Counterfactual Reasoning in Observational Studies

by

Negar Hassanpour

A thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Department of Computing Science

University of Alberta

Abstract

As one of the main tasks in studying causality, the goal of Causal Inference is to determine "whether" (and perhaps "how much") the value of a certain variable (*i.e.*, the **effect**) would change, had another specified variable (*i.e.*, the cause) changed its value. A prominent example is the counterfactual question "Would this patient have **lived longer** had she received an alternative treatment?".

The first challenge with causal inference is the *unobservability* of the counterfactual outcomes — *i.e.*, outcomes obtained by applying the treatments that were *not* administered. The second common challenge is that the training data is often an observational study that exhibits *selection bias* — *i.e.*, the treatment assignment can depend on the subjects' attributes.

In this dissertation, I have explored ways to address the above-mentioned challenges. Specifically, my Research Contributions (RCs) are the following: My first RC addresses the first challenge:

1. Unobservable counterfactuals prohibit proper evaluation of different methods' performance in estimating treatment effects. We provide an algorithm that can synthesize realistic observational datasets that exhibit various degrees of selection bias, then demonstrate that it can effectively assess various contextual bandit methods in the literature.

The remaining RCs are related to the second challenge:

2. Learning a common representation space that makes the transformed dataset close to a Randomized Controlled Trial (RCT), is a good strategy to *reduce*

selection bias. We devise a method that further alleviates selection bias (attempting to *account for* it) by incorporating appropriate re-weighting schemes and show that it outperforms its competitors in the literature.

3. Without loss of generality, we assume that three non-noise underlying factors generate any observational data. We devise a method that explicitly models these sources and argue that such model can better deal with selection bias. We then demonstrate its superior performance compared to the competing causal inference methods in the literature.

4. The majority of current causal effect estimation methods fall under the category of *discriminative* approaches. A promising direction is to consider developing *generative* models, in an attempt to shed light on the true underlying data generating mechanism, which in turn is useful for the downstream task of counterfactual regression. We develop such a method and show empirically that it significantly outperforms state-of-the-art.

Preface

The following is the list of papers related to this dissertation that I have (co-)authored:

Principal author

1. **Hassanpour, N.**, Greiner, R., "Variational Auto-Encoder Architectures that Excel at Causal Inference", *NeurIPS Workshop on Causal Discovery and Causality-Inspired Machine Learning*, December 11, 2020. [34]

2. **Hassanpour, N.**, Greiner, R., "Learning Disentangled Representations for CounterFactual Regression", *International Conference on Learning Representations (ICLR)*, April 27-30, 2020. [33]

3. **Hassanpour, N.**, Greiner, R., "CounterFactual Regression with Importance Sampling Weights", *The 28th International Joint Conference on Artificial Intelligence (IJCAI)*, August 10-16, 2019, Macao, China. [32]

4. **Hassanpour, N.**, "Counterfactual Reasoning in Observational Studies", *The 24th AAAI/SIGAI Doctoral Consortium*, January 27 - February 1, 2019, Honolulu, Hawaii, USA. [30]

5. **Hassanpour, N.**, Greiner, R., "A Novel Evaluation Methodology for Assessing Off-Policy Learning Methods in Contextual Bandits", *The 31st Canadian Conference on Artificial Intelligence*, May 8-11, 2018, Toronto, Canada, pp. 31-44. [31]

Co-author

6. Zhang, Z., Lan, Q., Ding, L., Wang, Y., **Hassanpour, N.**, Greiner, R., "Reducing Selection Bias in Counterfactual Reasoning for Individual Treatment Effects Estimation", *NeurIPS Workshop "Do the Right Thing": Machine Learning and Causal Inference for Improved Decision Making*, December 14, 2019, Vancouver, Canada. [120]

7. Wen, J., **Hassanpour, N.**, Greiner, R., "Weighted Gaussian Process for Estimating Treatment Effect", *NeurIPS Workshop: What If? Inference and Learning of Hypothetical and Counterfactual Interventions in Complex Systems*, December 10th, 2016, Barcelona, Spain. [112]

To Samad

For his unconditional love and support.

No man ever steps in the same river twice,
for it's not the same river and he's not the same man.

– Heraclitus

Acknowledgements

Throughout my PhD studies, I have received a great deal of support and assistance that I'm grateful for.

I would like to give a special thank you to my supervisor Professor Russell Greiner for his unwavering support and guidance. I couldn't have asked for a more supportive mentor.

I would also like to express my sincere gratitude to the University of Alberta for University of Alberta President's Doctoral Prize of Distinction, the Natural Sciences and Engineering Research Council of Canada for Postgraduate Doctoral Scholarship, and Alberta Machine Intelligence Institute for their continued support, financially and otherwise.

Last but not least, from the bottom of my heart, I would like to thank my husband Samad Kardan for all his unconditional support in this very intense academic endeavor.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

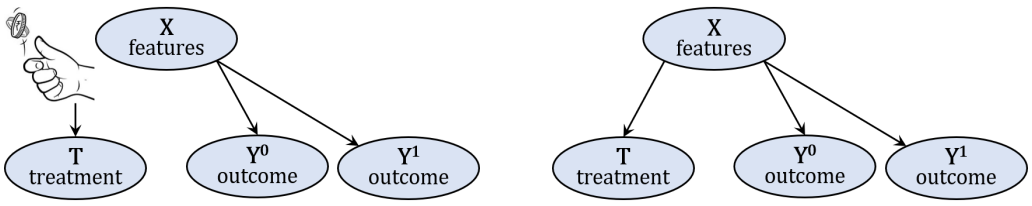
As we rely more and more on Artificial Intelligence (AI) to automate the decision making processes, accurately estimating the effects of taking different actions gains an essential role. A prominent example is precision medicine — *i.e.*, the customization of health-care tailored to each individual patient. In precision medicine, the goal is to identify which medical procedure t (aka *treatment* or *action*) will benefit a certain patient x the most, in terms of a certain health outcome $y \in \mathbb{R}$. Learning such models requires answering counterfactual questions [79], [90] such as: *"Would this patient have lived longer [and by how much], had she received an alternative treatment?"*.

This type of counterfactual analysis is not limited to the health-care domain. It can be of interest in any field where personalized action selection is of value; including: econometrics [4], intelligent tutoring systems [66], [87], recommender systems [92], news article recommender systems [61], ad-placement systems [10], and webpage recommendation by search engines [60].

For notation: A dataset

$$\mathcal{D} = \{ \ [x_i, \, t_i, \, y_i] \ \}_{i=1}^{N} \tag{1.1}$$

used for treatment effect estimation has the following format: for the i^{th} instance (*e.g.*, a patient), we have some context information $x_i \in \mathcal{X} \subseteq \mathbb{R}^K$ (*e.g.*, that patient's age, BMI, blood work, etc.), the administered treatment t_i chosen from a set of treatment options $\mathcal{T}$ (*e.g.*, $\{0\colon \text{medication}, 1\colon \text{surgery}\}$), and the associated observed outcome $y_i \in \mathcal{Y}$ (*e.g.*, survival time: $\mathcal{Y} \subseteq \mathbb{R}^+$) as a



(a) Randomized Controlled Trial (b) Observational Study

Figure 1.1: Belief net structure for (a) randomized controlled trials and (b) observational studies (of course, there can also be stochasticity here by having a noise variable pointing to T). Here, Y^0 (Y^1) is the outcome of applying $T=$ treatment#0 (#1) to the individual represented by X.

result of receiving treatment t_i. Note that $\mathcal{D}$ only contains the outcome of the administered treatment (aka *observed* outcome: y_i), but not the outcome(s) of the alternative treatment(s) (aka *counterfactual* outcome(s); that is, y_i^t for $t \in \mathcal{T} \setminus \{t_i\}$), which are inherently unobservable [41]. For the binary-treatment case $t \in \{0, 1\}$, we denote the alternative treatment as $\neg t_i = 1 - t_i$.

Pearl [79] demonstrates that, in general, causal relationships can only be learned by experimentation (on-line exploration), or running a Randomized Controlled Trial (RCT), where the treatment assignment T does not depend on the individual X — see Figure 1.1a. In many cases, however, collecting RCT data is expensive, unethical, or even infeasible.

A possible approach is to approximate treatment effects from off-line datasets collected through Observational Studies. In such datasets, however, the administered treatment T can depend on some attributes of individual X — see Figure 1.1b. Here, as $\Pr(T \mid X) \neq \Pr(T)$, we say these datasets exhibit *selection bias* [46]. Figure 1.2 illustrates selection bias in an example (synthetic) observational dataset. Here, to treat heart disease, a doctor typically prescribes surgery ($t=1$) to younger patients ($\bullet$) and medication ($t=0$) to older ones ($+$). Note that instances with larger (resp., smaller) x values have a higher chance to be assigned to the $t=0$ (resp., 1) treatment arm; hence we have selection bias. The counterfactual outcomes (only to be used for evaluation purpose) are illustrated by small faint $\cdot$ ($+$) for $\neg t=1$ (0).

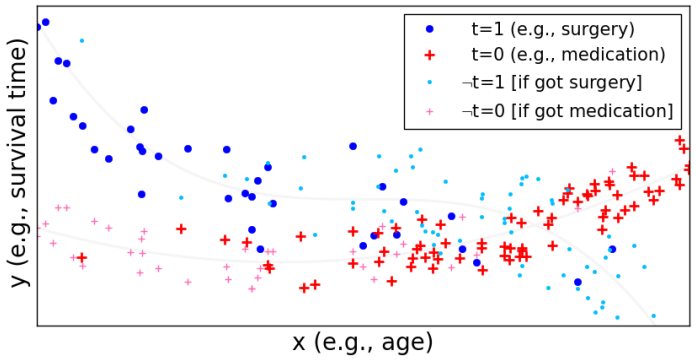


Figure 1.2: An example observational dataset (synthetic). Points in • represent a patient who actually got surgery ($t = 1$) and indicate their respective *factual* outcome. Points in • represent patients who in reality got medication but indicate their *counterfactual* outcome had they got surgery ($\neg t = 1$).

The main focus of this research is on finding the Individual Treatment Effect (ITE) for each instance i — *i.e.*, estimating $\mathrm{e}_i = y_i^1 - y_i^0$. We frame the solution as a regression task — *i.e.*, learning the function $f : \mathcal{X} \times \mathcal{T} \rightarrow \mathcal{Y}$ that can accurately predict the outcomes (both observed $\hat{y_i}^{t_i}$ as well as counterfactuals $\hat{y_i}^{\neg t_i}$) given the context information x_i for each individual. There are two challenges associated with this task:

1. The fact that counterfactual outcomes for any specific instance x_i are **unobservable** (*i.e.*, not present in any training data) [41] makes estimating treatment effects more difficult than the generalization[1] problem in the typical supervised learning paradigm. [2]

2. Often, the training data is an observational study, which means the data ...

   (a) is **off-line** — *i.e.*, we cannot make interventions to explore the effect of various treatments on the outcome, effectively preventing discovery of the causal relationships; and

[1] *I.e.*, how well a trained model can make predictions about unseen data.

[2] In supervised learning, given $\{[x_i, y_i]\}_{i=1..N}$, we want to predict y_j for an unseen x_j. In causal inference, however, given $\{[x_i, t_i, y_i^{t_i}]\}_{i=1..N}$ we not only want to estimate all $y_i^{\neg t_i}$s but also all y_js (of all treatments t_j) of an unseen x_j.

(b) is likely to exhibit **selection bias** — *i.e.*, the treatment assignment can depend on the subjects' attributes. This, in turn, creates skewed datasets, which have lots of instances in parts of the domain and fewer instances in other parts. The challenge is that although the accuracy and confidence of a fitted regression model is expected to be high in the former, it would be less so in the latter.

In my PhD studies, I have explored ways to address the above-mentioned challenges associated with counterfactual reasoning for causal effect estimation. Specifically, this research makes the following four Research Contributions (RCs):

RC1. Realistic Synthetic Observational Datasets The fact that the ground truth for counterfactuals are unobservable in real-world observational datasets, makes it non-trivial and challenging to properly evaluate different methods in terms of their performance in estimating treatment effects. Therefore, we require algorithms that can generate *realistic* synthetic observational datasets that exhibit various degrees of selection bias.

RC2. Accounting for Selection Bias Learning a common representation space Φ (*cf.*, [7]), shared between treatment arms, can be a good strategy to *reduce* the selection bias [49], [95]. This is effective if the distributions of the transformed instances $\Phi(\, x \,)$ belonging to every treatment arm are similar — making the (transformed) dataset close to an RCT. However, this learned representation might not remove all the selection bias, due to the existence of confounders (*i.e.*, variables that determine both T and Y). Reasonably, it should be possible to further alleviate the selection bias (in an attempt to *account for* it) by incorporating appropriate re-weighting schemes.

RC3. Identifying the Underlying Factors of Observational Data Without loss of generality, we can assume that the random variable X follows a(n unknown) joint probability distribution $\Pr(\, X \,|\, \Gamma, \Delta, \Upsilon, \Xi \,)$, treatment T follows $\Pr(\, T \,|\, \Gamma, \Delta \,)$, and outcome Y^T follows $\Pr_T(\, Y^T \,|\, \Delta, \Upsilon \,)$, where Γ, Δ,

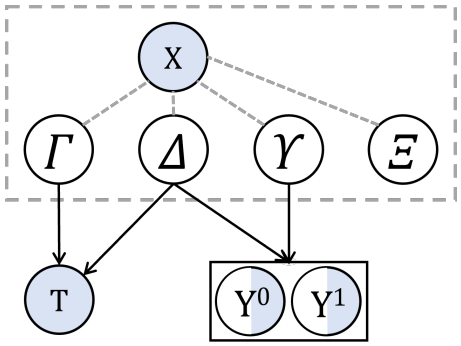


Figure 1.3: Underlying (latent) factors of X; Γ are factors that partially determine only t, but not the other variables; Υ are factors that partially determine y; and Δ are confounders (factors that partially determine both t and y). Selection bias is induced by Γ and Δ. Ξ represents noise. Here, we only consider binary treatment options $\{T^0, T^1\}$.

and Υ represent the three underlying factors that are not noise[3] that generate an observational dataset $\mathcal{D}$ (see Figure 1.3). We hypothesize that explicit identification of the underlying factors $\{\Gamma, \Delta, \Upsilon\}$ in observational datasets offers great insight to guide designing models that better handle selection bias and consequently achieve better performance in terms of estimating causal effects.

RC4. Generative Models for Causal Effect Estimation The majority of methods proposed to estimate treatment effects fall under the category of *discriminative* approaches — *i.e.*, learning a direct conditional model of y given x. A promising direction is to consider developing *generative* models, in an attempt to shed light on the true underlying data generating mechanism. We hypothesize that generative models can be employed to efficiently learn disentangled representations of the underlying factors of observational studies,

[3] Examples for:

(Γ) **wealth**: rich patients receiving the expensive treatment while poor patients receiving the cheap one, although outcomes of the possible treatments are not particularly dependent on the patients' wealth status.

(Δ) **age**: young patients receiving surgery while old patients receiving medication.

(Υ) **genetic information** that determines the efficacy of various medications, however, such relationships are unknown to the attending physician.

which in turn is useful for the downstream task of counterfactual regression.

The rest of this document is organized as follows: Chapter 2 reviews the background of causality and elaborates on the related works. Chapter 3 discusses the evaluation metrics and benchmarks used for empirical experiments. Chapter 4 explains RC1 and is based on a paper published in Canadian AI 2018 [31]; Chapter 5 elaborates on RC2 and is based on a paper published in IJCAI 2019 [32]; Chapter 6 gives details for RC3 and is based on a paper published in ICLR 2020 [33]; and Chapter 7 describes RC4 [34]. Chapter 8 concludes this dissertation by providing possible future directions of this research and highlighting the list of my contributions.

Chapter 2

Background and Related Works

There are two main tasks in studying causality [28], [82]:

1. Causal **Discovery**, where the goal is to determine changing **which** variables would *cause* other variables to change their values (*i.e.*, *effect*) [35]. In other words, given the data, the goal is to determine the causal graph.

2. Causal **Inference**, where the goal is to figure out **how much** the value of a certain variable would change (*i.e.*, *effect*), had a certain variable (*i.e.*, *cause*) changed its value. In other words, given the causal graph and data, the goal is to estimate the causal effects.

As an example, Sewell and Shah [94] studied factors that would motivate high school students to attend college. They looked at gender, socioeconomic status, IQ (intelligence quotient) score, parental encouragement, and college plans. The goal for **causal discovery** is to understand the causal relationships among these variables (*e.g.*, whether the student's gender would have an effect on the amount of parental encouragement that she would receive); and the goal of **causal inference** is to figure out the amount of effect that one of these variables would have on another (*e.g.*, in the context of receiving the encouragement of a diligent parent, would the student end up developing a serious college plan for herself?).

My PhD research is focused on **causal inference**. The most fundamental challenge with causal inference from empirical data is *confounding* [79] —

	Placebo Group		Drug D Group	
Heart Attack?	Yes	No	Yes	No
Female	5% (1)	95% (19)	**7.5%** (3)	92.5% (37)
Male	30% (12)	70% (28)	**40%** (8)	60% (12)
Total	**21.7%** (13)	78.3% (47)	18.3% (11)	81.7% (49)

Table 2.1: Fictitious data illustrating the Simpson's paradox (taken from [81]).

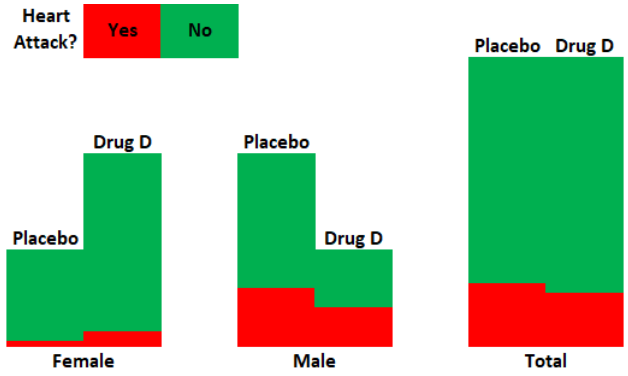


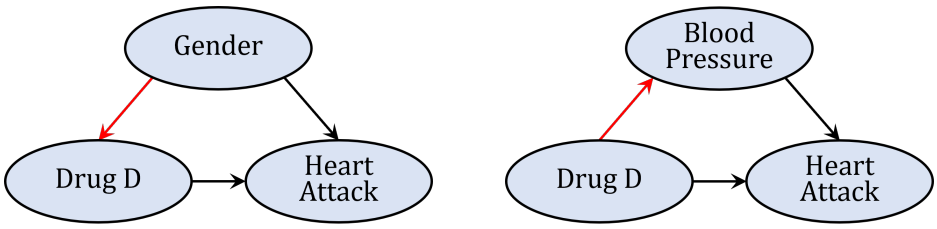
Figure 2.1: Bar-charts of the data (absolute values) in Table 2.1.

finding it and adjusting for it.[1] A discussion on the Simpson paradox (reversal) can elucidate the meaning of confounding: First described by Simpson [99], this is a phenomenon in which the statistical association that holds for an entire population is reversed in some sub-populations. Pearl and Mackenzie [81] give an example of this phenomenon (see Table 2.1): *"We might see that, in the general population, drug D reduces the risk of heart attack (from 21.7% to 18.3%), however, increases the risk in either sub-populations of males (from 30% to 40%) or females (from 5% to 7.5%)".* [2]

This reversal can happen numerically and should not confuse us. What is important is to figure out whether to adjust for gender or not — *i.e.*, partitioning the population into *homogeneous* sub-populations based on gender. To answer this question, we need to look at the underlying causal graph of the

[1] Pearl *et al.* [80] define confounding as anything that leads to a discrepancy between $\Pr(Y \mid T)$ and $\Pr(Y \mid do(T))$. The $do(\cdot)$ operator will be defined in Section 2.1.1.

[2] Also note that, as shown in Figure 2.1, taking drug D is imbalanced among the two populations; *i.e.*, more women have taken it than men (which might be for example due to targeted campaigns towards women).



(a) With a confounding variable (b) With a mediator variable

Figure 2.2: The underlying causal graphs for the Simpson's paradox examples (adapted from [81]). Note the direction of the arc connecting the top node and Drug D: In (a), it is FROM the top node (Gender), but in (b) it is TO the top node (Blood Pressure).

data (see Figure 2.2a). We find out that gender *confounds* T (here "drug D") with Y (here "heart attack") as gender points to both drug as well as heart attack. Therefore, as [81] points out, we must *adjust* for the confounding variable and deduce that drug D is indeed bad for everyone.[3]

2.1 Paradigms for Studying Causality

There are two major paradigms for studying causality: One is Pearl's Structural Causal Model (SCM) [79], in which the causal relationships between the variables are represented by a set of structural equations. This paradigm attempts to "learn the underlying causal *structure*" in the form of a graphical model. Section 2.1.1 provides a brief overview of SCMs. The Potential Outcome Framework [76], [90] is another paradigm for studying causality; this framework attempts to "infer/learn the causal *effects*" from data. Section 2.1.2 elaborates on this paradigm.

[3]We emphasize that the underlying causal graph determines whether we should or should not adjust for a variable [81]. In the above example, if, instead of male/female gender we had high/low blood pressure (BP), we should *not* have adjusted for BP, since BP is not a confounding variable but it is a mediator variable (drug points to BP and BP points to heart attack — see Figure 2.2b). We therefore conclude that in this case, the drug is indeed good for everyone.

2.1.1 Structural Causal Model

Pearl [79] defines a Structural Causal Model (SCM) as a set of equations of the form:

$$x_i = f_i(pa_i, u_i) \quad \text{for} \quad i = 1, \ldots, n \tag{2.1}$$

where pa_i (denoting *pa*rents) stands for the set of variables (endogenous) that directly determine the value of the random variable X_i; and U_i represents noise (which may be either due to unobserved factors (exogenous) or inherent stochasticity of the child variable). These Structural Equations (SEs) are formatted such that the dependent variables are placed on the left-hand-side and the explanatory variables (*i.e.*, causes) are placed on the right-hand-side. With SCMs, casual discovery corresponds to formulating the SEs and causal inference corresponds to solving them.

The set of SEs can also be represented as a Causal Bayesian Network[4], where the nodes represent the variables and the directed edges represent a causal influence from one variable to another, which are governed by the structural equations. Intervention on a variable (*e.g.*, T) is noted as $do(T = t)$, and is the result of forcing the random variable T to take on the value t.[5] This corresponds to a "mutilated (causal) Bayesian network", which matches the earlier structure, but has removed all the edges that lead into the node representing T.

We know, from the Simpson paradox, that the first step for causal inference is to identify the confounders. The Back-door Criterion [79] provides a simple graphical test to determine whether observing a set of variables $Z \subseteq X$ (in the previous example, this would be the "gender") is sufficient for identifying the causal effect of treatment $T = t$ on outcome Y — *i.e.*, $\Pr(\, y \,|\, do(T = t)\,)$. In other words, it checks whether the causal effect can be determined from observational data only, without requiring experimentation — see Equation 2.2.

[4]For a detailed description of graphical models in general, see [58]; and for causal Bayesian networks, see the first chapter of [79].

[5]Note this is different from conditioning on t; as the former is experimental and the latter is observational.

Definition: Back-door.

A set of variables $Z \subseteq X$ satisfies the back-door criterion relative to a pair of variables (T, Y) in a Directed Acyclic Graph (DAG) $G = \big[[X, T, Y], E\big]$[6] if:

(i) no node in Z is a descendant of T; and

(ii) Z blocks[7] every path between T and Y that contains an arrow into T.

For example, see Figure 2.2a in which T is drug D, X is heart attack, and Z is gender.

Back-door Adjustment. If a set of variables Z satisfies the back-door criterion relative to (T, Y), then the causal effect of T on Y is identifiable and is given by the formula:

$$\Pr(\, y \,|\, do(T = t)\,) \;=\; \sum_{z} \Pr(\, y \,|\, t, z\,) \Pr(\, z\,) \tag{2.2}$$

The intuition behind conditioning on the confounders is to remove the spurious (back-door) paths, such that only the genuinely causal path remains active after the back-door adjustment; making the measured effect unbiased (*cf.*, [81] pages 186–189).

Figure 2.3 gives an example of a couple of sets of variables that do satisfy the back-door criterion and another couple of sets that do not.

2.1.2 Potential Outcome Framework

The Potential Outcome Framework was first proposed by Neyman [76] and later popularized by Rubin [90]; it led to the framework now called the Neyman-Rubin Causal Model. Let us start with a formal definition of the *Potential Outcome*:

Definition: Potential Outcome.

Given the treatment variable T, the subjects under study X, and the outcome variable Y, the potential outcome of the instance $x \in X$, namely $Y^t(x)$, is defined as the value of Y for instance x if T had been set to t.[8]

[6] where E is the set of directed edges, with respect to the nodes $X \cup \{T, Y\}$

[7] in a Bayesian net sense

[8] *E.g.*, how much salary $Y^t(x)$ would Mr. Smith (x) get in a job later, if he enrolls in a certain job training program (t).

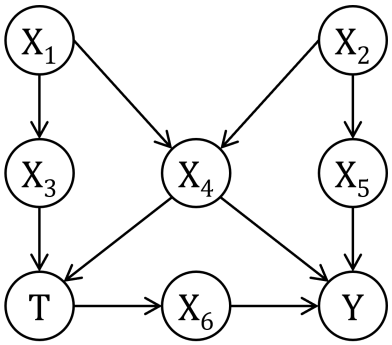


Figure 2.3: An example diagram to illustrate the back-door criterion. Variables $\{X3, X4\}$ or $\{X4, X5\}$ satisfy the back-door criterion and adjusting for them yields a consistent estimate of $\Pr(\, Y \,|\, do(T)\,)$. Variables $\{X4\}$ or $\{X6\}$ do not satisfy the back-door criterion and adjusting for them would yield a biased estimate (*i.e.*, knowing $\{X4\}$ or $\{X6\}$ does not make the causal effect identifiable). Taken from [79].

This definition reveals one of the fundamental challenges of causal inference: only one potential outcome can be observed — *i.e.*, the one corresponding to the administered treatment. The other potential outcomes are never observed[9]; these are called *counterfactuals*.

Definition: Individual Treatment Effect (ITE).

This quantity is defined for a binary treatment $t \in \{0, 1\}$ as:

$$e(x) \quad = \quad Y^1(x) - Y^0(x) \tag{2.3}$$

Definition: Average Treatment Effect (ATE).

The ATE is defined as the expectation of ITE over the entire population $\mathcal{D}$:

$$\text{ATE} \quad = \quad \mathbb{E}_{x \sim \mathcal{D}}[Y^1(x) - Y^0(x)] \tag{2.4}$$

[9]Mr. Smith in the previous example in Footnote 8 did in fact enroll in the job training program and because of that, gained a certain amount of salary. Now, we can never know what his salary would have been, had he not enrolled in that job training program because we cannot go back in time and change his actions.

Assumptions

In order for ITE and ATE to be identifiable[10], the potential outcome framework needs three assumptions to hold [45]:

Assumption 1: Stable Unit Treatment Value Assumption (SUTVA). The potential outcomes for any unit (think "patient") do not vary with the treatments assigned to other units.[11] Moreover, for each unit, there are no differences in forms or versions of each treatment level, that lead to different potential outcomes.[12]

Assumption 2: Unconfoundedness. There are no *unobserved* confounders — *i.e.*, covariates that contribute to both treatment selection procedure as well as determination of outcomes. Formally,

$$\{Y^t\}_{t\in\mathcal{T}} \perp\!\!\!\perp T \mid X \tag{2.5}$$

Assumption 3: Overlap. Every individual x should have a non-zero chance of being assigned to any treatment arm. That is,

$$0 < \Pr(T{=}t \,|\, X{=}x) < 1 \qquad \forall t \in \mathcal{T},\ \forall x \in \mathcal{X} \tag{2.6}$$

Assumptions 2 and 3 together are called *Strong Ignorability* [88] for short.

2.2 Addressing the Challenges of Causal Inference

As mentioned above, there are two main challenges associated with estimating treatment effects:

(i) **Counterfactual outcomes** are unobservable [41] (*i.e.*, not present in any training data; *e.g.*, instances indicated by • or + in Figure 1.2).

[10] *I.e.*, the causal effect(s) can be uniquely determined from observational data [79].

[11] *I.e.*, patients do not compete to get a certain treatment. Hence, SUTVA does not apply to cases such as organ transplantation, where there is limited supply of organs (so if patient A gets the organ, then patient B does not).

[12] *I.e.*, patients in each treatment arm get the exact same treatment (*e.g.*, dosage, procedure, etc.).

This makes estimating treatment effects a different (harder) problem than the generalization problem in the supervised learning paradigm. The missingness of counterfactuals is an inherent characteristic of causal inference.

In the literature, there are two main approaches to address this challenge: Matching (see Section 2.2.2) and Regression (see Section 2.2.3).

(ii) **Selection bias** in observational datasets means that the administered treatment T depends on some or all attributes of individual X — *i.e.*, $\Pr(T \mid X) \neq \Pr(T)$; see Figure 1.1b. This means, in some regions of the domain, we may have relatively fewer instances with treatment $t = 1$ than with $t = 0$ — *e.g.*, all men get $t = 1$ and all women get $t = 0$. This sparsity, in turn, would decrease the accuracy and confidence of predicting the outcome of alternative treatment(s) (*i.e.*, counterfactuals) at those regions.

Selection bias is equivalent to a domain adaptation scenario where a model is trained on the observed data distribution (source), but should perform well on the counterfactual one (target) — *cf.*, Section 2.4.2 for a detailed account. The current causal inference literature deals with selection bias via two main approaches: Re-weighting (see Section 2.2.4) and Representation learning (see Section 2.2.5).

Before diving into further details, note that many matching and regression approaches require propensity scores (*i.e.*, $\pi_\vartheta(t = 1 \mid x) = \pi_\vartheta(x)$ which is the probability that patient x gets treatment $t = 1$) as part of their algorithm. Section 2.2.1 provides a brief literature review on calculating propensity scores from data. The following four sections elaborate on matching, regression, re-weighting, and representation learning approaches (*i.e.*, Sections 2.2.2, 2.2.3, 2.2.4, and 2.2.5 respectively) and each lists some of the related papers.

2.2.1 Propensity Scores

A popular and standard choice in the literature is to estimate the propensity scores by Logistic Regression (LR) — *i.e.*,

$$\pi_\vartheta(x) = \frac{\exp\left(x^\top \vartheta\right)}{1 + \exp\left(x^\top \vartheta\right)} \tag{2.7}$$

Finding the parameters ϑ is often done by maximizing the log-likelihood function — *i.e.*, setting the parameters:

$$\hat{\vartheta}_{\text{MLE}} = \arg\max_{\vartheta \in \Theta} \frac{1}{N} \sum_{i=1}^{N} t_i \log\{\pi_\vartheta(x_i)\} + (1 - t_i) \log\{1 - \pi_\vartheta(x_i)\} \tag{2.8}$$

$$\longrightarrow \sum_{i=1}^{N} \left[\frac{t_i}{\pi_\vartheta(x_i)} - \frac{1 - t_i}{1 - \pi_\vartheta(x_i)} \right] \pi'_\vartheta(x_i) = 0 \tag{2.9}$$

where $\pi'_\vartheta(x_i) = \frac{\partial \pi_\vartheta(x_i)}{\partial \vartheta^\top}$.

The main issue with this approach is that the propensity score model (*e.g.*, LR) may be misspecified[13], which in turn might yield substantially biased estimates of treatment effects [20].

Imai and Ratkovic [44] proposed the Covariate Balancing Propensity Score (CBPS) method, which focuses on improving estimation of the propensity scores. We know that propensity is a balancing score [88] — *i.e.*,

$$X \perp\!\!\!\perp T \,|\, \pi_\vartheta(X) \tag{2.10}$$

However, the authors noted that, in order for the covariates to be balanced, Equation (2.9) must hold for *any* function of covariates $f(x)$, and not just $\pi'_\vartheta(x)$. For example, by setting $f(x) = x$, the first moment of each covariate would be balanced even if the model is misspecified. CBPS learns ϑ such that the covariates are balanced according to not only Equation (2.9), but also all the covariate moments. This makes CBPS robust to mild misspecification of the parametric propensity score model.

[13] *I.e.*, the selected functional form for the association of covariates with treatment selection is incorrect; *e.g.*, model assumes it to be linear, but in reality it is quadratic.

McCaffrey *et al.* [74] proposed a multivariate non-parametric regression technique, namely Generalized Boosted Models (GBM), to estimate the propensity score. GBM aggregates many weak learners (*i.e.*, regression trees with limited depth) to estimate a smooth function of a large number of covariates. In order to simplify the required computations, GBM models the log-odds of treatment assignment:

$$g(x) = \log\big(\pi(x)/[1{-}\pi(x)]\big) \tag{2.11}$$

as opposed to directly modeling propensity scores. They maximize the log-likelihood:

$$l(g) = \sum_{i=1}^{N} t_i g(x_i) - \log\big(1 + \exp[g(x_i)]\big) \tag{2.12}$$

to find $g(x)$, from which they can calculate the propensities:

$$\pi(x) = 1/\big(1 + \exp[-g(x)]\big) \tag{2.13}$$

It is noteworthy that model misspecification is not a problem with GBM (since it is a non-parametric method). Moreover, McCaffrey *et al.* [74] demonstrated that GBM can handle high dimensional data.

2.2.2 Finding Counterfactual Outcomes by Matching

Matching is based on the intuitive idea of estimating the counterfactual for a "treated" unit by seeking its most *similar* counterpart in the controlled group; and vice versa for a "controlled" unit [36], [89], [100]. Various definitions of the similarity measure (*i.e.*, closeness) give rise to different matching techniques. Here, we briefly discuss two basic matching methods, one performing on the raw covariates space and another on the 1D propensity score space: Nearest Neighbour Matching (NNM) and Propensity Score Matching (PSM), respectively.

Nearest Neighbour Matching (NNM)

Given a distance metric, NNM finds the one nearest neighbor in the $\neg t$ group in order to estimate the counterfactual outcome of the unit who received t. Alternative methods can use K-nearest neighbours and aggregate the results via

averaging. Two widely-used distance metrics for NNM are Euclidean distance and Mahalanobis distance.

Propensity Score Matching (PSM)

Another important similarity measure is the propensity score — *i.e.*, the probability of receiving treatment $t=1$ for each unit. The Propensity Score Matching (PSM) method pairs the units from two treatment groups with similar scores [18], [88].

2.2.3 Finding Counterfactual Outcomes by Regression

Another approach for estimating the counterfactuals is to fit a regression model — *i.e.*, learning a function $f : \mathcal{X} \times \mathcal{T} \to \mathcal{Y}$. Using the covariates as inputs, either one model can be learned with the treatment as an input feature as well, *e.g.*,

- Balancing Neural Network (BNN) [49]

or multiple separate models can be learned, one for each treatment arm, *e.g.*,

- Weighted Gaussian Process (WGP) [112]
- CounterFactual Regression Network (CFR Net) [95]
- Propensity Dropout (PD) [2]
- Causal Multi-task Gaussian Processes (CMGP) [1]
- Causal Effect Variational AutoEncoder (CEVAE) [69]
- Similarity preserved Individual Treatment Effect (SITE) [114]
- Deep Treat [3]
- Generative Adversarial Nets for inference of Individualised Treatment Effects (GANITE) [115]
- CFR with Importance Sampling Weights (CFR-ISW) [32]
- Dragon Net [96]
- Reducing Selection Bias Net (RSB Net) [120]
- Disentangled Representations for CFR (DR-CFR) [33]
- Treatment Effect by Disentangled Variational AutoEncoder (TEDVAE) [119]

- Variational AutoEncoder for Causal Inference (VAE-CI) [34]

2.2.4 Addressing Selection Bias by Re-weighting

Re-weighting is a common statistical method for addressing selection bias [71]. This strategy attempts to overcome the problem of data sparseness in parts of the subspace of features by up-weighting the few available instances in those regions and down-weighting the others. Inverse Propensity Score Weighting is a famous method that weights instances such that the synthesized dataset resembles an RCT. In summary, re-weighting is an attempt to **account for** the selection bias.

2.2.5 Addressing Selection Bias by Representation Learning

Representation learning [7] can be used to **reduce** selection bias. The idea here is to learn a common representation space $\Phi(\cdot)$ for both treatment arms by making the distributions $\Pr(x \mid t=0)$ and $\Pr(x \mid t=1)$ as close to each other as possible. Obviously, the learned representation should also retain enough information to be capable of accurately estimating the observed outcomes.

Figure 2.4 illustrates an example of reduction in selection bias using representation learning. Here, the $t=1$ and $t=0$ distributions of the transformed instances $\Phi(x)$ (*i.e.*, the distribution of + versus • on the x-axis of Figure 2.4-right) are much closer to each other compared to those distributions in the original x space (Figure 2.4-left). In particular, while both subfigures include the same number of •s and +s, they are scattered differently left-to-right: Figure 2.4-right has more •s on its right-half, and more +s on its left-half. Of course, the observed outcomes y (on the y-axis) remain unchanged through this transformation.

Johansson *et al.* [49] enforced this closeness by including an Integral Probability Metric (IPM)[14] [73] in the objective function (to be minimized) that measures the distance between the joint distributions of Φ and t (factual)

[14]Maximum Mean Discrepancy (MMD) [26] and Wasserstein distance [17] are two well-known methods to measure the discrepancy between two probability distributions.

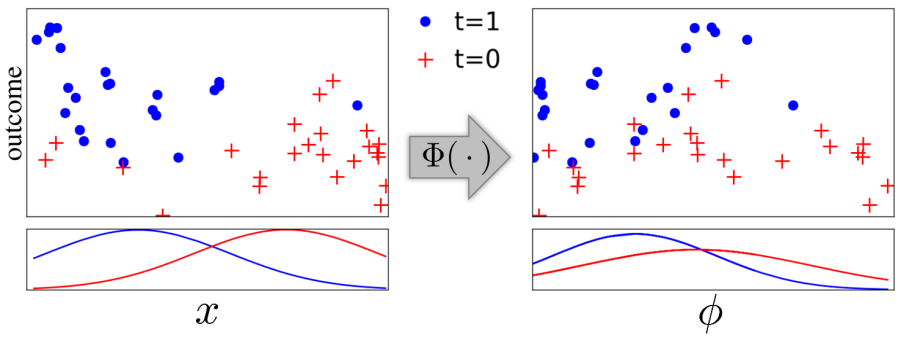


Figure 2.4: The learned representation has reduced the selection bias. That is, the $t=1$ and $t=0$ distributions of the transformed instances $\Phi(\,x\,)$ — here, the distribution of $+$ versus $\bullet$ on the x-axis — are much closer to each other compared to those distributions in the original x space. Also note that the observed outcomes y (on the y-axis) remain unchanged through this transformation.

versus Φ and $\neg t$ (counterfactual):

$$\texttt{disc} \;=\; \text{IPM}\Big(\big\{[\,\Phi(x_i), t_i\,]\big\}_{i=1}^{N}, \big\{[\,\Phi(x_i), \neg t_i\,]\big\}_{i=1}^{N}\Big) \tag{2.14}$$

where `disc` denotes discrepancy.

This makes sense in theory: if the factual and counterfactual joint distributions are hard to distinguish, it means that the data is close to RCT. However, since the two joint distributions only differ in their treatment bit (*i.e.*, t versus $\neg t$, while $\Phi(\,x\,)$ is the same for both), the numerical value of `disc` would naturally be small. Therefore, its contribution to the objective would be negligible. Moreover, a high dimensional $\Phi(\,\cdot\,)$ can overshadow the information in the treatment bit, which results in an even smaller `disc`.

Shalit *et al.* [95] addressed these issues by defining `disc` between the two distributions $\Pr(\,\Phi(\,x\,)\,|\,t=0\,)$ and $\Pr(\,\Phi(\,x\,)\,|\,t=1\,)$:

$$\texttt{disc} \;=\; \text{IPM}\big(\{\Phi(x_i)\}_{i:t_i=0}, \{\Phi(x_i)\}_{i:t_i=1}\big) \tag{2.15}$$

2.3 Detailed Discussions of the Literature

This section elaborates the ideas presented in the previous section by discussing their main contributions and gaps.

2.3.1 Notable Matching Methods

Balanced and Nonlinear Representations (BNR)

Li and Fu [64] proposed to train an ordinal classifier of quantized outcomes in a Reproducing Kernel Hilbert Space (RKHS). The kernel $\phi(x)$ is trained to learn representations that minimize the within-class — according to the (quantized) y values — scatter, while maximizing the noncontiguous-class scatter. This is referred to as Ordinal Scatter Discrepancy (OSD). A second constraint attempts to balance the learned representation via MMD according to the `disc` term in Equation (2.15).

Note, however, that OSD ignores the treatment bit altogether; which means instances with similar outcomes would be close in the kernel space, irrespective of their received treatment. This does not make sense since the treatment bit has a key role in determining the value of outcome. This appears to be a major flaw in the BNR algorithm.

Causal Forests (CF)

Wager and Athey [109] extended Random Forest [11] for estimating heterogeneous treatment effects. The paper develops two algorithms for growing causal trees and aggregating the results for the entire forest. CF can be categorized under matching approaches, because the trees' leaf nodes provide an *adaptive* set of nearest neighbors. The "adaptive" term refers to the trees' growing procedure (specifically, various splitting criteria), which determines the most important covariates to consider in selecting the nearest neighbors.

2.3.2 Notable Regression Methods

Bayesian Additive Regression Trees (BART)

First introduced in [14], [15], BART is a Bayesian "sum-of-trees" model where each tree is constrained by a regularization prior, so that it is trained to be a weak learner. Most interestingly, [14] showed (with extensive empirical experiments) that a *default* prior that is minimally dependent on the data, performs indistinguishably from (i) a prior whose parameters are selected via

cross-validation; and (ii) many other methods that rely on cross-validation to choose model parameters.

The fact that BART-*default* can train well without needing cross-validation is of great advantage in settings with partial information data, such as ours, where the counterfactuals are not available for training but required for evaluation. Hill [38] realized this potential and used BART for the task of causal inference.

Targeted Maximum Likelihood Estimator (TMLE)

Van der Laan and Rose [106] proposed a two step procedure to estimate the causal effects. In the first step, their super-learner algorithm is used to obtain an initial estimate of the outcomes y given t and x. The super-learner algorithm is similar to ensemble learning, where multiple weaker regressors are trained and combined to improve the overall accuracy of the model. In the second step, this initial fit is updated, while focusing on the bias-variance trade-off for the parameter of interest (here, ATE). TMLE is a doubly robust estimator[15].

CounterFactual Regression Network (CFR Net)

Shalit *et al.* [95] reduced the selection bias by learning a common representation space $\Phi(\,\cdot\,)$ that tries to make $\Pr(\,\Phi(x)\,|\,t=0\,)$ and $\Pr(\,\Phi(x)\,|\,t=1\,)$ as close to each other as possible (see Figure 2.4), provided that $\Phi(\,x\,)$ retains enough information that all $|\,\mathcal{T}\,|$ learned regressors $h^t(\,\Phi\,)$ can generalize well on the observed outcomes. Φ and h^t are implemented as neural networks and learned by minimizing:

$$\begin{aligned} J(\Phi, h^0, h^1) \;&= \frac{1}{N}\sum_{i=1}^{N} \omega(t_i)\cdot\mathcal{L}\big[\,y_i,\; h^{t_i}\big(\Phi(x_i)\big)\big] \\ &\quad + \alpha\cdot\texttt{disc}\big(\{\Phi(x_i)\}_{i:t_i=0}, \{\Phi(x_i)\}_{i:t_i=1}\big) \\ &\quad + \lambda\cdot\mathfrak{Reg}(h^0, h^1) \end{aligned} \tag{2.16}$$

[15]Doubly Robust (DR) estimators combine outcome regression and propensity score methods to estimate the causal effects [23]. In this case, the DR would yield an unbiased estimate of the causal effect even if only one of the methods is correctly specified.

where $\mathfrak{Reg}(h)$ is the regularization term for penalizing model complexity; `disc` term is the IPM$(\cdot)$ as defined in Equation (2.15); and $L\big[\, y_i,\, h^{t_i}\big(\Phi(x_i)\big)\big]$ is the loss of predicting the observed outcome for sample i, weighted by ω_i, derived via $\omega_i = \frac{t_i}{2u} + \frac{1-t_i}{2(1-u)}$, where $u = \frac{1}{N}\sum_{i=1}^{N} t_i = \Pr(\, t{=}1\,)$. This is effectively setting:

$$\omega_i \;=\; \frac{1}{2\Pr(\, t_i\,)} \;=\; \frac{1}{2}\big[\, 1 + \frac{\Pr(\, \neg t_i\,)}{\Pr(\, t_i\,)}\,\big] \tag{2.17}$$

where $\Pr(\, t_i\,)$ is the probability of selecting treatment $t_i \in \{0, 1\}$ over the entire population.

Similarity preserved Individual Treatment Effect (SITE)

Yao *et al*. [114] proposed SITE, which extends [95]'s framework by adding a local similarity preserving component. This component acts as a regularization term, that attempts to retain the same neighbourhood relationships in the learned representation space as exhibited in the original space, by matching the propensity scores $\Pr(\, t = 1 \,|\, x\,)$ and $\Pr(\, t = 1 \,|\, \phi\,)$. This, however, results in learning sub-optimal representations when there exists factors that induce selection bias but do not determine the outcomes ($\Gamma \neq \emptyset$ in Figure 1.3). This is because SITE tries to keep instances whose Γs are far apart in the original space, also far apart in ϕ. This is bad, since SITE does not discard the irrelevant information in Γ (effectively not removing the unnecessary selection bias) even when doing so does not hurt the outcome estimation at all.

Deep Treat

Atan *et al*. [3] used an auto-encoder network to learn a representation space $\Phi(\,\cdot\,)$ that attempts to reduce the selection bias by minimizing the cross entropy loss between $\Pr(\, t\,)$ and $\Pr(\, t \,|\, \Phi(\, x\,)\,)$. However, by training an auto-encoder, they force their network to be able to reproduce *all* the covariates in x from Φ; which effectively neutralizes the merit of using representation learning for reducing the selection bias when $\Gamma \neq \emptyset$.

Generative Adversarial Nets for inference of Individualised Treatment Effects (GANITE)

Yoon *et al.* [115] proposed the counterfactual GAN, whose generator G when given $[x, t, y^F]$, tries to estimate counterfactual outcomes ($\hat{y}^{CF}$); and whose discriminator D tries to identify the factual outcome given $[x, (y^F, \hat{y}^{CF})]$.[16] However, it is not clear why D should learn to distinguish factual from counterfactual outcome as opposed to learning the treatment selection mechanism — *i.e.*, just learn the logger function $g(x) = t$ that determines which treatment to apply. Yoon *et al.* [115] assume the former while the latter seems more plausible. Although this work is one of the few generative approaches for causal inference in the literature, it seems that the adversarial training designed in GANITE provides no advantage in terms of accurate estimation of counterfactuals.

Dragon Net

The main objective of the work by Shi *et al.* [96] was to estimate the ATE, which they explain requires a two stage procedure (similar to TMLE): (i) fit models that predict the outcomes and (ii) find a downstream estimator of the effect. Their method is based on a classic result from strong ignorability (*i.e.*, Theorem 3 in [88]), which states:

$$\begin{array}{lll} (y^1, y^0) \perp\!\!\!\perp t \,|\, x & \& & \Pr(\, t = 1 \,|\, x \,) \in (0, 1) \\ (y^1, y^0) \perp\!\!\!\perp t \,|\, b(x) & \& & \Pr(\, t = 1 \,|\, b(x) \,) \in (0, 1) \end{array} \quad \Longrightarrow$$

where $b(x)$ is a balancing score (see Equation 2.10 for a definition). They consider propensity score as a balancing score and deduce from this theorem that only the parts of X relevant for predicting the treatment are required for the estimation of the causal effect.[17]

Their ideal proposed structure is a neural network that is first fitted to predict treatments (using the *propensity* head of the neural network), then

[16] *I.e.*, given x, as well as the (estimated) values of y^0 and y^1, determine which one is in fact the factual outcome.

[17] They do acknowledge that this would hurt the predictive performance for individual outcomes.

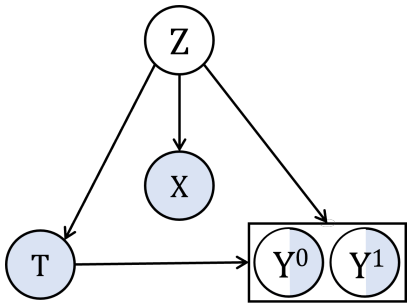


Figure 2.5: Graphical model of the CEVAE method [69]

removes this head and uses the learned representations from previous step (fixed) to estimate the outcomes (using the *outcome* heads of the neural network). However, it seems that their interpretation of the theorem is wrong. The theorem only provides a way to *match* treated and control instances to find potential counterfactual outcomes in order to calculate ATE; however, Shi *et al*. [96] appeared to misuse this theorem to find minimal representations on which to *regress* and find the counterfactuals. Clearly, if true, this either requires a proof or empirical evidence.

Causal Effect Variational AutoEncoder (CEVAE)

Louizos *et al*. [69] used VAE to extract latent confounders from their observed proxies in X. While this is a step in the right direction, empirical results show that it does not always accurately estimate treatment effects. The authors note that this may be because CEVAE is not able to address the problem of selection bias. Another reason for CEVAE's sub-optimal performance might be its assumed graphical model of the underlying data generating mechanism, depicted in Figure 2.5. This model assumes that there is only one latent variable Z (confounding T and Y) that generates the entire observational data; however, [33], [59] have shown the possibility of involving more factors (see Figure 1.3) and the advantages of accounting for them.

Treatment Effect by Disentangled Variational AutoEncoder (TEDVAE)

Similar to our DR-CFR [33] (see Chapter 6), Zhang *et al*. [119] proposed TEDVAE in an attempt to learn disentangled factors but using a *generative* model instead (*i.e*., a VAE with a three-headed encoder, one for each underlying factor). While their method proposed an interesting intuition on how to achieve this task, according to the reported empirical results (see their Figure 4c), the authors found that TEDVAE was not successful in identifying the risk factors z_y (equivalent to our Υ — see Figure 1.3). This might be because their model does not have a mechanism for distinguishing between the risk factors and confoundings z_c (equivalent to our Δ — see Figure 1.3). The evidence is in TEDVAE's objective function (Equation (8)), which would allow z_y to be degenerate and have all information embedded in z_c.

2.4 Closely Related Fields

2.4.1 Off-policy Learning from Logged Bandit Feedback

Learning treatment effects from observational datasets is closely related to "off-policy learning from logged bandit feedback" — *cf*., [102], whose goal is to learn an optimal policy that selects the best personalized treatment for each individual.

One solution strategy is Outcome Prediction (OP)[18] — *i.e*., estimating $y(x,t)$ for each x and every t, then select the treatment that promises the best outcome $\pi(\,t\,|\,x\,) = \underset{t}{\operatorname{argmax}}\; y(x,t)$. OP is equivalent to what is done for ITE estimation. While this approach is overkill (as computing an optimal policy only requires *ranking* the potential treatments) we consider this beneficial, since predicting the exact outcomes is valuable to both patients as well as insurance companies: knowing the margin of effect would hopefully increase compliance in the former and persuade the latter to accommodate the more expensive treatment.

[18] Also known as the Direct Method (DM) [21].

Another strategy bypasses the outcome prediction step altogether and directly obtains the optimal policy by maximizing a utility function. This is similar to the "expected return" in Reinforcement Learning (RL) [101]. The following sections touch on the most notable methods under this category.

Inverse Propensity Score Weighting

Inverse Propensity Score (IPS) weighting is an *importance sampling* technique that adjusts the weights of different instances in order to address the selection bias problem. Here, we study a variant of the early works of [42] and [88], since we are interested in evaluating a stochastic policy $\pi(\, t \,|\, x \,)$ as opposed to a deterministic one [71]. This variant has been used in many closely-related applications, such as off-policy RL [101], off-policy learning for contextual bandits [61], [62], and counterfactual learning with causal graphs [10]. First, we formulate a utility function with importance sampling weights $\frac{\pi(\cdot)}{\pi_0(\cdot)}$ as:

$$\widehat{U}_{IPS}(\pi) \quad = \quad \frac{1}{N} \sum_{i=1}^{N} \frac{\pi(\, t_i \,|\, x_i \,)}{\pi_0(\, t_i \,|\, x_i \,)} \; y(\, x_i,\, t_i \,) \tag{2.18}$$

where N is the number of instances, $\pi_0(\, t_i \,|\, x_i \,)$ is the policy used to sample treatments in the training set (*i.e.*, the "logging policy" or "behaviour policy" in RL), $\pi(\, t_i \,|\, x_i \,)$ is the probability of selecting t_i given x_i by the proposed policy $\pi(\cdot)$, and $y(\, x_i,\, t_i \,)$ is the observed outcome. Note that IPS is an unbiased estimator of the true [unknown] utility $U(\pi)$, meaning $\mathbb{E}_{\mathcal{D}}[\; \widehat{U}_{IPS}(\pi) \;] = U(\pi)$ for any $\pi(\cdot)$ provided that $\pi_0(\cdot)$ has a non-zero value everywhere in its support [88]. Ultimately, the optimal policy π^* is obtained by:

$$\pi^* \quad = \quad \underset{\pi \in \Pi}{\arg\max}\, \widehat{U}(\pi) \quad = \quad \underset{\pi \in \Pi}{\arg\min}\, \widehat{R}(\pi) \tag{2.19}$$

where Π is the hypothesis space for all possible policies and $\widehat{R}(\pi) = -\widehat{U}(\pi)$ is the empirical risk.

Although unbiased, the IPS estimator has a high variance due to the $\pi_0(\, t_i \,|\, x_i \,)$ term in its denominator. That is, some importance sampling weights (*i.e.*, $\frac{\pi(\cdot)}{\pi_0(\cdot)}$) will be very large for any instance with small π_0 — *i.e.*, treatments that had a small chance of being selected, but were selected anyway. The most

common approach is to clip the weights [10]:

$$w_i = \min\left\{M, \frac{\pi(t_i|x_i)}{\pi_0(t_i|x_i)}\right\} \tag{2.20}$$

where the M hyperparameter is an upper bound for the importance sampling weights. The risk calculated with the clipped weights is denoted as $\widehat{R}^M(\pi)$.

Doubly Robust Estimator

As discussed earlier, methods based on OP enjoy a low variance estimate at the cost of a high bias. On the other hand, although IPS is an unbiased estimator, it suffers from a high variance. This motivated [86] to propose the Doubly Robust (DR) estimator, which leverages the strengths and mitigates the weaknesses of the above mentioned methods (see also [5]). Dudík *et al.* [21] provided non-asymptotic analysis of this method, and formulated DR's utility function as:

$$\widehat{U}_{DR}(\pi) \quad = \quad \frac{1}{N}\sum_{i=1}^{N}\left[\frac{\pi(t_i|x_i)}{\pi_0(t_i|x_i)}\Big(y(x_i,t_i) - \widehat{y}(x_i,t_i)\Big) \;+\; \mathbb{E}_{t\sim\pi|x_i}\big[\widehat{y}(x_i,t)\big]\right] \tag{2.21}$$

where $\widehat{y}(x,t)$ is the regression fit (obtained from an OP method) that predicts the (counter)factual outcome for any given patient x and treatment $t \in T$.

Self-Normalized Estimator

In order to alleviate the high variance problem, the doubly robust estimator employs a regression fit to predict counterfactual outcomes to be used as *additive* terms in the utility function — see Equation (2.21). Alternatively, Swaminathan and Joachims [104] take a *multiplicative* approach by proposing the Self-Normalized (SN) estimator, which is a stochastic variant of the method proposed by Hirano *et al.* [39] for evaluating deterministic policies with a binary choice of treatment. This Self-Normalized (SN) estimator uses the fact that:

$$\mathbb{E}\left[\sum_{i=1}^{N}\frac{\pi(t_i|x_i)}{\pi_0(t_i|x_i)}\right] = N \tag{2.22}$$

which motivates replacing N with $\sum_{i=1}^{N}\frac{\pi(t_i|x_i)}{\pi_0(t_i|x_i)}$ in Equation (2.18), leading to:

$$\widehat{U}_{SN}(\pi) = \sum_{i=1}^{N}\frac{\pi(t_i|x_i)}{\pi_0(t_i|x_i)}\,y(x_i,t_i)\Big/\sum_{i=1}^{N}\frac{\pi(t_i|x_i)}{\pi_0(t_i|x_i)} \tag{2.23}$$

The intuition is: since variance is mainly due to the importance sampling weights appearing in the numerator, having a similar factor in the denominator may cancel out some of the variability.

Counterfactual Risk Minimization

Swaminathan and Joachims [102], [103] studied the variance of the IPS estimator with clipped weights under the Empirical Risk Minimization (ERM) principle — see Equation (2.19) — to prove the following generalization bound:

$$\Pr\left[\forall\,\pi\in\Pi:\ R(\pi)\ \leq\ \widehat{R}^{M}(\pi)+\alpha(n,\gamma)\sqrt{\frac{\widehat{Var}\big(\,u(\cdot)\,\big)}{N}}+\beta(M,n,\gamma)\right]\quad\geq\quad 1-\gamma \tag{2.24}$$

where $R(\pi)$ is the true risk, $u(\cdot)=y(x,t)\min\left\{M,\frac{\pi(t|x)}{\pi_0(t|x)}\right\}$ and $\widehat{Var}(\cdot)$ is the estimated variance of its argument — here $u(\cdot)$. This suggests adding the square-root term to the ERM objective function as a penalizing factor:

$$\pi^{*}\ =\ \underset{\pi\in\Pi}{\arg\min}\left\{\widehat{R}^{M}(\pi)\ +\ \lambda\sqrt{\frac{\widehat{Var}\big(\,u(\pi)\,\big)}{N}}\right\} \tag{2.25}$$

This addition to the objective function yields the Counterfactual Risk Minimization (CRM) principle [103], which is designed to penalize high empirical variance in weighted observed outcomes. The CRM principle can be used along with any of the utility functions described in the IPS family of methods.

2.4.2 Domain Adaptation

One of the main assumptions of traditional machine learning is that both training and test data (aka *source* and *target*, denoted by subscripts $_S$ and $_T$ respectively) are sampled from the same distribution. This assumption, however, is often violated in practice. The field of Domain Adaptation (DA) attempts to address this in a systematic manner [48]. DA has applications in many areas, including computer vision [78], [111], natural language processing [63], and bio-informatics [113].

DA addresses the following two main scenarios:

- $\Pr_S(x) \neq \Pr_T(x), \quad \Pr_S(y \mid x) = \Pr_T(y \mid x)$
  This scenario is known as Covariate Shift [97].

- $\Pr_S(y) \neq \Pr_T(y), \quad \Pr_S(x \mid y) = \Pr_T(x \mid y)$
  This scenario is known as Class Imbalance [19] [47].

Sample Selection Bias

Zadrozny [116] formulated the problem from a different perspective. She assumed a new binary variable s that controls whether the respective instance is selected to be part of the dataset (*i.e.*, $s = 1$ for a selected instance). Zadrozny [116] studied four possible Sample Selection Bias scenarios:

1. $s \perp\!\!\!\perp x$ and $s \perp\!\!\!\perp y$: No bias — aka missing completely at random (MCAR) in the statistics literature [65].

2. $s \perp\!\!\!\perp y \mid x \rightarrow \Pr(s \mid x, y) = \Pr(s \mid x)$: Bias depends only on x — missing at random (MAR).

3. $s \perp\!\!\!\perp x \mid y \rightarrow \Pr(s \mid x, y) = \Pr(s \mid y)$: Bias depends only on y.

4. No independence assumptions can be made — missing not at random (MNAR)

Note that we can also deduce $\Pr(y \mid x, s) = \Pr(y \mid x)$ from the left-hand-side of item 2; meaning that as long as x is observed, the selection function that determines s (and hence the *bias*) cannot change $\Pr(y \mid x)$. In other words, if bias only depends on x, then $\Pr(y \mid x)$ remains the same in both source and target domains. This is equivalent to Covariate Shift. A similar case holds for item 3, which is equivalent to the Class Imbalance case.

2.4.3 Fairness in Machine Learning

In the Machine Learning (ML) literature, a system is considered **fair** if *"individuals who are similar in their non-protected attributes should be classified similarly"* [117]. It is only natural that models trained on off-line datasets

[19] Closely related to this scenario is the [Generalized] Target Shift [118].

Table 2.2: Sample scenario to illustrate fairness in the context of selection bias.

Covariates	**Treatment**	**Outcome**
(W, S_1)	T_1	Y_+
(B, S_1)	T_0	Y_-
(B, S_2)	T_1	Y_+

such as observational studies would inherit the biases that are embedded in them due to the data collection policy. *Fair ML algorithms* attempt to actively prevent learning from incorporating such biases into their trained models — either via fair representation learning (see [70], [117]) or through optimization with fairness constraints (see [91]).

Here, we define fairness in the context of selection bias. To illustrate this, imagine the following scenario: a white patient ($\mathbf{W}$) with symptoms S_1 receives treatment T_1. However, a black patient ($\mathbf{B}$) with similar symptoms S_1 initially does not receive any treatment T_0; until her conditions worsens to S_2, only then she will receive T_1. This scenario can be modeled as an observational dataset with three instances (*i.e.*, rows), as shown in Table 2.2. Here, for the same symptoms, we assume that the same care was required. However, due to selection bias, appropriate care was provided only to $\mathbf{W}$, but not $\mathbf{B}$.

Defining fairness in the context of selection bias makes sense (at least for us: data scientists), because treatment is the only intervention that we can apply and therefore, we should select the one that is fair; hence, fairness is in fact related to selection bias. However, this definition is not universal. Assume in the previous example that $\mathbf{B}$ does not see a doctor until she became very ill; perhaps because she is very poor or located far from a healthcare provider facility. Therefore, information corresponding to the second instance in Table 2.2 is never recorded and missing from the dataset. We leave the study of fairness in such cases to policy-makers (who can read between the lines) and stick to analysing the cases for which unfairness happened as a consequence of the observed intervention(s).

Chapter 3

Evaluation Settings

3.1 Datasets

For our proposed evaluation methodology in Chapter 4, we experimented the proposed framework with two RCT datasets (see Sections 3.1.1 and 3.1.2). We also considered three observational datasets (see Sections 3.1.3, 3.1.4, and 3.1.5) for evaluating the proposed methods in Chapters 5, 6, and 7.

3.1.1 Acupuncture

The Acupuncture RCT [107], [108] was designed to study the potential benefit of acupuncture (in addition to the standard care) for treatment of chronic headache disorders. It has 18 features, all measured prior to applying any treatment. There were two main outcomes: "severity score" and "headache frequency", each measured at two points in time: "immediately after the treatment is completed" and "at one year follow-up". Out of 401 participants, we use the 295 subjects with no missing values. In this dissertation, we only report the performance results on one of the main outcomes (*i.e.*, severity score at one year follow-up), but the others are similar.

3.1.2 Hypericum

The Hypericum RCT [43] was designed to assess the acute efficacy of a standardized extract of the herb St. John's Wort in treatment of patients with major depression disorder. This study has three arms (placebo, hypericum,

and an SSRI medication). The primary outcome measure is Hamilton Depression scale at the end of week 8. We compiled 278 features from assessment forms. In our experiments, we use the "hypericum" and "SSRI" as the binary treatment options (82 and 79 patients in each arm respectively).

3.1.3 Infant Health and Development Program (IHDP)

IHDP is a synthetic binary-treatment dataset, designed to evaluate the effect of specialist home visits on future cognitive test scores of premature infants. Hill [2011] induced selection bias by removing a non-random subset of the treated population from the original RCT data in order to create a realistic observational dataset. The resulting dataset contains 747 instances (608 control, 139 treated) with 25 covariates that measure different attributes of infants and their mothers.

We worked with the same dataset provided by and used in [49], [50], [95], in which outcomes are simulated as setting "A" of the Non-Parametric Causal Inference (NPCI) package [19]. The noiseless outcomes are used to compute the true individual effects (available for evaluation purpose only).

3.1.4 Atlantic Causal Inference Conference 2018 (ACIC'18)

ACIC'18 is a collection of 24 synthetic binary-treatment datasets released for a data challenge. Each dataset is created according to a unique data generating process (unknown to the challenge participants) that describe the relationship between the treatment assignment, the outcomes, and the covariates. For the evaluation purposes, the organizers had not only supplied the factual outcomes but also the counterfactual outcomes as well. The benchmark includes 24 datasets each with number of instances $n_m \in \{1, 2.5, 5, 10, 25, 50\} \times 10^3$ (four datasets in each category) for $m \in \{1, ..., 24\}$. The covariates matrix for each of these datasets are sub-sampled from a covariates table of real-world medical measurements taken from the Linked Birth and Infant Death Data (LBIDD) [72], that contains information corresponding to 100,000 subjects, each described with 177 features.

3.1.5 Synthetic Benchmark

We generated our synthetic datasets according to the following process, which takes as input the sample size N; dimensionalities $[m_\Gamma, m_\Delta, m_\Upsilon] \in \mathcal{Z}^{+(3)}$; for each factor $L \in \{\Gamma, \Delta, \Upsilon\}$, the means and covariance matrices (μ_L, Σ_L); and a scalar ζ that determines the slope of the logistic curve.

- For each latent factor $L \in \{\Gamma, \Delta, \Upsilon\}$
  - Form L by drawing N instances (each of size m_L) from $\mathcal{N}(\mu_L, \Sigma_L)$
  - Concatenate Γ, Δ, and Υ to make the covariates matrix X [of size $N \times (m_\Gamma + m_\Delta + m_\Upsilon)$]
  - Concatenate Γ and Δ to make Ψ [of size $N \times (m_\Gamma + m_\Delta)$]
  - Concatenate Δ and Υ to make Φ [of size $N \times (m_\Delta + m_\Upsilon)$]
- For treatment T:
  - Sample $m_\Gamma + m_\Delta$ tuple of coefficients θ from $\mathcal{N}(0,1)^{m_\Gamma + m_\Delta}$
  - Define the logging policy as $\pi_0(t = 1 \mid z) = \frac{1}{1+\exp(-\zeta z)}$, where $z = \Psi \cdot \theta$
  - For each instance x_i, sample treatment t_i from the Bernoulli distribution with parameter $\pi_0(t = 1 \mid z_i)$
- For outcomes Y^0 and Y^1:
  - Sample $m_\Delta + m_\Upsilon$ tuple of coefficients ϑ^0 and ϑ^1 from $\mathcal{N}(0,1)^{m_\Delta + m_\Upsilon}$
  - Define $y^0 = (\Phi \circ \Phi \circ \Phi + 0.5) \cdot \vartheta^0 / (m_\Delta + m_\Upsilon) + \varepsilon$ and $y^1 = (\Phi \circ \Phi) \cdot \vartheta^1 / (m_\Delta + m_\Upsilon) + \varepsilon$, where ε is a white noise sampled from $\mathcal{N}(0, 0.1)$ and $\circ$ is the symbol for element-wise (Hadamard/Schur) product.

We considered all the viable datasets in a mesh generated by $m_\Gamma, m_\Delta, m_\Upsilon \in \{0, 4, 8\}$. This creates 24 scenarios[1] that consider all possible situations in

[1] There are not $2^3 = 27$ scenarios because we removed the three tuples: $(0,0,0)$, $(4,0,0)$, and $(8,0,0)$, as any scenario with $\Delta = \Upsilon = \emptyset$ would generate outcomes that are pure noise.

terms of the relative sizes of the factors Γ, Δ, and Υ. For each scenario, we synthesized several datasets with various initial random seeds.

3.2 Evaluation Criteria

There are two categories of performance measures for evaluating causal effect estimation algorithms: individual-based and population-based. Our main focus here is producing models with high individual-based performance, as measured by

"Precision in Estimation of Heterogeneous Effect" (PEHE) [38]

$$\text{PEHE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\text{e}_i - \hat{\text{e}}_i)^2} \tag{3.1}$$

and

"Effect-Normalized Root Mean Squared Error" (ENoRMSE) [54], [98]

$$\text{ENoRMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(1 - \frac{\hat{\text{e}}_i}{\text{e}_i}\right)^2} \tag{3.2}$$

where $\hat{\text{e}}_i = \hat{y}_i^1 - \hat{y}_i^0$ is the predicted effect and $\text{e}_i = y_i^1 - y_i^0$ is the true effect.

We also consider a population-based performance measure, namely,

Bias of the "Average Treatment Effect (ATE)"

$$\epsilon_{\text{ATE}} = \left|\text{ATE} - \widehat{\text{ATE}}\right| \tag{3.3}$$

where

$$\text{ATE} = \frac{1}{N}\sum_{i=1}^{N} y_i^1 - \frac{1}{N}\sum_{j=1}^{N} y_j^0 \tag{3.4}$$

in which y_i^1 and y_j^0 are the true outcomes for the treatment and control arms respectively. Note that we can calculate ATE here since we work with a synthetic dataset and so have access to both observed and counterfactual outcomes. $\widehat{\text{ATE}}$ is calculated based on the estimated outcomes.

Chapter 4

An Evaluation Methodology for Assessing Off-Policy Learning Methods in Contextual Bandits[1]

As described earlier, observational studies inherently contain partial information, as we only observe the outcome for the one treatment that was administered to each patient, based on a policy in-place for data collection (*i.e.*, logger policy; aka "behaviour policy" in reinforcement learning [101]). When evaluating a new learned policy, however, due to the variance of the utility estimators, it is challenging to know if the new proposed policy is indeed better than the logger policy.

The best way to determine the effectiveness of a new policy is to actually deploy it on-site, record the factual outcomes for a reasonable period of time, and then analyze the results. However, it is not neither ethical, nor allowed by the health-care community, to deploy a policy that has a chance of producing results that may reduce the patients' quality of life.[2] Therefore, we need to synthesize bandit datasets in such a way that their counterfactual outcomes are also known, merely for the purpose of evaluation.

The whole setup can be viewed in a *contextual bandit* setting [110] where, given a vector of attributes $x_i \in X$ describing patient i and her received treatment $t_i \in T$, we observe an outcome value $y(\, x_i,\, t_i\,) \in \mathbb{R}$. Treatment is selected according to an established clinical pathway represented by a conditional prob-

[1]The material of this chapter is taken from my Canadian AI 2018 paper [31].

[2]Similar scenarios hold for other applications such as finance, social welfare, etc.

ability distribution $\pi_0(\, t \,|\, x \,)$. Following the literature, we refer to this as the *logger policy*.

Also note that, for each patient, we only get to observe the outcome $y(\, x_i,\, t_i \,)$ associated with the *received* treatment t_i and not the outcomes associated with the alternative treatment(s) $t \neq t_i$. Since such *counterfactual* outcome(s) are inherently "unobservable" (and not just "unobserved") [41], there is no way to truly determine the best personalized treatment for each patient. As examples, the case of $y(\, x_i,\, t_i \,) \in \mathbb{R}^{\geq 0}$ might represent life expectancy after treatment, while $y(\, x_i,\, t_i \,) \in \{0, 1\}$ might indicate whether a patient would die within a week or not.

In general, such contextual bandit datasets have the following information: $\mathcal{D} = \{\, [x_i,\, t_i,\, y(\, x_i,\, t_i \,),\, \pi_0(\, t_i \,|\, x_i \,)] \,\}_{i=1..n}$. The goal here is to find the best policy, $\pi^*(\, t \,|\, x \,)$, whose most likely selected treatment $t^* = \arg\max_t \pi^*(\, t \,|\, x \,)$ for patient x, indeed matches the best one. In other words, had we known all outcomes (observed *and* counterfactual(s)), we want:

$$t^* \quad = \quad \arg\max_t y(\, x,\, t \,) \tag{4.1}$$

In the following two sections, we first review the existing evaluation methodology [8] and point out its shortcomings; and then explain our proposed evaluation methodology, which allows for a more comprehensive assessment of a proposed algorithm in terms of robustness to various degrees of selection bias.

4.1 The Existing Approach

Beygelzimer *et al.* [8] proposed an approach that converts the training partition of a full-information binary multi-label[3] supervised dataset $\mathcal{D}^* = \{\, [\,(x_i,\, t_i^*)] \,\}_{i=1..n}$ with $t_i^* \in \{0, 1\}^k$ into a partial-information bandit dataset for training off-policy learning methods.[4] They view each label t_i^* as the best possible treatment for patient i — *i.e.*, the outcome value-function $y(\, x_i,\, t_i \,)$ is defined such that $y(\, x_i,\, t_i^* \,) > y(\, x_i,\, \neg t_i \,)$, where $\neg t_i$ is any of the treatments other than

[3] These are not one-hot encoded as there may be instances with multiple associated labels — *e.g.*, a news article concerning political initiatives on climate change.

[4] Note that the test set remains intact for evaluating the learned policy.

t_i^*, as this ensures that Equation 4.1 holds, and so the optimal policy $\pi^*(t|x_i)$ will prefer t_i^*. One can convert this supervised dataset into a bandit dataset by sampling a set of new labels $t_i \sim h_0(\, t \,|\, x_i \,)$ for each x_i, where $h_0(\cdot)$ is the underlying mechanism that decides treatment assignments for this observational study. This allows a single subject to appear many ($r \geq 2^k$) times in the dataset, each time associated with a different treatment.

In many applications, such as ad-placement, the underlying treatment assignment mechanism (*i.e.*, the deployed algorithm / logger policy) is known [10]. To mimic the same situation, Swaminathan and Joachims [102] set $h_0(t|x)$ to be a logistic regression function, whose parameters are learned from a small portion (*e.g.*, 5%) of the supervised training set. This $h_0(\cdot)$ is then used to guide the sampling process of new labels t_i for each x_i, and record the propensities $\pi_0(t_i|x_i) = h_0(t_i|x_i)$. Finally, the outcome $y(x_i, t_i)$ is calculated as the Jaccard index between the supervised (true) label t_i^* and the bandit label(s) t_i(s). This completes the procedure of generating a bandit dataset $\mathcal{D} = \{[x_i, t_i, y(x_i, t_i), \pi_0(t_i|x_i)]\}_{i=1..n}$.

There are several reasons why this evaluation framework is not appropriate for assessing off-policy learning methods for medical observational studies:

1. It is not clear how to map the concept of binary multi-label $\in \{0, 1\}^k$ to treatment; *e.g.*, letting each bit in a label vector to be 1 (resp. 0) refer to taking (resp. not taking) a certain medication, a multi-label target would mean a combination of several drugs. However, due to drug interactions, such combinations might neutralize the effect of the treatment or worse, be detrimental to the patient's health. Therefore, unless there is a principled way to consider such interactions, a single class label seems more appropriate.

2. Using the Jaccard index to define outcomes implies assigning equal importance to various treatment options. However, this assumption does not hold in medicine since receiving the wrong treatment might be catastrophic for some cases while minor for others. Here, continuous measures such as *survival time* ($y \in \mathbb{R}^{\geq 0}$) that directly correspond to the conse-

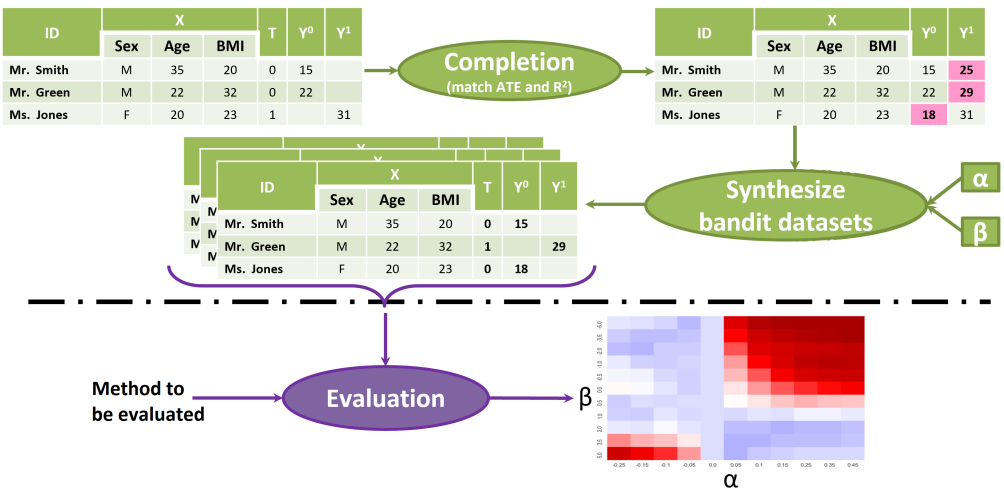


Figure 4.1: Pipeline of the proposed evaluation methodology

quences of the assigned treatment on the patient's health status seem more appropriate.

3. Unlike applications such as ad-placement, where the underlying mechanism of action selection is known, it may not be fully understood in medical observational studies (*e.g.*, clinical pathways). In reality, we never have access to (even a small) subset of data with ground truth labels. Hence, the propensities $\pi_0(\cdot)$ have to be calculated directly from the bandit dataset, as opposed to readily deriving them from $h_0(\cdot)$ (*i.e.*, estimated from 5% of supervised data).

4.2 The Proposed Approach

This section discusses our proposed evaluation methodology and its advantages over the existing approach. In addition to overcoming the shortcomings of the existing approach, we want to address the following requirements: (i) design a bandit dataset that is as realistic as possible in terms of similarity to an actual medical observational study (Section 4.2.1); and (ii) include a procedure to generate many different observational studies from a single RCT dataset to allow for comprehensive evaluation of learning methods for contextual bandits (Section 4.2.2). Figure 4.1 illustrates the pipeline of the proposed approach.

4.2.1 Designing a Bandit Dataset

We require that the designed bandit dataset be as similar as possible to a real medical observational dataset. Therefore, instead of converting a supervised dataset to a bandit dataset, we directly work with a real-world RCT dataset[5] as the source and from it synthesize various observational studies[6] with different degrees of selection bias. This makes sense because there is no selection bias in RCT datasets and therefore, one can often estimate the counterfactual outcomes reliably. In addition to the primary constraints (*e.g.*, $t \in \{0, 1\}$ and $y \in \mathbb{R}^{\geq 0}$), we want to preserve the statistical characteristics of the original (source) RCT dataset; characteristics such as:

(i) Average Treatment Effect: $ATE = \frac{1}{N_1} \sum y(x_i, 1) - \frac{1}{N_0} \sum y(x_i, 0)$, where N_1 (resp., N_0) is the number of subjects assigned to $t = 1$ (resp. $t = 0$); and

(ii) Coefficient of Determination: $R_t^2 = 1 - \frac{\sum [y(x_i,t) - \hat{y}(x_i,t_i)]^2}{\sum [y(x_i,t) - \bar{y}]^2}$, one for each treatment arms, where $\hat{y}(x_i, t_i)$ is the estimated outcome and $\bar{y}$ is the mean of y. Coefficient of Determination is calculated on each treatment arm separately and measures the amount of variance in the response variable that can be explained by the observed explanatory variables.[7]

Given a RCT dataset with two treatment arms (*i.e.*, $t \in \{0, 1\}$), we first fit two Gaussian Process (GP) [84] models $f_t(\cdot)$ based only on the observed outcomes; one for each treatment arm. More concretely, $f_t(x)$ provides a mean $\mu_t(x)$ along with a standard deviation $\sigma_t(x)$ that indicates the confidence of estimation at any point x in the function domain. We can now calculate the counterfactual outcome for each subject. For example, for a subject whose assigned treatment in the respective synthetic observational dataset was $t = 1$

[5] This means that the X values in the synthetic observational dataset would be realistic. By contrast, we do not know whether the X values from a supervised dataset look like realistic [medical] observational studies.

[6] These synthetic datasets have the same sample size as the original RCT data that we are working with.

[7] A low R^2 measure suggests that there must exist [some] unobserved covariate(s) that [significantly] contribute to the outcome.

(with $y(x_i, 1)$ as its observed outcome), we define the the counterfactual outcome as:

$$\widehat{y}(x_i, 0) = \mu_0(x_i) + k_0 \times \sigma_0(x_i) \tag{4.2}$$

where k_0 is determined such that the average *personalized* treatment effect calculated on the N_1 subjects who received treatment $t = 1$ (*i.e.*, $\widehat{ATE}_1 = \frac{1}{N_1} \sum_i \text{ s.t. } _{t_i=1} \big(y(x_i, 1) - \widehat{y}(x_i, 0)\big)$) matches the ATE calculated on the original RCT dataset. Solving for $\widehat{ATE}_1 = \widehat{ATE}_0 = ATE$ yields:

$$k_t = (2t-1)\Big(ATE - (2t-1)\frac{1}{N_{\neg t}} \sum \big(\mu_t(x_i) - y(x_i, \neg t_i)\big)\Big) \Big/ \frac{1}{N_{\neg t}} \sum \sigma_t(x_i) \tag{4.3}$$

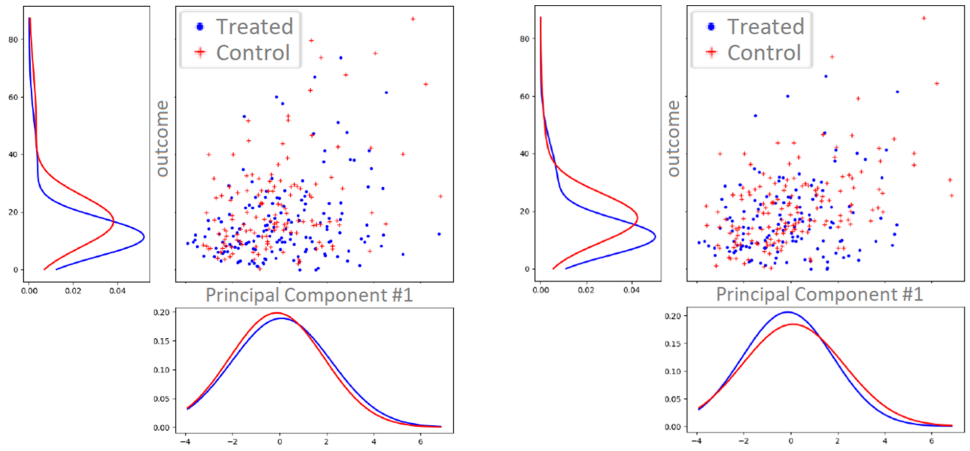
This procedure ensures that any synthetic RCT data generated by random re-assignment of treatments will have an $\widehat{ATE}$ close to the original ATE.

We also want our synthetic datasets to match the R_t^2 measure on every treatment arm t. To do so, we first calculate $\widehat{R}_t^2$ for each treatment arm t, on all subjects, using either observed, or counterfactual outcomes as derived in the previous step. Then, if the $\widehat{R}_t^2$ was higher than the original R_t^2 value, we modify the counterfactual outcomes by adding noise to them as follows:

$$\hat{y}(x_i, t) \quad += \quad e_t \times \epsilon_i \quad , \quad t \neq t_i \tag{4.4}$$

where e_t is the amplitude of the noise (tuned such that $\widehat{R}_t^2$ matches R_t^2) and $\epsilon \sim U(-0.5, 0.5)$. As $\mathbb{E}[\epsilon] = 0$, $\hat{y}(x_i, t)$'s expected increase is 0, and therefore we expect that $\widehat{ATE}$ would not change.

With this complete set of outcomes (observed as well as counterfactual), we can determine the best treatment for each patient (*i.e.*, ground truth labels), following Equation 4.1. It is also possible to synthesize any observational study (including RCT) by simply designing an appropriate $h_0(\cdot)$ function. Figures 4.2a and 4.2b respectively show the scatter plots of an original RCT dataset and a sample synthetic RCT generated from it, following the procedure described above. The next section explains how our proposed evaluation methodology can synthesize various observational studies, covering a wide range of selection bias.



(a) Original RCT (b) Synthetic RCT

Figure 4.2: The proposed method generates a synthetic RCT dataset (right) that is very similar to a real RCT dataset (left)

4.2.2 Various Generating Policies $h_0(\cdot)$

Unlike [102]–[104], our proposed evaluation methodology decouples the generating policy $h_0(\cdot)$ from the supervised dataset. This means we can easily design different $h_0(\cdot)$ policies with various degrees of selection bias and/or conservatism, which in turn enables us to study the behaviours/robustness of different learning algorithms under such various circumstances. In order to create a bandit dataset, our basis function for sampling labels (*i.e.*, treatments) is a sigmoid function:

$$\sigma(z) \quad = \quad \sigma_{\alpha,\beta}(z) \quad = \quad \frac{1}{1+e^{-\alpha(z-\beta)}} \tag{4.5}$$

where $z = y^1 - y^0$ which is positive for $t^* = 1$ class and negative for $t^* = 0$ class.[8] For instances in $t^* = 1$ class, the treatments t are then drawn according to $\sigma(z)$ via rejection sampling and for $t^* = 0$ class according to $1 - \sigma(z)$.

Parameter α in Equation 4.5 controls the degree of selection bias. With $\alpha = 0$, $\sigma(z)$ is a uniform distribution, which results in synthesizing a RCT dataset. Increased α creates a more biased dataset; at the limit of $\alpha = \infty$, $\sigma(z)$ becomes a step function at β. At $\beta = 0$, a larger α increases the chance that a sampled treatment t is equal to its respective ground truth label t^*. We

[8] In other words, $|z|$ is the amount of improvement in outcome for a patient in case she receives the best treatment.

can simulate the tendency towards prescribing a certain treatment more than the alternative(s) (*i.e.*, conservatism) by modifying β. As such, a $\beta > 0$ would assign treatment 0 to more patients and treatment 1 to fewer ones, and vice versa for $\beta < 0$.

4.3 Empirical Results and Discussions

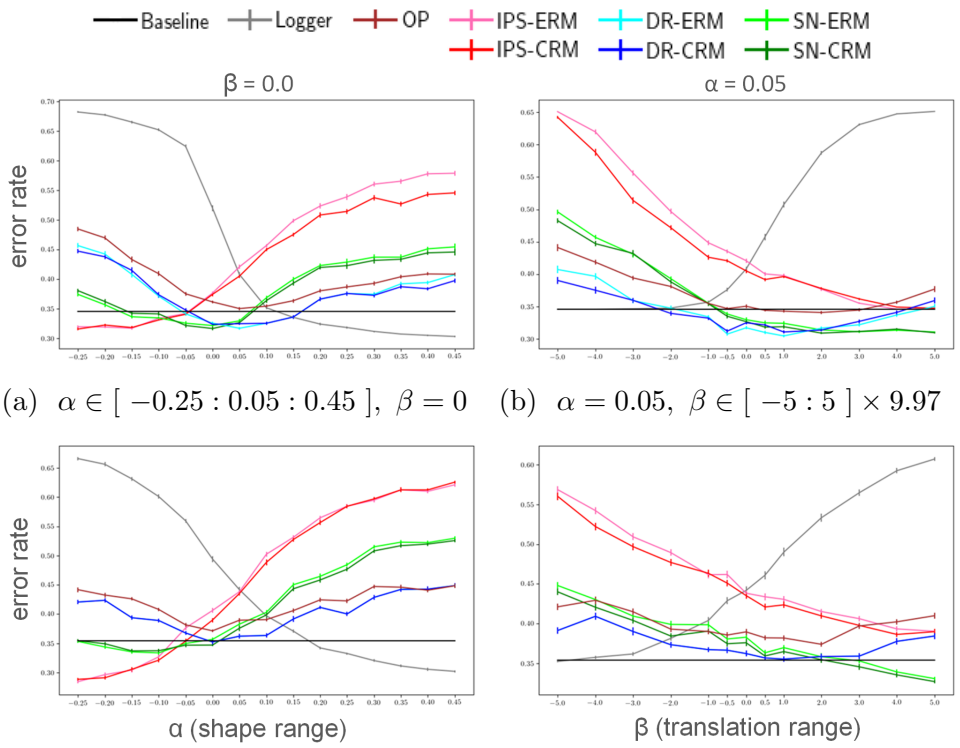
For the Acupuncture RCT [107], [108], $ATE = -6.15$, $R_0^2 = 0.68$, and $R_1^2 = 0.33$, while our synthesized RCTs has $\widehat{ATE} = -6.17(0.76)$[9], $\hat{R}_0^2 = 0.60(0.05)$, and $\hat{R}_1^2 = 0.36(0.07)$. For the Hypericum RCT [43], $ATE = -2.25$, $R_0^2 = 0.24$, and $R_1^2 = 0.00$. Our synthesized RCTs has $\widehat{ATE} = -2.65(0.97)$, $\hat{R}_0^2 = 0.15(0.10)$, and $\hat{R}_1^2 = 0.04(0.09)$.

The following methods are compared in terms of classification accuracy on the ground truth labels (*i.e.*, optimal treatment) derived following the procedure described in Section 4.2.1.

- **Baseline:** predict the majority class.
- **Logger:** use the logger policy $\pi_0(\cdot)$ as the classifier.
- **Outcome Prediction:** find a regression fit with a simple linear least squares method with L2 regularization (OP), then use Equation 4.1 to predict the best label (*i.e.*, treatment).
- **Inverse Propensity Scoring:** use the ERM objective function (IPS-ERM), as well as the CRM objective function (IPS-CRM) to learn a new policy $\pi(t|x)$ that acts as our classifier.[10]
- **Doubly Robust:** use the same regression function as OP for the regression component with either ERM (DR-ERM) or CRM (DR-CRM) objective functions to obtain $\pi(t|x)$.
- **Self-Normalized:** use either ERM or CRM objective functions (SN-ERM and SN-CRM respectively) to obtain $\pi(t|x)$.

[9] Each a(b) pair of numbers is mean(standard deviation).

[10] Our implementation of IPS (and SN below) is obtained from Policy Optimizer for Exponential Models (POEM [102]). We extended POEM substantially to include a way to deal with the missing components (*i.e.*, OP and DR), as well as implementation of the proposed evaluation methodology.



(a) $\alpha \in [\, -0.25 : 0.05 : 0.45 \,],\ \beta = 0$ (b) $\alpha = 0.05,\ \beta \in [\, -5 : 5 \,] \times 9.97$

(c) $\alpha \in [\, -0.25 : 0.05 : 0.45 \,],\ \beta = 0$ (d) $\alpha = 0.05,\ \beta \in [\, -5 : 5 \,] \times 6.01$

Figure 4.3: Mean and $\frac{1}{10}\times$standard deviation of the classification error rates on the "Acupuncture" (top) and "Hypericum" (bottom) datasets; best viewed in color.

Figure 4.4 summarizes the performance results; showing the effect of changing α at $\beta = 0$ in Figures 4.3a and 4.3c, and changing β at $\alpha = 0.05$ in Figures 4.3b and 4.3d. Each point on the plots represents the mean classification error rate across 25 runs and its respective error bar indicates a fraction (10%) of the standard deviation of the error rates (in order to maintain the plots' clarity). Also note that the Baseline accuracy for neither of the datasets is 1.0, meaning that not all patients benefit from receiving "acupuncture" or "SSRI"; indeed, for some, "no acupuncture" or "St. John's Wort" achieves a better outcome, *i.e.*, personalized medicine.

Effects of changing α

As α increases, it is trivial that the Logger's accuracy would improve since a higher α produces a bandit dataset with a higher tendency towards sampling the ground truth treatments more often. Moreover, OP's prediction of

(counter)factual outcomes is not accurate as α moves away from 0, resulting in a bad performance for DR as well. IPS's performance is also correlated with α and it tends to do worse as α increases, as opposed to that of Logger's. SN tends to perform worse as $|\alpha|$ increases since this imposes π_0 to be small in parts of its domain that, due to weight clipping, results in $\mathbb{E}\left[\sum_{i=1}^{n} \frac{\pi(t_i|x_i)}{\pi_0(t_i|x_i)}\right] \neq n$. Large $|\alpha|$ is however at odds with the fundamental idea of SN which suggests that SN is only useful for datasets that are close to RCT (*cf.*, Section 2.4.1).

Effects of changing β

We know that the effect of changing β varies relative to the degree of class imbalance in a dataset. In ours, since the majority class is labeled as "1", a $\beta > 0$ would result in assigning fewer samples with label $t = 1$ (hence the error rate of Logger keeps increasing as β increases). This, in turn, would generate a bandit dataset that is more exploratory (which SN seems to prefer); but, on the other hand, is far from the true underlying label distribution (which is not desirable for OP and DR). As a result, SN outperforms the rest of the methods in larger β values. DR closely follows for $0 \leq \beta \leq 2$ but diverges afterwards.

ERM versus CRM principle

We found that the CRM principle often improves the performance over ERM for all three IPS family of methods (*i.e.*, IPS, DR, and SN). Although these results are not statistically significant, it seems that the additional variance penalty is helpful in performance improvement.

The detailed results of changing α and β values on the performance of the contending methods are illustrated in Figures 4.4 and 4.5 for the Acupuncture and Hypericum datasets respectively, where blue (red) means a higher (lower) accuracy. In general, our results indicate that if a reliable OP method is available, then DR is the most effective and robust method for various α and β values. However, we should remember that most OP (and as a result DR) methods require more processing power than IPS and SN. Therefore, we face

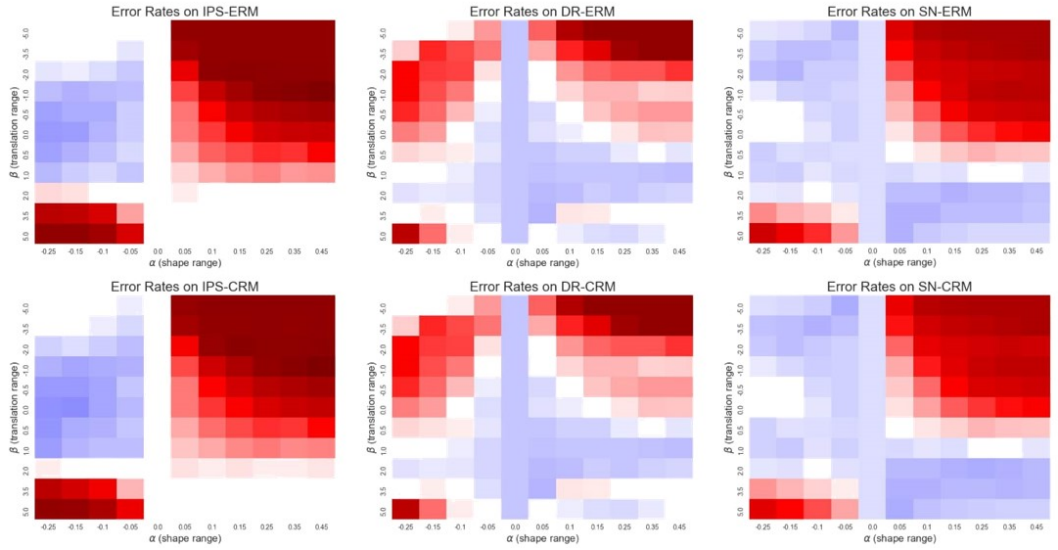


Figure 4.4: Detailed results on the Acupuncture dataset.

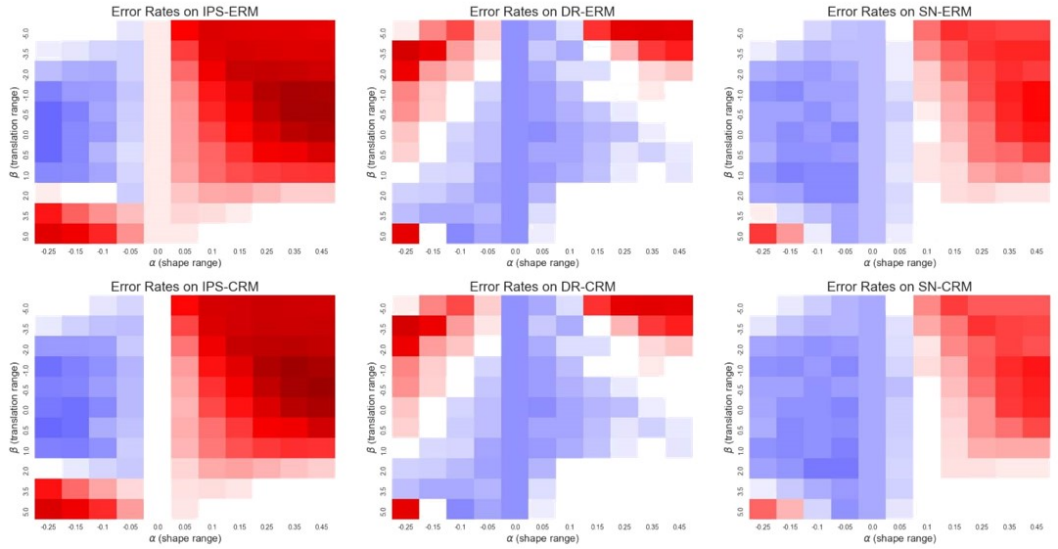


Figure 4.5: Detailed results on the Hypericum dataset.

a trade-off between a quick response versus a more accurate one. Overall, SN appears to perform well on a wider range of α and β values than the other methods.

4.4 Conclusion

In this chapter, we proposed a novel evaluation methodology for assessing off-policy learning methods in contextual bandits. Unlike the existing methodology (*cf.*, [8]), our approach allows for a comprehensive assessment of the learning methods in terms of performance and robustness with respect to various degrees of selection bias. Moreover, it does not require the underlying mechanism for data generation to be known, and it better matches medical applications as it allows the outcomes to be more realistic ($y \in \mathbb{R}^{\geq 0}$).

Using the proposed evaluation methodology, we assessed several prominent off-policy learning methods in contextual bandits — namely, outcome prediction, Inverse Propensity Scoring, Doubly Robust [21], Self-Normalized [104], and Counterfactual Risk Minimization principle [103] — on observational datasets synthesized using two RCT datasets. Our analyses identify the conditions under which a certain off-policy learning method performs best (*e.g.*, SN is preferable for a close-to-RCT dataset). Such analysis was not possible with [8]'s evaluation methodology as it has no means to generate such diverse observational datasets in terms of selection bias. Thus, we believe the proposed evaluation methodology should become a standard way for comprehensive assessment of new off-policy learning methods in contextual bandits, especially in costly applications such as precision medicine where deploying a bad policy can have devastating effects.

Chapter 5

Context-aware Importance Weighting for Counterfactual Regression[1]

As mentioned earlier in Chapter 1, there are two challenges associated with estimating ITEs:

(i) Training data never includes the counterfactual outcomes $y^{\neg t}$ for any training instances; which makes estimating causal effects a significantly different (and more complicated) problem than the common tasks in standard supervised machine learning.

(ii) Selection bias in observational datasets implies having fewer instances within each treatment arm at some specific regions of the domain. This sparsity, in turn, would decrease the accuracy and confidence of estimating the counterfactual outcomes at those regions.

The first challenge is an inherent characteristic of this task. We focus on the following ways to mitigate the second challenge:

- **Representation learning** [7] — The idea here is to learn a representation space $\Phi(\cdot)$ in which the selection bias is reduced as much as possible but not at the expense of a decrease in accuracy of predicting the observed outcomes. In other words, assuming X is generated from three

[1]The material of this chapter is taken from my IJCAI 2019 paper [32].

non-noise underlying factors as shown in Figure 1.3, this would ideally be conducted by identifying $\{\Gamma, \Delta, \Upsilon\}$ factors and then removing Γ.

- **Re-weighting** — This is a common statistical method for addressing covariate shift [97] and domain adaptation in general. It is easy to show that selection bias in observational studies translates into a domain adaptation scenario where we want to learn a model from the "source" (observed) data distribution that will perform well in the "target" (counterfactual) one.

Main contribution: In this chapter, we propose a new context-aware weighting scheme based on importance sampling technique, on top of a representation learning module, to alleviate the problem of selection bias in ITE estimation.

Our analysis relies on the following assumptions, repeated from Chapter 2, Section 2.1.2: (i) **SUTVA** (potential outcome of one unit should be unaffected by treatment assignment to other units), (ii) **unconfoundedness** (there are no unobserved confounders), and (iii) **overlap** (every individual x should have a non-zero chance of being assigned to any treatment arm).

5.1 The Existing Approach

Shalit *et al.* [95] attempt to reduce selection bias by learning a common representation space $\Phi(\,\cdot\,)$ that tries to make $\Pr(\,\Phi(x)\,|\,t=0\,)$ and $\Pr(\,\Phi(x)\,|\,t=1\,)$ as close to each other as possible[2] (see Figure 2.4), provided that $\Phi(\,x\,)$ retains enough information that all $|\,\mathcal{T}\,|$ learned regressors $h^t(\,\Phi\,)$ can generalize well on the observed outcomes. Φ and h^t are implemented as neural networks and learned by minimizing:

$$\begin{aligned} J(h,\Phi) \;&= \frac{1}{N}\sum_{i=1}^{N} \omega_i \cdot L\big[\; y_i,\, h^{t_i}\big(\Phi(x_i)\big)\;\big] + \lambda \cdot \mathfrak{R}(h) \\ &\quad + \alpha \cdot \mathrm{IPM}\big(\{\Phi(x_i)\}_{i:t_i=0}, \{\Phi(x_i)\}_{i:t_i=1}\big) \end{aligned} \tag{5.1}$$

[2]Johansson *et al.* [51], however, show in their domain-invariant representations work that matching densities might be too strong of a constraint and that having overlap in support (*i.e.*, both are non-zero for much of the domain) should suffice.

where $L\big[\ y_i,\, h^{t_i}\big(\Phi(x_i)\big)\ \big]$ is the loss of predicting the observed outcome for sample i, weighted by ω_i, derived via:

$$\omega_i \;=\; \frac{t_i}{2u} \;+\; \frac{1-t_i}{2(1-u)} \tag{5.2}$$

where $u = \frac{1}{N}\sum_{i=1}^{N} t_i \;=\; \Pr(\,t=1\,)$. Also, $\mathfrak{R}(h)$ in Equation (5.1) is the regularization term for penalizing model complexity, and the final term

$$\texttt{disc} \;=\; \mathrm{IPM}\big(\{\Phi(x_i)\}_{i:t_i=0}, \{\Phi(x_i)\}_{i:t_i=1}\big) \tag{5.3}$$

is the *discrepancy* — calculated by an Integral Probability Metric (IPM) — that measures the distance between the two distributions $\Pr(\,\Phi(\,x\,)\,|\,t=0\,)$ and $\Pr(\,\Phi(\,x\,)\,|\,t=1\,)$. See Figure 5.1 for [95]'s model architecture.

Shalit *et al.* [95]'s model is closely related to its predecessor [49], which defined `disc` between the joint distributions of Φ and t (factual) versus Φ and $\neg t$ (counterfactual) — *i.e.*,

$$\texttt{disc} \;=\; \mathrm{IPM}\Big(\big\{[\,\Phi(x_i), t_i\,]\big\}_{i=1}^{N}, \big\{[\,\Phi(x_i), \neg t_i\,]\big\}_{i=1}^{N}\Big) \tag{5.4}$$

This makes sense in theory: if the factual and counterfactual joint distributions are hard to distinguish, it means that the data is close to RCT. However, since the two joint distributions only differ in their treatment bit (*i.e.*, t versus $\neg t$, while $\Phi(\,x\,)$ is the same for both), the numerical value of `disc` would naturally be small. Therefore, its contribution to the objective would be negligible. Moreover, a high dimensional $\Phi(\,\cdot\,)$ can overshadow the information in the treatment bit, which results in an even smaller `disc`.

Perhaps the work most related to ours is the work by Johansson *et al.* [50], which also applies sample re-weighting on top of representation learning to balance their source and target domains by minimizing `disc` between the factual joint distribution $p_\mu(\Phi(x), t)$ and a weighted $(\,\omega\,)$ counterfactual one $\omega \cdot p_\pi(\Phi(x), \neg t)$. However, this method is also susceptible to and suffers from the same issue with small `disc` as discussed above.

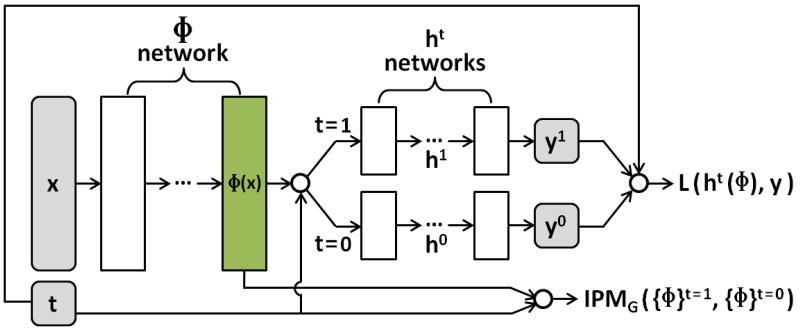


Figure 5.1: Shalit *et al.* [95]'s model architecture.

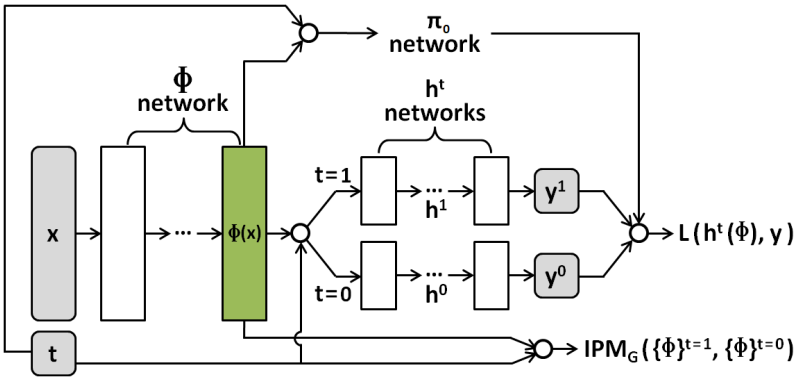


Figure 5.2: Our proposed model named CounterFactual Regression with Importance Sampling Weights (CFR-ISW). Note the addition of the propensity network (for π_0 — in the top right) in our method versus that of [95] (see Figure 5.1).

5.2 The Proposed Approach

Observe that $J(h, \Phi)$'s first term in Equation (5.1) tries to minimize a weighted sum of the factual losses — *i.e.*, a standard supervised machine learning objective. We can re-write this term as:

$$\begin{aligned}
&\frac{1}{N}\sum_{i=1}^{N}\omega_i \cdot L\big[\, y_i,\, h^{t_i}\big(\Phi(x_i)\big)\,\big] \\
&\quad = \frac{1}{N}\sum_{t\in\mathcal{T}} N_t \frac{1}{N_t}\sum_{j=1}^{N_t}\omega_j \cdot L\big[\, y_j,\, h^{t}\big(\Phi(x_j)\big)\,\big] \\
&\quad = \sum_{t\in\mathcal{T}} \hat{\Pr}(\,t\,)\frac{1}{N_t}\sum_{j=1}^{N_t}\omega_j \cdot L\big[\, y_j,\, h^{t}\big(\Phi(x_j)\big)\,\big] \qquad (5.5)
\end{aligned}$$

where N_t is the number of instances assigned to the treatment arm $t \in \{0, 1\}$.

Using Equation (5.2), Shalit *et al.* [95] is basically setting $\omega_i = \frac{1}{2\,\hat{\Pr}(\,t_i\,)}$, where $\hat{\Pr}(\,t_i\,)$ is simply the observed probability of using the treatment $t_i \in \{0, 1\}$ over the entire population. This effectively reduces the loss term in Equation (5.5) to the macro-average

$$\frac{1}{2}\sum_{t\in\mathcal{T}}\frac{1}{N_t}\sum_{j=1}^{N_t} L\big[\, y_j,\, h^{t_j}\big(\Phi(x_j)\big)\,\big] \qquad (5.6)$$

In other words, different treatment arms contribute equally to the objective, irrespective of their sample size. This somewhat makes sense since, at test time, we want to estimate the outcomes of *all* possible treatments.

Such weights, however, do not account for the remaining selection bias in $\Phi(\,x\,)$ due to the presence of confounding factors Δ (see Figure 1.3).[3] In our work, inspired by the importance sampling technique, we propose *context-aware* weights that incorporate the valuable context information of each instance $\Phi(\,x\,)$, thus further mitigating the impact of selection bias on estimating ITEs.

[3]The `disc` term tries to balance the two distributions by pushing to eliminate factors Γ and Δ from Φ, while the factual loss term fights to keep Δ in Φ. Due to this trade-off, we anticipate that Φ will learn to eliminate Γ and keep Δ and Υ. Note it is critical that Φ includes Δ as it contributes to accurately predicting the outcome (y) and is critical to correctly modeling the un-removable part of selection bias.

Importance sampling is used to compute $\mathbb{E}_{x\sim p(x)}\big[f(x)\big]$ when in fact we observe samples that are drawn from an alternative distribution $q(x)$, where p and q are called the "nominal" and "importance" distributions respectively. It is easy to show that:

$$\mathbb{E}_{x\sim p(x)}\big[f(x)\big] = \mathbb{E}_{x\sim q(x)}\big[f(x)\frac{p(x)}{q(x)}\big] \tag{5.7}$$

Proof Here, we want to show $\mathbb{E}_{x\sim p(x)}\big[f(x)\big] = \mathbb{E}_{x\sim q(x)}\big[f(x)\frac{p(x)}{q(x)}\big]$, where p and q are probability density functions defined on $\mathbb{R}^d$, with an assumption on p that every sample x within its support $\mathcal{D}$ should have a non-zero probability and zero otherwise (*i.e.*, $p(x) \neq 0$ for $x \in \mathcal{D}$ and $p(x) = 0$ for $x \in \mathcal{D}^c$), and another assumption on q that we define its support $\mathcal{Q}$ (*i.e.*, $q(x) \neq 0$ for $x \in \mathcal{Q}$) wherever $f(x)p(x) \neq 0$; then:

$$\begin{aligned}
&\mathbb{E}_{x\sim q(x)}\big[f(x)\frac{p(x)}{q(x)}\big] \\
&\quad= \int_{\mathcal{Q}} \frac{f(x)p(x)}{q(x)} q(x)\mathrm{d}x \\
&\quad= \int_{\mathcal{D}} f(x)p(x)\mathrm{d}x + \int_{\mathcal{D}^c\cap\mathcal{Q}} f(x)p(x)\mathrm{d}x - \int_{\mathcal{D}\cap\mathcal{Q}^c} f(x)p(x)\mathrm{d}x \\
&\quad= \int_{\mathcal{D}} f(x)p(x)\mathrm{d}x \quad = \quad \mathbb{E}_{x\sim p(x)}\big[f(x)\big]
\end{aligned}$$

since the second integral term is zero because $p(x) = 0$ for $x \in \mathcal{D}^c \cap \mathcal{Q}$ and the third integral is zero because $f(x) = 0$ for $x \in \mathcal{D} \cap \mathcal{Q}^c$. □

In the task of ITE estimation, we have a similar problem. Therefore, we need to first identify the importance distribution that generated the data, then design a nominal distribution that helps improve the performance.

Re-visiting Equation (5.5), our solution strategy is to learn an independent regression function $h^t\big(\,\Phi(\,x\,)\,\big)$ for each treatment arm $t \in \{0, 1\}$ that predicts the outcome of the respective treatment t for subject x. By decoupling the weights from $J(h, \Phi)$'s parameters via setting $\phi = \Phi(\,x\,)$, we arrive at the following belief net: $t \leftarrow x \rightarrow \phi \rightarrow \{y^1, y^0\}$. The importance distribution of $L\big[\,y,\, h^t(\,\phi\,)\,\big]$ is then:

$$\Pr(\,y, \phi \,|\, t\,) \;=\; \Pr(\,y \,|\, \phi\,) \cdot \Pr(\,\phi \,|\, t\,)$$

We choose $\Pr(\, y, \phi \,|\, \neg t \,)$ as our nominal distribution in order to emphasize those instances that are important for predicting accurate **counterfactual** outcomes. This yields the likelihood ratio of

$$\frac{\Pr(\, y,\, \phi \,|\, \neg t \,)}{\Pr(\, y,\, \phi \,|\, t \,)} = \frac{\Pr(\, y \,|\, \phi \,) \cdot \Pr(\, \phi \,|\, \neg t \,)}{\Pr(\, y \,|\, \phi \,) \cdot \Pr(\, \phi \,|\, t \,)} = \frac{\Pr(\, \phi \,|\, \neg t \,)}{\Pr(\, \phi \,|\, t \,)} \tag{5.8}$$

Moreover, to ensure that our model also performs well on the observed instances (associated with t_i), we add $\frac{\Pr(\, \phi_i \,|\, t_i \,)}{\Pr(\, \phi_i \,|\, t_i \,)} = 1$ to the derived likelihood ratio so that our objective accounts for the **factual** loss as well. Our weights would then be:

$$\omega_i \;=\; 1 \;+\; \frac{\Pr(\, \phi_i \,|\, \neg t_i \,)}{\Pr(\, \phi_i \,|\, t_i \,)} \tag{5.9}$$

Note these ω_i weights depend on ϕ_i whose numerical values are derived from $\Phi(\, x_i \,)$. This means that estimating these weights adds a nested optimization loop (for learning the $\omega(\,\cdot\,)$ parameters) within the main optimization loop (for learning the $\Phi(\,\cdot\,)$ and $h^t(\,\cdot\,)$ parameters). This motivates us to devise an efficient method for learning the weights. In this sense, learning the weights directly is not desirable because:

- It requires fitting two density functions, $\Pr(\, \phi \,|\, t \,)$ and $\Pr(\, \phi \,|\, \neg t \,)$ that doubles the necessary computations.
- Efficient approximations, such as fitting simple multivariate Gaussians, are anticipated to yield inaccurate densities.[4]
- More flexible solutions, such as fitting Gaussian mixture models, are of high computational complexity.

To circumvent these issues, we use the Bayes theorem to learn $\Pr(\, \phi \,|\, t \,)$ indirectly from $\pi_0\big(t \,|\, \phi\big)$ — *i.e.*, probability of selecting the assigned treatment t given the context ϕ — which can be efficiently estimated by fitting a Logistic Regression (LR) model. Here, the counterfactual part of our proposed weight

[4]Since the data generating process is rarely based on a simple multivariate Gaussian.

function can be simplified as follows:

$$\frac{\Pr(\,\phi_i \,|\, \neg t_i\,)}{\Pr(\,\phi_i \,|\, t_i\,)} \;=\; \frac{\dfrac{\pi_0\big(\neg t_i \,|\, \phi_i\big)\cdot \Pr(\,\phi_i\,)}{\Pr(\,\neg t_i\,)}}{\dfrac{\pi_0\big(t_i \,|\, \phi_i\big)\cdot \Pr(\,\phi_i\,)}{\Pr(\,t_i\,)}}$$

$$= \frac{\Pr(\,t_i\,)}{\Pr(\,\neg t_i\,)}\cdot\frac{\pi_0\big(\neg t_i \,|\, \phi_i\big)}{\pi_0\big(t_i \,|\, \phi_i\big)} \;=\; \frac{\Pr(\,t_i\,)}{1-\Pr(\,t_i\,)}\cdot\frac{1-\pi_0\big(t_i \,|\, \phi_i\big)}{\pi_0\big(t_i \,|\, \phi_i\big)} \tag{5.10}$$

where $\pi_0\big(t \,|\, \phi\big)$ is parametrized by LR with $[\,W, b\,]$ as:

$$\pi_0\big(t \,|\, \phi\big) \;=\; \frac{1}{1+e^{-(\,2t-1\,)(\,\phi\cdot W+b\,)}}$$

and parameters $[\,W, b\,]$ are learned by minimizing:

$$C(W,b) \;=\; \frac{1}{N}\sum_{i=1}^{N} -\log\big[\,\pi_0\big(t_i \,|\, \phi_i\big)\,\big] \tag{5.11}$$

Since π_0 depends on Φ, we update $[\,W, b\,]$ with every update of the parameters of Φ and h. Hence, this is a multi-objective optimization problem with two objectives — *i.e.*, Equations (5.1) and (5.11) — that we try to solve alternatingly. That is, each training iteration consists of two steps:

(i) Minimizing Equation (5.1) using stochastic gradient descent to update the parameters of the representation and hypothesis networks — *i.e.*, U and V (see Algorithm 1). Note that ω_is in the factual loss term are calculated based on Equations (5.9) and (5.10), with parameters W and b held fixed during optimization.

(ii) Minimizing Equation (5.11) to update parameters of the propensity score function $\pi_0\big(t \,|\, \phi\big)$ — *i.e.*, W and b — with parameters U and V held fixed.

Algorithm 1 describes this procedure in more detail. Note that both objective functions are computed for one mini-batch at a time. Figure 5.2 illustrates our network architecture.

5.2.1 Intuition of the Proposed Weighting Scheme

To illustrate the idea (in a trivialized fashion), imagine subject S received treatment $T0$, but his 10 clones $\{S_1, \ldots, S_{10}\}$ were each observed to receive

Algorithm 1 CFR-ISW: CounterFactual Regression with Importance Sampling Weights

Input: Factual samples $\{[x_1, t_1, y_1], ..., [x_N, t_N, y_N]\}$, batch size m, scaling parameter $\alpha > 0$, regularization parameter $\lambda > 0$, loss function $L(\cdot, \cdot)$, representation network Φ_U with initial weights $[U]$, outcome networks $h_V^{\{0,1\}}$ with initial weights $[V]$, function family for IPM, propensity network π_0 with initial weights $[W, b]$, and limit on the total number of iterations I.
Estimate probabilities $\hat{\Pr}(t)$ for $t \in \{0, 1\}$
for $iter = 1$ **to** I **do**
  Sample mini-batch $\{i_1, i_2, ..., i_m\} \subset \{1, 2, ..., N\}$
  Calculate the gradient of the discrepancy term:
    $g_d = \nabla_U \text{IPM}(\{\Phi_U(x_{i_j})\}_{t_{i_j}=0}, \{\Phi_U(x_{i_j})\}_{t_{i_j}=1})$
  Calculate the proposed importance sampling weights ω_{i_j} from W, b, and $\hat{\Pr}(t)$ following Equation (5.10)
  Calculate the gradients of the empirical loss:
    $g_U = \nabla_U \frac{1}{m} \sum_j \omega_{i_j} \cdot L\big[h_V^{t_{i_j}}\big(\Phi_U(x_{i_j})\big), y_{i_j}\big]$
    $g_V = \nabla_V \frac{1}{m} \sum_j \omega_{i_j} \cdot L\big[h_V^{t_{i_j}}\big(\Phi_U(x_{i_j})\big), y_{i_j}\big]$
  Obtain step size scalar or matrix η_1 with standard neural net methods (*e.g.*, `Adam` [55])
  Update weights of the representation and hypothesis networks:
    $[U, V] \leftarrow \big[U - \eta_1(\alpha g_d + g_U),\ V - \eta_1(g_V + 2\lambda V)\big]$
  Calculate gradients of the propensity network's cost function:
    $g_W = \nabla_W \frac{1}{m} \sum_j \log\big[1 + e^{-(2t_{i_j}-1)(\Phi_U(x_{i_j})\cdot W + b)}\big]$
    $g_b = \nabla_b \frac{1}{m} \sum_j \log\big[1 + e^{-(2t_{i_j}-1)(\Phi_U(x_{i_j})\cdot W + b)}\big]$
  Obtain $\eta_2 \in \mathbb{R}^+$ according to its decay scheduling function
  Update the propensity network's weights:
    $[W, b] \leftarrow [W, b] - \eta_2[g_W, g_b]$
end for
Output: $[U, V]$

treatment $T1$. How much should we weight our estimate of $h^{T0}(S)$? One component is based on the fact that we observed $[S,\ T0]$, which should contribute — *i.e.*, $\Pr(\Phi(S) \,|\, T0)$. But later, to estimate the ITE for each clone S_i, our algorithm will want to know what would have happened had S_i received $T0$. In this situation, that would also be $h^{T0}(S)$. Hence, the weight should also include the density of instances that look like S, but received the other treatment — *i.e.*, $\Pr(\Phi(S) \,|\, T1)$ — which here would be based on the 10 clones S_i. Of course, the real situation is much more complicated, as we will not typically have exact clones. In general, this suggests that the weight associated

with observing $[\phi_i,\ t_i]$ should be $\Pr(\,\phi_i \,|\, t_i\,) + \Pr(\,\phi_i \,|\, \neg t_i\,)$, normalized in the expectation by dividing by $\Pr(\,\phi_i \,|\, t_i\,)$.

5.3 Experiments

As mentioned earlier, an inherent characteristic of causal inference datasets is that counterfactual outcomes are unobservable, which makes it difficult to evaluate any proposed algorithm. The common solution in the literature is to synthesize datasets where the outcomes of all possible treatments are available. Some entries are then discarded in order to create a proper observational dataset with characteristics (such as selection bias) similar to a real-world one — see for example [31] and [8]. To make performance comparison easier, however, we do not synthesize our own datasets here. Instead, we use two publicly available benchmarks — see Section 5.3.2.

5.3.1 Hyperparameter Selection

As counterfactuals are unobserved, it is impossible for our learning algorithm to perform standard internal cross-validation, to set the hyperparameters. Therefore, our learner needs to obtain some estimate $\hat{\mathrm{e}}_i$ of the true effect $\mathrm{e}_i = y_i^1 - y_i^0$, so that it can calculate a surrogate for its desired performance measure. Shalit *et al.* [95] estimated the outcome of $y(x_i, \neg t_i)$ as the observed outcome $y_{j(i)}^{\neg t_i}$, where $j(i)$ is the nearest neighbor of x_i who received treatment $\neg t_i$ (*i.e.*, 1-NN based on a distance metric defined on the original x space). The surrogate effect would then be $\hat{\mathrm{e}}_{\text{1-NN}} = (2t_i - 1)(y_i^{t_i} - y_{j(i)}^{\neg t_i})$.

However, as our empirical results also confirm, this method is quite unlikely to select good hyperparameters. This is expected since, due to selection bias, the nearest neighbor $j(i)$ in the alternative treatment arm might not be a good enough representative of the counterfactual outcome. Hence, its estimated surrogate effect might not be reliable for finding the best set of hyperparameters.

A better solution is to employ a stronger counterfactual regression method such as Bayesian Additive Regression Trees (BART) [15]. BART is particularly

Parameter name	Range
Imbalance parameter α	1E{-2, -1, 0, 1}
Num. of representation layers	{3, 5}
Num. of hypothesis layers	{3, 5}
Dim. of representation layers	{50, 100, 200}
Dim. of hypothesis layers	{50, 100, 200}
Batch size	{100, 300}

Table 5.1: Hyperparameters and ranges

desirable since it is not very sensitive to various set of hyperparameters, and that a default one would work effectively (as mentioned in Section 2.3.2). This is interesting because, even though our empirical results (see Section 5.3.2) show that BART's performance is not as good as either CFR or CFR-ISW, $\hat{e}_{\text{BART}}$ identifies a set of hyperparameters (via $\text{PEHE}_{\text{BART}}$ or $\text{ENoRMSE}_{\text{BART}}$) that are better than $\hat{e}_{\text{1-NN}}$.

We trained CFR-ISW's π_0 logistic regression function with gradient descent optimizer and a learning rate of 1E-3. For both CFR and CFR-ISW, we trained the Φ and h^t networks with regularization coefficient λ=1E-3, `elu` [16] as the non-linear activation function, `Adam` optimizer [55], learning rate of 1E-3, and maximum number of iterations of 3000. We used the Maximum Mean Discrepancy (MMD) [26] as our IPM to calculate `disc` between the $\Pr(\Phi \mid t=1)$ and $\Pr(\Phi \mid t=0)$ distributions. See Table 5.1 for details on our hyperparameter search space.

5.3.2 Results and Discussion

In this chapter, we empirically compare the proposed CFR-ISW with the following ITE estimation methods[5]:

- 1-NN: One Nearest Neighbor method (as described in Section 5.3.1) — the baseline.
- BART: Bayesian Additive Regression Trees method [15].

[5]While these are only a subset of the many methods in the literature, Table 1 of [95] establishes that CFR outperforms several notable ones such as Random Forest [11], Causal Forest [109], and Targeted Maximum Likelihood Estimation [27].

- CFR: CounterFactual Regression method [95].
- RCFR: Re-weighted CFR [50].[6]

Table 5.2 reports ENoRMSE, PEHE, and ϵ_{ATE} performances of the considered methods on the IHDP benchmark with 1000 datasets. Our results show that CFR-ISW outperforms CFR and RCFR in PEHE and ϵ_{ATE} evaluation measures (showcasing the positive contribution of the proposed context-aware weights), no matter whether the hyperparameter selection is done according to $\hat{\text{e}}_{\text{BART}}$ or $\hat{\text{e}}_{\text{1-NN}}$. However, note that $\hat{\text{e}}_{\text{BART}}$ selects better hyperparameters than $\hat{\text{e}}_{\text{1-NN}}$: compare PB and P1 rows (PEHE of 0.55 and 0.77 respectively as well as ENoRMSE of 1.87 and 2.65 respectively). Also note that we should use a proper surrogate measure for hyperparameter selection depending on the performance measure that we would like to optimize — compare PB and EB rows (*e.g.*, ENoRMSE of 2.50 and 0.88 respectively for CFR-ISW). This is expected, since, there is no way to encode such a criterion in the objective function that is being optimized.

For each of the 24 datasets of the ACIC'18 benchmark, we have access to both factual and counterfactual tables. For each subject, factual tables contain the treatment bit and the respective observed outcome. Counterfactual tables (only to be used for evaluation purpose) contain the true outcomes $\{y^0,\ y^1\}$ for treatments 0 and 1 respectively. For each synthetic dataset, a Data Generating Process (DGP) determines t, y^0, and y^1 for each sampled x instance. The challenge organizers have not revealed the DGPs they used. Here, we look at two evaluation measures: (i) the aggregated ENoRMSE for datasets with the same number of instances (*i.e.*, A_n for $n \in S = \{1, 2.5, 5, 10, 25, 50\} \times 10^3$), where S is the set of different dataset sizes; and (ii) the aggregated ENoRMSE of all the 24 datasets (*i.e.*, A). A_n and A respectively are calculated as follows:

$$A_n = \sqrt{\frac{1}{|D_n|} \sum_{i \in D_n} \big[\text{ENoRMSE}(i)\big]^2}\ ,\ A = \sqrt{\frac{1}{\sum_{n \in S} n} \sum_{n \in S} n A_n^2}$$

where D_n is set of all datasets that have n instances.

[6] As RCFR's code is unavailable, we are limited in comparing its performance against contenders to what is reported in their paper.

	Methods	ENoRMSE	PEHE	ϵ_{ATE}
	1-nn	24.6 (189)	4.85 (6.29)	0.67 (1.27)
	BART	2.13 (11.3)	1.57 (2.41)	0.22 (0.30)
	CFR†		0.78 (0.0)	0.31 (0.01)
	RCFR‡		0.65 (0.04)	
P1	**CFR**	**2.65** (1.67)	0.88 (0.10)	0.20 (0.03)
P1	**CFR-ISW**	3.82 (3.17)	**0.77** (0.10)	**0.19** (0.03)
PB	**CFR**	**1.87** (1.29)	0.65 (0.05)	0.21 (0.03)
PB	**CFR-ISW**	2.50 (2.05)	**0.55** (0.05)	**0.20** (0.03)
EB	**CFR**	1.18 (0.29)	0.84 (0.07)	0.23 (0.03)
EB	**CFR-ISW**	**0.88** (0.29)	**0.66** (0.05)	**0.16** (0.02)

Table 5.2: ENoRMSE, PEHE, and ϵ_{ATE} performance measures (lower is better), each of the form "mean (standard error)" on the **IHDP** benchmark. Symbols † and ‡ indicate results reported in [95] and [50] respectively. Rows P1, PB, and EB report results of our runs for CFR and CFR-ISW whose hyperparameters were selected based on $\text{PEHE}_{\text{1-NN}}$, $\text{PEHE}_{\text{BART}}$, and $\text{ENoRMSE}_{\text{BART}}$ respectively. Comparing CFR-ISW with CFR, entries in **bold** indicate the best performance in each category.

Table 5.3 summarizes the macro-average performances of the four methods on the ACIC'18 datasets in terms of aggregated ENoRMSE. Our empirical results indicate that incorporating the proposed context-aware importance sampling weights into the network's objective function improves the aggregated ENoRMSE on almost all dataset categories by a large margin (except for $25k$ in which CFR-ISW's performance is very close to that of CFR). We also computed the micro-average performances (not shown here) which confirms that, as expected, CFR-ISW outperforms CFR in almost all categories as well.

5.4 Conclusion

In this work, we proposed a context-aware importance sampling weighting scheme that helps mitigate the negative effect of selection bias on the accuracy of models that estimate Individual Treatment Effects (ITEs). Additionally, we proposed a hyperparameter selection procedure, which plays an important

Datasets		1-nn	BART	CFR	CFR-ISW
	All	54.56	9.35	5.43 (5.78)	**1.03** (0.27)
# Instances	**1** k	66.70	73.66	7.08 (8.97)	**1.54** (0.87)
	2.5 k	33.31	15.12	8.33 (14.78)	**0.68** (0.31)
	5 k	31.89	8.15	2.00 (2.28)	**0.88** (0.35)
	10 k	31.46	2.60	0.86 (1.00)	**0.74** (0.39)
	25 k	19.47	1.27	**0.85** (0.30)	1.00 (0.28)
	50 k	75.43	12.27	8.23 (8.63)	**1.13** (0.23)

Table 5.3: Aggregated ENoRMSE (lower is better) on the **ACIC'18** benchmark. Model hyperparameters for both CFR and CFR-ISW methods are selected according to $\text{ENoRMSE}_{\text{BART}}$. Comparing CFR-ISW with CFR, entry in **bold** indicates the best performance.

role in determining the model performance. The proposed improvements were applied to the CounterFactual Regression (CFR) framework [95], leading to our method: CFR with Importance Sampling Weights (CFR-ISW).

We evaluated CFR-ISW against 1-NN (baseline), Bayesian Additive Regression Trees (BART) [15], and the state-of-the-art methods CFR [95] and Re-weighted CFR [50] on two publicly available synthetic benchmarks: (i) Infant Health and Development Program (IHDP) [38] and (ii) Atlantic Causal Inference Conference 2018 (ACIC'18) data challenge [98]. The empirical results demonstrated that CFR-ISW outperforms all the contender methods in terms of three common measures of performance for estimating causal effects, namely: Precision in Estimation of Heterogeneous Effect (PEHE), Effect-Normalized Root Mean Squared Error (ENoRMSE), and bias of the Average Treatment Effect (ϵ_{ATE}).

Chapter 6

Disentangling the Underlying Factors of an Observational Study[1]

This chapter attempts to address the second challenge of causal effect estimation by investigating the root causes of selection bias, by dissecting and identifying the underlying factors that can generate an observational dataset $\mathcal{D}$, and leveraging this knowledge to reduce, as well as account for, the negative impact of selection bias on estimating the treatment effects from $\mathcal{D}$. In this work, we borrow ideas from the representation learning literature [7] in order to reduce selection bias and from the domain adaptation literature [97] in order to account for the remainder selection bias that (might) still exist after its reduction. Our analysis relies on the assumptions stated in Chapter 2, Section 2.1.2.

Without loss of generality, we assume that the random variable X follows a(n unknown) joint probability distribution $\Pr(X \,|\, \Gamma, \Delta, \Upsilon, \Xi)$, treatment T follows $\Pr(T \,|\, \Gamma, \Delta)$, and outcome Y^T follows $\Pr_T(Y^T \,|\, \Delta, \Upsilon)$, where Γ, Δ, and Υ represent the three underlying factors that are not noise[2] that generate an observational dataset $\mathcal{D}$. This graphical model is illustrated in Figure 1.3. Conforming with the statements above, note that the graphical model also suggests that selection bias is induced by factors Γ and Δ, where Δ represents the confounding factors between T and Y. The inductive bias here is

[1] The material of this chapter is taken from my ICLR 2020 paper [33].

[2] See examples for Γ, Δ, and Υ in Footnote 3 of Chapter 1.

that there are non-overlapping underlying factors that, if identified, can be utilized to reduce selection bias, which in turn makes re-weighting more effective. This inductive bias is part of the representation learning module of the proposed causal effect estimator. In this (respectively next) chapter, we design a discriminative (respectively generative) model for the representation learning module.

Main contribution: We argue that explicit identification of the underlying factors $\{\Gamma, \Delta, \Upsilon\}$ in observational datasets offers great insight to guide designing models that can better handle selection bias and consequently achieve better performance in terms of estimating ITEs. In this chapter, we propose a model, named Disentangled Representations for CounterFactual Regression (DR-CFR), that is optimized to do exactly that. We also present experiments that demonstrate the advantages of this perspective; and show empirically that the proposed method outperforms state-of-the-art models in a variety of data generation scenarios with different dimensionality of factors.

Note that both [95] and [32] use Φ to model the concatenation of factors Δ and Υ (see Figure 1.3). Although it does make sense that there should be no discrepancy between conditional distributions of Υ — *i.e.*, $\Pr(\Upsilon \mid T=0)$ and $\Pr(\Upsilon \mid T=1)$ — the Δ factor should model the confounding factors, which by definition, must embed some information about treatment assignment. This would result in a positive discrepancy between conditional distributions of Δ — *i.e.*, $\Pr(\Delta \mid T=0)$ and $\Pr(\Delta \mid T=1)$ — that should not be minimized. Thus, minimizing Equation (2.15) with respect to Φ can lead to problematic results as it discards some of the confounders.

Our work has similarities to [59], who decomposed X into two subsets: confounding and adjustment variables, which are similar to our Δ and Υ factors respectively. They then used an optimization algorithm for identifying these variables, to ultimately find an unbiased estimate of the Average Treatment Effect (ATE). We extend their work in three ways:

(i) In addition to confounders and adjustment variables, we also identify

the factors (*i.e.*, Γ) that determine the treatment (thus contributing to selection bias) but have no effect on the outcome.

(ii) While [59] takes a linear approach for tagging the raw features as either confounders or adjustment variables, our proposed method has the capacity to learn non-linear representations of the underlying factors.

(iii) Our method facilitates estimating both ATE as well ITE, whereas [59] cannot provide estimates of ITEs. This is because [59] proposes a novel method based on Inverse Propensity Weighting that directly estimates the ATE, and does not provide estimation for individual counterfactual outcomes.

6.1 The Proposed Approach

We assume, without loss of generality, that any dataset of the form $\{ X, T, Y \}$ is generated from four underlying factors $\{ \Gamma, \Delta, \Upsilon, \Xi \}$ (where Ξ indicates noise) as illustrated in Figure 1.3. [3]

Observe that the factor Γ (resp., Υ) partially determines only T (resp., Y), but not the other variables; and Δ includes the confounding factors between T and Y. This graphical model suggests that selection bias is induced by factors Γ and Δ. It also shows that the outcome depends on the factors Δ and Υ. Inspired by this graphical model, our model architecture incorporates the following components:

- Three representation learning networks; one for each underlying factor: $\Gamma(x)$, $\Delta(x)$, and $\Upsilon(x)$.

[3] Note that the assumption of unconfoundedness (*i.e.*, there are no unobserved confounders) still holds; here is why:
Short: Observing X (that includes Δ) blocks the path from T to Y, which supports the unconfoundedness assumption.
Long: Once the representation networks are learned from the observational data, we can compute the latent factors $\{ \Gamma, \Delta, \Upsilon \}$ from X only. Therefore, although these factors are not explicitly observed, they are effectively observed, in that they are derived directly from the observed X, and so should not be categorized as "unobserved confounders". For example, the latent factor for "zip code" in X is "socio-economic status" (perhaps in Δ). In other words, "socio-economic status" can be inferred from "zip code" which can be viewed as a proxy for it.

- Two regression networks; one for each treatment arm: $h^0(\,\Delta(x), \Upsilon(x)\,)$ and $h^1(\,\Delta(x), \Upsilon(x)\,)$.
- Two logistic networks: $\pi_0\big(t \,|\, \Gamma(x), \Delta(x)\big)$ to model the logging policy — aka behaviour policy in Reinforcement Learning; *cf*., [101] — and $\pi\big(t \,|\, \Delta(x)\big)$ to design weights that account for the confounders' impact.[4]

We therefore try to minimize the following objective function:

$$
\begin{aligned}
J(\Gamma, \Delta, \Upsilon, h^0, h^1, \pi_0) \;= & \frac{1}{N} \sum_{i=1}^{N} \omega\big(t_i, \Delta(x_i)\big) \cdot \mathcal{L}\big[\; y_i,\, h^{t_i}\big(\Delta(x_i), \Upsilon(x_i)\big)\,\big] && (6.1)\\
& + \alpha \cdot \texttt{disc}\big(\{\Upsilon(x_i)\}_{i:t_i=0}, \{\Upsilon(x_i)\}_{i:t_i=1}\big) && (6.2)\\
& + \beta \cdot \frac{1}{N} \sum_{i=1}^{N} -\log\big[\,\pi_0\big(t_i \,|\, \Gamma(x_i), \Delta(x_i)\big)\,\big] && (6.3)\\
& + \lambda \cdot \mathfrak{Reg}(\Gamma, \Delta, \Upsilon, h^0, h^1, \pi_0) && (6.4)
\end{aligned}
$$

where $\omega\big(t_i, \Delta(x_i)\big)$ is the re-weighting function; $\mathcal{L}\big[\; y_i,\, h^{t_i}\big(\Delta(x_i), \Upsilon(x_i)\big)\,\big]$ is the prediction loss for the observed outcomes (*i.e*., factual loss); the second term $\texttt{disc}\big(\{\Upsilon(x)\}_{i:t_i=0}, \{\Upsilon(x)\}_{i:t_i=1}\big)$ calculates the discrepancy between conditional distributions of Υ given $t = 0$ versus given $t = 1$; the third term $-\log \pi_0(\,\cdot\,)$ is the cross entropy loss of predicting the assigned treatments given the learned context; and $\mathfrak{Reg}(\,\cdot\,)$ is the regularization term for penalizing model complexity. Figure 6.1 illustrates the architecture of the proposed method. The following sections elaborate on each of these terms.

6.1.1 Factual Loss: $\mathcal{L}\big[\; y,\, h^t\big(\Delta(x), \Upsilon(x)\big)\,\big]$

Similar to [49], [95], [114], and our proposed method in Chapter 5 [32], we train two regression networks h^0 and h^1, one for each treatment arm. As guided by the graphical model in Figure 1.3, the inputs to these regression networks are the outputs of the $\Delta(x)$ and $\Upsilon(x)$ representation networks and their outputs are the predicted outcomes for their respective treatments.

Note that the prediction loss $\mathcal{L}$ can only be calculated on the observed outcomes (hence the name *factual loss*), as counterfactual outcomes are not

[4] It is imperative that we re-weight the factual loss according to the correct adjustment set (here Δ) to achieve unbiased results [93].

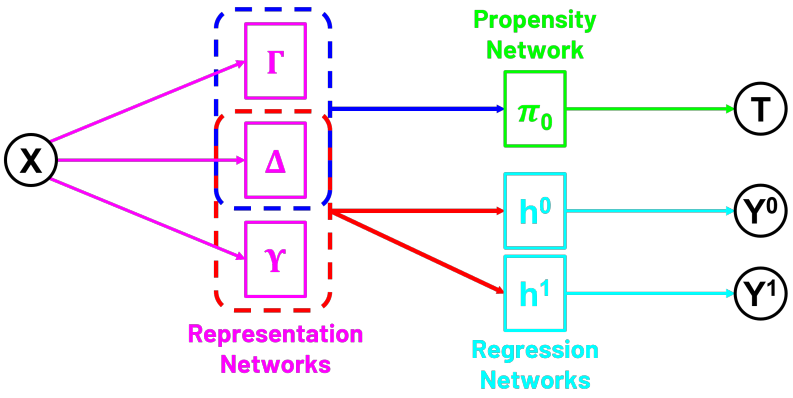


Figure 6.1: DR-CFR's model architecture.

available in any training set. This would be an L2-loss for real-valued outcomes and a log-loss for binary outcomes. By minimizing the factual loss, we ensure that the union of the learned representations $\Delta(x)$ and $\Upsilon(x)$ retain the information needed for accurate estimation of the observed outcomes.

6.1.2 Re-Weighting Function: $\omega\big(t, \Delta(x)\big)$

We follow our proposed method in [32] for deriving these weights, as re-stated in Equation (5.10), with the modification that we employ Δ to calculate the weights instead of Φ. Although following the same design, we anticipate our weights should perform better in practice than those in [32] as: (i) no confounders are discarded due to minimizing the imbalance loss (because our `disc` is defined based on Υ, not Φ); and (ii) only the legitimate confounders are used to derive the weights (*i.e.*, Δ), not the ones that have not contributed to treatment selection (*i.e.*, Υ).

Notably, the weights design in Equation (5.10) is different from the common practice in re-weighting techniques (*e.g.*, IPW) in that the weights are calculated based on all factors that determine T (*i.e.*, Γ as well as Δ). However, we argue that incorporation of Γ in the weights might result in emphasizing the wrong instances. In other words, since the factual loss $\mathcal{L}$ is only sensitive to factors Δ and Υ, and not Γ, re-weighting $\mathcal{L}$ according to Γ would yield a wrong objective function to be optimized.

6.1.3 Imbalance Loss: $\texttt{disc}\big(\{\Upsilon(x_i)\}_{i:t_i=0}, \{\Upsilon(x_i)\}_{i:t_i=1}\big)$

According to Figure 1.3, Υ should be independent of T due to the collider structure at Y. Therefore,

$$\Upsilon \perp\!\!\!\perp T \quad \Longrightarrow \quad \Pr(\,\Upsilon \,|\, T\,) = \Pr(\,\Upsilon\,) \quad \Longrightarrow \quad \Pr(\,\Upsilon \,|\, T{=}0\,) = \Pr(\,\Upsilon \,|\, T{=}1\,) \tag{6.5}$$

We used Maximum Mean Discrepancy (MMD) [26] to calculate dissimilarity between the two conditional distributions of Υ given $t{=}0$ versus $t{=}1$.

By minimizing the imbalance loss, we ensure that the learned factor Υ embeds no information about T and all the confounding factors are retained in Δ. Capturing all the confounders in Δ (and only in Δ) is the hallmark of the proposed method, as we will use it for optimal re-weighting of the factual loss term. Note that this differs from [95]'s approach in that they do not distinguish between the independent factors Δ and Υ; and minimizing the loss defined on only one factor Φ might erroneously suggest discarding some of the confounders in Δ.

6.1.4 Cross Entropy Loss: $-\log\big[\,\pi_0\big(t \,|\, \Gamma(x), \Delta(x)\big)\,\big]$

We model the logging policy as a logistic regression network parameterized by $[\,W_0, b_0\,]$ as follows: $\pi_0(\,t\,|\,\psi\,) = \big[1 + e^{-(\,2t-1\,)(\,\psi\cdot W_0 + b_0\,)}\big]^{-1}$, where ψ is the concatenation of matrices Γ and Δ. Minimizing the cross entropy loss enforces learning Γ and Δ in a way that allows $\pi_0(\,\cdot\,)$ to predict the assigned treatments. In other words, the union of the learned representations of Γ and Δ retain enough information to recover the logging policy that guided the treatment assignments.

6.2 Experiments

6.2.1 Hyperparameters

We trained DR-CFR's π_0 logistic regression function with gradient descent optimizer and a learning rate of 1E-3. We trained the Γ, Δ, Υ, and h^t networks with regularization coefficient λ=1E-3, three layers for representation and hy-

Parameter name	Range
Imbalance parameter α	1E{-2, -1, 0, 1}
Cross-entropy parameter β	1E{-2, -1, 0, 1}

Table 6.1: Hyperparameters and ranges

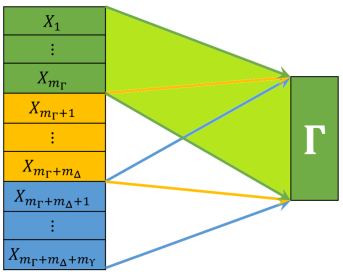
potheses networks each with 200 hidden units, `elu` [16] as the non-linear activation function, `Adam` optimizer [55] with learning rate of 1E-3, batch size of 300, and maximum number of iterations of 5000. We used the Maximum Mean Discrepancy (MMD) [26] as our IPM to calculate `disc` between the $\Pr(\Upsilon \mid t=1)$ and $\Pr(\Upsilon \mid t=0)$ distributions. See Table 6.1 for details on our hyperparameter search space.

6.2.2 Results and Discussions

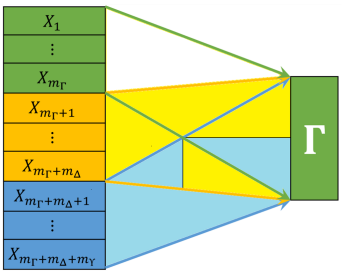
Evaluating Identification of Factors $\{\Gamma, \Delta, \Upsilon\}$

First, we want to determine if the proposed method is able to identify the variables that belong to each underlying factor. To do so, we look at the weight matrix in the first layer of each representation network, which is of size $(m_\Gamma + m_\Delta + m_\Upsilon) \times K$, where K is the number of neurons in the first hidden layer of the respective representation network. For example, to check if Γ is identified properly, we partition the weights matrix into two slices, as shown in Figure 6.2, and calculate the average of the absolute values of the weights in each slice. The first slice (referred to as S_Γ; highlighted in Figure 6.2a) pertains to "Γ's ground truth variables in X" and the second slice ($\mathrm{S}_{\neg\Gamma}$; Figure 6.2b) pertains to "variables in X that do not belong to Γ". Constructing S_Δ, $\mathrm{S}_{\neg\Delta}$, S_Υ, and $\mathrm{S}_{\neg\Upsilon}$ follow a similar procedure.

If the proposed method achieves a good identification, then we expect the average of the absolute values of weights in S_Γ should be higher than that of $\mathrm{S}_{\neg\Gamma}$; the same should hold for $(\mathrm{S}_\Delta, \mathrm{S}_{\neg\Delta})$ and $(\mathrm{S}_\Upsilon, \mathrm{S}_{\neg\Upsilon})$ as well. Note that only the relative relationships between the average absolute values of the weights in either of the slices matter; since this analysis is checking whether, for example, the respective representation network has indeed learned to emphasize on "Γ's ground truth variables in X" more than the other variables in X. Figure 6.3

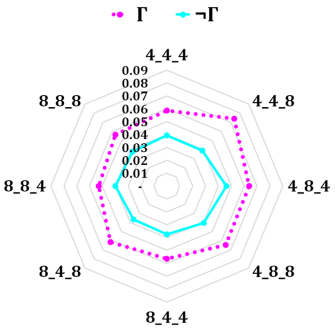


(a) Slice of the weights matrix that connects "the Γ variables in X" to "the first layer of the representation network that attempts to identify Γ". The size of this slice is $m_\Gamma \times K$.

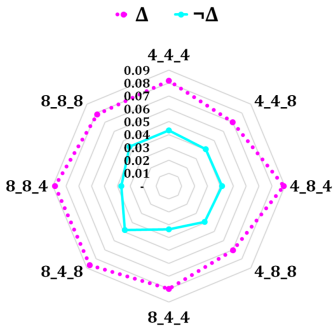


(b) Slice of the weights matrix that connects "the ¬Γ variables in X (*i.e.*, concatenation of Δ and ϒ)" to "the first layer of the representation network that attempts to identify Γ". The size of this slice is $(m_\Delta + m_\Upsilon) \times K$.

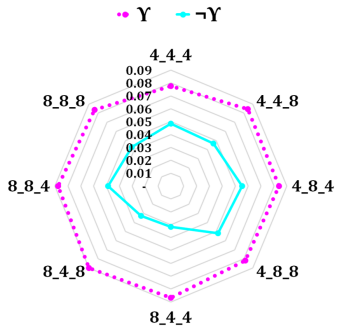
Figure 6.2: Visualization of slicing the learned weights matrix in the first layer of the representation network (number of neurons: K) for identifying Γ (best viewed in color).



(a) Identification of Γ



(b) Identification of Δ



(c) Identification of ϒ

Figure 6.3: Radar charts that visualize the capability of DR-CFR in identifying the underlying factors Γ, Δ, and ϒ. Each vertex on the polygons is identified with the factors' dimension sequence (m_Γ_m_Δ_m_Υ) of the associated synthetic dataset. The polygons' radii are scaled between $0 : 0.09$ and quantify the average weights of the first slice (in dotted magenta) and the second slice (in cyan).

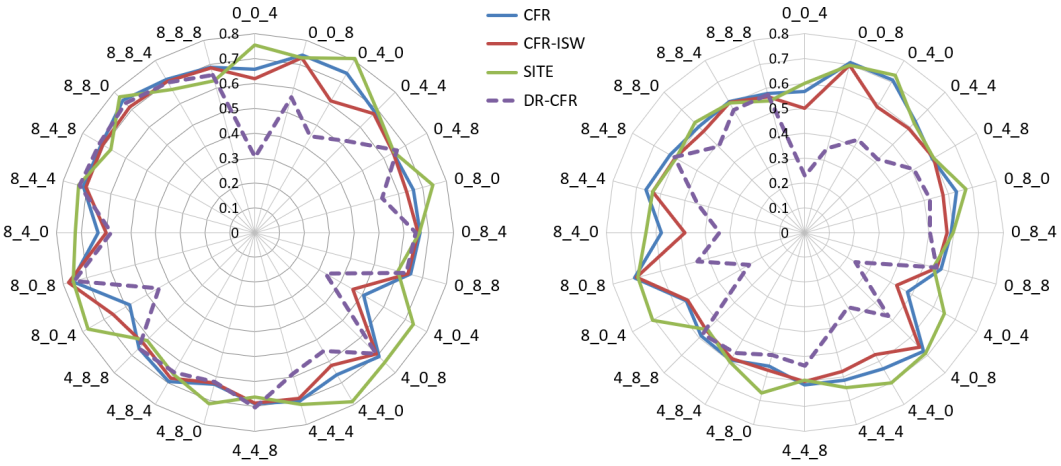


Figure 6.4: Radar charts for visualizing the PEHE performance results on the synthetic datasets. Training sample size on the left chart is 2,500 and on the right chart is 10,000. Each vertex on the polygons is identified with the factors' dimension sequence (m_Γ_m_Δ_m_Υ) of the associated group of datasets. The polygons' radii are scaled between $0:0.8$ to quantify the PEHE values (*i.e.*, the closer to the centre, the smaller the PEHE). The dashed purple curve illustrates the results of our proposed method.

illustrates the identification performance of DR-CFR according to this analysis; showing empirically that the proposed method successfully identifies all the three underlying factors, for all synthetic datasets.

Evaluating Estimation of Treatment Effects

In this chapter, we compare performances of the following treatment effect estimation methods: [5]

- **CFR**: CounterFactual Regression [95].
- **CFR-ISW**: CFR with Importance Sampling Weights [32].
- **SITE**: Similarity preserved Individual Treatment Effect [114].
- **DR-CFR**: Disentangled Representations for CFR – our proposed method.

[5]Note that all four methods share the same core code-base: based on CFR (developed by [49] and [95]) and so they share very similar model architectures. To allow for fair comparison, we searched their respective hyperparameter spaces, constrained to ensure that all had the same model complexity.

Table 6.2: Synthetic datasets (24×5 with $N=10{,}000$)

Methods	PEHE	ϵ_{ATE}
CFR	0.61 (0.05)	0.021 (0.018)
CFR-ISW	0.58 (0.06)	0.017 (0.009)
SITE	0.63 (0.05)	0.035 (0.039)
DR-CFR	**0.45 (0.11)**	**0.013 (0.006)**

Table 6.3: IHDP datasets (100 with $N=747$)

Methods	PEHE	ϵ_{ATE}
CFR	0.81 (0.30)	0.13 (0.12)
CFR-ISW	0.73 (0.28)	0.11 (0.10)
SITE	0.73 (0.33)	0.10 (0.09)
DR-CFR	**0.65 (0.37)**	**0.03 (0.04)**

PEHE and ϵ_{ATE} performance measures (lower is better) represented in the form of "mean (standard deviation)".

Figure 6.4 visualizes the PEHE measures in radar charts for these four methods, trained on the synthetic datasets (see Section 3.1.5) of size $N=2{,}500$ (left) and $N=10{,}000$ (right). As expected, all methods perform better with observing more training data; however, DR-CFR took the most advantage by reducing PEHE the most (by 0.15, going down from 0.60 to 0.45), while CFR, CFR-ISW, and SITE reduced PEHE by 0.07, 0.08, and 0.08 respectively.

Table 6.2 summarizes the PEHE and ϵ_{ATE} measures (lower is better) for all scenarios, in terms of mean and standard deviation of all the 24×5 datasets, in order to give a unified view on the performance. DR-CFR achieves the best performance among the contending methods. These results are statistically significant based on the Welch's unpaired t-test with $\alpha=0.05$.[6]

Table 6.3 summarizes the PEHE and ϵ_{ATE} measures on the IHDP benchmark. The results are reported in terms of mean and standard deviation over the 100 datasets with various realizations of outcomes. Again, DR-CFR achieves the best performance (statistically significant for ϵ_{ATE}) among the contending methods.

6.3 Conclusion

In this chapter, we studied the problem of estimating treatment effects from observational studies. We argued that not all factors in the observed covariates

[6]Since we are looking at two different performance measures (*i.e.*, PEHE and ϵ_{ATE}), this is a case of multiple comparison; thus α is corrected according to the Bonferroni correction method [9] (*i.e.*, divided by 2).

X might contribute to the procedure of selecting treatment T, or more importantly, determining the outcomes Y. We modeled this using three underlying sources of X, T, and Y, and showed that explicit identification of these sources offers great insight on designing models that better handle selection bias in observational datasets.

We proposed an algorithm, Disentangled Representations for CounterFactual Regression (DR-CFR), that can (i) identify disentangled representations of the above-mentioned underlying sources, and (ii) leverage this knowledge to reduce as well as account for the negative impact of selection bias on estimating the treatment effects from observational data. Our empirical results showed that the proposed method achieves state-of-the-art performance in both individual and population based evaluation measures.

Chapter 7

Variational Auto-encoders for Causal Inference.

Like any other machine learning task, we can employ either of the two general approaches to address the problem of causal inference: (i) discriminative modeling, or (ii) generative modeling, which differ in how the input features x and their target values y are modeled [77]:

Discriminative methods focus solely on modeling the conditional distribution $\Pr(\, y \,|\, x \,)$ with the goal of direct prediction of the target y for each instance x. For prediction tasks, discriminative approaches to learning are often more accurate since they use the model parameters more efficiently than generative approaches. Most of the current causal inference approaches are discriminative, including the matching-based methods such as Deep Match [53] and Counterfactual Propagation [29], as well as the regression-based methods such as Balancing Neural Network (BNN) [49], CounterFactual Regression Network (CFR-Net) [95] and its extensions (*cf.*, [32], [114]), Similarity preserved Individual Treatment Effect (SITE) [114], and Dragon-Net [96].

Generative methods, on the other hand, describe the relationship between x and y by their joint probability distribution $\Pr(\, x, y \,)$. This, in turn, allows the generative model to answer arbitrary queries, including coping with missing features using the marginal distribution $\Pr(\, x \,)$ or (similar to discriminative models) predicting the unknown target values y via $\Pr(\, y \,|\, x \,)$. A promising direction forward for causal inference is developing *generative* models, using either Generative Adverserial Networks (GAN) [24] or Variational

Auto-Encoders (VAE) [57], [85]. This has led to three generative approaches for causal inference: GANs for inference of Individualised Treatment Effects (GANITE) [115], Causal Effect VAE (CEVAE) [69], and Treatment Effect by Disentangled VAE (TEDVAE) [119]. However, these generative methods either do not achieve competitive performance compared to the discriminative approaches or come short of fully disentangling the underlying factors of observational data (see Figure 1.3).

Our motivation to use a generative model for learning representations of the underlying factors as opposed to a discriminative model is the intuition that learning the underlying data generating process is quite important for causal inference due to the interventional queries that we need to make. That is, knowing how the data was generated would facilitate accurate estimation of what would have happened had a certain variable had taken a different value.

Moreover, although discriminative models have excellent predictive performance, they often suffer from two drawbacks: (i) overfitting, and (ii) making highly-confident predictions, even for instances that are "far" from the observed training data. Generative models based on Bayesian inference, on the other hand, can handle both of these drawbacks: issue (i) can be minimized by taking an average over the posterior distribution of model parameters; and issue (ii) can be addressed by explicitly providing model uncertainty via the posterior [25]. Although the exact inference is often intractable, efficient approximations to the parameter posterior distribution is possible through variational methods. In this work, we use the Variational Auto-Encoder (VAE) framework [57], [85] to tackle this.

Contributions: We propose three interrelated Bayesian models (namely Series, Parallel, and Hybrid) that employ the VAE framework to address the task of causal inference for binary treatments. We demonstrate that all three of these models significantly outperform the state-of-the-art in terms of estimating treatment effects on two publicly available benchmarks, as well as a fully synthetic dataset that allows for detailed performance analyses. We also show that our proposed Hybrid model is the best at decomposing the underlying

factors of any observational dataset.

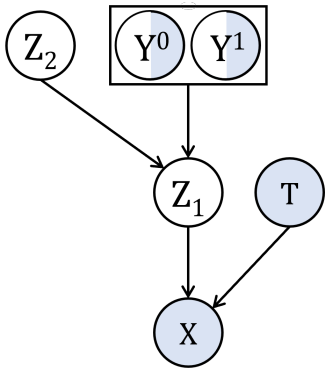
Our analysis in this chapter relies on the assumptions stated in Chapter 2, Section 2.1.2.

7.1 Method

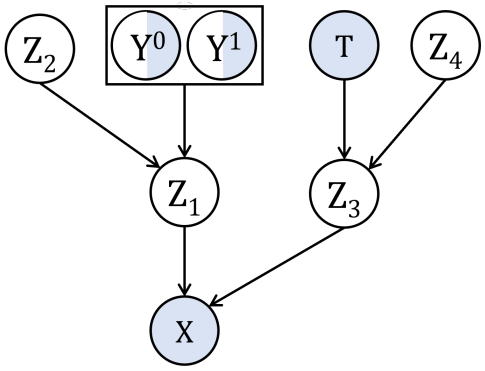
Following [33], [59], we assume that the random variable X follows an unknown joint probability distribution $\Pr(X \mid \Gamma, \Delta, \Upsilon, \Xi)$, where Γ, Δ, Υ, and Ξ are non-overlapping independent factors (see Figure 1.3). We emphasize that the belief-net in Figure 1.3 is built without loss of generality; *i.e.*, it also covers the scenarios where any of the latent factors is degenerate (*i.e.*, has a zero-dimensionality; effectively making it non-existent in X). Therefore, if we can design a method that has the capacity to capture all of these latent factors, it would be successful in all scenarios — even in the ones that have degenerate factors (and in fact this is true; see the experimental setting and performance results on the Synthetic benchmark in Sections 3.1.5 and 7.2.3 respectively).

Our goal is to design a generative model architecture that encourages learning decomposed representations of these underlying latent factors (see Figure 1.3). In other words, it should be able to decompose and separately learn the three underlying factors that are responsible for determining "T only" (Γ), "Y only" (Υ), and "both T and Y" (Δ). To achieve this, we propose a progressive sequence of three models (namely Series, Parallel, and Hybrid; as illustrated in Figures 7.1a, 7.1b, and 7.1c respectively), where each is an *improvement* over the previous one. Every model employs several stacked M2 or M1+M2 VAEs [56], that each includes a decoder (generative model) and an encoder (variational posterior), which are parametrized as deep neural networks.[1] Appendix B presents an overview of the M1 and M2 VAEs.

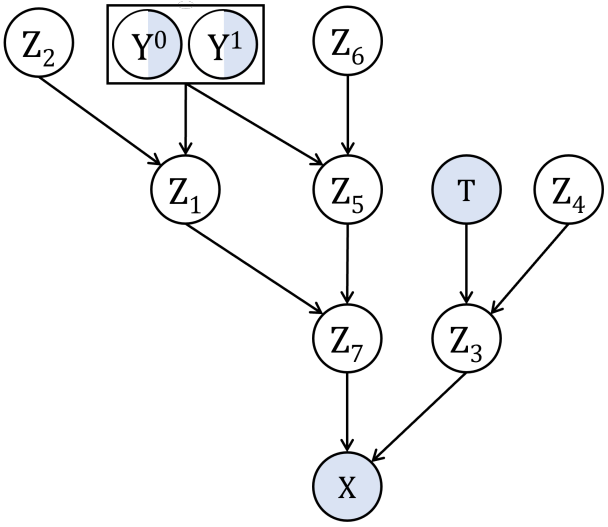
[1]Note that the M2 and M1+M2 components in the proposed architectures are capable of employing the extra supervision from Y and T to help learning disentangled representations of the underlying factors. In other words, in the proposed architectures, Y and T guide each of the representation nodes to capture the appropriate factor.



(a) The **Series** Model. We expect Z_1 to capture Δ and Υ.



(b) The **Parallel** Model. We expect Z_1 to capture Δ and Υ, and Z_3 to capture Γ.



(c) The **Hybrid** Model. We expect Z_1 to capture Υ, Z_5 to capture Δ, and Z_3 to capture Γ.

Figure 7.1: Belief-nets of the proposed models.

7.1.1 The Variational Auto-Encoder Component

The Series Model

The belief-net of the Series model is illustrated in Figure 7.1a. Louizos *et al.* [70] proposed a similar architecture to address fairness in machine learning, but using a binary sensitive variable S (*e.g.*, gender, race, etc.) rather than the treatment T. Here, we employ this architecture for causal inference and explain why it should work. We hypothesize that this structure functions as a fractionating column[2]: the bottom M2 VAE attempts to decompose Γ (guided by T) from Δ and Υ (captured by Z_1); and the top M2 VAE attempts to learn Δ and Υ (guided by Y).

The decoder and encoder components of the Series model — $p(\cdot)$ and $q(\cdot)$ parametrized by θ_s and ϕ_s respectively — involve the following distributions:

Priors		**Likelihood**	**Posteriors**	
$p_{\theta_s}(z_2)$	$p_{\theta_s}(z_1\|y,z_2)$	$p_{\theta_s}(x\|z_1,t)$	$q_{\phi_s}(z_1\|x,t)$	$q_{\phi_s}(y\|z_1)$
			$q_{\phi_s}(z_2\|y,z_1)$	

Hereafter, we drop the θ and ϕ subscripts for brevity.

The goal is to maximize the conditional log-likelihood of the observed data[3] (left-hand-side of the following inequality) by maximizing the Evidence Lower BOund (ELBO; right-hand-side):

$$\sum_{i=1}^{N} \log p(x_i|t_i,y_i) \quad \geq \quad \sum_{i=1}^{N} \mathbb{E}_{q(z_1|x,t)}\big[\log p(x_i|z_{1_i},t_i)\big] \tag{7.1}$$

$$- \mathrm{KL}\big(\, q(z_1|x,t) \;||\; p(z_1|y,z_2) \,\big)$$

$$- \mathrm{KL}\big(\, q(z_2|y,z_1) \;||\; p(z_2) \,\big) \tag{7.2}$$

where KL denotes the Kullback-Leibler divergence, $p(z_2)$ is the unit multivariate Gaussian (*i.e.*, $\mathcal{N}(0,\mathbb{I})$), and the other distributions are assumed to be

[2]In chemistry, a fractionating column is used for separating different liquid compounds in a mixture; see `https://en.wikipedia.org/wiki/Fractionating_column` for more details. In our work, similarly, we can separate different factors from the pool of features using the proposed architectures (especially, the Hybrid model).

[3]We try to maximize the lower bound on the likelihood of the observed data (*i.e.*, $P(X|Y,T)$), in order to find the underlying data generating process from which our data is sampled. This is the generative part of the algorithm. After learning the representations of the underlying factors, we then use a discriminative approach to estimate the outcomes.

multivariate Gaussian whose μ and Σ (diagonal) are parameterized as deep neural networks.

The Parallel Model

The Series model is composed of two M2 stacked models. However, Kingma *et al.* [56] showed that an M1+M2 stacked architecture learns better representations than an M2 model alone for a downstream prediction task. This motivated us to design a double M1+M2 Parallel model; where one arm is for the outcome to guide the representation learning via Z_1 and another for the treatment to guide the representation learning via Z_3. Figure 7.1b shows the belief-net of this model. We hypothesize that Z_1 would learn Δ and Υ, and Z_3 would learn Γ (and perhaps partially Δ).

The decoder and encoder components of the Parallel model — $p(\cdot)$ and $q(\cdot)$ parametrized by θ_p and ϕ_p respectively — involve the following distributions:

Priors		**Likelihood**	**Posteriors**	
$p(z_2)$	$p(z_1\|y, z_2)$	$p(x\|z_1, z_3)$	$q(z_1\|x, t)$	$q(y\|z_1)$
$p(z_4)$	$p(z_3\|t, z_4)$		$q(z_2\|y, z_1)$	$q(t\|z_3)$
			$q(z_3\|x, y)$	
			$q(z_4\|t, z_3)$	

Here, the conditional log-likelihood can be upper bounded by:

$$\sum_{i=1}^{N} \log p(x_i|t_i, y_i) \quad \geq \quad \sum_{i=1}^{N} \mathbb{E}_{q(z_1,z_3|x,t,y)}\big[\log p(x_i|z_{1_i}, z_{3_i})\big] \tag{7.3}$$

$$- \mathrm{KL}\big(\, q(z_1|x, t) \;||\; p(z_1|y, z_2)\,\big)$$

$$- \mathrm{KL}\big(\, q(z_2|y, z_1) \;||\; p(z_2)\,\big)$$

$$- \mathrm{KL}\big(\, q(z_3|x, y) \;||\; p(z_3|t, z_4)\,\big)$$

$$- \mathrm{KL}\big(\, q(z_4|t, z_3) \;||\; p(z_4)\,\big) \tag{7.4}$$

The Hybrid Model

The final model, Hybrid (see Figure 7.1c), attempts to combine the best capabilities of the previous two architectures. The backbone of the Hybrid model has a Series architecture, that separates Γ (factors related to the treatment

T; captured by the right module with Z_3 as its head) from Δ and Υ (factors related to the outcome Y; captured by the left module with Z_7 as its head). The left module, itself, consists of a Parallel model that attempts to proceed one step further and decompose Δ from Υ. This is done with the help of a discrepancy penalty (see Section 7.1.3).

The decoder and encoder components of the Hybrid model — $p(\cdot)$ and $q(\cdot)$ parametrized by θ_h and ϕ_h respectively — involve the following distributions:

Priors		**Likelihood**	**Posteriors**	
$p(z_2)$	$p(z_1\|y, z_2)$	$p(x\|z_3, z_7)$	$q(z_1\|z_7)$	$q(y\|z_1, z_5)$
$p(z_4)$	$p(z_3\|t, z_4)$		$q(z_2\|y, z_1)$	$q(t\|z_3)$
$p(z_6)$	$p(z_5\|y, z_6)$		$q(z_3\|x, y)$	
	$p(z_7\|z_1, z_5)$		$q(z_4\|t, z_3)$	
			$q(z_5\|z_7)$	
			$q(z_6\|y, z_5)$	
			$q(z_7\|x, t)$	

Here, the conditional log-likelihood can be upper bounded by:

$$\begin{aligned}\sum_{i=1}^{N} \log p(x_i|t_i, y_i) \quad \geq \quad & \sum_{i=1}^{N} \mathbb{E}_{q(z_3,z_7|x,t,y)}\big[\log p(x_i|z_{3_i}, z_{7_i})\big] \qquad (7.5)\\ & - \mathrm{KL}\big(\, q(z_1|z_7) \;||\; p(z_1|y, z_2)\,\big)\\ & - \mathrm{KL}\big(\, q(z_2|y, z_1) \;||\; p(z_2)\,\big)\\ & - \mathrm{KL}\big(\, q(z_3|x, y) \;||\; p(z_3|t, z_4)\,\big)\\ & - \mathrm{KL}\big(\, q(z_4|t, z_3) \;||\; p(z_4)\,\big)\\ & - \mathrm{KL}\big(\, q(z_5|z_7) \;||\; p(z_5|y, z_6)\,\big)\\ & - \mathrm{KL}\big(\, q(z_6|y, z_5) \;||\; p(z_6)\,\big)\\ & - \mathrm{KL}\big(\, q(z_7|x, t) \;||\; p(z_7|z_1, z_5)\,\big) \qquad (7.6)\end{aligned}$$

For all three of these models, we refer to the first term in the ELBO (*i.e.*, right-hand-side of Equations (7.1), (7.3), or (7.5)) as the Reconstruction Loss (RecL) and the next term(s) (*i.e.*, Equations (7.2), (7.4), or (7.6)) as the KL Divergence (KLD). Concisely, the ELBO can be viewed as maximizing:

$$\mathrm{RecL} - \mathrm{KLD} \qquad (7.7)$$

7.1.2 Further Disentanglement with β-VAE

As mentioned earlier, we want the learned latent variables to be disentangled, to match our assumption of non-overlapping factors Γ, Δ, and Υ. To further encourage this (in addition to the proposed architecture), we employ the β-VAE [37], which adds a hyperparameter β as a multiplier of the KLD part of the ELBO. This adjustable hyperparameter facilitates a trade-off that helps balance the latent channel capacity and independence constraints with the reconstruction accuracy. In other words, a higher β coefficient for the KL divergence term would enforce the approximate posterior to be closer to the unit Gaussian prior, which in turn would encourage independence between the learned latent variables.[4] We therefore hypothesize that including the β hyperarameter should grant a better control over the level of disentanglement in the learned representations [12]. Therefore, the generative objective to be minimized becomes:

$$\mathcal{L}_{\text{VAE}} \quad = \quad -\text{RecL} \; + \; \beta \cdot \text{KLD} \tag{7.8}$$

Although Higgins *et al.* [37] suggest setting β greater than 1 in the original paper, Hoffman *et al.* [40] show that having a $\beta < 1$ weight on the KLD term can be interpreted as optimizing the ELBO under an alternative prior, which functions as a regularization term to reduce the chance of degeneracy.

7.1.3 Discrepancy

Although all three proposed models encourage statistical independence between T and Z_1 in the marginal posterior $q_\phi(Z_1|T)$ where X is not given — see the collider structure (at X): $T \rightarrow X \leftarrow Z_1$ in Figure 7.1a — an information leak is quite possible due to the correlation between the outcome Y and treatment T in the data.[5] We therefore require an extra regularization term on $q_\phi(Z_1|T)$ in order to penalize the discrepancy (denoted by `disc`) between the

[4] This is due to the zero non-diagonal terms in the covariance matrix of the unit Gaussian prior.

[5] In other words, since Y is one of Z_1's parents, and Y is either the outcome of treatment 1 or 0, we have an information leak from T to Z_1 through Y.

conditional distributions of Z_1 given $T=0$ versus given $T=1$. [6] [7] To achieve this regularization, we calculate the `disc` using an Integral Probability Metric (IPM) [73] [8] (*cf.*, [70], [95], [114], etc.) that measures the distance between the two above-mentioned distributions:

$$\mathcal{L}_{\texttt{disc}} \quad = \quad \text{IPM}\big(\ \{z_1\}_{i:t_i=0},\ \{z_1\}_{i:t_i=1}\ \big) \tag{7.9}$$

7.1.4 Predictive Loss

Note, however, that neither the VAE nor the `disc` losses contribute to training a predictive model for outcomes. To remedy this, we extend the objective function to include a discriminative term for the regression loss of predicting y: [9]

$$\mathcal{L}_{\text{pred}} \quad = \quad \frac{1}{N}\sum_{i=1}^{N} \omega_i \cdot \mathcal{L}\big[\, y_i,\, \hat{y}_i \big] \tag{7.10}$$

where the predicted outcome $\hat{y}_i$ is set to be the mean of the $q_\phi^{t_i}(y_i|z_{1_i})$ distribution for the Series and Parallel models and the mean of the $q_\phi^{t_i}(y_i|z_{1_i}, z_{5_i})$ distribution for the Hybrid model; $\mathcal{L}\big[\, y_i,\, \hat{y}_i \,\big]$ is the factual loss (*i.e.*, L2 loss for real-valued outcomes and log loss for binary-valued outcomes); and ω_i represent the weights that attempt to account for selection bias. We consider two approaches in the literature to derive these ω_i weights: (i) the *Population-Based* (PB) weights as proposed in [95]; and (ii) the *Context-Aware* (CA) weights as proposed in [32][10]. Note that disentangling Δ from Υ is only beneficial when using the CA weights, since we need just the Δ factors to derive them [33].

[6] Note that even for the Hybrid model (see Figure 7.1c), we apply the `disc` penalty only on Z_1 and not Z_7. This is because we want Z_1 to capture Υ and Z_5 to capture Δ (therefore, Z_5 should have a non-zero `disc`). Hence, Z_7 must include both Δ and Υ (and therefore, it also should have a non-zero `disc`) to be able to reconstruct X.

[7] Similarly, we could think of enforcing $(Z_3 \perp\!\!\!\perp Y)$; however, `disc` would not work here since Y is not binary. It is possible to enforce this independence by minimizing the mutual information (*e.g.*, via [6]) for either of Y^0 and Y^1 against Z_3 by adding two more independence penalty terms. This is left to future work.

[8] In this work, we use the Maximum Mean Discrepancy (MMD) [26] as our IPM.

[9] This is similar to the way Kingma *et al.* [56] included a classification loss in their Equation (9).

[10] Note that we use Z_1 (which captures both Δ and Υ) for deriving the weights for the series and parallel models and Z_5 (which captures Δ) for the hybrid model.

7.1.5 Final Model(s)

Putting everything together, the overall objective function to be minimized is:

$$\mathcal{J} \quad = \quad \mathcal{L}_{\text{pred}} \; + \; \alpha \cdot \mathcal{L}_{\texttt{disc}} \; + \; \gamma \cdot \mathcal{L}_{\text{VAE}} \; + \; \lambda \cdot \mathfrak{Reg} \tag{7.11}$$

where $\mathfrak{Reg}$ penalizes the model complexity.

This objective function is motivated by the work of [75], which suggested optimizing a convex combination of discriminative and generative losses would indeed improve predictive performance. As an empirical verification, note that for $\gamma = 0$, the Series and Parallel models effectively reduce to CFR-Net [95]. However, our empirical results (see Section 7.2) suggest that the generative term in the objective function helps learning representations that embed more relevant information for estimating outcomes than that of Φ in CFR-Net.

We refer to the family of our proposed methods as VAE-CI (Variational Auto-Encoder for Causal Inference); specifically: **{S, P, H}-VAE-CI**, for **S**eries, **P**arallel, and **H**ybrid respectively. We anticipate that each method is an *improvement* over the previous one in terms of estimating causal effects, culminating in **H-VAE-CI**, which we expect can best decompose the underlying factors and accurately estimate the outcomes of all treatments.

7.2 Experiments, Results, and Discussion

7.2.1 Hyperparameters

For all CFR, DR-CFR, and VAE-CI methods, we trained the neural networks with 3 layers (each consisting of 200 hidden neurons)[11], non-linear activation function `elu` [16], regularization coefficient of λ=1E-4, `Adam` optimizer [55] with a learning rate of 1E-3, batch size of 300, and maximum number of iterations of $10,000$. See Table 7.1 for our hyperparameter search space.

[11] In addition to this basic configuration, we also performed our grid search for all the contending methods with an updated number of layers and/or number of neurons in each layer which guaranteed that all methods enjoy a similar model complexity. Therefore, any improved performance should be due to the superiority of the respective model.

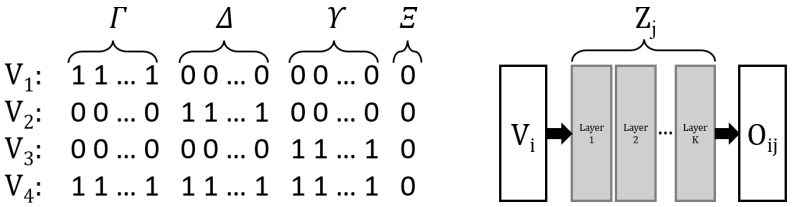


Figure 7.2: The four dummy x-like vectors (left); and the input/output vectors of the representation networks (right).

Table 7.1: Hyperparameters and ranges

Hyperparameter	Range
Discrepancy coefficient α	$\{0, 1E\{-3, -2, -1, 0, 1\}\}$
KLD coefficient β	$\{0, 1E\{-3, -2, -1, 0, 1, 2\}\}$
Generative coefficient γ	$\{0, 1E\{-5, -4, -3, -2, -1, 0\}\}$

7.2.2 Identification of the Underlying Factors

Procedure for Evaluating Identification of the Underlying Factors

To evaluate any representation network Z_j in terms of its disentanglement quality of the learned representations of the underlying factors, we use a fully synthetic dataset with $m_\Gamma = m_\Delta = m_\Upsilon = 8$ and $m_\Xi = 1$ (see Section 3.1.5). We then ran the learned model on four dummy test instances $V_i \in \mathbb{R}^{m_\Gamma + m_\Delta + m_\Upsilon + m_\Xi}$ as depicted on the left-side of Figure 7.2. The first to third vectors had "1" (constant) in the positions associated with Γ, Δ, and Υ respectively, and the remaining $2 \times 8 + 1 = 17$ positions were filled with "0". The fourth vector was all "1" except for the last position (the noise) which was "0". The use of these dummy signals as input helps measure the maximum amount of information that is allowed to reach to the final layer of each representation network.

Next, each vector V_i is fed as input to each trained network Z_j (as if it was x). We let O_{ij} be the output (here, $\in \mathbb{R}^{200}$, with an activation function of `elu`) of the encoder network Z_j when its input is set to V_i. The average of the 200 values of the O_{ij} (*i.e.*, $Avg(O_{ij})$) represents the power of signal that was produced by the Z_j channel on the input V_i.[12] The values reported in

[12] *E.g.*, if a representation network Z_j is supposed to, for example, capture Δ, we expect

the tables illustrated in Figure 7.3 are the ratios of $Avg(O_{1j})$, $Avg(O_{2j})$, and $Avg(O_{3j})$ divided by $Avg(O_{4j})$ for each of the learned representation networks. Note that, a larger ratio indicates that the respective representation network Z_j has allowed more of the input signal V_i to pass through; thus, Z_j has in fact captured the respective underlying factor.[13]

If the model could perfectly learn each underlying factor in a disentangled manner, we expect to see one element in each column to be significantly larger than the other elements in that column. For example, for H-VAE-CI, the weights of the Γ row should be highest for Z_3, as that means Z_3 has captured the Γ factors. Similarly, we would want the Z_5 and Z_1 entries on the Δ and Υ rows to be largest respectively.

Results' Analysis

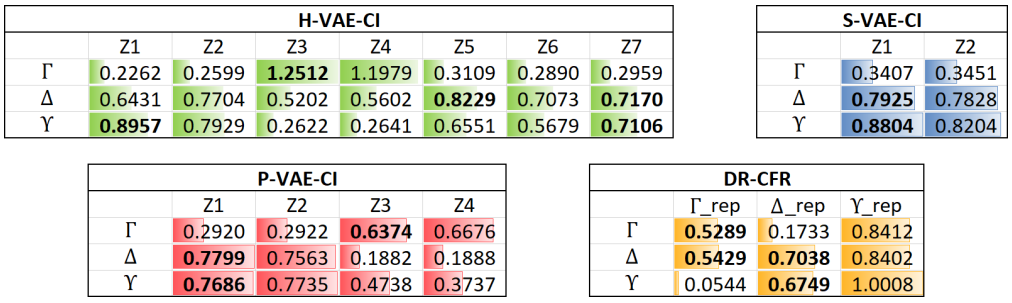
As expected, Figure 7.3 shows that Z_3 and Z_4 capture Γ (*e.g.*, the Z_3 ratios for Γ in the {P, H}-VAE-CI tables are largest), and Z_1, Z_2, Z_5, Z_6, and Z_7 capture Δ and Υ. Note that decomposition of Δ from Υ has not been achieved by any of the methods except for H-VAE-CI, which captures Υ by Z_1 and Δ by Z_5 (note the ratios are largest for Z_1 and Z_5). This decomposition is vital for deriving context-aware importance sampling weights because they must be calculated from Δ only [33]. Also observe that {P, H}-VAE-CI are each able to separate Γ from Δ. However, DR-CFR, which tried to disentangle all factors, failed not only to disentangle Δ from Υ, but also Γ from Δ.

7.2.3 Treatment Effect Estimation

Here, we compare performances of the proposed **{S, P, H}-VAE-CI** versus the following treatment effect estimation methods: **CFR-Net** [95], **DR-CFR**

to see the output of Z_j network to be high for input of V_2 and low for inputs of V_1 and V_3 dummy variables.

[13]Unlike the evaluation strategy presented in [33] that only examined the first layer's weights of each representation network, we propagate the values through the entire network and check how much of each factor is exhibited in the final layer of every representation network. Yet, the proposed procedure still crudely evaluates the quality of disentanglement of the underlying factors in observational studies. We did explore using the Mutual Information [6] for this task; however, it appears that it does not work for high-dimensional data such as ours. All in all, more research is needed to address this task.



H-VAE-CI	Z1	Z2	Z3	Z4	Z5	Z6	Z7
Γ	0.2262	0.2599	**1.2512**	1.1979	0.3109	0.2890	0.2959
Δ	0.6431	0.7704	0.5202	0.5602	**0.8229**	0.7073	**0.7170**
Υ	**0.8957**	0.7929	0.2622	0.2641	0.6551	0.5679	**0.7106**

S-VAE-CI	Z1	Z2
Γ	0.3407	0.3451
Δ	**0.7925**	0.7828
Υ	**0.8804**	0.8204

P-VAE-CI	Z1	Z2	Z3	Z4
Γ	0.2920	0.2922	**0.6374**	0.6676
Δ	**0.7799**	0.7563	0.1882	0.1888
Υ	**0.7686**	0.7735	0.4738	0.3737

DR-CFR	Γ_rep	Δ_rep	Υ_rep
Γ	**0.5289**	0.1733	0.8412
Δ	**0.5429**	**0.7038**	0.8402
Υ	0.0544	**0.6749**	1.0008

Figure 7.3: Performance analysis for decomposition of the underlying factors on the **Synthetic** dataset with $m_\Gamma = m_\Delta = m_\Upsilon = 8$ and $m_\Xi = 1$. The color shading in each cell represents the value of that cell, with a longer colored bar for larger values.

[33], **Dragon-Net** [96], **GANITE** [115], **CEVAE** [69], and **TEDVAE** [119]. The basic search grid for hyperparameters of the CFR-Net based algorithms (including our methods) is available in Section 7.2.1. For the other algorithms, we searched around their default hyperparameter settings.

We ran the experiments for the contender methods using their publicly available code-bases; note the following points regarding these runs:

- Since Dragon-Net is designed to estimate ATE only, we did not report its performance results for the PEHE measure (which, as expected, were significantly inaccurate compared to the rest of the methods).
- Original GANITE code-base could only deal with binary outcomes. We modified the code (losses, etc.) to allow it to process real-valued outcomes also.
- We were surprised that CEVAE diverged when running on the ACIC'18 datasets. To avoid this, we had to run the ACIC'18 experiments on the binary covariates only.

Results' Analysis

Tables 7.2, 7.3, and 7.4 summarizes the mean and standard deviation of the PEHE and ϵ_{ATE} measures (lower is better) on the IHDP, ACIC'18, and Synthetic benchmarks respectively. VAE-CI achieves the best performance among the contending methods. These results (best ones shown in **bold**) are

Table 7.2: PEHE and ϵ_{ATE} performance measures (lower is better) of the IHDP benchmark represented in the form of "mean (standard deviation)".

Method	IHDP	
	PEHE	$\boldsymbol{\epsilon_{\text{ATE}}}$
CFR-Net	0.75 (0.57)	0.08 (0.10)
DR-CFR	0.65 (0.37)	0.03 (0.04)
Dragon-Net	NA	0.14 (0.15)
GANITE	2.81 (2.30)	0.24 (0.46)
CEVAE	2.50 (3.47)	0.18 (0.25)
TEDVAE	1.61 (2.37)	0.18 (0.23)
S-VAE-CI	0.51 (0.37)	**0.00 (0.02)**
P-VAE-CI	0.52 (0.36)	0.01 (0.03)
H-VAE-CI (PB)	0.49 (0.36)	0.01 (0.02)
H-VAE-CI (CA)	**0.48 (0.35)**	0.01 (0.01)

statistically significant (based on the Welch's unpaired t-test with $\alpha = 0.05$) for the IHDP (ϵ_{ATE}) and Synthetic benchmarks (both PEHE and ϵ_{ATE}). Although VAE-CI also achieves the best performance on the ACIC'18 benchmark, the results are not statistically significant due to the high standard deviation of the performances of the contending methods.

Figure 7.4 visualizes the PEHE measures on the entire synthetic datasets with sample size of $N = 10{,}000$. We observe that both plots corresponding to H-VAE-CI method (PB as well as CA) are completely within the plots of all other methods, showcasing H-VAE-CI's superior performance under every possible selection bias scenario.

Note that for scenarios where $m_\Delta = 0$ — *i.e.*, the ones of the form m_Γ_0_m_Υ on the perimeter of the radar chart in Figure 7.4 (0_0_4, 0_0_8, 4_0_4, 4_0_8, 8_0_4, and 8_0_8) — the performances of H-VAE-CI (PB) and H-VAE-CI (CA) are almost identical. This is expected, since for these scenarios, the learned representation for Δ would be degenerate, and therefore, the context-aware weights would reduce to population-based ones. On the other hand, for scenarios where $m_\Delta \neq 0$, the H-VAE-CI (CA) often outperforms H-VAE-CI (PB). This may be because H-VAE-CI has correctly disentangled Δ from Υ. This facilitates learning good CA weights that better account for selection bias,

Table 7.3: PEHE and ϵ_{ATE} performance measures (lower is better) of the ACIC'18 benchmark represented in the form of "mean (standard deviation)".

Method	ACIC'18	
	PEHE	$\boldsymbol{\epsilon_{\text{ATE}}}$
CFR-Net	5.13 (5.59)	1.21 (1.81)
DR-CFR	3.86 (3.39)	0.80 (1.41)
Dragon-Net	NA	0.48 (0.77)
GANITE	3.55 (2.27)	0.69 (0.65)
CEVAE	5.30 (5.52)	3.29 (3.50)
TEDVAE	6.63 (8.69)	3.74 (5.00)
S-VAE-CI	2.73 (2.39)	0.51 (0.82)
P-VAE-CI	2.62 (2.26)	**0.37 (0.75)**
H-VAE-CI (PB)	1.78 (1.27)	0.44 (0.77)
H-VAE-CI (CA)	**1.66 (1.30)**	0.39 (0.75)

Table 7.4: PEHE and ϵ_{ATE} performance measures (lower is better) of the Synthetic benchmark represented in the form of "mean (standard deviation)".

Method	Synthetic	
	PEHE	$\boldsymbol{\epsilon_{\text{ATE}}}$
CFR-Net	0.39 (0.08)	0.027 (0.020)
DR-CFR	0.26 (0.07)	0.007 (0.004)
Dragon-Net	NA	0.007 (0.005)
GANITE	1.28 (0.43)	0.036 (0.015)
CEVAE	1.39 (0.32)	0.287 (0.217)
TEDVAE	0.25 (0.07)	0.013 (0.007)
S-VAE-CI	0.28 (0.05)	0.004 (0.003)
P-VAE-CI	0.28 (0.05)	0.004 (0.003)
H-VAE-CI (PB)	0.20 (0.03)	**0.003 (0.002)**
H-VAE-CI (CA)	**0.18 (0.02)**	**0.003 (0.002)**

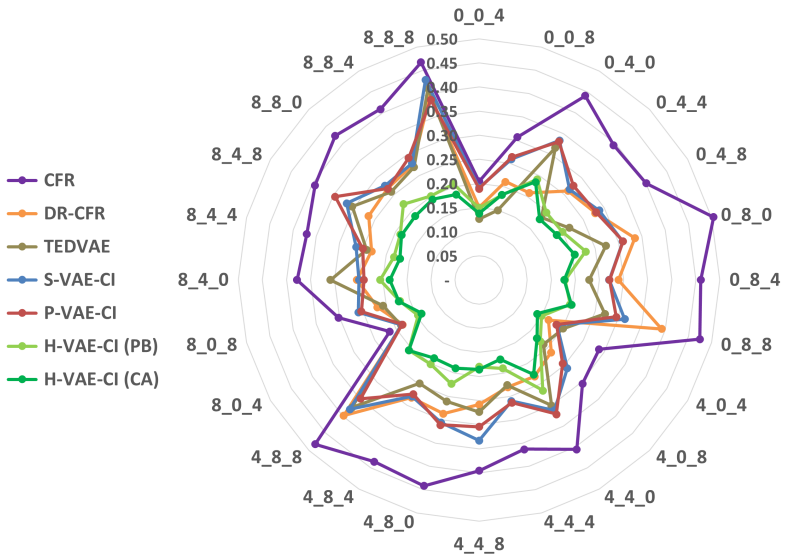


Figure 7.4: Radar graphs of PEHE (on the radii; closer to the center is better) for the entire **Synthetic** benchmark (24×3 with $N = 10{,}000$; each vertex denotes the respective dataset). Figure is best viewed in color.

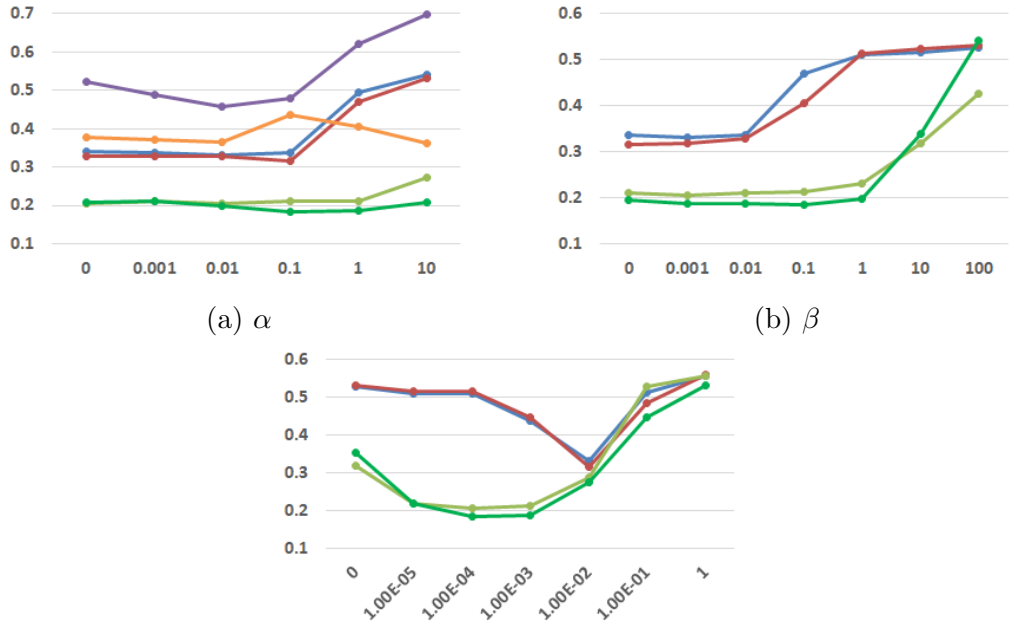
which in turn, results in a better causal effect estimation performance.

7.2.4 Hyperparameters' Sensitivity Analyses

Figure 7.5 illustrates the results of our hyperparameters' sensitivity analyses (in terms of PEHE). In the following, we discuss the insights we gained from these ablation studies:

Hyperparameter α (coefficient of the discrepancy penalty)

Figure 7.5a suggests that DR-CFR and H-VAE-CI methods have the most robust performance throughout various values of α. This is expected, because, unlike CFR-Net and {S, P}-VAE-CI, DR-CFR and H-VAE-CI possess an independent node for representing Δ. This helps them still capture Δ as α grows; since for them, α only affects learning a representation of Υ. Comparing H-VAE-CI (PB) with (CA), we observe that for all $\alpha > 0.01$, (CA) outperforms (PB). This is because the discrepancy penalty would force Z_1 to only capture Υ and Z_5 to only capture Δ. This results in deriving better CA weights (that should be learned from Δ; here, from its learned representation Z_5). H-VAE-CI (PB), on the other hand, cannot take advantage of this



(a) α (b) β

(c) γ

Figure 7.5: Hyperparameters' (x-axis) sensitivity analysis based on PEHE (y-axis) on the **synthetic** dataset with $m_{\Gamma,\Delta,\Upsilon} = 8, m_{\Xi} = 1$. Legend is the same as Figure 7.4: purple for CFR-Net, orange for DR-CFR, blue for S-VAE-CI, red for P-VAE-CI, and light and dark green for H-VAE-CI (PB) and (CA) respectively. Plots are best viewed in color.

disentanglement, which explains its sub-optimal performance.

Hyperparameter β (coefficient of KL divergence penalty)

Figure 7.5b shows that various β values do not make much difference for H-VAE-CI (except for $\beta \geq 1$, since this large value means the learned representations will be close to Gaussian noise). We initially thought using β-VAE might help *further* disentangle the underlying factors. However, Figure 7.5b suggests that close-to-zero or even zero βs also work effectively. Our hypothesis is that the H-VAE-CI's architecture already takes care of decomposing the Γ, Δ, and Υ factors, without needing the help of a KLD penalty. Appendix C includes more evidence and a detailed discussion on why this interpretation should hold. Moreover, the KLD penalty attempts to disentangle the learned representations *within* each underlying factor; however, what we really need is the disentanglement (*i.e.*, [non-linear] independence) *across* the learned representations of different underlying factors.

Hyperparameter γ (coefficient of the generative loss penalty)

Figure 7.5c demonstrates that for a wide range, the hyperparameter γ achieves the best and most stable performance in H-VAE-CI compared to that of {S, P}-VAE-CI. Observing that H-VAE-CI outperforms {S, P}-VAE-CI for $\gamma \leq 0.01$, suggests that having the generative loss term (*i.e.*, $\mathcal{L}_{\text{VAE}}$) is more important for {S, P}-VAE-CI than it is for H-VAE-CI to perform well. Note an extreme case happens at $\gamma = 0$, where the latter performs significantly (statistical) better than the former. We hypothesize that although $\mathcal{L}_{\text{VAE}}$ is helpful (note the drop in PEHE from $\gamma = 0$ to $0 < \gamma \leq 0.01$), the other terms in the objective function can *partially* impose the decomposition and learn expressive representations Z_3 and Z_7 in H-VAE-CI. This is in contrast to Z_1 in S-VAE-CI, and Z_1 and Z_3 in P-VAE-CI.

7.3 Conclusion

The goal of this chapter was to estimate causal effects (either for individuals or the entire population) from observational data. We designed three models that employ Variational Auto-Encoders (VAE) [57], [85], namely Series, Parallel, and Hybrid. Each model was an improvement over the previous one, in terms of identifying the underlying factors of any observational data as well as estimating the causal effects. Our proposed methods employed Kingma *et al*. [56]'s M1 and M2 models as their building blocks. Our Hybrid model performed best, and succeeded at learning decomposed representations of the underlying factors Γ, Δ, and Υ. This, in turn, helped to accurately estimate the outcomes of all treatments. Our empirical results demonstrated the superiority of the proposed methods, compared to both state-of-the-art discriminative as well as generative approaches in the literature.

Chapter 8

Future Directions and Contributions

In this chapter, we first provide some thoughts on possible avenues to extend this research. We then conclude this dissertation with a summary of contributions.

8.1 Future Directions

8.1.1 Counterfactual Regression for Non-Binary Treatments

The approaches we developed in Chapters 5, 6 and 7 can only be applied to *binary*-treatment datasets. A possible future direction of this research is to develop methods that accommodate counterfactual regression when the treatment options are not binary — *e.g.*, when those treatments are:

1. categorical (*e.g.*, $\mathcal{T} = \{\text{bypass}, \text{stent}, \text{medication}\}$ for curing heart disease).

2. multiple-binary (*e.g.*, $\mathcal{T} = \{0,1\}^k$ — *i.e.*, combination[1] of a subset of medications for controlling depression).

3. real-valued $\mathcal{T} \subseteq \mathbb{R}$ (*e.g.*, the right dosage of insulin for a diabetic patient).

[1] However, the algorithm should be mindful of drug interactions; since some combinations might neutralize the effect of the treatment or worse, be detrimental to the patient's health.

8.1.2 Further Disentanglement of the Underlying Factors

Despite the success of the methods proposed in Chapters 6 and 7 in addressing causal inference for treatment effect estimation, no known algorithms can yet learn to perfectly decompose factors Δ and Υ. This goal is important because isolating Δ facilitates learning Context-Aware (CA) weights, which in turn can be exploited to enhance the quality of the causal effect estimation performance — *e.g.*, note the superior performance of H-VAE-CI (CA).

The results of our ablation study in Figure 7.5b, however, revealed that the currently used β-VAE does not help much with disentanglement of the underlying factors. Therefore, the proposed architectures and objective function ought to be responsible for most of the achieved decomposition. A future direction is to explore the use of better disentangling constraints (*e.g.*, works of [13] and [68]) to see if that would yield sharper results.

8.1.3 Survival Prediction

Many observational datasets deal with survival times, and include right-censored instances.[2] These instances should not be ignored as many datasets are $>70\%$ censored, and therefore, simply discarding those instances is not only data inefficient, but also causes estimation bias [52]. Moreover, selection bias is present in many observational survival datasets — *e.g.*, patients who need a liver transplant are ranked based on their MELD score[3] [105].

The Competing Risks (CR) framework [83] studies situations where more than one type of event compete to occur. For example, imagine trying to estimate the time until a patient, on the wait-list for a new liver, will live. Here, receiving a new liver is a competing risk besides death. The CR framework can be used to predict who should get the liver by estimating the expected

[2]That is, we only know a lower bound of the outcomes (*i.e.*, survival times), which is often time to *event* (*e.g.*, death). For instance, we may know that patient i survived at least 5 months, either because she was lost to follow-up, or the data collection period ended without the *event* occurring.

[3]The MELD score estimates a patient's chance of three months survival. Organ allocation is determined based on this score; *i.e.*, the sickest patients gets the organ first.

utility of a transplant. However, the scope of this framework does not cover addressing selection bias in order to answer the "*what if*" question — *e.g.*, patient P1 received a liver graft after being on the wait-list for 1 year; how long would she have lived *without this graft*? Developing survival prediction methods that can handle selection bias while exploiting the censored samples is still an open research question and the methods developed in this dissertation may provide a partial solution (see for example [22]).

8.1.4 Synthetic Observational Data Generation for Evaluation

The proposed synthetic data generation methodology in Chapter 4 can be extended in three directions:

1. The current pool size of possible treatments is $|T| = 2$ (*i.e.*, $t \in \{0, 1\}$). We can increase the pool size to have more options — this could cover combining several medications or other medical interventions. This is similar to the future work described in Section 8.1.1.

2. Explore ways to extend this methodology for sequential observational studies (*i.e.*, following a course of treatment). This is not trivial since the space of all possible decisions grows exponentially as we progress through the course of treatment. This is closely related to off-line Reinforcement Learning [101].

3. Create observational datasets that contain censored samples, *i.e.*, only a *lower bound* of the survival time of some patients is available. Such datasets can then be used to develop and evaluate methods that can not only address survival analysis/prediction tasks but also handle the intrinsic selection bias in observational data.

8.2 Contributions

This dissertation attempted to answer the question:

How can we improve estimating [individual] treatment effects from off-line datasets collected through observational studies?

To explore this question, we tried to address the following two challenges:

1. As counterfactual outcomes are **unobservable** [41], estimating treatment effects is more difficult than the generalization problem in the typical supervised learning paradigm.

2. Standard training data is based on an observational study, which means this data is: (i) **off-line** — *i.e.*, we cannot explore the effect of various treatments on the outcome, and (ii) likely to exhibit **selection bias** — *i.e.*, the treatment assignment can depend on the subjects' attributes.

This dissertation provides some solutions to these challenges in the form of the following four contributions:

The first contribution addressed the first challenge:

1. As mentioned earlier, counterfactual outcomes are unobservable in real-world observational datasets. This makes it challenging to properly evaluate different methods in terms of their performance in estimating treatment effects. In **Chapter 4**, we proposed an algorithm that can generate *realistic* synthetic observational datasets that exhibit specific degrees of selection bias. We then employed this algorithm to assess the performance of various contextual bandit methods in the literature.

The remaining contributions were related to the second challenge:

2. In order to *reduce* the selection bias, Johansson *et al*. [49] and Shalit *et al*. [95] proposed first learning a common representation space Φ (*cf*., [7]) shared between treatment arms. This is effective if the distributions of the transformed instances $\Phi(x)$ belonging to every treatment arm are similar — making the (transformed) dataset close to an RCT. However, this learned representation might not remove all the selection

bias, due to the existence of confounders. Reasonably, it should be possible to further alleviate the selection bias (in an attempt to *account for* it) by incorporating appropriate re-weighting schemes. In **Chapter 5**, we proposed the CounterFactual Regression with Importance Sampling Weights (CFR-ISW) method as a solution. We showed that CFR equipped with our proposed context-aware weights outperforms the vanilla CFR method.

3. Recall a general observational dataset includes random variables (X, T, Y), where without loss of generality, we can assume that X follows a(n unknown) joint probability distribution $\Pr(\, X \,|\, \Gamma, \Delta, \Upsilon, \Xi \,)$, where treatment T follows $\Pr(\, T \,|\, \Gamma, \Delta \,)$, and outcome Y^T follows $\Pr_T(\, Y^T \,|\, \Delta, \Upsilon \,)$, using Γ, Δ, and Υ to represent the three non-noise underlying (latent) factors that generate an observational dataset $\mathcal{D}$ (see Figure 1.3). We hypothesize that explicit identification of the underlying factors $\{\, \Gamma, \Delta, \Upsilon \,\}$ in observational datasets offers great insight to guide designing models that better handle selection bias and consequently achieve better performance in terms of estimating causal effects. In **Chapter 6**, we proposed the Disentangled Representations for CounterFactual Regression (DR-CFR) method to address this task and demonstrated that DR-CFR is [partially] successful in disentangling the underlying factors and significantly (statistical) outperforms the contending methods.

4. The majority of methods proposed to estimate treatment effects fall under the category of *discriminative* approaches — *i.e.*, learning a direct conditional model of y given x. A promising direction is to consider developing *generative* models, in an attempt to shed light on the true underlying data generating mechanism. We hypothesize that generative models can be employed to efficiently learn disentangled representations of the underlying factors of observational studies, which in turn is useful for the downstream task of counterfactual regression. Notable generative approaches (such as [69]) are not yet capable of addressing the problem of selection bias. In **Chapter 7**, we proposed the Variational

AutoEncoders for Causal Inference (VAE-CI) methods to address this shortcoming of the current state-of-the-art. Our empirical studies show that the Hybrid VAE-CI: (i) is successful in learning disentangled representations of Γ, Δ, and Υ, and (ii) significantly (statistical) outperforms all discriminative as well as generative contending methods in the literature.

References

[1] A. M. Alaa and M. van der Schaar, "Bayesian inference of individualized treatment effects using multi-task gaussian processes," in *Neural Information Processing Systems (NeurIPS)*, 2017.

[2] A. M. Alaa, M. Weisz, and M. Van Der Schaar, "Deep counterfactual networks with propensity-dropout," *arXiv preprint:1706.05966*, 2017.

[3] O. Atan, J. Jordon, and M. van der Schaar, "Deep-treat: Learning optimal personalized treatments from observational data using neural networks," in *Association for the Advancement of Artificial Intelligence (AAAI)*, 2018.

[4] S. Athey and G. W. Imbens, "The state of applied econometrics: Causality and policy evaluation," *Journal of Economic Perspectives*, vol. 31, no. 2, pp. 3–32, 2017.

[5] H. Bang and J. M. Robins, "Doubly robust estimation in missing data and causal inference models," *Biometrics*, vol. 61, no. 4, 2005.

[6] M. I. Belghazi, A. Baratin, S. Rajeshwar, *et al.*, "MINE: Mutual information neural estimation," in *International Conference on Machine Learning (ICML)*, 2018.

[7] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 35, no. 8, 2013.

[8] A. Beygelzimer and J. Langford, "The offset tree for learning with partial labels," in *ACM Special Interest Group on Knowledge Discovery and Data Mining (SIGKDD)*, ACM, 2009.

[9] C. Bonferroni, "Teoria statistica delle classi e calcolo delle probabilita," *Pubblicazioni del R Istituto Superiore di Scienze Economiche e Commericiali di Firenze*, vol. 8, pp. 3–62, 1936.

[10] L. Bottou, J. Peters, J. Q. Candela, *et al.*, "Counterfactual reasoning and learning systems: The example of computational advertising," *Journal of Machine Learning Research (JMLR)*, vol. 14, no. 1, 2013.

[11] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, 2001.

[12] C. Burgess, I. Higgins, A. Pal, *et al.*, "Understanding disentangling in β-VAE," *arXiv preprint:1804.03599*, 2018.

[13] R. T. Chen, X. Li, R. B. Grosse, and D. K. Duvenaud, "Isolating sources of disentanglement in variational autoencoders," in *Neural Information Processing Systems (NeurIPS)*, 2018.

[14] H. A. Chipman, E. I. George, and R. E. McCulloch, "Bayesian ensemble learning," in *Neural Information Processing Systems (NeurIPS)*, 2007.

[15] H. A. Chipman, E. I. George, R. E. McCulloch, *et al.*, "BART: Bayesian additive regression trees," *The Annals of Applied Statistics*, vol. 4, no. 1, 2010.

[16] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (elus)," *arXiv preprint arXiv:1511.07289*, 2015.

[17] M. Cuturi and A. Doucet, "Fast computation of wasserstein barycenters," in *International Conference on Machine Learning (ICML)*, 2014.

[18] R. H. Dehejia and S. Wahba, "Propensity score-matching methods for nonexperimental causal studies," *Review of Economics and statistics*, vol. 84, no. 1, 2002.

[19] V. Dorie, *NPCI: Non-parametrics for causal inference*, `https://github.com/vdorie/npci`, 2016.

[20] C. Drake, "Effects of misspecification of the propensity score on estimators of treatment effect," *Biometrics*, 1993.

[21] M. Dudık, J. Langford, and L. Li, "Doubly robust policy evaluation and learning," *International Conference on Machine Learning (ICML)*, 2011.

[22] M. Engelhard and R. Henao, "Disentangling whether from when in a neural mixture cure model for failure time data," in *International Conference on Artificial Intelligence and Statistics (AIStats)*, Proceedings of Machine Learning Research (PMLR), 2022, pp. 9571–9581.

[23] M. J. Funk, D. Westreich, C. Wiesen, T. Stürmer, M. A. Brookhart, and M. Davidian, "Doubly robust estimation of causal effects," *American Journal of Epidemiology*, vol. 173, no. 7, pp. 761–767, 2011.

[24] I. Goodfellow, J. Pouget-Abadie, M. Mirza, *et al.*, "Generative adversarial nets," in *Neural Information Processing Systems (NeurIPS)*, 2014.

[25] J. Gordon and J. M. Hernández-Lobato, "Combining deep generative and discriminative models for Bayesian semi-supervised learning," *Pattern Recognition*, vol. 100, 2020.

[26] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, "A kernel two-sample test," *Journal of Machine Learning Research (JMLR)*, vol. 13, no. March, 2012.

[27] S. Gruber and M. van der Laan, "TMLE: An R package for targeted maximum likelihood estimation," *Journal of Statistical Software*, vol. 51, 2012.

[28] R. Guo, L. Cheng, J. Li, P. R. Hahn, and H. Liu, "A survey of learning causality with data: Problems and methods," *ACM Computing Surveys (CSUR)*, vol. 53, no. 4, pp. 1–37, 2020.

[29] S. Harada and H. Kashima, "Counterfactual propagation for semi-supervised individual treatment effect estimation," *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD)*, 2020.

[30] N. Hassanpour, "Counterfactual reasoning in observational studies," *Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI)*, vol. 33, pp. 9886–9887, Jul. 2019.

[31] N. Hassanpour and R. Greiner, "A novel evaluation methodology for assessing off-policy learning methods in contextual bandits," in *Canadian AI*, 2018.

[32] ——, "Counterfactual regression with importance sampling weights," in *International Joint Conferences on Artificial Intelligence (IJCAI)*, 2019.

[33] ——, "Learning disentangled representations for counterfactual regression," in *International Conference on Learning Representations (ICLR)*, 2020.

[34] ——, "Variational auto-encoder architectures that excel at causal inference," *arXiv preprint arXiv:2111.06486*, 2021.

[35] D. Heckerman, C. Meek, and G. Cooper, "A Bayesian approach to causal discovery," *Computation, Causation, and Discovery*, vol. 19, 1999.

[36] J. J. Heckman, H. Ichimura, and P. Todd, "Matching as an econometric evaluation estimator," *The Review of Economic Studies*, vol. 65, no. 2, 1998.

[37] I. Higgins, L. Matthey, A. Pal, *et al.*, "β-VAE: Learning basic visual concepts with a constrained variational framework.," *International Conference on Learning Representations (ICLR)*, 2017.

[38] J. L. Hill, "Bayesian nonparametric modeling for causal inference," *Journal of Computational and Graphical Statistics*, vol. 20, no. 1, 2011.

[39] K. Hirano, G. W. Imbens, and G. Ridder, "Efficient estimation of average treatment effects using the estimated propensity score," *Econometrica*, vol. 71, no. 4, 2003.

[40] M. Hoffman, C. Riquelme, and M. Johnson, "The β-VAE's implicit prior," 2017. [Online]. Available: `http://bayesiandeeplearning.org/2017/papers/66.pdf`.

[41] P. W. Holland, "Statistics and causal inference," *Journal of the American statistical Association*, vol. 81, no. 396, 1986.

[42] D. G. Horvitz and D. J. Thompson, "A generalization of sampling without replacement from a finite universe," *Journal of the American Statistical Association*, vol. 47, no. 260, 1952.

[43] Hypericum Depression Trial Study Group and others, "Effect of Hypericum perforatum (St. John's Wort) in major depressive disorder: A randomized controlled trial," *Journal of the American Medical Association (JAMA)*, vol. 287, no. 14, 2002.

[44] K. Imai and M. Ratkovic, "Covariate balancing propensity score," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 76, no. 1, 2014.

[45] G. W. Imbens and J. M. Wooldridge, "Recent developments in the econometrics of program evaluation," *Journal of Economic Literature*, vol. 47, no. 1, 2009.

[46] G. W. Imbens and D. B. Rubin, *Causal Inference for Statistics, Social, and Biomedical Sciences: An Introduction*. Cambridge University Press, 2015.

[47] N. Japkowicz and S. Stephen, "The class imbalance problem: A systematic study," *Intelligent Data Analysis*, vol. 6, no. 5, 2002.

[48] J. Jiang, "A literature survey on domain adaptation of statistical classifiers," *URL: http://sifaka. cs. uiuc. edu/jiang4/domainadaptation/survey*, vol. 3, 2008.

[49] F. Johansson, U. Shalit, and D. Sontag, "Learning representations for counterfactual inference," in *International Conference on Machine Learning (ICML)*, 2016.

[50] F. D. Johansson, N. Kallus, U. Shalit, and D. Sontag, "Learning weighted representations for generalization across designs," *arXiv preprint:1802.08598*, 2018.

[51] F. D. Johansson, D. Sontag, and R. Ranganath, "Support and invertibility in domain-invariant representations," in *The 22nd International Conference on Artificial Intelligence and Statistics*, Proceedings of Machine Learning Research (PMLR), 2019, pp. 527–536.

[52] J. D. Kalbfleisch and R. L. Prentice, *The Statistical Analysis of Failure Time Data*. John Wiley & Sons, 2011, vol. 360.

[53] N. Kallus, "Deepmatch: Balancing deep covariate representations for causal inference using adversarial training," in *International Conference on Machine Learning (ICML)*, 2020, pp. 5067–5077.

[54] E. Karavani, Y. Shimoni, and C. Yanover, *IBM causal inference benchmarking framework*, `https://github.com/IBM-HRL-MLHLS/IBM-Causal-Inference-Benchmarking-Framework`, 2018.

[55] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations (ICLR)*, 2015.

[56] D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling, "Semi-supervised learning with deep generative models," in *Neural Information Processing Systems (NeurIPS)*, 2014.

[57] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *International Conference on Learning Representations (ICLR)*, 2014.

[58] D. Koller and N. Friedman, *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[59] K. Kuang, P. Cui, B. Li, M. Jiang, S. Yang, and F. Wang, "Treatment effect estimation with data-driven variable decomposition," in *Association for the Advancement of Artificial Intelligence (AAAI)*, 2017.

[60] L. Li, S. Chen, J. Kleban, and A. Gupta, "Counterfactual estimation and optimization of click metrics in search engines: A case study," in *Proceedings of the 24th International Conference on World Wide Web*, ACM, 2015.

[61] L. Li, W. Chu, J. Langford, and R. E. Schapire, "A contextual-bandit approach to personalized news article recommendation," in *Proceedings of the 19th International Conference on World Wide Web*, ACM, 2010.

[62] L. Li, W. Chu, J. Langford, and X. Wang, "Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms," in *Proceedings of the 4th International Conference on Web Search and Data Mining*, Hong Kong, 2011.

[63] Q. Li, "Literature survey: Domain adaptation algorithms for natural language processing," *Department of Computer Science The Graduate Center, The City University of New York*, 2012.

[64] S. Li and Y. Fu, "Matching on balanced nonlinear representations for treatment effects estimation," in *Neural Information Processing Systems (NeurIPS)*, 2017.

[65] R. J. Little and D. B. Rubin, *Statistical analysis with missing data*. John Wiley & Sons, 2019, vol. 793.

[66] Y.-E. Liu, T. Mandel, E. Brunskill, and Z. Popovic, "Trading off scientific knowledge and user learning with multi-armed bandits.," in *Educational Data Mining (EDM)*, 2014, pp. 161–168.

[67] F. Locatello, S. Bauer, M. Lucic, *et al.*, "Challenging common assumptions in the unsupervised learning of disentangled representations," in *International Conference on Machine Learning (ICML)*, 2019.

[68] R. Lopez, J. Regier, M. I. Jordan, and N. Yosef, "Information constraints on auto-encoding variational bayes," in *Neural Information Processing Systems (NeurIPS)*, 2018.

[69] C. Louizos, U. Shalit, J. M. Mooij, D. Sontag, R. Zemel, and M. Welling, "Causal effect inference with deep latent-variable models," in *Neural Information Processing Systems (NeurIPS)*, 2017.

[70] C. Louizos, K. Swersky, Y. Li, M. Welling, and R. Zemel, "The variational fair autoencoder," *arXiv preprint:1511.00830*, 2015.

[71] J. K. Lunceford and M. Davidian, "Stratification and weighting via the propensity score in estimation of causal treatment effects: A comparative study," *Statistics in Medicine*, vol. 23, no. 19, 2004.

[72] M. F. MacDorman and J. O. Atkinson, "Infant mortality statistics from the 1996 period linked birth/infant death dataset," *Monthly Vital Statistics Report*, vol. 46, no. 12, 1998.

[73] Y. Mansour, M. Mohri, and A. Rostamizadeh, "Domain adaptation: Learning bounds and algorithms," *arXiv preprint:0902.3430*, 2009.

[74] D. F. McCaffrey, G. Ridgeway, and A. R. Morral, "Propensity score estimation with boosted regression for evaluating causal effects in observational studies.," *Psychological Methods*, vol. 9, no. 4, 2004.

[75] A. McCallum, C. Pal, G. Druck, and X. Wang, "Multi-conditional learning: Generative/discriminative training for clustering and classification," in *Association for the Advancement of Artificial Intelligence (AAAI)*, 2006.

[76] J. Neyman, "Sur les applications de la théorie des probabilités aux experiences agricoles: Essai des principes," *Roczniki Nauk Rolniczych*, vol. 10, 1923.

[77] A. Y. Ng and M. I. Jordan, "On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes," in *Neural Information Processing Systems (NeurIPS)*, 2002.

[78] V. M. Patel, R. Gopalan, R. Li, and R. Chellappa, "Visual domain adaptation: A survey of recent advances," *IEEE Signal Processing Magazine*, vol. 32, no. 3, 2015.

[79] J. Pearl, *Causality*. Cambridge University Press, 2009.

[80] J. Pearl, M. Glymour, and N. P. Jewell, *Causal inference in statistics: A primer*. John Wiley & Sons, 2016.

[81] J. Pearl and D. Mackenzie, *The book of why: the new science of cause and effect*. Basic Books, 2018.

[82] J. Peters, D. Janzing, and B. Schölkopf, *Elements of causal inference: foundations and learning algorithms*. MIT press, 2017.

[83] M. Pintilie, *Competing risks: a practical perspective*. John Wiley & Sons, 2006, vol. 58.

[84] C. E. Rasmussen and C. K. Williams, *Gaussian Processes For Machine Learning*. MIT Press Cambridge, 2006, vol. 1.

[85] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," *International Conference on Machine Learning (ICML)*, 2014.

[86] J. M. Robins, A. Rotnitzky, and L. P. Zhao, "Estimation of regression coefficients when some regressors are not always observed," *J. Am. Stat. Assoc*, vol. 89, no. 427, 1994.

[87] J. Rollinson and E. Brunskill, "From predictive models to instructional policies," *International Educational Data Mining Society*, 2015.

[88] P. R. Rosenbaum and D. B. Rubin, "The central role of the propensity score in observational studies for causal effects," *Biometrika*, 1983.

[89] D. B. Rubin, "Matching to remove bias in observational studies," *Biometrics*, 1973.

[90] ——, "Estimating causal effects of treatments in randomized and nonrandomized studies," *Journal of Educational Psychology*, vol. 66, no. 5, 1974.

[91] C. Russell, M. J. Kusner, J. Loftus, and R. Silva, "When worlds collide: Integrating different counterfactual assumptions in fairness," in *Neural Information Processing Systems (NeurIPS)*, 2017.

[92] T. Schnabel, A. Swaminathan, A. Singh, N. Chandak, and T. Joachims, "Recommendations as treatments: Debiasing learning and evaluation," in *International Conference on Machine Learning (ICML)*, New York, NY, USA, 2016.

[93] B. Schölkopf and J. von Kügelgen, "From statistical to causal learning," *arXiv preprint arXiv:2204.00607*, 2022.

[94] W. H. Sewell and V. P. Shah, "Social class, parental encouragement, and educational aspirations," *American journal of Sociology*, vol. 73, no. 5, 1968.

[95] U. Shalit, F. D. Johansson, and D. Sontag, "Estimating individual treatment effect: Generalization bounds and algorithms," in *International Conference on Machine Learning (ICML)*, 2017.

[96] C. Shi, D. Blei, and V. Veitch, "Adapting neural networks for the estimation of treatment effects," in *Neural Information Processing Systems (NeurIPS)*, 2019.

[97] H. Shimodaira, "Improving predictive inference under covariate shift by weighting the log-likelihood function," *Journal of Statistical Planning And Inference*, vol. 90, no. 2, 2000.

[98] Y. Shimoni, C. Yanover, E. Karavani, and Y. Goldschmnidt, "Benchmarking framework for performance-evaluation of causal inference analysis," *arXiv preprint:1802.05046*, 2018, `www.cmu.edu/acic2018/data-challenge/`.

[99] E. H. Simpson, "The interpretation of interaction in contingency tables," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 13, no. 2, 1951.

[100] E. A. Stuart, "Matching methods for causal inference: A review and a look forward," *Statistical Science: a Review Journal of the Institute of Mathematical Statistics*, vol. 25, no. 1, 2010.

[101] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 1. MIT Press Cambridge, 1998, vol. 1.

[102] A. Swaminathan and T. Joachims, "Batch learning from logged bandit feedback through counterfactual risk minimization," *Journal of Machine Learning Research (JMLR)*, vol. 16, 2015.

[103] ——, "Counterfactual risk minimization: Learning from logged bandit feedback," in *International Conference on Machine Learning (ICML)*, 2015.

[104] ——, "The self-normalized estimator for counterfactual learning," in *Neural Information Processing Systems (NeurIPS)*, 2015.

[105] UNOS, *United Network for Organ Sharing. Model for End-Stage Liver Disease (MELD)*, `https://optn.transplant.hrsa.gov/resources/allocation-calculators/meld-calculator/`.

[106] M. J. Van der Laan and S. Rose, *Targeted learning: causal inference for observational and experimental data*. Springer Science & Business Media, 2011.

[107] A. J. Vickers, "Whose data set is it anyway? sharing raw data from randomized trials," *Trials*, vol. 7, no. 1, 2006.

[108] A. J. Vickers, R. W. Rees, C. E. Zollman, *et al.*, "Acupuncture for chronic headache in primary care: Large, pragmatic, randomised trial," *BMJ*, vol. 328, no. 7442, 2004.

[109] S. Wager and S. Athey, "Estimation and inference of heterogeneous treatment effects using random forests," *Journal of the American Statistical Association*, vol. 113, no. 523, 2018.

[110] C.-C. Wang, S. R. Kulkarni, and H. V. Poor, "Bandit problems with side observations," *IEEE Transactions on Automatic Control*, vol. 50, no. 3, 2005.

[111] M. Wang and W. Deng, "Deep visual domain adaptation: A survey," *Neurocomputing*, vol. 312, 2018.

[112] J. Wen, N. Hassanpour, and R. Greiner, "Weighted gaussian process for estimating treatment effect," in *What If? Inference and Learning of Hypothetical and Counterfactual Interventions in Complex Systems Workshop*, Neural Information Processing Systems (NeurIPS), 2016.

[113] Q. Xu and Q. Yang, "A survey of transfer and multitask learning in bioinformatics," *Journal of Computing Science and Engineering*, vol. 5, no. 3, 2011.

[114] L. Yao, S. Li, Y. Li, M. Huai, J. Gao, and A. Zhang, "Representation learning for treatment effect estimation from observational data," in *Neural Information Processing Systems (NeurIPS)*, 2018.

[115] J. Yoon, J. Jordon, and M. van der Schaar, "GANITE: Estimation of individualized treatment effects using generative adversarial nets," in *International Conference on Learning Representations (ICLR)*, 2018.

[116] B. Zadrozny, "Learning and evaluating classifiers under sample selection bias," in *International Conference on Machine Learning (ICML)*, ACM, 2004.

[117] R. Zemel, Y. Wu, K. Swersky, T. Pitassi, and C. Dwork, "Learning fair representations," in *International Conference on Machine Learning (ICML)*, 2013.

[118] K. Zhang, B. Schölkopf, K. Muandet, and Z. Wang, "Domain adaptation under target and conditional shift," in *International Conference on Machine Learning (ICML)*, 2013.

[119] W. Zhang, L. Liu, and J. Li, "Treatment effect estimation with disentangled latent factors," in *Association for the Advancement of Artificial Intelligence (AAAI)*, 2021.

[120] Z. Zhang, Q. Lan, L. Ding, Y. Wang, N. Hassanpour, and R. Greiner, "Reducing selection bias in counterfactual reasoning for individual treatment effects estimation," *Neural Information Processing Systems (NeurIPS) Workshop "Do the Right Thing": Machine Learning and Causal Inference for Improved Decision Making*, 2019.

Appendix A

How to distinguish Υ from Δ in DR-CFR?

It seems if you replace Δ by (Δ, Υ) and Υ by 0, the loss would always reduce since the discrepancy term reduces and the other terms do not change. So how could you distinguish Υ from Δ by minimizing the objective function?

First, the regularization term $Reg(\Gamma, \Delta, \Upsilon, h^0, h^1, \pi_0)$ assures that the intersection between each pair of the representations learned for Γ, Δ, and Υ is empty.

The discrepancy term ensures that the representation learned for Υ (let us call it $Rep(\Upsilon)$) embeds no information about the confounders Δ; and only embeds information about Υ factors, if any. However, based on the objective function in Equations (6.1-6.4), it appears that there are no explicit constraints on what information $Rep(\Delta)$ can learn to embed: it could be Δ and even some Υ.

This question (in the title) is about the extreme case: why optimization does not lead to learning a $Rep(\Upsilon)$ that embeds no information (*i.e.*, learns only noise) and a $Rep(\Delta)$ that learns to embed both Δ and Υ.

We do not have a theoretical proof that this won't occur, but have a hypothesis that is also supported by empirical evidence. As mentioned earlier, the discrepancy term works as a sieve, allowing only the Υ factors (ones that are related to only Y, but not X nor T) to be learned by $Rep(\Upsilon)$. If all Υ are represented by $Rep(\Upsilon)$, then due to regularization, $Rep(\Delta)$ will not represent

	A	B	C	D
1	dataset	30, 30	45, 15	60, 0
2	0_0_4	0.1906	0.2045	0.3886
3	0_0_8	0.2847	0.3536	0.6634
4	0_4_0	0.3427	0.3732	0.3875
5	0_4_4	0.3204	0.3593	0.4377
6	0_4_8	0.3682	0.4461	0.5673
7	0_8_0	0.345	0.3565	0.3863
8	0_8_4	0.3639	0.4087	0.5168
9	0_8_8	0.4963	0.5618	0.5676
10	4_0_4	0.202	0.216	0.2679
11	4_0_8	0.2302	0.2724	0.4534
12	4_4_0	0.2868	0.2976	0.3554
13	4_4_4	0.2406	0.2756	0.3441
14	4_4_8	0.4069	0.5001	0.5051
15	4_8_0	0.3041	0.3569	0.3776
16	4_8_4	0.4069	0.4524	0.4727
17	4_8_8	0.5403	0.5604	0.5718
18	8_0_4	0.2019	0.2503	0.2759
19	8_0_8	0.2556	0.3121	0.3708
20	8_4_0	0.2336	0.2618	0.3583
21	8_4_4	0.3208	0.3111	0.3547
22	8_4_8	0.4069	0.4918	0.5669
23	8_8_0	0.3187	0.3288	0.3706
24	8_8_4	0.4535	0.5091	0.5654
25	8_8_8	0.5789	0.5843	0.5851
26	AVG	0.3375	0.3769	0.4463

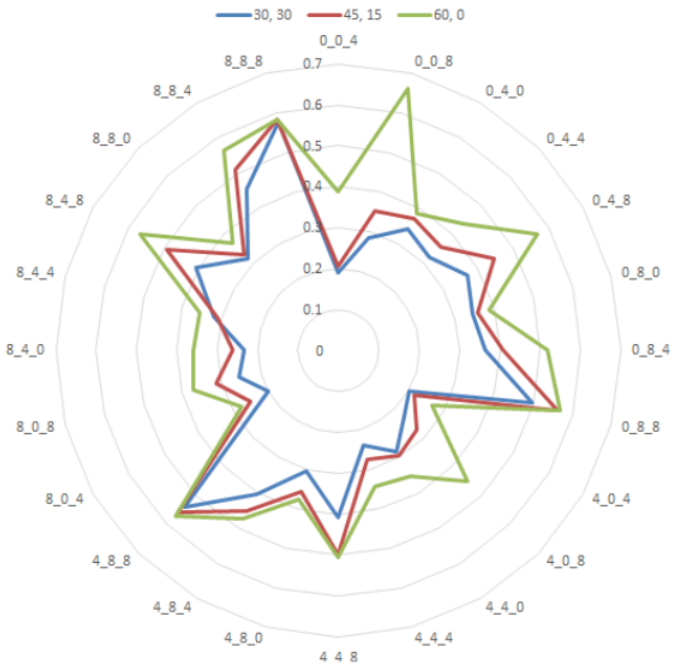


Figure A.1

any Υ and would only represent Δ. This is the desired case. However, if some Υ are left out and not represented by $Rep(\Upsilon)$, $Rep(\Delta)$ must compensate and represent them, which effectively will reduce the capacity of $Rep(\Delta)$ to represent Δ — *i.e.*, factors that cannot be represented by any of the other components.

Our hypothesis is that this bottle-neck implicitly derives the optimization procedure to learn to distinguish Δ from Υ.

We empirically tested this hypothesis by limiting the capacity of the $Rep(\Upsilon)$ network (by reducing the number of hidden neurons from 30 to 15 to 0) while increasing the capacity of $Rep(\Delta)$ (from 30 to 45 to 60 respectively). Our performance results are best at the (30, 30) setting and worsen as the capacity of $Rep(\Upsilon)$ is decreased, regardless of whether the capacity of $Rep(\Delta)$ is increased. A particularly important observation is that the increase in the capacity of $Rep(\Delta)$ does not replace the need for having a separate $Rep(\Upsilon)$ network: It must be $Rep(\Upsilon)$ that embeds Υ, not $Rep(\Delta)$.

Appendix B

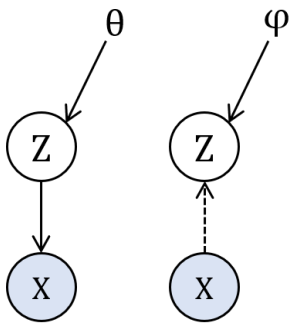
M1 and M2 Variational Auto-Encoders

As the first proposed model, the M1 VAE is the conventional model that is used to learn representations of data [57], [85]. These features are learned from the covariate matrix X only (*i.e.*, unsupervised). Figure B.1a illustrates the decoder and encoder of the M1 VAE. Note the graphical model on the left depicts the decoder; and the one on the right depicts the encoder, which has arrows going the other direction.

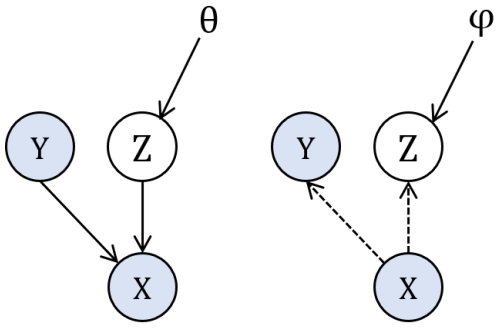
Proposed by Kingma *et al.* [56], the M2 model was an attempt to incorporate the information in target Y into the representation learning procedure. This results in learning representations that separate specifications of individual targets from general properties shared between various targets. In case of digit generation, this translates into separating specifications that distinguish each digit from writing style or lighting condition. Figure B.1b illustrates the decoder and encoder of the M2 VAE.

Stacking the M1 and M2 models produced their best results (see Figure B.1c): first learn a representation Z_1 from the raw covariates, then find a second representation Z_2, now learning from Z_1 (instead of the raw data) as well as the target information. In our work, the target information includes the treatment bit T as well as the observed outcome Y.[1] This additional information helps the model to learn more expressive representations, which was not possible with the unsupervised M1 model.

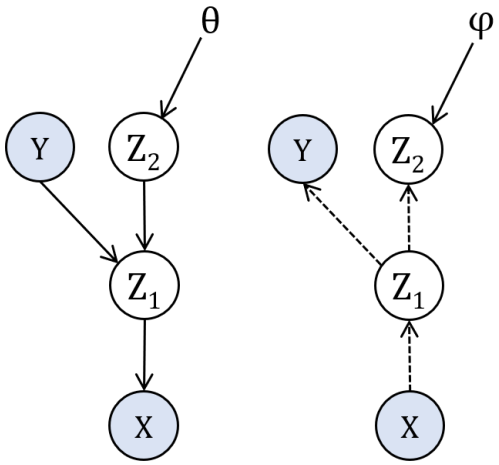
[1]Therefore, we require multiple stacked models here.



(a) M1 model



(b) M2 model



(c) M1+M2 model

Figure B.1: Decoders (parametrized by θ) and encoders (parametrized by φ) of the M1, M2, and M1+M2 VAEs.

Appendix C

Analysis of the Effect of $\beta = 0$ in H-VAE-CI

Our initial hypothesis in using β-VAE in Chapter 7 was that it might help *further* disentangle the underlying factors, in addition to the other constraints already in place (*i.e.*, the architecture as well as the discrepancy penalty). However, Figure 7.5b suggests that close-to-zero or even zero βs also work effectively. Our hypothesis is that the H-VAE-CI's architecture already takes care of decomposing the Γ, Δ, and Υ factors, without needing the help of a KLD penalty.[1]

In order to validate this hypothesis, we examined the decomposition tables of H-VAE-CI (similar to the performance reported in the green table in Figure 7.3) for extreme configurations with $\beta = 0$ and observed that they were all effective at decomposing the underlying factors Γ, Δ, and Υ. Figure C.1 shows several of these tables. This means either of the following is happening:

(i) β-VAE is not the best performing disentangling method and other disentangling constraints should be used instead — *e.g.*, works of Chen *et al.* [13] and Lopez *et al.* [68].

(ii) It is theoretically impossible to achieve disentanglement without some supervision [67], which might not be possible to provide in this task.

Exploring these options is however out of the scope of this study and is left to

[1] Therefore, it appears that we can safely drop out the KLD term altogether; which can significantly reduce the model and time complexity.

H-VAE-CI							
	Z1	Z2	Z3	Z4	Z5	Z6	Z7
Γ	0.2181	0.2185	**1.8791**	1.7711	0.2164	0.2190	0.2039
Δ	0.6041	0.6051	0.6142	0.6308	**0.8688**	0.8315	**0.6138**
ϒ	**0.8523**	0.8552	0.3321	0.3834	0.7552	0.7859	**0.7384**

H-VAE-CI							
	Z1	Z2	Z3	Z4	Z5	Z6	Z7
Γ	0.2242	0.2182	**0.7439**	0.6770	0.2373	0.2583	0.2189
Δ	0.3014	0.2963	0.2612	0.3169	**0.6051**	0.5845	**0.7048**
ϒ	**0.5385**	0.5430	0.3211	0.3303	0.4394	0.4412	**0.4571**

H-VAE-CI							
	Z1	Z2	Z3	Z4	Z5	Z6	Z7
Γ	0.4254	0.4493	**0.7090**	0.6872	0.3823	0.3771	0.3874
Δ	0.6438	0.6461	0.2750	0.3129	**0.7452**	0.7569	**0.8237**
ϒ	**0.7760**	1.1200	0.3137	0.3480	0.7240	0.7464	**0.6717**

H-VAE-CI							
	Z1	Z2	Z3	Z4	Z5	Z6	Z7
Γ	0.3646	0.3643	**1.1457**	0.8659	0.3942	0.4069	0.3166
Δ	0.5127	0.5307	0.6463	0.5794	**0.7016**	0.6717	**0.7652**
ϒ	**0.5565**	0.5780	0.4260	0.3964	0.4119	0.4234	**0.4534**

H-VAE-CI							
	Z1	Z2	Z3	Z4	Z5	Z6	Z7
Γ	0.8821	0.8752	**0.5805**	0.5782	0.3326	0.3309	0.4006
Δ	1.2542	1.2480	0.2843	0.3488	**0.8553**	0.8568	**0.9392**
ϒ	**1.914**	1.923	0.4498	0.4797	0.7791	0.7757	**0.7969**

H-VAE-CI							
	Z1	Z2	Z3	Z4	Z5	Z6	Z7
Γ	0.0850	0.0875	**1.8791**	1.7711	0.1464	0.1459	0.1833
Δ	0.6107	0.6085	0.6142	0.6308	**0.7851**	0.7937	**0.6878**
ϒ	**0.8349**	0.8242	0.3321	0.3834	0.5177	0.5073	**0.6832**

Figure C.1: Decomposition tables for H-VAE-CI with $\beta = 0$.

future work.