods. In this section, we briefly review Variational Inference (VI) (Jordan et al., 1999; Wainwright, Jordan, et al., 2008), a method that approximates a wide range of posterior densities through optimization, rather than statistical estimation, for latent models. The basic idea of VI is to choose a candidate distribution $q(z)$ from a family of easy-to-learn distributions Q to approximate the hard-to-compute posteriors $p^\star(z|x)$ with respect to KL divergence. Figure 2.11 (Polykovskiy, 2019) presents a concise visualization of this procedure. A more comprehensive review of VI is provided by Blei et al., 2017.

Choosing the family of candidate distribution is not straightforward: while large families are hard to estimate, small families are not general enough. Under the VI framework, the family is chosen to be the mean-field variational family

$$\{q(\mathbf{z}) : q(\mathbf{z}) = \prod_j q_j(z_{(j)})\} \tag{2.19}$$

where $z_{(j)}$ is the j^{th} coordinate of $\mathbf{z}$, and q_j is an independent density function with respect to $z_{(j)}$. We emphasize that the member of the mean-field variational family is not a model of the observed data – in fact, the observed data $\mathbf{x}$ does not exist in Equation 2.19. We will see in Section 2.3.3 that it is the E-step

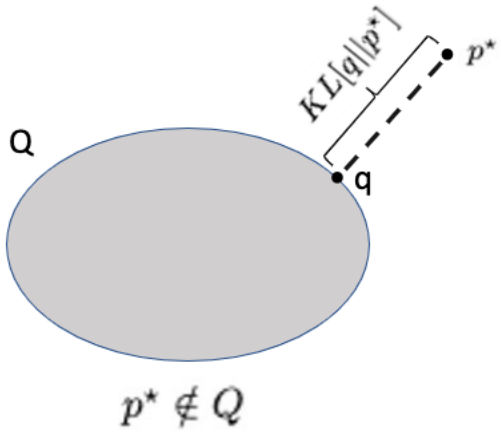

Figure 2.11: Basic idea of Variational Inference

in the EM-algorithm that connects fitted variational density to the data.

We have not specified the parametric form of $q_j(z_{(j)})$. In principle, each $q_j(z_{(j)})$ can have any appropriate parametric form corresponding to the type of $\mathbf{z}$. For example, a Gaussian form for $q_j(z_{(j)})$ might be an appropriate choice for a continuous $\mathbf{z}$; any categorical distribution is suitable for $q_j(z_{(j)})$ if $\mathbf{z}$ is categorical.

2.3.3 Variational EM-algorithm

In this section, we take a deeper look at the Encoder structure of VAEs and have a better understanding of how the variational EM-algorithm trains a VAE. In the rest of the section, we assume the latent variable $\mathbf{z}_i$ for the i^{th} subject is continuous for convenience. We start by describing the latent model corresponding to a VAE. Following the general setting for a latent model with continuous latent variables, a VAE assumes a standard Gaussian prior distribution for $\mathbf{z}_i$. To make the model more general, VAE assumes each i^{th} subject has its own prior distribution $p_i(\mathbf{z}_i)$ and

$$p_i(\mathbf{z}_i) \sim N(0, I)$$

As a result, each subject also has its own approximation $q_i(\mathbf{z}_i)$ for the posterior $p_i(\mathbf{z}_i|\mathbf{x}_i, \omega)$. The Encoder of a VAE designs $q_i(\mathbf{z}_i)$ using VI by factorizing $q_i(\mathbf{z}_i)$ with respect to coordinates of $\mathbf{z}_i$ and assumes that $q_i(\mathbf{z}_i)$ has the structure

$$q_i(\mathbf{z}_i; \phi) \sim N(f_1(\mathbf{x}_i; \phi_1), diag(f_2(\mathbf{x}_i; \phi_2))) \tag{2.20}$$

where f_1 and f_2 are different components of the Encoder consisting of multiple neural network layers with decreasing number of nodes with parameters ϕ_1, ϕ_2 respectively. Here, we also intentionally name the parameters $\phi = (\phi_1, \phi_2)$ same as ϕ in the green box in Figure 2.10. Notice that $q_i(\mathbf{z}_i; \phi)$ in Equation 2.20 are not independent among subjects since they have shared parameters $\phi = (\phi_1, \phi_2)$. This reduces the numbers of parameters by large amounts and avoid potential over-fitting problem, compared with assuming that each $q_i(\mathbf{z}_i; \phi)$ has its own ϕ. Also, for each latent variable $\mathbf{z}_i$, its coordinates are not independent since they share parameters ϕ_2, which makes $q_i(\mathbf{z}_i; \phi)$ more general than the mean-field variational family where coordinates of $\mathbf{z}_i$ are independent with each other.

Keen readers may notice that VAE could degenerate to a standard AE if $diag(f_2(\mathbf{x}_i; \phi_2))$ shrinks to a zero matrix during the training process. We argue that this is unlikely since if we rewrite ELBO as

$$\begin{aligned} L_{ELBO}(\omega, q) &= \sum_{i=1}^{N} E_{q_i(\mathbf{z}_i)} \left[\log \frac{p(\mathbf{x}_i, \mathbf{z}_i; \omega)}{q_i(\mathbf{z}_i)} \right] \\ &= \sum_{i=1}^{N} E_{q_i(\mathbf{z}_i)} \left[\log \frac{p(\mathbf{x}_i | \mathbf{z}_i; \omega) p_i(\mathbf{z}_i)}{q_i(\mathbf{z}_i)} \right] \\ &= \sum_{i=1}^{N} [E_{q_i(\mathbf{z}_i)} \log p(\mathbf{x}_i | \mathbf{z}_i, \omega) \\ &\quad - KL(q_i(\mathbf{z}_i) || p_i(\mathbf{z}_i))] \end{aligned} \tag{2.21}$$

we can see that the second term in Equation 2.21 is a KL divergence and acts like a regularization term, which allows noise in the model and thus prevents VAE shrink down to standard AE.

Now we can have a clear understanding of the Variational EM-algorithm if we go back to Figure 2.10. When the variational EM-algorithm trains the VAE and updates parameters ϕ in the Encoder, it is similar to finding the best candidate $q_i(z_i)$ to approximate $p_i(z_i | x_i, \omega)$ in the E-step; when it updates the parameters ω in the Decoder, the procedure corresponds to the M-step and maximizes the ELBO.

2.3.4 Reparameterization Trick

In theory, we can sample from $q_i(\mathbf{z}_i)$ in Equation 2.20 once it is updated and feed the sampled data to the Decoder in the forward propagation step. However,

it is impossible to implement this procedure because backward propagations cannot be defined for a random sampling process before feeding the data to the Decoder. The reparametrization trick is then applied to redefine the latent random variable $\mathbf{z}_i$ and extract its random part outside of the neural network node. In this way, all nodes remain deterministic while we still include the random mechanism. More specifically, it randomly samples a value ϵ from a standard Gaussian $N(0, I)$ and scale it by the latent distribution variance $diag(f_2(\mathbf{x}_i|\phi_2))$ and shift it by the mean $f_1(\mathbf{x}_i|\phi_1)$. Now, we have transformed the sampling process as something done outside what the backward propagation pipeline handles, and the sampled value ϵ is similar to another input of the model fed into the bottleneck, as shown in Figure 2.12 (Paul, 2020).

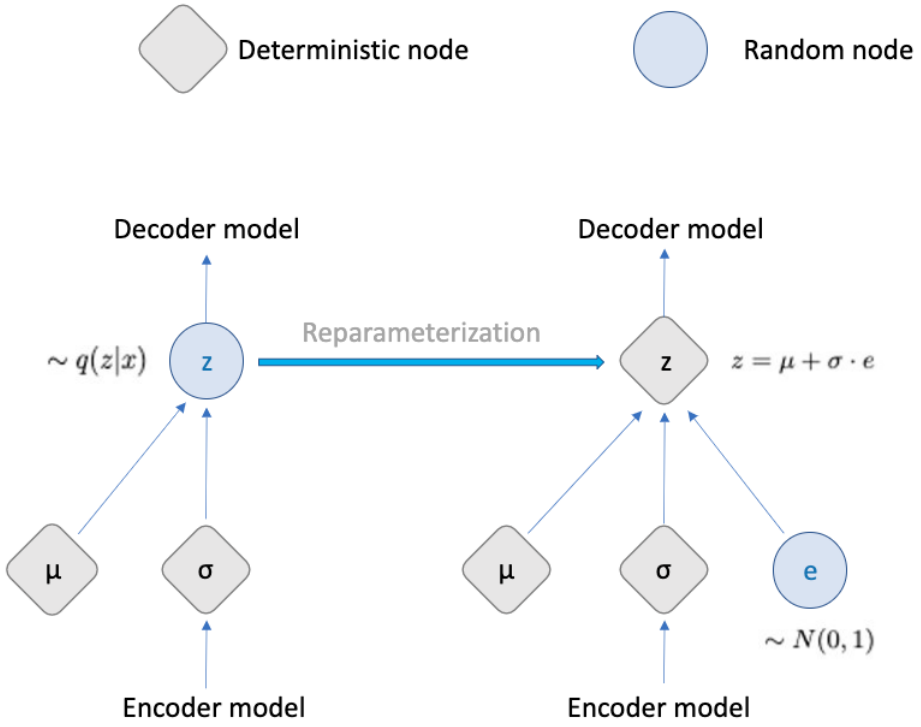

Figure 2.12: A graphical view of reparameterization trick

Chapter 3

Causal Effect Variational Auto-Encoder with Importance Sampling Weight

This chapter will propose our model and elaborate on some technical details. Section 3.1 introduces some previous work related to our model in detail. We propose our model, named Importance-sampling Causal Effect with Disentangled Auto-Encoder (ICEDVAE), in Section 3.2.

3.1 Related work

3.1.1 VAE for Causal Inference

VAE has not been introduced for causal inference until recently. The pioneering work is from Louizos et al., 2017 where they apply VAE to analyze the treatment effect in the observational study and propose the Causal Effect Variational Auto-Encoder (CEVAE). By translating the confounders as latent variables, they propose a CBN shown in Figure 3.1 describing the causal relations among variables. In Figure 3.1, **Z** refers to the unobserved confounders, **X** means the observed covariates, and T, Y represent the treatment and the outcome, respectively.

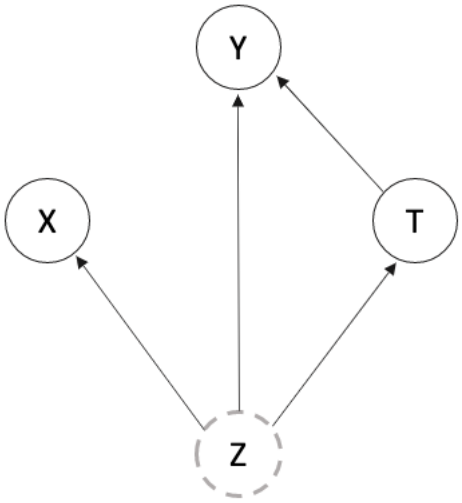

Figure 3.1: Bayesian network for CEVAE

Recall that training a VAE requires to maximize the ELBO

$$\mathcal{L}_{ELBO}(\omega, q) = \sum_{i=1}^{N} E_{q(\mathbf{z}_i|\mathbf{x}_i;\phi)} \left[\log \frac{p(\mathbf{x}_i, \mathbf{z}_i; \omega)}{q(\mathbf{z}_i|\mathbf{x}_i; \phi)} \right] \tag{3.1}$$

In the CBN shown in Figure 3.1, the joint probability distribution becomes $p(y_i, \mathbf{x}_i, \mathbf{z}_i, t_i|\omega)$, which can be further factorized by the Local Markov Assumption,

$$p(y_i, \mathbf{x}_i, \mathbf{z}_i, t_i; \omega) = p(y_i|t_i, \mathbf{z}_i; \omega)p(\mathbf{x}_i|\mathbf{z}_i; \omega)p(t_i|\mathbf{z}_i; \omega)p(\mathbf{z}_i)$$

where ω represents the set of parameters in the Decoder. The posterior distribution of $\mathbf{z}_i$ is replaced with $q(\mathbf{z}_i|t_i, y_i, \mathbf{x}_i; \phi)$ where ϕ is the set of parameters in the Encoder. After substitution for Equation 3.1, the ELBO becomes

$$\begin{aligned}\mathcal{L}_{ELBO} = \sum_{i=1}^{N} E_{q(\mathbf{z}_i|\mathbf{x}_i,t_i,y_i;\phi)}[&\log p(\mathbf{x}_i|\mathbf{z}_i; \omega) \\ &+ \log p(t_i|\mathbf{z}_i; \omega) + \log p(y_i|\mathbf{z}_i, t_i; \omega) + \log p(\mathbf{z}_i) - \log q(\mathbf{z}_i|\mathbf{x}_i, t_i, y_i; \phi)]\end{aligned}$$

The latent variable $\mathbf{z}_i$ is assumed to have the standard normal prior distribution

$$p(\mathbf{z}_i) = \prod_{j=1}^{D_z} \mathcal{N}(z_{i(j)}|0, 1)$$

where D_z is the dimension of $\mathbf{z}_i$ and $z_{i(j)}$ is the j^{th} coordinate of $\mathbf{z}_i$. $p(\mathbf{x}_i|\mathbf{z}_i)$ is also factorized as

$$p(\mathbf{x}_i|\mathbf{z}_i;\omega) = \prod_i^{D_x} p(x_{i(j)}|\mathbf{z}_i;\omega) \tag{3.2}$$

where D_x is the dimension of $\mathbf{x}_i$. Each $p(x_{i(j)}|\mathbf{z}_i)$ can be adjusted to either continuous or discrete distribution catering to the data type of $x_{i(j)}$. Since the treatment is treated as a binary variable, $p(t_i|\mathbf{z}_i)$ is assumed to have a Bernoulli distribution

$$p(t_i|\mathbf{z}_i;\omega) = Bern(\sigma(f_1(\mathbf{z}_i)))$$

where $\sigma(\cdot)$ is the sigmoid function, and f_1 is a neural network with fully connected layers. For a continuous outcome, $p(y_i|t_i,\mathbf{z}_i)$ is parameterized as Gaussian with its mean given by a TAR-net (Shalit et al., 2017) architecture, i.e. a treatment specific function, and its variance fixed to v shared by all subjects irrelevant to the treatment,

$$p(y_i|t_i,\mathbf{z}_i,\omega) = \mathcal{N}(\mu=\mu_i, s^2=v), \quad \mu_i = t_i f_2(\mathbf{z}_i)+(1-t_i)f_3(\mathbf{z}_i) \tag{3.3}$$

where f_2 and f_3 are FC layers. For the binary outcome, Gaussian distribution is replaced with Bernoulli distribution with structures of f_2, f_3 unchanged,

$$Bern(\pi=\pi_i), \quad \pi_i = \sigma(t_i f_2(\mathbf{z}_i) + (1-t_i)f_3(\mathbf{z}_i)) \tag{3.4}$$

By assuming the posterior distributions of $\mathbf{z}_i$ can be factorized with respect to its coordinates following the idea of VI, Louizos et al., 2017 construct $q(\mathbf{z}_i|\mathbf{x}_i,t_i,y_i)$ as

$$q(\mathbf{z}_i|\mathbf{x}_i,t_i,y_i;\phi) = \mathcal{N}(\mu_i, diag(\mathbf{s}_i^2)) = \prod_{j=1}^{D_z} \mathcal{N}(\mu_{i(j)}, s_{i(j)}^2) \tag{3.5}$$

where $\mu_{i(j)}$ and $s_{i(j)}^2$ are the j^{th} coordinate of μ_i and $\mathbf{s}_i^2$, respectively. To reflect the different generation process in different treatment arms, μ_i and $\mathbf{s}_i^2$ are further modeled as

$$\mu_i = t_i\mu_{1i} + (1-t_i)\mu_{0i}; \quad \mathbf{s}_i^2 = t_i\mathbf{s}_{1i}^2 + (1-t_i)\mathbf{s}_{0i}^2 \tag{3.6}$$

$$\mu_{0i}, \mathbf{s}_{0i}^2 = g_2 \circ g_1(\mathbf{x}_i, y_i); \quad \mu_{1i}, \mathbf{s}_{1i}^2 = g_3 \circ g_1(\mathbf{x}_i, y_i) \tag{3.7}$$

where g_i are different FC layers, μ_{0i} is the mean value for the i^{th} subject if it is assigned to the control group, and μ_{1i} is for the subject assigned to the treatment group. As the output vector of g_2 in Equation 3.7, the first half of coordinates

are used to generate μ_{0i} and the second are for $\mathbf{s}_{0i}^2$ using an extra layer, respectively. $\mu_{1i}, \mathbf{s}_{1i}^2$ are constructed in a similar way with the output vector of g_3. Since the variance $\mathbf{s}_{1i}^2, \mathbf{s}_{0i}^2$ should be positive, a *soft-plus* function

$$SoftPlus(x, \beta) = \frac{1}{\beta} \times \log(1 + \exp(\beta x))$$

is applied to satisfy the positivity constraint, where β is set to the default value 1 in the model. Notice that for out-of-sample predictions, the treatment assignment t along with its outcome y are required before inferring the distribution over $\mathbf{z}$ in Equation 3.5. For this reason, two auxiliary distributions are introduced to predict t_i, y_i for new samples. More specifically, the auxiliary distribution for treatment assignment t is constructed with another Bernoulli distribution

$$q_{aux}(t_i|\mathbf{x}_i) = Bern(\pi = \sigma(g_4(\mathbf{x}_i))) \tag{3.8}$$

and auxiliary distributions for continuous and binary outcome y are also constructed accordingly similar to Equation 3.3,3.4, and 3.7,

$$\begin{aligned}
q_{aux}(y_i|t_i, \mathbf{x}_i) &= \mathcal{N}(\mu = \mu_{aux}, \mathbf{s}_{aux}^2 = v) && (3.9)\\
q_{aux}(y_i|t_i, \mathbf{x}_i) &= Bern(\pi = \pi_{aux}) && (3.10)\\
\mu_{aux} &= t_i(g_6 \circ g_5(\mathbf{x}_i)) + (1 - t_i)(g_7 \circ g_5(\mathbf{x}_i)) && (3.11)\\
\pi_{aux} &= t_i(g_6 \circ g_5(\mathbf{x}_i)) + (1 - t_i)(g_7 \circ g_5(\mathbf{x}_i)) && (3.12)
\end{aligned}$$

The final loss function incorporates both $\mathcal{L}_{ELBO}$ and auxiliary distributions

$$\mathcal{L}_{CEVAE} = \mathcal{L}_{ELBO} + \sum_{i=1}^{N} [q_{aux}(t_i|\mathbf{x}_i) + q_{aux}(y_i|t_i, \mathbf{x}_i)] \tag{3.13}$$

3.1.2 Counterfactual Regression with Importance Sampling Weight

Recall that *importance sampling* (Kloek and Van Dijk, 1978; Van Dijk and Kloek, 1983) is used to compute $E_{x\sim p(x)}[f(x)]$ when in fact we observe samples that are drawn from an alternative distribution $q(x)$. It is easy to show that

$$E_{x\sim p(x)}[f(x)] = E_{x\sim q(x)}\left[f(x)\frac{p(x)}{q(x)}\right] \tag{3.14}$$

In practice, we first need to identify the distribution q that generates the data, then design a distribution p that helps improve the evaluation performance.

The CounterFactual Regression (CFR) (Hassanpour and Greiner, 2019) is the pioneering work of incorporating importance sampling to construct weights and enhance the prediction power on the outcome Y. More specifically, CFR first uses FC layers $\Phi(\cdot)$ to encode covariate vector $\mathbf{X}$ into a representation space and generates two similar latent distribution $\{\Phi(\mathbf{x}_i)\}_{i:t_i=0}$, $\{\Phi(\mathbf{x}_i)\}_{i:t_i=1}$ imitating the RCT setting. Then, $\Phi(\mathbf{x}_i)$ are used to predict the outcome Y. The final loss function is

$$\frac{1}{N}\sum_{i=1}^{N} w_i L(y_i, h^{t_i}(\Phi(\mathbf{x}_i))) + \alpha \times IPM(\{\Phi(\mathbf{x}_i)\}_{i:t_i=0}, \{\Phi(\mathbf{x}_i)\}_{i:t_i=1}) + \lambda \times Reg \tag{3.15}$$

where the second term measures the discrepancy between distributions $\{\Phi(\mathbf{x}_i)\}_{i:t_i=0}$, $\{\Phi(\mathbf{x}_i)\}_{i:t_i=1}$ using the Integral Probability Metric (IPM) (Müller, 1997; Sriperumbudur et al., 2012), and the third term is related to model regularization. What we focus on is the first term $\frac{1}{N}\sum_{i=1}^{N} w_i L(y_i, h^{t_i}(\Phi(\mathbf{x}_i)))$, an estimator of $E_{p(y,\Phi(\mathbf{x})|t)}[wL(y, h^t(\Phi(\mathbf{x})))]$, that corresponds to the prediction power on the outcome. Two separate FC layers h^{t_i} are designed for each treatment group and take learned representation vectors $\Phi(\mathbf{x}_i)$ for outome prediction. The importance sampling weight w_i is introduced here to enhance the prediction power of h^{t_i}. To construct w_i, Hassanpour and Greiner, 2019 design a CBN

$$T \leftarrow \mathbf{X} \rightarrow \Phi(\mathbf{X}) \rightarrow (Y(0), Y(1))$$

Notice that $T \perp Y|\Phi(\mathbf{X})$, the distribution "$q(x)$" from which the data actually come in Equation 3.14 is then

$$P(y, \Phi(\mathbf{x})|t) = P(y|\Phi(\mathbf{x}), t)p(\Phi(\mathbf{x})|t) = P(y|\Phi(\mathbf{x}))P(\Phi(\mathbf{x})|t)$$

To emphasize the prediction power for both factual and counterfactual outcomes, the distribution density "$p(x)$" in Equation 3.14 is chosen to be

$$p(\mathbf{x}) = p(y, \Phi(\mathbf{x})|t) + p(y, \Phi(\mathbf{x})|\neg t)$$

where $\neg t$ means the counterfactual treatment. This yields the likelihood ratio

$$\begin{aligned}\frac{p(\mathbf{x})}{q(\mathbf{x})} &= \frac{p(y,\Phi(\mathbf{x})|t)+p(y,\Phi(\mathbf{x})|\neg t)}{p(y,\Phi(\mathbf{x})|t)} \\ &= 1+\frac{p(y|\Phi(\mathbf{x}))p(\Phi(\mathbf{x})|\neg t)}{p(y|\Phi(\mathbf{x}))p(\Phi(\mathbf{x})|t)} \\ &= 1+\frac{p(\Phi(\mathbf{x})|\neg t)}{p(\Phi(\mathbf{x})|t)}\end{aligned}$$

As a result, the weight for each subject becomes

$$w_i = 1+\frac{p(\Phi(\mathbf{x}_i)|\neg t_i)}{p(\Phi(\mathbf{x}_i)|t_i)} \tag{3.16}$$

With Bayes formula, Equation (3.16) can be expressed as

$$\begin{aligned}w_i &= 1+\frac{\frac{p(\neg t_i|\Phi(\mathbf{x}_i))p(\Phi(\mathbf{x}_i))}{p(\neg t_i)}}{\frac{p(t_i|\Phi(\mathbf{x}_i))p(\Phi(\mathbf{x}_i))}{p(t_i)}} \\ &= 1+\frac{p(t_i)}{1-p(t_i)}\cdot\frac{1-p(t_i|\Phi(\mathbf{x}_i))}{p(t_i|\Phi(\mathbf{x}_i))}\end{aligned}$$

where $p(t_i|\Phi(\mathbf{x}_i))$ is parameterized by additional parameters $[\mathbf{W}, b]$ as

$$p(t_i|\Phi(\mathbf{x}_i)) = \frac{1}{1+e^{-(2t_i-1)[\Phi(\mathbf{x}_i)\cdot\mathbf{W}+b]}}$$

and parameters $[\mathbf{W}, b]$ are learned by minimizing the loss function of the traditional logistic regression

$$C(\mathbf{W}, b) = \frac{1}{N}\sum_{i=1}^{N} -log[p(t_i|\Phi(\mathbf{x}_i))]$$

Since updating parameters in $\Phi(\cdot)$ needs fixed $[\mathbf{W}, b]$ for estimating ω_i's, and updating ω_i's requires fixed $\Phi(\mathbf{x}_i)$'s, Hassanpour and Greiner, 2019 propose a training algorithm to update $\Phi(\cdot)$ and $[\mathbf{W}, b]$ alternately.

3.1.3 DR-CFR

Instead of treating $\mathbf{X}$ as a whole and feeding it into $\Phi(\cdot)$ to predict the outcome in CFR, Disentangled Representation (DR)-CFR uses three different neural networks and decomposes $\mathbf{X}$ into three parts (Figure 3.2): $\Gamma(\mathbf{X})$ (affect treatment only), $\Upsilon(\mathbf{X})$ (affect outcome only), and confounding part $\Delta(\mathbf{X})$.

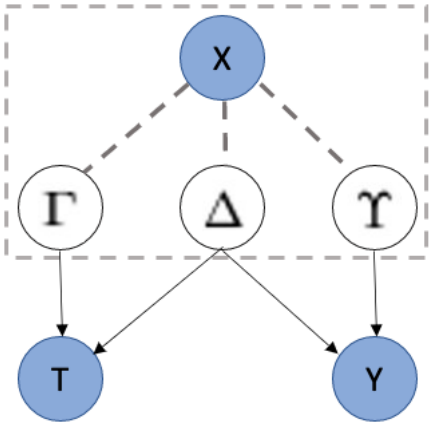

Figure 3.2: Decomposition of DR-CFR

Compared with CFR, DR-CFR is different in many aspects. First of all, the representation predictors for Y changes from $\Phi(\mathbf{X})$ to $\Delta(\mathbf{X})$ and $\Upsilon(\mathbf{X})$, i.e. $L(h^{t_i}(\Phi(\mathbf{x}_i)), y_i)$ changes to $L(h^{t_i}(\Delta(\mathbf{x}_i), \Upsilon(\mathbf{x}_i)), y_i)$ in Equation 3.15. Secondly, due to the Collider structure at the outcome Y in Figure 3.2, Υ is independent with the treatment T, i.e. the discrepancy between two distributions $\{\Upsilon(\mathbf{x}_i)\}_{i:t_i=0}$ and $\{\Upsilon(\mathbf{x}_i)\}_{i:t_i=1}$ should be as small as possible. To reflect this independence, an additional term $disc(\{\Upsilon(\mathbf{x}_i)\}_{i:t_i=0}, \{\Upsilon(\mathbf{x}_i)\}_{i:t_i=1})$ is added to Equation 3.15. Thirdly, an extra cross entropy loss $-\log \pi_0(T|\Gamma(\mathbf{X}), \Delta(\mathbf{X}))$ is added to Equation 3.15 to increase the prediction power on T with $(\Gamma(\mathbf{X}), \Delta(\mathbf{X}))$ and strengthen the link from $(\Gamma(\mathbf{X}), \Delta(\mathbf{X}))$ to T. Finally, DR-CFR only incorporates $\Delta(\mathbf{X})$ instead of the whole $\Phi(\mathbf{X})$ to construct the importance sampling weights.

3.2 Proposed model

3.2.1 Main Idea

Following the improvement steps of DR-CFR on CFR, we decide to modify CEVAE similarly, that is, further decomposing $\mathbf{Z}$ in Figure 3.1. Then, we apply importance sampling weights to increase the model's outcome prediction power.

The first part has been done by Zhang et al., 2021 in their Treatment Effect by Disentangled Variational AutoEncoder (TEDVAE) model, and we will follow

their decomposition framework. More specifically, we start by decomposing $\mathbf{Z}$ into three factors named as the $\mathbf{Z}_t$, $\mathbf{Z}_y$, and $\mathbf{Z}_c$, and we assume that $\{\mathbf{Z}_t$,$\mathbf{Z}_y, \mathbf{Z}_c\}$ commonly generate the feature space $\mathbf{X}$. Moreover, $\mathbf{Z}_t$ also has a direct causal effect on the treatment T, and the causal effect also exists from $\mathbf{Z}_y$ to Y; $\mathbf{Z}_c$ acts like a "confounder" and has causal effects on both T and Y. A detailed CBN describing the causal relationships among $\{\mathbf{Z}_t$,$\mathbf{Z}_y, \mathbf{Z}_c, T, Y\}$ is depicted in Figure 3.3. This disentanglement is crucial since both the bias (Abadie and Imbens,

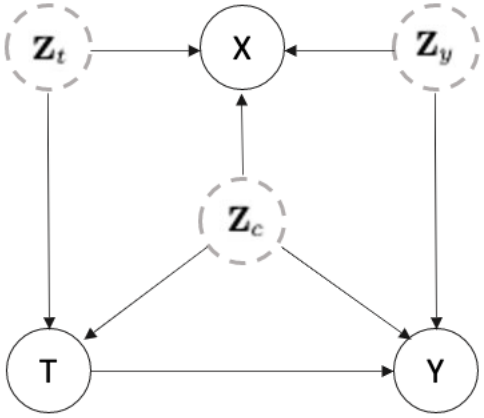

Figure 3.3: Graphical model for data generation

2006) and the variance (Hahn, 1998) of treatment effect estimation will increase if we include variables unrelated to the outcome as confounders. Furthermore, This CBN also relax the ignorability assumption since if we conditioned on X, the Collider structure around X is unblocked and thus T and $Y(T)$ becomes dependent. We then apply the framework of Louizos et al., 2017 to train a VAE accordingly and assign the importance sampling weights to the auxiliary term for better outcome prediction.

The second part is original, which makes our work different from Zhang et al., 2021: in their work, fixed hyper-parameters are used to penalize the outcome and treatment prediction errors. Finding the best hyper-parameters is always time-consuming since we need to try different combinations of them by cross-validation. In contrast, the importance sampling weights can be automatically learned during the model training process, which reduces the number of hyper-parameters and thus saves time on cross-validation. As we will see, the components of importance sampling weights are already constructed in the VAE part, so we don't introduce extra model parameters which increase the model complexity.

3.2.2 Technical Details

The first thing we check is whether the causal effect from T to Y is identified in Figure 3.3. The following theorem proves that we can estimate the causal effect from this causal Bayesian network if $\mathbf{Z}_c$ is identifiable.

Theorem 3.2.1. *(Zhang et al., 2021) The effect of T on Y can be identified if we recover the confounding factors $\mathbf{Z}_c$ from the data.*

Proof. From Figure 3.3 we know that $\mathbf{Z}_t, \mathbf{Z}_c$ are the parents of the treatment T satisfying the backdoor criterion, so we have

$$p(y|do(t)) = \sum_{\mathbf{z}_t} \sum_{\mathbf{z}_c} p(y|t, \mathbf{z}_t, \mathbf{z}_c) p(\mathbf{z}_t) p(\mathbf{z}_c)$$

Since $Y \perp \mathbf{Z}_t | t, \mathbf{Z}_c$, we have

$$\begin{aligned} p(y|do(t)) &= \sum_{\mathbf{z}_t} \sum_{\mathbf{z}_c} p(y|t, \mathbf{z}_c) p(\mathbf{z}_t) p(\mathbf{z}_c) \\ &= \sum_{\mathbf{z}_t} p(\mathbf{z}_t) \sum_{\mathbf{z}_c} p(y|t, \mathbf{z}_c) p(\mathbf{z}_c) \\ &= \sum_{\mathbf{z}_c} p(y|t, \mathbf{z}_c) p(\mathbf{z}_c) \end{aligned}$$

□

Our goal is to find a suitable variational mean-field family and choose the best candidate to approximate the posterior distribution of latent variables $p(\mathbf{z}|\mathbf{x}, t, y)$ with $\mathbf{z} = \{\mathbf{z}_t, \mathbf{z}_y, \mathbf{z}_c\}$. Notice that $\mathbf{Z}_t, \mathbf{Z}_y, \mathbf{Z}_c$ are independent in the CBN shown in Figure 3.3 since all paths connecting any two of them are blocked by Colliders. Thus, we assume the distribution in the mean-field variational family $q(\mathbf{z}_t, \mathbf{z}_c, \mathbf{z}_y)$ can be factorized with respect to each latent variable, i.e. $q(\mathbf{z}_t, \mathbf{z}_c, \mathbf{z}_y) = q^t(\mathbf{z}_t) q^c(\mathbf{z}_c) q^y(\mathbf{z}_y)$. Using the standard ELBO expression in Equation 2.21 and factorizing the joint distribution $p(\mathbf{x}, t, y, \mathbf{z}_t, \mathbf{z}_c, \mathbf{z}_y; \omega)$, we can rewrite the ELBO for the CBN shown in Figure 3.3 as

$$\begin{aligned} \mathcal{L}_{ELBO} &= E_{q^t, q^c, q^y} \log p(\mathbf{x}|\mathbf{z}_t, \mathbf{z}_c, \mathbf{z}_y) \\ &\quad + E_{q^c, q^y} \log p(y|t, \mathbf{z}_y, \mathbf{z}_c) + E_{q^t, q^c} \log p(t|\mathbf{z}_t, \mathbf{z}_c) \\ &\quad - KL(q^t(\mathbf{z}_t)||p(\mathbf{z}_t)) - KL(q^c(\mathbf{z}_c)||p(\mathbf{z}_c)) \\ &\quad - KL(q^y(\mathbf{z}_y)||p(\mathbf{z}_y)) \end{aligned} \tag{3.17}$$

Following standard VAE design, the prior distributions $p(\mathbf{z}_t), p(\mathbf{z}_c), p(\mathbf{z}_y)$ are chosen as Gaussian distributions (Kingma and Welling, 2013).

$$p(\mathbf{z}_t) = \mathcal{N}(0, I_{D_{\mathbf{z}_t}}) = \prod_j^{D_{\mathbf{z}_t}} \mathcal{N}(z_{t(j)}; \mu = 0, \sigma^2 = 1)$$
$$p(\mathbf{z}_c) = \mathcal{N}(0, I_{D_{\mathbf{z}_c}}) = \prod_j^{D_{\mathbf{z}_c}} \mathcal{N}(z_{c(j)}; \mu = 0, \sigma^2 = 1)$$
$$p(\mathbf{z}_y) = \mathcal{N}(0, I_{D_{\mathbf{z}_y}}) = \prod_j^{D_{\mathbf{z}_y}} \mathcal{N}(z_{y(j)}; \mu = 0, \sigma^2 = 1)$$

where $D_{\mathbf{z}_t}, D_{\mathbf{z}_c}, D_{\mathbf{z}_y}$ are dimensions of latent variables $\mathbf{z}_t, \mathbf{z}_c, \mathbf{z}_y$ respectively. Since the treatment variable T is binary, we assume $p(t|\mathbf{z}_t, \mathbf{z}_c)$ has a Bernoulli distribution

$$p(t|\mathbf{z}_t, \mathbf{z}_c) = Bern(\pi = \sigma(f_1(\mathbf{z}_t, \mathbf{z}_c)) \tag{3.18}$$

where f_1 is a fully connected neural network layers. The generative model for $\mathbf{X}$ is

$$p(\mathbf{x}|\mathbf{z}_t, \mathbf{z}_c, \mathbf{z}_y) = \prod_{j=1}^{D_{\mathbf{x}}} p(x_{(j)}|\mathbf{z}_t, \mathbf{z}_c, \mathbf{z}_y) \tag{3.19}$$

with D_x being the dimension of x and $x_{(j)}$ the j^{th} coordinate of $\mathbf{x}$. $p(x_{(j)}|\mathbf{z}_t, \mathbf{z}_c, \mathbf{z}_y)$ caters to the distribution of $x_{(j)}$ similar to Equation 3.2. For the continuous outcome Y, we parameterize $p(y|t, \mathbf{z}_c, \mathbf{z}_y)$ similar to Equation 3.3 using a Gaussian distribution

$$\begin{aligned} p(y|t, \mathbf{z}_c, \mathbf{z}_y) &= \mathcal{N}(\mu, \mathbf{s}^2) \\ \mu &= t \times f_2(\mathbf{z}_c, \mathbf{z}_y) + (1-t) \times f_3(\mathbf{z}_c, \mathbf{z}_y) \\ \mathbf{s}^2 &= t \times f_4(\mathbf{z}_c, \mathbf{z}_y) + (1-t) \times f_5(\mathbf{z}_c, \mathbf{z}_y) \end{aligned} \tag{3.20}$$

where f_2, f_3, f_4, f_5 are different FC layers parameterized by their own parameters. Notice that we also cater $\mathbf{s}^2$ to different treatment arms instead of a fixed one compared with Equation 3.3. The parameterization of $p(y|t, \mathbf{z}_c, \mathbf{z}_y)$ for binary outcome Y is similar to Equation 3.4

$$\begin{aligned} &p(y|t, \mathbf{z}_c, \mathbf{z}_y) = Bern(p) \\ &p = t \times \sigma(f_6(\mathbf{z}_c, \mathbf{z}_y)) + (1-t) \times \sigma(f_7(\mathbf{z}_c, \mathbf{z}_y)) \end{aligned}$$

To parameterize $q^t(\mathbf{z}_t), q^c(\mathbf{z}_c), q^y(\mathbf{z}_y)$, we start by checking the loss function in Equation 3.17. In the CEVAE model, two auxiliary functions are constructed

to predict out-of-sample (T, Y), which are required to infer the variational posterior distribution of Z in Equation 3.5. These two auxiliary functions and additional terms related to T and Y in Equations3.5-3.7 increase the number of model parameters and thus the risk of over-fitting. To make the model more robust, we propose that the variational approximations of the posteriors learn only from $\mathbf{X}$. Specifically, similar to Equation 3.5, we define $q^t(\mathbf{z}_t), q^c(\mathbf{z}_c), q^y(\mathbf{z}_y)$ as

$$\begin{aligned} q^t(\mathbf{z}_t) &= \mathcal{N}(\mu_t, diag(\mathbf{s}_t^2)) \\ q^c(\mathbf{z}_c) &= \mathcal{N}(\mu_c, diag(\mathbf{s}_c^2)) \\ q^y(\mathbf{z}_y) &= \mathcal{N}(\mu_y, diag(\mathbf{s}_y^2)) \end{aligned}$$

where μ_t, μ_c, μ_y and $\mathbf{s}_t^2$, $\mathbf{s}_c^2$, $\mathbf{s}_y^2$ are the means and variances of the Gaussian distributions parameterized by different FC layers f_8, f_9, f_{10}:

$$\mu_t, \mathbf{s}_t^2 = f_8(\mathbf{x}) \quad \mu_c, \mathbf{s}_c^2 = f_9(\mathbf{x}) \quad \mu_y, \mathbf{s}_y^2 = f_{10}(\mathbf{x}) \tag{3.21}$$

Notice that we only have $\mathbf{x}$ as the input in Equation 3.21 which significantly reduces the number of parameters without constructing different neural networks for each treatment arm.

To guarantee the prediction power on Y, we directly rely on $E_{q^c q^y} \log p(y|t, \mathbf{z}_y, \mathbf{z}_c)$ in Equation 3.17 and add a penalty coefficient in front of it. This is based on Theorem 2 in Zhang et al., 2021 and the fact that

$$\begin{aligned} p(y|do(t), \mathbf{X}) &= \sum_{\mathbf{z}_t} \sum_{\mathbf{z}_c} \sum_{\mathbf{z}_y} p(y, \mathbf{z}_t, \mathbf{z}_c, \mathbf{z}_y|do(t), \mathbf{X}) \\ &= \sum_{\mathbf{z}_t} \sum_{\mathbf{z}_c} \sum_{\mathbf{z}_y} p(y|\mathbf{z}_c, \mathbf{z}_y, t, \mathbf{X}) p(\mathbf{z}_t|\mathbf{X}) p(\mathbf{z}_c|\mathbf{X}) p(\mathbf{z}_y|\mathbf{X}) \\ &= \sum_{\mathbf{z}_c} \sum_{\mathbf{z}_y} p(y|\mathbf{z}_c, \mathbf{z}_y, t) p(\mathbf{z}_c|\mathbf{X}) p(\mathbf{z}_y|\mathbf{X}) \end{aligned}$$

Unlike the work of Zhang et al., 2021 where penalty coefficients are treated as hyper-parameters and need manually tuned, we propose to apply the importance sampling weights for the penalty coefficient construction which can be learned automatically during the training process. More specifically, we define the weight w similar to Equation 3.16 as

$$
\begin{aligned}
w(y,t,\mathbf{z}) &= 1+\frac{p(y|\neg t,\mathbf{z})}{q(y|t,\mathbf{z})} \\
&= 1+\frac{p(y,\mathbf{z}|\neg t)}{p(y,\mathbf{z}|t)} = 1+\frac{p(y,\mathbf{z})-p(y,t,\mathbf{z})}{p(y,t,\mathbf{z})}\frac{p(t)}{p(\neg t)} \\
&= 1+\frac{p(y|\mathbf{z})-p(y,t|\mathbf{z})}{p(y,t|\mathbf{z})}\frac{p(t)}{1-p(t)} \\
&= 1+\frac{p(y|\mathbf{z})-p(y|t,\mathbf{z})p(t|\mathbf{z})}{p(y|t,\mathbf{z})p(t|\mathbf{z})}\frac{p(t)}{1-p(t)} \qquad (3.22)
\end{aligned}
$$

where $\frac{p(y|\neg t,\mathbf{z})}{q(y|t,\mathbf{z})}$ represents the likelihood ratio weight function in Equation (3.16), and $\mathbf{z}$ refers to latent variables in VAE which includes $\mathbf{z}_t$, $\mathbf{z}_c$ and $\mathbf{z}_y$. Notice that we already have $p(t|\mathbf{z})$ and $p(y|t,\mathbf{z})$ in the VAE part, i.e. $p(t|\mathbf{z}) = p(t|\mathbf{z}_t,\mathbf{z}_c)$ and $p(y|t,\mathbf{z}) = p(y|t,\mathbf{z}_y,\mathbf{z}_c)$ from Equation 3.18 and 3.20, respectively. However, we do not have $p(y|\mathbf{z})$ and we have to construct another neural network to estimate this term. To further decrease the number of parameters in our model and make our model more robust, we made some approximation when constructing the importance sampling weight and assume that $P(y|\mathbf{z},t) \approx P(y|\mathbf{z})$. Although this "homogeneity" approximation might be strong and can cause some potential inaccuracy of our model, the heterogeneous causal effect has already been considered in Equation 3.20, thus this approximation should not cause too much bias. Another interpretation of this approximation is $Y \perp T|\mathbf{Z}$, which implies that the treatment effect is weak. In other words, the assumption is reasonable if we are certain the treatment effect is small. Small treatment effects are usually difficult to estimate and can happen in mortality analysis of diseases, such as studies comparing treatment-related mortality rates for two chemotherapies for breast cancer (McGough and Faraone, 2009). As we will see in Section 4, this approximation will help our model perform well in estimating small treatment effects. Once we can estimate small treatment effects more accurately based on observational data, we can then better calculate sample sizes needed to provide sufficient power to detect meaningful treatment effects via RCT.

With this "homogeneity" approximation, Equation (3.22) can be simplified as

$$
\frac{p(y|\mathbf{z})-p(y|t,\mathbf{z})p(t|\mathbf{z})}{p(y|t,\mathbf{z})p(t|\mathbf{z})}\frac{p(t)}{1-p(t)} = \frac{1-p(t|\mathbf{z})}{p(t|\mathbf{z})}\frac{p(t)}{1-P(t)} \qquad (3.23)
$$

which corresponds to the one in (Hassanpour and Greiner, 2019). The weighted objective function is shown in Equation (3.24) and we name our method Importance-sampling Causal Effect with Disentangled Variational Auto-Encoder (ICED-VAE).

$$
\begin{aligned}
\mathcal{L}_{ICEDVAE} \quad = \quad & E_{q^t,q^c,q^y} \log p(x|\mathbf{z}_t, \mathbf{z}_c, \mathbf{z}_y) \\
& + E_{q^c,q^y} w \cdot \log p(y|t, \mathbf{z}_y, \mathbf{z}_c) + E_{q^t,q^c} \log p(t|\mathbf{z}_t, \mathbf{z}_c) \\
& - KL(q^t(\mathbf{z}_t)||p(\mathbf{z}_t)) - KL(q^c(\mathbf{z}_c)||p(\mathbf{z}_c)) \\
& - KL(q^y(\mathbf{z}_y)||p(\mathbf{z}_y)) \qquad (3.24)
\end{aligned}
$$

Chapter 4

Numerical studies

This chapter will train our model on several datasets and compare its performance with other State-Of-The-Art (SOTA) models. We will first describe the metric used to evaluate the model performance in Section 4.1 and then briefly introduce those SOTA models in Section 4.2.

4.1 Evaluation Metrics

The treatment effect for the i^{th} individual is defined as $\tau_i = Y_i(1) - Y_i(0)$. Due to the counterfactual problem, we never observe $Y_i(1)$ and $Y_i(0)$ simultaneously and thus τ_i is not observed for any individual. However, under some circumstances, CATE is easier for estimation which is defined in Equation 2.9:

$$CATE(\mathbf{x}) = \tau(\mathbf{x}) = E[Y(1) - Y(0)|\mathbf{X} = \mathbf{x}]$$

For evaluating the performance of CATE estimation, we use the Precision in Estimation of Heterogeneous Effect (PEHE) (Dorie et al., 2019; Hill, 2011; Louizos et al., 2017; Shalit et al., 2017) which measures the root mean squared distance between the estimated and true CATE when ground truth is available:

$$\epsilon_{PEHE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{\tau}(\mathbf{x}_i) - \tau(\mathbf{x}_i))^2}$$

where $\tau(\mathbf{x})$ is the ground truth CATE for subjects with observed variables $\mathbf{x}$.

Another metric we will use is the Area Under Curve of the Receiver Operating Characteristic (AUC_ROC), which shows the performance of a classification model with all classification thresholds. The ROC curve uses False Positive

Rate (FPR) and True Positive Rate (TPR) as x-axis and y-axis, respectively, in a plot, where FPR and TPR are defined as

$$FPR = \frac{FP}{FP + TN}, \quad TPR = \frac{TP}{TP + FN}$$

according to the *Confusion Matrix* shown in Table 4.1.

Table 4.1: Confusion Matrix

	Predicted Positive	Predicted Negative
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

Intuitively speaking, FPR conveys the message, "Among all actual negative units, what proportion of them are predicted as positive?" and TPR says, "Among all actual positive units, what proportion of them are predicted as positive?" As we lower the probability threshold of labeling a unit as positive, the amount of "Predicted Positive" units will increase, and so will the FPR and TPR. For an ideal classification model, the increasing rate of TPR should be much faster than the FPR rate. Since the threshold probability decreases along the positive direction of the x-axis when drawing a ROC curve, a higher increasing rate of TPR means the ROC curve is more concave, which leads to a larger AUC. To summarize, the higher the AUC's value, the better the classifier is.

Since AUC_ROC is a more widely used metric for evaluating binary classifiers' performance, we will follow the work of Louizos et al., 2017; Yao et al., 2018 and use AUC_ROC to evaluate model performances on the real-world dataset *Twins* with binary outcomes.

4.2 Introduction of State-Of-The-Art Models

Besides CEVAE, TEDVAE, CFR, and DR-CFR we mentioned in Chapter 3, we will also introduce some other models frequently used for estimating causal effects.

4.2.1 Marginal Structural Models (MSM)

MSM (Robins et al., 2000) is similar to the classical linear model. Suppose the outcome is continuous; a straightforward way to estimate ATE is to use a linear

regression model with treatment T as the predictor and check the corresponding coefficient β_1 in Equation 4.1,

$$E[Y|T] = \beta_0 + \beta_1 \times T \tag{4.1}$$

If we want to estimate CATE, we can add interaction terms between treatment T and covariates X. The model then becomes

$$E[Y|T, \mathbf{X}] = \beta_0 + \beta_1 T + \beta_3 \mathbf{X} + \beta_4 T * \mathbf{X} \tag{4.2}$$

and the estimated ITE is

$$E[Y|1, \mathbf{X}] - E[Y|0, \mathbf{X}] = \beta_1 + \beta_4 \mathbf{X}$$

However, this estimation process can be biased due to the confounding variables. Besides, we may introduce a pseudo-causal effect if there are Collider structures on $\mathbf{X}$. For example, suppose we have the causal graph $T \longrightarrow \mathbf{X} \longleftarrow Y$; when we fix $\mathbf{X}$ and interpret the coefficient of T, we also unblock the Collider and thus introduce a pseudo-causal effect even though there is no direct path between T and Y. To overcome this difficulty, we can use IPW to create balanced pseudo-populations such that we not only remove the bias but also break the causal path $T \longrightarrow \mathbf{X}$. This IPW procedure results in a four-step procedure for training an MSM: First, we estimate the propensity score for each subject using a separate model, e.g., logistic regression; Second, we create IPW using the formula

$$w_i = \frac{1}{t_i P(t_i = 1|\mathbf{x}_i) + (1 - t_i) P(t_i = 0|\mathbf{x}_i)}$$

Then, we specify MSM of interest, either for ATE (Equation 4.1) or CATE (Equation 4.2); Finally, we use the constructed IPW to fit a weighted version of Equation 4.1 or Equation 4.2.

4.2.2 X-learner

X-learner is extremely useful when estimating CATE. If the ignorability assumption holds, we can rewrite Equation 2.10 as follows by Theorem 2.2.1

$$\begin{aligned} \tau(\mathbf{x}) &= E[Y|T = 1, \mathbf{X} = \mathbf{x}] - E[Y|T = 0, \mathbf{X} = \mathbf{x}] \\ &\triangleq \mu_1(\mathbf{x}) - \mu_0(\mathbf{x}) \end{aligned}$$

Although it is straightforward to estimate $\tau(\mathbf{x})$ by training a model on all data, including both control and treated groups, the treatment effect will be biased towards zero if the dimension of $\mathbf{X}$ is too large (Künzel et al., 2019). One way to solve this issue is to partition the original data set into treated and control groups and fit models separately. This approach is called "T-learner" ("T" means "Two" here) and has been analyzed when the models are tree-based methods (Athey and Imbens, 2016). However, the "T-learner" approach does not fully utilize the data in each training branch and can cause high variance compared with the ones trained on complete data.

To increase the data utilization efficiency, Shalit et al., 2017 propose the Treatment-Agnostic Representation Network (TAR-Net). The main idea is to first train the model on X with all data for representation constructions and then branch off into two branches – one for the treated group and another for the control group. TAR-Net fully utilizes the data in the first step, but data is still separated into two parts in the second step, which can lead to limited improvement.

To further improve the efficiency, Künzel et al., 2019 propose the "X-learner" to further mix the control and treated group data. Specifically, we first choose two models to estimate $\mu_1(\mathbf{x})$ and $\mu_0(\mathbf{x})$ and obtain $\hat{\mu}_1(\mathbf{x})$ and $\hat{\mu}_0(\mathbf{x})$ using treated and control group data, respectively. Then we compute CATEs for subjects in the treated group using the observed outcome $y_i(1)$ and estimated potential outcome $\hat{\mu}_0(\mathbf{x}_i)$

$$\hat{\tau}_{1,i} = y_i(1) - \hat{\mu}_0(\mathbf{x}_i)$$

Notice that μ_0 is the model trained on control group data, and $\hat{\mu}_0(\mathbf{x}_i)$ is an extrapolation value with treated group data as the model input. Similarly, we can also compute imputed ITEs for control group subjects,

$$\hat{\tau}_{0,i} = \hat{\mu}_1(\mathbf{x}_i) - y_i(0)$$

After obtaining $\hat{\tau}_{1,i}, \hat{\tau}_{0,i}$, we treat them as the response and fit another two models $\hat{\tau}_1(\mathbf{x}), \hat{\tau}_0(\mathbf{x})$ using $\mathbf{x}_i$'s from treated and control group, respectively, as model inputs. Finally, we compute CATE for the i^{th} individual using a weighted sum of $\hat{\tau}_1(\mathbf{x}_i), \hat{\tau}_0(\mathbf{x}_i)$

$$\hat{\tau}(\mathbf{x}_i) = \hat{p}(\mathbf{x}_i) \times \hat{\tau}_0(\mathbf{x}_i) + [1 - \hat{p}(\mathbf{x}_i)] \times \hat{\tau}_1(\mathbf{x}_i)$$

where $\hat{p}(\mathbf{x}_i)$ is the estimated propensity score of the i^{th} subject.

Different models can be applied to estimate $\mu_0(\cdot)$, $\mu_1(\cdot)$, $\tau_0(\cdot)$, and $\tau_1(\cdot)$. Following Künzel et al., 2019 who choose Random Forest (RF) as the baseline and name their model X-RF, we choose another tree-based model, XGboost, to construct the X-learner and name it X-XGB.

4.2.3 SITE

Similar to the idea of TAR-Net, local Similarity preserved Individual Treatment Effect estimation (SITE) also starts by constructing a feed-forward neural network $f(\cdot)$ to map the pre-treatment covariate vector $\mathbf{X}$ into a representation vector $\mathbf{Z}$, i.e. $\mathbf{Z} = f(\mathbf{X})$. However, SITE focuses on the loss function instead of manipulating the neural network architecture to improve the model performance. To be more specific, the loss function consists of four parts:

$$L = L_{FL} + \beta L_{PDDM} + \gamma L_{MPDM} + \lambda ||W||_2$$

where L_{FL} is the factual loss between the estimated and observed factual outcomes; L_{PDDM} corresponds to the Position-Dependent Deep Metric (PDDM) metric (Huang et al., 2016), which is used to preserve the local similarities among the "hard cases" of $\mathbf{z}_i$'s to improve the training efficiency; L_{MPDM} plays a similar role with IPM (or MMD) and measure the distance between two distributions, which is also based on "hard cases" of $\mathbf{z}_i$'s.

4.3 Simulated Dataset

The 2016 Atlantic Causal Inference Challenge (ACIC2016) (Dorie et al., 2019) provides 77 different settings of benchmark datasets designed to test causal inference algorithms under a diverse range of real-world scenarios. The data in each setting are generated by a model with a combination of six model parameters:$1)$ **model.trt**; $2)$ **root.trt**; $3)$ **overlap.trt**; $4)$ **model.rsp**; $5)$ **alignment**; $6)$ **te.hetero**. Detailed descriptions of each parameter and the combination for each setting can be found in the Appendix $A.1$ of Dorie et al., 2019. Here, we only provide a brief explanation for each model parameter, which will help us better understand the model performance.

- **model.trt** determines the function used when building the treatment assignment mechanism $P(T = 1|\mathbf{X})$. This parameter can take three values: $linear, polynomial, step$, where *linear* implies that some pre-

dictors are added to the assignment mechanism model as linear terms with random coefficients; *polynomial* gives a chance to continuous predictors to be added in quadratic or cubic forms besides in the form of a "main effect"; *step* potentially adds "jumps" and "kinks" of the form $I(x_{(j)} \leq A)$ and $(x_{(j)} - B)I(x_{(k)} \leq C)$.

- **root.trt** is the baseline percentage of observations receiving the treatment taking two values 35% and 65%.

- **overlap.trt** describes the covariate space similarity between treated and control groups. It can take two values: *full* and *one-term*. When taking the value *one-term*, a penalty term $A \cdot I(x_{(j)} \geq B)$ is added to the treatment assignment function $P(T = 1|\mathbf{X})$, where A is a large negative value and B is a marginal quantile for the j^{th} covariate $x_{(j)}$. This term can forcibly assign some "corner" points in the covariate space to the control group and make it harder to find a similar subjects in the treatment group.

- **model.rsp** is similar to the parameter **model.trt** and determines the function used for $E[Y|\mathbf{X}, T]$. It can take four values: *linear*, *polynomial*, *exponential*, and *step*.

- **alignment** describes the degree of similarity between the generation function for $P(T = 1|\mathbf{X})$ and $E[Y|\mathbf{X}, T]$. It can take two values: 25% and 75%, where the value represents the percentage of a term in $P(T = 1|\mathbf{X})$ being copied to $E[Y|\mathbf{X}, T]$. This parameter reflects the dimension of the confounder space, the higher the percentage is, the lower dimension the confounder space has.

- **te.hetero** controls the treatment effect heterogeneity by specifying the number of terms of $\mathbf{X}$ interacting with the treatment. It takes four values: *none*, *low*, *med*, *high*, where *none* implies that treatment is a single, additive term in model, *low* implies that treatment is interacted with approximately three of the terms in the response model, and *high* yields around six interactions.

The dataset of each setting contains 4802 observations and 65 columns including treatment effect "z", observed outcome "y", factual (counterfactual) outcome "$y.0(y.1)$", ground truth factual (counterfactual) outcome "$mu.0(mu.1)$", propensity score "e", and 58 covariates. Datasets can be accessed at https://github.com/-vdorie/aciccomp/tree/master/2016.

For faster and more informative model performance comparisons, we do not exactly follow the evaluation procedure of Zhang et al., 2021 where 77 PEHEs are computed and averaged for each model. This is because each model can perform quite differently in different settings. Taking the average can miss some important information about the performance of a specific model. Instead, we choose 10 out of 77 settings, that is, 2, 7, 13, 14, 15, 19, 26, 37, 47, and 69. A more explicit parameter comparison for each setting is shown in Table 4.2, where we gather settings in different groups in which only one model parameter is changed. For example, the first group contains setting $13, 15, 69$ and only $model.trt$ takes different values $linear, polynomial, step$. By doing this, we can compare different models more explicitly and understand which model performs better (or worse) under what circumstances.

Table 4.2: Parameter comparison for ACIC dataset

setting	model.trt	root.trt	overlap.trt	model.rsp	alignment	te.hetero
13	linear	0.65	one-term	exponential	0.75	high
15	polynomial	0.65	one-term	exponential	0.75	high
69	step	0.65	one-term	exponential	0.75	high
7	polynomial	0.35	one-term	exponential	0.75	high
15	polynomial	0.65	one-term	exponential	0.75	high
15	polynomial	0.65	one-term	exponential	0.75	high
47	polynomial	0.65	full	exponential	0.75	high
14	polynomial	0.65	one-term	linear	0.75	high
15	polynomial	0.65	one-term	exponential	0.75	high
37	polynomial	0.65	one-term	step	0.75	high
14	polynomial	0.65	one-term	linear	0.75	high
19	polynomial	0.65	one-term	linear	0.25	high
2	polynomial	0.35	one-term	exponential	0.75	
7	polynomial	0.35	one-term	exponential	0.75	high
26	polynomial	0.35	one-term	exponential	0.75	med

Following the work of Zhang et al., 2021, we train our ICEDVAE model on 10 repetitions under each setting and compute the mean of the corresponding 10 PEHEs and their standard error. We also compare our model with greedy match, MSM, X-learner, CEVAE, TEDVAE, and SITE. The comparison details are presented in Figures 4.1-4.6. We demonstrate the result with a vertical box containing three horizontal bars for each method: the middle bars reflect the mean PEHE of 10 replications. The upper and lower bars represent 1 standard deviation away from the mean value. Comparing all plots in Figures 4.1-4.6, we find that X-learner usually performs the best and has the lowest mean PEHE.

CFR and DR-CFR have similar performances over all settings. Although MSM is relatively the simplest model, it still has a comparable performance with all other sophisticated deep learning models, especially with SITE, and has lower PEHE in settings 14, 19, 26, and 69. Among all VAE models, TEDVAE usually has the best performance. However, in setting 2 with no treatment heterogeneity, our proposed model ICEDVAE has the lowest PEHE and beats all other models. This is not surprising since we made "treatment homogeneity" assumptions in Equations **??**-**??** and the construction of importance sampling weights, i.e. $P(y|\mathbf{z}, t) \approx P(y|\mathbf{z})$. To further justify this finding, we also compare all models on setting 3, another setting with no treatment heterogeneity. The result is shown in Figure 4.7, from which we can find that ICEDVAE indeed performs relatively well when there is no treatment heterogeneity. The hardest setting for all models is setting 7, where all models have relatively high PEHE. This makes sense due to the non-linear generation mechanism for both $P(T = 1|\mathbf{X})$ and $E[Y|\mathbf{X}, T]$, and the low overlap of covariate distributions from different treatment groups.

Figure 4.1: ACIC setting: 13, 15, 69

4.4 Semi Real-world Dataset

The Infant Health and Development Program (IHDP) dataset is a randomized controlled study designed to evaluate the home visit effects of specialist doctors on cognitive test scores of premature infants. The dataset is first used for benchmarking treatment effect estimation algorithms in Hill, 2011. This dataset can be accessed at https://github.com/WeijiaZhang24/TEDVAE. There are two settings in this dataset, named "Setting A" and "Setting B," where the outcomes follow a linear relationship with covariates in "Setting A" and an exponential

Figure 4.2: ACIC setting: 7, 15

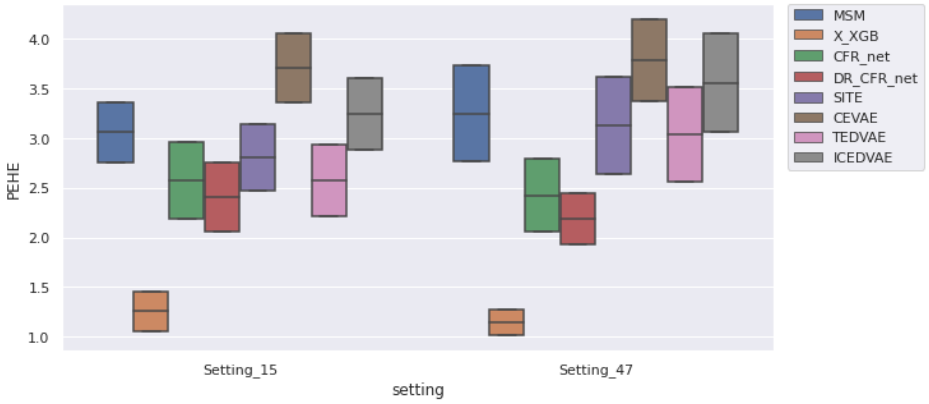

Figure 4.3: ACIC setting: 15, 47

relationship in "Setting B." To mimic the characteristics of observational data, Hill et al. (Hill, 2011) reconstructed the data by throwing away a non-random portion of the treatment group while keeping the control group unchanged. The final dataset contains 747 subjects and 25 variables (6 continuous variables and 19 binary variables) that describe both the features of the infants and their mothers. The outcomes, i.e., the cognitive test scores, are the only simulated part of this dataset. The 25 confounding variables are used to generate two different outcomes:

- In "setting A", the outcome follows the distribution

$$Y(0) \sim \mathcal{N}(\mathbf{X}\beta_A, 1), \quad Y(1) \sim \mathcal{N}(\mathbf{X}\beta_A + 4, 1) \tag{4.3}$$

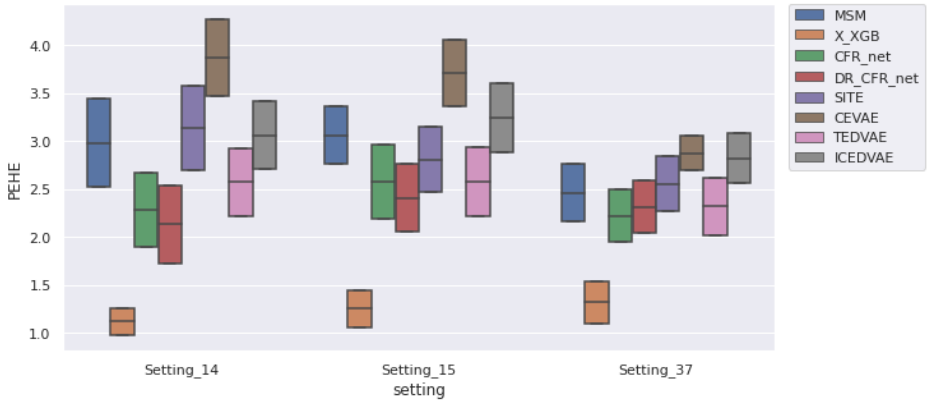

Figure 4.4: ACIC setting: 14, 15, 37

Figure 4.5: ACIC setting: 14, 19

where $\mathbf{X}$ represents the covariate matrix. The coefficients in β_A are randomly sampled values $(0, 1, 2, 3, 4)$ with probabilities (0.5, 0.2, 0.15, 0.1, 0.05), which make smaller coefficients more likely.

- In "setting B", the outcome has the distribution

$$Y(0) \sim \mathcal{N}(\exp((\mathbf{X} + \mathbf{W})\beta_B), 1), \quad Y(1) \sim \mathcal{N}(\mathbf{X}\beta_B - \omega_B^s, 1) \tag{4.4}$$

where $\mathbf{W}$ is an offset matrix of the same dimension as $\mathbf{X}$ with every value equal to 0.5; the coefficients in β_B are randomly samples values from $(0, 0.1, 0.2, 0.3, 0.4)$ with probabilities $(0.6, 0.1, 0.1, 0.1, 0.1)$. For the s^{th} simulation, ω_B^s was chosen such that the estimated Causal Average

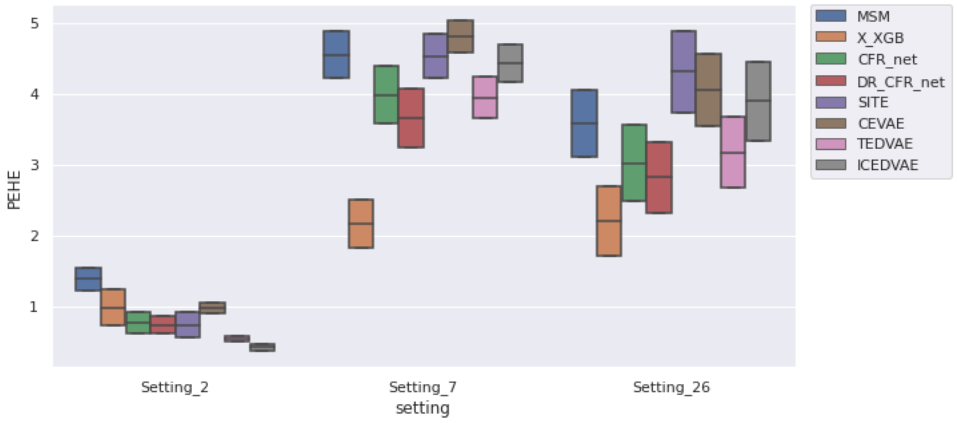

Figure 4.6: ACIC setting: 2, 7, 26

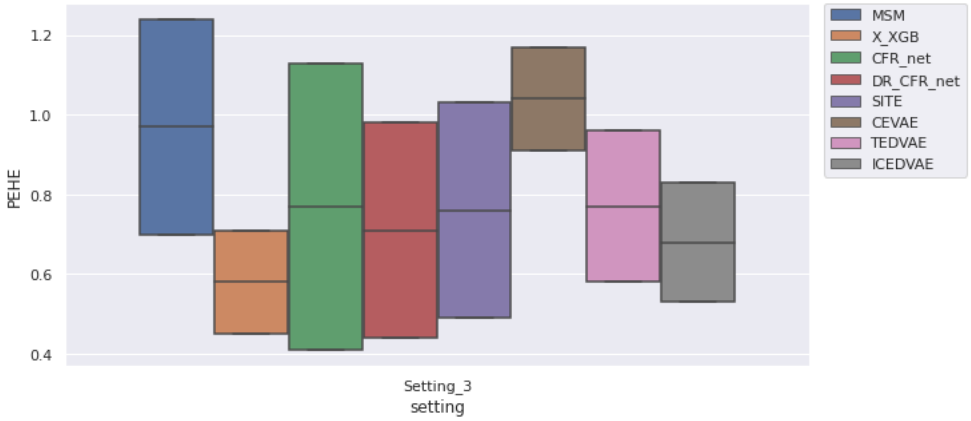

Figure 4.7: ACIC setting 3

effect of the Treatment on the Treated (CATT) and estimated Causal Average effect of the Treatment on the Controls (CATC) equal 4.

We average the reported performances over 100 replications with a training/test split proportions of 90%/10% shown in Table 4.3, where PEHE is used as the evaluation metric since the outcome is continuous. Our ICEDVAE model outperforms other methods in "setting A," where the relationship between the outcome and covariates is linear. We also observe that our model has the lowest increasing PEHE from training to test data set under both settings, which provides evidence for the effectiveness of the importance sampling weights in alleviating the over-fitting problem. However, ICEDVAE does not perform quite well under "setting B," which is not a surprise since the relations between the outcome and covariates in "setting B" are more complex than in "setting A."

Also, the assumption $P(y|\mathbf{z}, t) \approx P(y|\mathbf{z})$ is not as applicable as in "setting A" where the treatment effect is more homogeneous than "setting B."

Table 4.3: Evaluation result on IHDP dataset

	Setting A		Setting B	
Methods	PEHE_train	PEHE_test	PEHE_train	PEHE_test
MSM	1.1 ± 0.07	1.12 ± 0.07	2.93 ± 0.05	3.03 ± 0.06
X-XGB	1.11 ± 0.08	1.15 ± 0.09	2.68 ± 0.03	2.90 ± 0.04
CFR	0.71 ± 0.03	0.83 ± 0.08	2.63 ± 0.04	2.75 ± 0.04
DR-CFR	0.90 ± 0.05	1.12 ± 0.11	3.05 ± 0.04	3.21 ± 0.05
SITE	0.61 ± 0.08	0.65 ± 0.07	2.79 ± 0.13	2.96 ± 0.15
CEVAE	1.15 ± 0.08	1.20 ± 0.09	3.19 ± 0.04	3.35 ± 0.06
TEDVAE	0.65 ± 0.04	0.66 ± 0.05	2.15 ± 0.02	2.26 ± 0.03
ICEDVAE	0.55 ± 0.04	0.55 ± 0.05	3.45 ± 0.05	3.47 ± 0.05

4.5 Real-world Dataset

The Twins dataset comes from all twins birth in the USA between 1989 and 1991 (Almond et al., 2005). We follow the work of Yao et al., 2018 and focus on the same sex twin-pairs whose weights are less than 2000g and remove some covariates, reducing the number of covariates from 52 to 40. More details can be found at https://github.com/Osier-Yi/SITE/issues/3. As a result, each record contains 40 pre-treatment covariates related to the parents, the pregnancy, and the birth. The treatment $T = 1$ is viewed as heavier in the twins, and $T = 0$ is lighter. The outcome is binary and represents the infant's survival status after one year. After removing the records with missing features and weights greater than 2000g, the final dataset contains 5409 records. In this setting, both factual and counterfactual outcomes can be observed. To generate selection bias, we follow Yao et al., 2018 and use the following distributions to selectively choose one of the twins as the observation and hide the other: $T_i|\mathbf{x}_i \sim Bern(Sigmoid(w^T\mathbf{x}_i + \epsilon))$, where $w \sim U((-0.1, 0.1)^{40\times 1})$ and $\epsilon \sim N(0, 0.1)$.

The results of model comparisons are shown in Table 4.4 where we use ROC_AUC as the evaluation metric since the outcome variable is binary. We can see that our model still performs better than most other compared methods. Although X-XGB has the best performance in the training data set, the AUC decreases significantly in the test set, exposing the potential over-fitting prob-

lem. Our ICEDVAE model performs slightly worse than TEDVAE, which is reasonable since ICEDVAE is a simplified version of TEDVAE.

Table 4.4: Evaluation result on Twins dataset

	ROC_AUC^{train}	ROC_AUC^{test}
MSM	0.86 ± 0.002	0.84 ± 0.009
X-XGB	0.90 ± 0.002	0.86 ± 0.006
CFR	0.83 ± 0.033	0.81 ± 0.030
DR-CFR	0.86 ± 0.003	0.84 ± 0.009
CEVAE	0.85 ± 0.008	0.84 ± 0.009
TEDVAE	0.90 ± 0.003	0.90 ± 0.003
ICEDVAE	0.88 ± 0.002	0.88 ± 0.002

4.6 Summary

Although exposing the over-fitting problem, X-learner usually has decent performance and even outperforms other methods on the synthetic ACIC dataset. As the only statistical method, MSM is still comparable with sophisticated deep learning models on synthetic datasets but not good enough on more complex, real-world cases. Although VAE models have similar performance, TEDVAE usually performs slightly better than CEVAE and ICEDVAE. However, our proposed model ICEDVAE outperforms all other models when the data contains no treatment heterogeneity, i.e., the treatment effect is the same for all subjects. Moreover, ICEDVAE is also more robust than other models since it has the lowest increasing PEHE/ROC_AUC from training to test dataset.

Chapter 5

Conclusion

The causal inference has been increasingly drawing attention during the past decades, especially in this pandemic when people care more about effective protection approaches against COVID-19. Although researchers have developed plenty of classical statistical and machine learning methods to estimate causal effects, they still face the Big-Data challenges from modern technologies. In this thesis, we try to utilize the deep learning architecture, a powerful tool to extract valuable information from large amounts of data, to improve the accuracy of causal effect estimations.

In Chapter 1, we discuss classical methods to estimate causal effects and elaborate on the challenges they faced, including the curse of dimensionality and the controversy of the Ignorability Assumption in the potential outcome framework. To overcome these two obstacles, we use Causal Bayesian Network as the paradigm to identify hidden confounders and apply the Variational Auto-Encoder to estimate the causal effect with high-dimensional input data. To further simplify the model and avoid the over-fitting problem, we propose our ICEDVAE in Chapter 3 and compute the importance sampling weight with existing model components to replace some hyper-parameters. Numerical studies are presented in Chapter 4 and we find that ICEDVAE is more robust and better at estimating homogeneous treatment effect than other SOTA methods.

Our proposed model still has space to improve. For example, we only tried the importance sampling weights to penalize the term for outcome prediction in the final objective function. We choose this weight because its effectiveness is theoretically guaranteed and can be easily computed with existing components of our model, which aligns with our goal of simplifying the model and improving the model's robustness. Nevertheless, other weights might exist and

be more suitable for estimating both homogeneous and heterogeneous treatment effects. One thought is to relax the "homogeneity assumptions" to make our model better at estimating heterogeneous treatment effects. However, this relaxation will inevitably introduce extra components to our model, which increase the model complexity and thus the risk of over-fitting. This is a classical trade-off scenario, and we will leave this as our future work.

There is also still much to improve in causal inference. From a micro-level where algorithmic and computational methods are focused on, precisely estimating the propensity score is critical for most potential outcome-based methods. Inaccurate estimation of propensity scores can cause bias, which is difficult to detect since it is usually separated from causal effect estimation, such as matching, stratification, weighting, etc. Statistical methods also suffer from the curse of dimensionality when they have high-dimensional data as the input to estimate the propensity score. Existing methods for estimating propensity scores are mainly based on regression methods, which require users to include and exclude variables manually. Variable selection methods can be applied, but the prediction accuracy cannot be guaranteed. As a result, users include all covariates to fit the regression model in most cases and hope for the best. Inspired by our work in this thesis, we think that decomposing the covariates is an alternative way that deserves exploration for estimating the propensity score, and classic statistical models can then be built upon it.

On the macro level, perhaps one of the most critical issues is code sharing in statistician communities. Unlike the machine learning communities where open-source codes are prevalent on GitHub, codes for newly proposed statistical methods are hard to find and hinder other researchers from reproducing the results. Another issue is the consistency of data used to evaluate the model performance. Although standard data sets are available online, authors of different models have different data pre-processing procedures, leading to different training and test data sets to evaluate the model performance. To solve this issue, authors can upload codes of their proposed models and the processed data set on GitHub such that other researchers can re-use them and make model comparisons more convincing.

Another area that deserves attention is making causal inference methods more attractive to a broader audience, especially researchers who are not familiar with causal inference concepts. Although this COVID-19 pandemic speeds up this process somehow, it is "unethical" to expect this to happen frequently. Most

causal methods have definitions and assumptions that are hard to follow for scientists of other fields, and they are mainly expressed as unintuitive notations and formulas. More causal inference concepts should be presented with graphs following their intuition, enabling more researchers to understand and further develop causal discovery methods in various fields.

Bibliography

Abadie, A., Drukker, D., Herr, J. L., & Imbens, G. W. (2004). Implementing matching estimators for average treatment effects in stata. *The stata journal*, *4*(3), 290–311.

Abadie, A., & Imbens, G. W. (2006). Large sample properties of matching estimators for average treatment effects. *econometrica*, *74*(1), 235–267.

Almond, D., Chay, K. Y., & Lee, D. S. (2005). The costs of low birth weight. *The Quarterly Journal of Economics*, *120*(3), 1031–1083.

An, J., & Cho, S. (2015). Variational autoencoder based anomaly detection using reconstruction probability. *Special Lecture on IE*, *2*(1), 1–18.

Athey, S., & Imbens, G. (2016). Recursive partitioning for heterogeneous causal effects. *Proceedings of the National Academy of Sciences*, *113*(27), 7353–7360.

Austin, P. C. (2011). An introduction to propensity score methods for reducing the effects of confounding in observational studies. *Multivariate behavioral research*, *46*(3), 399–424.

Baiocchi, M., Cheng, J., & Small, D. S. (2014). Instrumental variable methods for causal inference. *Statistics in medicine*, *33*(13), 2297–2340.

Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, *35*(8), 1798–1828.

Bengio, Y., Yao, L., Alain, G., & Vincent, P. (2013). Generalized denoising auto-encoders as generative models. *Advances in neural information processing systems*, *26*.

Bishop, C. M., & Nasrabadi, N. M. (2006). *Pattern recognition and machine learning* (Vol. 4). Springer.

Blei, D. M., Kucukelbir, A., & McAuliffe, J. D. (2017). Variational inference: A review for statisticians. *Journal of the American statistical Association*, *112*(518), 859–877.

Bloniarz, A., Liu, H., Zhang, C.-H., Sekhon, J. S., & Yu, B. (2016). Lasso adjustments of treatment effect estimates in randomized experiments. *Proceedings of the National Academy of Sciences*, *113*(27), 7383–7390.

Blyth, C. R. (1972). On simpson's paradox and the sure-thing principle. *Journal of the American Statistical Association*, *67*(338), 364–366.

Caliendo, M., & Kopeinig, S. (2008). Some practical guidance for the implementation of propensity score matching. *Journal of economic surveys*, *22*(1), 31–72.

Chang, Y., & Dy, J. (2017). Informative subspace learning for counterfactual inference. *Proceedings of the AAAI Conference on Artificial Intelligence*, *31*(1).

Chipman, H., George, E., & McCulloch, R. (2006). Bayesian ensemble learning. *Advances in neural information processing systems*, *19*.

Chipman, H. A., George, E. I., & McCulloch, R. E. (2010). Bart: Bayesian additive regression trees. *The Annals of Applied Statistics*, *4*(1), 266–298.

Chu, Z., Rathbun, S. L., & Li, S. (2020). Matching in selective and balanced representation space for treatment effects estimation. *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 205–214.

Chung, J., Kastner, K., Dinh, L., Goel, K., Courville, A. C., & Bengio, Y. (2015). A recurrent latent variable model for sequential data. *Advances in neural information processing systems*, *28*.

Dorie, V., Hill, J., Shalit, U., Scott, M., & Cervone, D. (2019). Automated versus do-it-yourself methods for causal inference: Lessons learned from a data analysis competition. *Statistical Science*, *34*(1), 43–68.

Ellis, B., & Wong, W. H. (2008). Learning causal bayesian network structures from experimental data. *Journal of the American Statistical Association*, *103*(482), 778–789.

EugenioTL. (2021). The basic scheme of a variational auto-encoder.

Fong, C., Hazlett, C., & Imai, K. (2018). Covariate balancing propensity score for a continuous treatment: Application to the efficacy of political advertisements. *The Annals of Applied Statistics*, *12*(1), 156–177.

Good, I. J., & Mittal, Y. (1987). The amalgamation and geometry of two-by-two contingency tables. *The Annals of Statistics*, 694–711.

Gregor, K., Danihelka, I., Graves, A., Rezende, D., & Wierstra, D. (2015). Draw: A recurrent neural network for image generation. *International Conference on Machine Learning*, 1462–1471.

Guo, R., Li, J., & Liu, H. (2020). Learning individual causal effects from networked observational data. *Proceedings of the 13th International Conference on Web Search and Data Mining*, 232–240.

Häggström, J. (2018). Data-driven confounder selection via markov and bayesian networks. *Biometrics*, *74*(2), 389–398.

Hahn, J. (1998). On the role of the propensity score in efficient semiparametric estimation of average treatment effects. *Econometrica*, 315–331.

Hansen, B. B. (2007). Optmatch: Flexible, optimal matching for observational studies. *New Functions for Multivariate Analysis*, *7*(2), 18–24.

Hassanpour, N., & Greiner, R. (2019). Counterfactual regression with importance sampling weights. *IJCAI*, 5880–5887.

Heckerman, D., Geiger, D., & Chickering, D. M. (1995). Learning bayesian networks: The combination of knowledge and statistical data. *Machine learning*, *20*(3), 197–243.

Hill, J. L. (2011). Bayesian nonparametric modeling for causal inference. *Journal of Computational and Graphical Statistics*, *20*(1), 217–240.

Hinton, G. E., & Zemel, R. (1993). Autoencoders, minimum description length and helmholtz free energy. *Advances in neural information processing systems*, *6*.

Huang, C., Loy, C. C., & Tang, X. (2016). Local similarity-aware deep feature embedding. *Advances in neural information processing systems*, *29*, 1262–1270.

Iacus, S. M., King, G., & Porro, G. (2012). Causal inference without balance checking: Coarsened exact matching. *Political analysis*, 1–24.

Imai, K., & Ratkovic, M. (2014). Covariate balancing propensity score. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *76*(1), 243–263.

Imbens, G. W. (2004). Nonparametric estimation of average treatment effects under exogeneity: A review. *Review of Economics and statistics*, *86*(1), 4–29.

Imbens, G. W., & Rubin, D. B. (2015). *Causal inference in statistics, social, and biomedical sciences*. Cambridge University Press.

Johansson, F., Shalit, U., & Sontag, D. (2016). Learning representations for counterfactual inference. *International conference on machine learning*, 3020–3029.

Johansson, F. D., Kallus, N., Shalit, U., & Sontag, D. (2018). Learning weighted representations for generalization across designs. *arXiv preprint arXiv:1802.08598*.

Jordan, M. I. (2004). Graphical models. *Statistical science*, *19*(1), 140–155.

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., & Saul, L. K. (1999). An introduction to variational methods for graphical models. *Machine learning*, *37*(2), 183–233.

Kingma, D. P., & Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Kipf, T. N., & Welling, M. (2016). Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Kloek, T., & Van Dijk, H. K. (1978). Bayesian estimates of equation system parameters: An application of integration by monte carlo. *Econometrica: Journal of the Econometric Society*, 1–19.

Kramer, M. A. (1991). Nonlinear principal component analysis using autoassociative neural networks. *AIChE journal*, *37*(2), 233–243.

Kuang, K., Cui, P., Li, B., Jiang, M., Yang, S., & Wang, F. (2017). Treatment effect estimation with data-driven variable decomposition. *Proceedings of the AAAI Conference on Artificial Intelligence*, *31*(1).

Kullback, S. (1997). *Information theory and statistics*. Courier Corporation.

Kullback, S., & Leibler, R. A. (1951). On information and sufficiency. *The annals of mathematical statistics*, *22*(1), 79–86.

Künzel, S. R., Sekhon, J. S., Bickel, P. J., & Yu, B. (2019). Metalearners for estimating heterogeneous treatment effects using machine learning. *Proceedings of the national academy of sciences*, *116*(10), 4156–4165.

Lauritzen, S. L., Dawid, A. P., Larsen, B. N., & Leimer, H.-G. (1990). Independence properties of directed markov fields. *Networks*, *20*(5), 491–505.

Lee, B. K., Lessler, J., & Stuart, E. A. (2011). Weight trimming and propensity score weighting. *PloS one*, *6*(3), e18174.

Li, F., Morgan, K. L., & Zaslavsky, A. M. (2018). Balancing covariates via propensity score weighting. *Journal of the American Statistical Association*, *113*(521), 390–400.

Liu, M.-Y., Breuel, T., & Kautz, J. (2017). Unsupervised image-to-image translation networks. *Advances in neural information processing systems*, *30*.

Loh, W.-Y. (2011). Classification and regression trees. *Wiley interdisciplinary reviews: data mining and knowledge discovery*, *1*(1), 14–23.

Louizos, C., Shalit, U., Mooij, J., Sontag, D., Zemel, R., & Welling, M. (2017). Causal effect inference with deep latent-variable models. *arXiv preprint arXiv:1705.08821*.

Louizos, C., Swersky, K., Li, Y., Welling, M., & Zemel, R. (2015). The variational fair autoencoder. *arXiv preprint arXiv:1511.00830*.

Ma, X., & Wang, J. (2020). Robust inference using inverse probability weighting. *Journal of the American Statistical Association*, *115*(532), 1851–1860.

McGough, J. J., & Faraone, S. V. (2009). Estimating the size of treatment effects: Moving beyond p values. *Psychiatry (Edgmont)*, *6*(10), 21.

Müller, A. (1997). Integral probability metrics and their generating classes of functions. *Advances in Applied Probability*, *29*(2), 429–443.

Paul, S. (2020).
reparameterization" trick in variational autoencoders.

Pearl, J. et al. (2000). Models, reasoning and inference. *Cambridge, UK: CambridgeUniversityPress*, *19*, 2.

Pearl, J. et al. (2009). Causal inference in statistics: An overview. *Statistics surveys*, *3*, 96–146.

Pearl, J. (2012). On a class of bias-amplifying variables that endanger effect estimates. *arXiv preprint arXiv:1203.3503*.

Pearl, J., & Mackenzie, D. (2018). *The book of why: The new science of cause and effect*. Basic books.

Pimentel, S. D. (2016). Large, sparse optimal matching with r package rebalance. *Observational Studies*, *2*(1), 4–23.

Polykovskiy, D. (2019). Why approximate inference.

Rezende, D. J., Mohamed, S., & Wierstra, D. (2014). Stochastic backpropagation and approximate inference in deep generative models. *International conference on machine learning*, 1278–1286.

Richard, B. (1961). Adaptive control processes: A guided tour. *Princeton, New Jersey, USA*.

Rifai, S., Vincent, P., Muller, X., Glorot, X., & Bengio, Y. (2011). Contractive auto-encoders: Explicit invariance during feature extraction. *Icml*.

Robins, J. M., Hernan, M. A., & Brumback, B. (2000). Marginal structural models and causal inference in epidemiology.

Robins, J. M., Rotnitzky, A., & Zhao, L. P. (1994). Estimation of regression coefficients when some regressors are not always observed. *Journal of the American statistical Association*, *89*(427), 846–866.

Rosenbaum, P. R. (1987). Model-based direct adjustment. *Journal of the American statistical Association*, *82*(398), 387–394.

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

Rosenbaum, P. R., & Rubin, D. B. (1984). Reducing bias in observational studies using subclassification on the propensity score. *Journal of the American statistical Association*, *79*(387), 516–524.

Rubin, D. B. (1973). Matching to remove bias in observational studies. *Biometrics*, 159–183.

Rubin, D. B. (1974). Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of educational Psychology*, *66*(5), 688.

Rubin, D. B. (2005). Causal inference using potential outcomes: Design, modeling, decisions. *Journal of the American Statistical Association*, *100*(469), 322–331.

Rubin, D. B. (2007). The design versus the analysis of observational studies for causal effects: Parallels with the design of randomized trials. *Statistics in medicine*, *26*(1), 20–36.

Rubin, D. B., & Thomas, N. (2000). Combining propensity score matching with additional adjustments for prognostic covariates. *Journal of the American Statistical Association*, *95*(450), 573–585.

Sauer, B. C., Brookhart, M. A., Roy, J., & VanderWeele, T. (2013). A review of covariate selection for non-experimental comparative effectiveness research. *Pharmacoepidemiology and drug safety*, *22*(11), 1139–1145.

Schwab, P., Linhardt, L., Bauer, S., Buhmann, J. M., & Karlen, W. (2020). Learning counterfactual representations for estimating individual dose-response curves. *Proceedings of the AAAI Conference on Artificial Intelligence*, *34*(04), 5612–5619.

Schwab, P., Linhardt, L., & Karlen, W. (2018). Perfect match: A simple method for learning representations for counterfactual inference with neural networks. *arXiv preprint arXiv:1810.00656*.

Shalit, U., Johansson, F. D., & Sontag, D. (2017). Estimating individual treatment effect: Generalization bounds and algorithms. *International Conference on Machine Learning*, 3076–3085.

Shalizi, C. R., & Thomas, A. C. (2011). Homophily and contagion are generically confounded in observational social network studies. *Sociological methods & research*, *40*(2), 211–239.

Sriperumbudur, B. K., Fukumizu, K., Gretton, A., Schölkopf, B., Lanckriet, G. R., et al. (2012). On the empirical estimation of integral probability metrics. *Electronic Journal of Statistics*, *6*, 1550–1599.

Stuart, E. A. (2010). Matching methods for causal inference: A review and a look forward. *Statistical science: a review journal of the Institute of Mathematical Statistics*, *25*(1), 1.

Stuart, E. A., King, G., Imai, K., & Ho, D. (2011). Matchit: Nonparametric preprocessing for parametric causal inference. *Journal of statistical software*.

Van Dijk, H. K., & Kloek, T. (1983). *Experiments with some alternatives for simple importance sampling in monte carlo integration* (tech. rep.).

Wager, S., & Athey, S. (2018). Estimation and inference of heterogeneous treatment effects using random forests. *Journal of the American Statistical Association*, *113*(523), 1228–1242.

Wainwright, M. J., Jordan, M. I. et al. (2008). Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, *1*(1–2), 1–305.

Wang, P., Sun, W., Yin, D., Yang, J., & Chang, Y. (2015). Robust tree-based causal inference for complex ad effectiveness analysis. *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, 67–76.

Wooldridge, J. (2009). *Should instrumental variables be used as matching variables* (tech. rep.). Citeseer.

Yan, X., Yang, J., Sohn, K., & Lee, H. (2016). Attribute2image: Conditional image generation from visual attributes. *European conference on computer vision*, 776–791.

Yao, L., Li, S., Li, Y., Huai, M., Gao, J., & Zhang, A. (2018). Representation learning for treatment effect estimation from observational data. *Advances in Neural Information Processing Systems*, *31*.

Yao, L., Li, S., Li, Y., Huai, M., Gao, J., & Zhang, A. (2019). Ace: Adaptively similarity-preserved representation learning for individual treatment effect estimation. *2019 IEEE International Conference on Data Mining (ICDM)*, 1432–1437.

Zhang, W., Liu, L., & Li, J. (2021). Treatment effect estimation with disentangled latent factors. *Proceedings of the AAAI Conference on Artificial Intelligence*, *35*(12), 10923–10930.

Zhao, Q. et al. (2019). Covariate balancing propensity score by tailored loss functions. *Annals of Statistics*, *47*(2), 965–993.

Zubizarreta, J. R. (2015). Stable weights that balance covariates for estimation with incomplete outcome data. *Journal of the American Statistical Association*, *110*(511), 910–922.