Graph Intervention Networks for Causal Effect Estimation

Anonymous Author(s)
Affiliation
Address
email

Abstract

We address the estimation of conditional average treatment effects (CATEs) when treatments are graph-structured (e.g., molecular graphs of drugs). Given a weak condition on the effect, we propose a plug-in estimator that decomposes CATE estimation into separate, simpler optimization problems. Our estimator (a) isolates the causal estimands (reducing regularization bias), and (b) allows one to plug in arbitrary models for learning. In experiments with small-world and molecular graphs, we show that our approach outperforms prior approaches and is robust to varying selection biases.

1 Introduction

Estimating covariate-level causal effects, so-called *conditional average treatment effects* (CATEs), from observational data is a fundamental problem across many domains. Examples include understanding the effects of non-pharmaceutical interventions on the transmission of COVID-19 in a specific region [13], on whether school meal programs improve child health [14], and the effects of chemotherapy drugs on cancer patients [49]. Supervised learning methods face two challenges in such settings: (i) *missing counterfactuals*, the fact that we only observe one treatment for each unit means models must extrapolate to new treatments without access to ground truth, and (ii) *confounding factors* that affect both treatment assignment and the outcome means that extrapolation from observation to intervention requires assumptions. Many approaches have been proposed to overcome these issues [1, 2, 3, 4, 5, 6, 7, 10, 11, 17, 20, 21, 23, 24, 25, 27, 29, 30, 34, 38, 39, 40, 42, 49, 52, 53, 57, 61, 64].

In many cases, treatments are naturally represented as graphs: the spatial network of regional transmission, the nutritional content of meals, and molecular structure of chemotherapy drugs. However, the vast majority of prior work operates with categorical (mostly binary) or continuous scalar treatments – we describe a notable exception [18] in Section 2. Adapting these methods is problematic: even if one was willing to suffer the exponential increases in runtime and/or memory, they would still not leverage the structure of graph data.

In this work, we propose *Graph Intervention Networks* (GIN), a CATE estimation method designed for *graph interventions*. Concretely, our contributions are:

- Given a weak condition on the structural form of the outcomes (Assumption 3), we generalize a well-known decomposition of CATE estimation [44] to graph treatments[1].
- This decomposition reveals a learnable pseudo-outcome target that isolates the causal component of the observed signal by partialling out confounding associations. To derive it, we introduce a generic variation of the propensity score idea [45], which we term *propensity features*.

[1] In fact, this generalization applies to any treatments that can be vectorized by feature extraction (pre-defined or learned). We focus on graph treatments as they are a generic representation with many practical applications.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

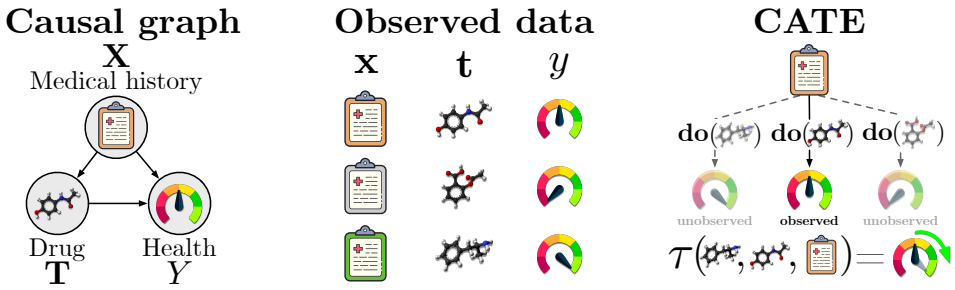

Figure 1: **Pictorial depiction for CATE estimation with graph interventions.** *Left*: Causal graph of problem setup with covariates $\mathbf{X}$, treatment $\mathbf{T}$, and outcome Y. *Center*: Observational data the estimator has access to, typically containing only one outcome per unit. *Right*: the CATE is defined as the difference between the expected outcomes given a fixed unit and a pair of treatments.

- Our method allows one to plug in any supervised learning methods for learning nuisance and target functions. We extend recent work on *plug-in estimators* [30] focusing on binary treatments to graph-structured treatments and integrating representation learning into CATE estimation.
- We show in experiments with small-world networks and molecular graphs that previous approaches can learn causal effects sub-optimally. In contrast, GIN successfully estimates these effects and is robust to varying levels of confounding.

2 Related Work

Graph interventions. Closest to our work is GraphITE [18], a method that learns representations of graph interventions for CATE estimation. They propose to minimize prediction loss plus a regularization term that aims to control for confounding based on the Hilbert-Schmidt Independence Criterion (HSIC) [16]. This technique suffers from two drawbacks: (i) $\mathcal{O}(b^2)$ *scaling*: the HSIC requires multiplication of kernel matrices and so scales quadratically in the batch size b, for each optimization step; (ii) *Hyper-parameter selection*: The HSIC term is multiplied by a weight λ; a hyper-parameter which controls the trade-off between minimizing loss and confounding. However, it is unclear how to select this hyper-parameter, as ground-truth CATEs are never observed, and empirical loss does not bound CATE estimation error [1, 40, 50].

Plug-in estimators. A recent line of work for CATE estimation derives *plug-in estimators* [9].[2] These work by decomposing CATE estimation into multiple sub-problems (so-called *nuisance components*), each solvable using any supervised learning method [11, 17, 28, 30, 40]. Currently, these approaches are limited to binary treatment setups. Our approach is inspired by these methods, extending plug-in estimation to graph treatment settings.

CATE estimation with neural networks. Neural network CATE estimators typically use separate prediction heads for each treatment option [26, 33, 39, 48, 49, 50, 51]. This architectural design reduces one source of regularization bias: the influence of the treatment indicator variable might be lost in the high-dimensional network representations. Extending this idea directly to graph-structured treatments would not only be computationally expensive, but would also not be able to make use of graph features or learn graph representations.

Multiple treatments. While Inverse Probability Weighting (IPW) [31, 32, 63] is a popular technique for estimating effects with multiple, categorical treatments, it requires estimating the propensity density which is infeasible in settings with hundreds or thousands of graph treatments; some of which may have not been seen during training. Nabi et al. [38] propose a framework for sufficient dimensionality reduction of high-dimensional treatments based on semiparametric inference theory. Besides relying on IPW, this approach is designed for average treatment effects (not CATEs).

[2] These are also called *meta-learners*. To avoid confusion with *meta-learning*, we call these *plug-in estimators*.

3 Preliminaries

3.1 Conditional Average Treatment Effects (CATEs)

Imagine a dataset where each example $(\mathbf{x}_i, \mathbf{t}_i, y_i) \in \mathcal{D}$ represents a hospital patient's medical history record $\mathbf{x}_i$, prescribed drug treatment $\mathbf{t}_i$, and health outcome y_i, as illustrated in Figure 1 (*Center*). Further, we wish to understand how changing the treatment changes a patient's health outcome. The CATE, $\tau\left(\mathbf{t}', \mathbf{t}_i, \mathbf{x}_i\right)$, describes the expected change in outcome for individuals with history $\mathbf{x}_i$, when treatment $\mathbf{t}_i$ is replaced by $\mathbf{t}'$, depicted in Figure 1 (*Right*). In real-world scenarios, we only observe one outcome for each patient at one treatment level. Further, the patient's pre-treatment health conditions $\mathbf{x}_i$ influence both the doctor's treatment prescription and outcome, thereby *confounding* the effect of the treatment on the outcome.

Formally, we have the dataset $\mathcal{D} = \left\{(\mathbf{x}_i, \mathbf{t}_i, y_i)\right\}_{i=1}^{n}$ sampled from a joint distribution $p\left(\mathbf{X}, \mathbf{T}, Y\right)$, where $Y = f\left(\mathbf{X}, \mathbf{T}\right) + \varepsilon$, as depicted in Figure 1 (*Left*). We define the causal effect of fixing *treatment* variable $\mathbf{T} \in \mathcal{T}$ to a value $\mathbf{t}$ on *outcome* variable $Y \in \mathbb{R}$ using the do-operator [41] as $\mathbb{E}\left[Y \mid \text{do}(\mathbf{T} = \mathbf{t})\right]$. Crucially, this estimate differs from the conditional expectation $\mathbb{E}\left[Y \mid \mathbf{T} = \mathbf{t}\right]$ in that it describes the effect of an external entity *intervening* on $\mathbf{T}$ by fixing it to a value $\mathbf{t}$. We further condition on pre-treatment *covariates* $\mathbf{X}$ to define the conditional causal estimand $\mathbb{E}\left[Y \mid \mathbf{X} = \mathbf{x}, \text{do}(\mathbf{T} = \mathbf{t})\right]$. The *conditional average treatment effect* (CATE) is the difference between expected outcomes at different treatment values $\mathbf{t}, \mathbf{t}'$ for given covariates $\mathbf{x}$,

$$\tau(\mathbf{t}', \mathbf{t}, \mathbf{x}) \triangleq \underbrace{\mathbb{E}\left[Y \mid \mathbf{X} = \mathbf{x}, \text{do}(\mathbf{T} = \mathbf{t}')\right]}_{=:\mu_{\mathbf{t}'}(\mathbf{x})} - \underbrace{\mathbb{E}\left[Y \mid \mathbf{X} = \mathbf{x}, \text{do}(\mathbf{T} = \mathbf{t})\right]}_{=:\mu_{\mathbf{t}}(\mathbf{x})}, \tag{1}$$

where $\mu_{\mathbf{t}}\left(\mathbf{x}\right)$ is defined as the *expected outcome* for a covariate vector $\mathbf{x}$ under treatment $\mathbf{t}$.

Because we do not observe both treatments $\mathbf{t}, \mathbf{t}'$ for a single covariate $\mathbf{x}$, we need to make assumptions that allow us to identify the CATE from observational data.

Assumption 1. *(Unconfoundedness) There are no confounders of the effect between* $\mathbf{T}$ *and* Y *beyond* $\mathbf{X}$*. Therefore,* $Pr\left(Y \leq y \mid \mathbf{x}, do(\mathbf{t})\right) = Pr\left(Y \leq y \mid \mathbf{x}, \mathbf{t}\right)$*, for all* $(\mathbf{x}, \mathbf{t}, y)$*.*

Assumption 2. *(Overlap) It holds that* $0 < p\left(\mathbf{t} \mid \mathbf{x}\right) < 1$*, for all* $(\mathbf{x}, \mathbf{t}, y)$*.*

Assumption 2 means that all sub-populations have some probability of receiving any value of treatment (otherwise, some $\tau(\mathbf{t}', \mathbf{t}, \mathbf{x})$ may be undefined or impossible to estimate.) These assumptions allow us to estimate the causal quantity $\tau(\mathbf{t}', \mathbf{t}, \mathbf{x})$ through statistical estimands:

$$\tau\left(\mathbf{t}', \mathbf{t}, \mathbf{x}\right) = \mu_{\mathbf{t}'}\left(\mathbf{x}\right) - \mu_{\mathbf{t}}\left(\mathbf{x}\right) = \mathbb{E}\left[Y \mid \mathbf{X} = \mathbf{x}, \mathbf{T} = \mathbf{t}'\right] - \mathbb{E}\left[Y \mid \mathbf{X} = \mathbf{x}, \mathbf{T} = \mathbf{t}\right]. \tag{2}$$

While one can model $\mu_t(\mathbf{x})$ with regression models, such approaches suffer from *regularization bias* [10, 28, 30] due to two factors: (i) associations between $\mathbf{X}$ and $\mathbf{T}$, which are to be expected under confounding, and (ii) regularizing for predictive performance instead of effect estimation. Mitigating such biases usually works by transforming the representation of $\mathbf{X}$ and $\mathbf{T}$ to a (regularized) regression problem that will still provide the causal effect, e.g., by using nuisance components.

3.2 Robinson Decomposition

The *Robinson decomposition* [44] is a mathematical expression which allows one to construct a learnable objective of the CATE in binary treatment settings. The *R-learner* [40] is a plug-in estimator that exploits this expression. Its modeling flexibility, combined with the avoidance of regularization bias, makes it a compelling estimator.

Let the treatment variable be $T \in \{0, 1\}$ and the outcome model $p\left(y \mid \mathbf{x}, \mathbf{t}\right)$ parameterized as

$$Y = f(\mathbf{X}, T) + \varepsilon \triangleq \mu_0(\mathbf{X}) + T \times \tau_b(\mathbf{X}) + \varepsilon, \tag{3}$$

where we define error term ε such that $\mathbb{E}\left[\varepsilon \mid \mathbf{x}, \mathbf{t}\right] = \mathbb{E}\left[\varepsilon \mid \mathbf{x}\right] = 0$, and $\tau_b\left(\mathbf{x}\right) \triangleq \tau\left(1, 0, \mathbf{x}\right)$.

Define the *propensity score* $e\left(\mathbf{x}\right) \triangleq p\left(T = 1 \mid \mathbf{x}\right)$ and the *conditional mean outcome* as

$$m\left(\mathbf{x}\right) \triangleq \mathbb{E}\left[Y \mid \mathbf{x}\right] = \mu_0\left(\mathbf{x}\right) + e\left(\mathbf{x}\right)\tau_{\text{b}}\left(\mathbf{x}\right). \tag{4}$$

From model (3) and the previous definitions, it follows that

$$Y - m(\mathbf{X}) = \big(T - e(\mathbf{X})\big)\,\tau_b(\mathbf{X}) + \varepsilon, \tag{5}$$

allowing us to define the estimator

$$\widehat{\tau}_{\mathrm{b}}(\cdot) = \arg\min_{\tau_{\mathrm{b}}} \left\{ \frac{1}{n}\sum_{i=1}^{n} \Big(\tilde{y}_i - \tilde{t}_i \times \tau_{\mathrm{b}}(\mathbf{x}_i)\Big)^2 + \Lambda\big(\tau_{\mathrm{b}}(\cdot)\big) \right\}, \tag{6}$$

where $\tilde{y}_i \triangleq y_i - \widehat{m}(\mathbf{x}_i)$ and $\tilde{t}_i \triangleq t_i - \widehat{e}(\mathbf{x}_i)$ are pseudo-data points defined through estimated nuisance functions $\widehat{m}(\cdot), \widehat{e}(\cdot)$, which can be learned separately with any supervised learning algorithm.

3.3 Graph Neural Networks

We consider interventions with relational structures represented by graphs, such as molecules consisting of atoms (nodes) joined by chemical bonds (edges). Special cases include images and sentences (graphs with grid-structure), as well as real-valued vectors (graphs with independent nodes) [8]. Here, a graph $\mathcal{G} := (\mathcal{V}, \mathcal{E})$ consists of a set of vertices $v^{(i)} \in \mathcal{V}$, labeled with vectors $\mathbf{v}^{(i)} \in \mathbb{R}^{m_v}$ and a set of directed edges $e^{ij} \in \mathcal{E}$ from vertex $v^{(i)}$ to $v^{(j)}$, labeled with vectors $\mathbf{e}^{ij} \in \mathbb{R}^{m_e}$. To enable our estimator to operate over graph-structured data, we resort to graph neural networks (GNNs), whose parameters, like those of other neural networks, can be learned through backpropagation.

4 Graph Intervention Networks

Our goal is to estimate the CATE $\tau(\mathbf{t}', \mathbf{t}, \mathbf{x})$ for graph interventions $\mathbf{t} \triangleq \mathcal{G}, \mathbf{t}' \triangleq \mathcal{G}'$. A good estimator should correct for the confounding of $\mathbf{X}$ on $\mathbf{T}$ and Y in the data. For this we appeal to the Robinson decomposition, which has inspired flexible CATE estimation for binary treatments [6, 10, 34, 40].

4.1 Generalizing the Robinson Decomposition

A generic way to extend the Robinson decomposition to arbitrary treatments is to learn a model $\hat{f}(\mathbf{X}, \mathbf{T})$ defined over the entire outcome surface, via mean outcome $\hat{m}(\mathbf{X})$ and treatment conditional density $p(\mathbf{T} \mid \mathbf{X})$. In this case, we fit the relationship

$$Y - m(\mathbf{X}) = f(\mathbf{X}, \mathbf{T}) - e^p(\mathbf{X}) + \varepsilon, \quad \text{where} \quad e^p(\mathbf{x}) \triangleq \mathbb{E}\left[f(\mathbf{X}, \mathbf{T}) \mid \mathbf{x}\right]. \tag{7}$$

To learn $f(\cdot,\cdot)$ from a dataset $\mathcal{D} = \big\{(\mathbf{x}_i, \mathbf{t}_i, y_i)\big\}_{i=1}^{n}$ we need to solve,

$$\hat{f} = \arg\min_{f \in \mathcal{F}} \frac{1}{N}\sum_{i=1}^{N} \Big[\big\{y_i - \hat{m}(\mathbf{x}_i)\big\} - \big(f(\mathbf{x}_i, \mathbf{t}_i) - \hat{e}^p(\mathbf{x}_i)\big)\Big]^2, \tag{8}$$

where $\mathcal{F}$ is some function space and $\hat{m}$ is a plug-in finite sample estimate of m. Because e^p contains f, we need to estimate it, which we denote $\hat{e}^p$. One solution is to estimate the propensity $p(\mathbf{T} \mid \mathbf{X})$ and use it to compute $\hat{e}^p$. However, this approach requires conditional density estimation over high-dimensional graphs, which remains an open research question [62], and is prone to high variance [54]. Further, to compute $\hat{e}^p$ from it, one has to resort to Monte Carlo evaluation. Another option is to solve for f, fix it, then estimate $\hat{e}^p$ using regression from finite samples, and iterate to a fixed point. However, there is a fundamental issue with this approach: we are typically interested in regularizing the causal effect directly as opposed to the generic regression function. This is why, for instance, the R-learner parameterizes $\mu_1(\mathbf{x})$ as a (nuisance) baseline $\mu_0(\mathbf{x})$ plus the CATE $\tau_b(\mathbf{x})$. The black-box $f(\mathbf{x}, \mathbf{t})$ does not capture the asymmetry between $\mathbf{x}$ and $\mathbf{t}$ in the implied CATE $f(\mathbf{x}, \mathbf{t}) - f(\mathbf{x}, \mathbf{t}')$. Further, unlike the binary case, in many applications, we do not have a baseline treatment $\mathbf{t}_0$ with respect to which we could parameterize f in terms of some $\tau(\mathbf{t}, \mathbf{t}_0, \mathbf{x})$.

To address these issues, we take a different route that (a) avoids conditional density estimation over high-dimensional spaces and, (b) partials-out the dependence of $\mathbf{X}$ on $\mathbf{T}$ on Y. For (a), we propose to learn *propensity features*, which, as we will see, have a natural connection to the propensity score, but make inference easier. For (b), we make the assumption that the causal effect of interest is a *product effect*: the outcome function $f(\mathbf{X}, \mathbf{T})$ can be written as an inner product of two high-dimensional functions depending on the covariates and the treatment, respectively. We consider this to be a mild assumption, as we can formally justify its universality.

Assumption 3. *(Product effect) We consider the following partial parameterization of $p(y \mid \mathbf{x}, \mathbf{t})$,*

$$Y = \underbrace{g(\mathbf{X})^\top h(\mathbf{T})}_{=:f(\mathbf{X},\mathbf{T})} + \varepsilon, \tag{9}$$

where $g : \mathcal{X} \to \mathbb{R}^d, h : \mathcal{T} \to \mathbb{R}^d$ and $\mathbb{E}[\varepsilon \mid \mathbf{x}, \mathbf{t}] = \mathbb{E}\left[\varepsilon \mid \mathbf{x}\right] = 0$, for all $(\mathbf{x}, \mathbf{t}) \in \mathcal{X} \times \mathcal{T}$.

The CATE follows as

$$\tau\left(\mathbf{t}', \mathbf{t}, \mathbf{x}\right) = g(\mathbf{x})^\top \left(h\left(\mathbf{t}'\right) - h(\mathbf{t})\right). \tag{10}$$

The following asserts that provided we allow the dimensionality of g and h to grow, we may approximate any arbitrary bounded continuous functions in $\mathcal{C}(\mathcal{X} \times \mathcal{T})$ where $\mathcal{X} \times \mathcal{T}$ is compact.

Proposition 1. *(Universality of product effect) Let $\mathcal{H}_{\mathcal{X}\times\mathcal{T}}$ be a Reproducing Kernel Hilbert Space (RKHS) on the set $\mathcal{X} \times \mathcal{T}$ with universal kernel k. Under any finite sample regime with N samples, any minimizer of the objective function in eq. (8) where $\mathcal{F} = \mathcal{H}_{\mathcal{X}\times\mathcal{T}}$ can be written as the inner product of two vector fields $g \in (\mathcal{H}_{\mathcal{X}})^d, h \in (\mathcal{H}_{\mathcal{T}})^d$ with $d \leq N^2$, where $\hat{m}$ and $\hat{e}^p$ are finite-sample, kernel-ridge-regression estimates of $\mathbb{E}\left[Y \mid \mathbf{X}\right]$ and $\mathbb{E}\left[f(\mathbf{X}, \mathbf{T}) \mid \mathbf{X}\right]$ respectively.*

We defer the proof to Appendix A.1.

4.2 The Robinson Decomposition for Product Effects

Define *propensity features* $e^h(\mathbf{x}) \triangleq \mathbb{E}\left[h(\mathbf{T}) \mid \mathbf{x}\right]$ and $m(\mathbf{x}) \triangleq \mathbb{E}\left[Y \mid \mathbf{x}\right] = g(\mathbf{x})^\top e^h(\mathbf{x})$.

Following the same steps as in Section 3.2, the generalized Robinson decomposition for eq. (9) is

$$\boxed{Y - m(\mathbf{X}) = g(\mathbf{X})^\top \left(h(\mathbf{T}) - e^h(\mathbf{X})\right) + \varepsilon.} \tag{11}$$

For a fixed $h(\cdot)$, a direct generalization of the R-learner to graph treatments is the solution of

$$\widehat{g}(\cdot) = \arg\min_g \left\{ \frac{1}{n} \sum_{i=1}^n \left(Y_i - \widehat{m}(\mathbf{X}_i) - g(\mathbf{X}_i)^\top \left(h(\mathbf{T}_i) - \widehat{e}^h(\mathbf{X}_i)\right)\right)^2 + \Lambda\left(g(\cdot)\right) \right\}. \tag{12}$$

Instead of fixing $h(\cdot)$, we propose to simultaneously learn $h(\cdot)$ using an alternating approach. This allows us to learn representations for both covariates $\mathbf{X}$ and treatments $\mathbf{T}$ that can adapt to each other. This poses the following challenge: learning $\widehat{e}^h(\cdot)$ is now entangled with learning $h(\cdot)$. While the R-learner is based on the idea of *cross-fitting*, where at each data point i we pick estimates of the nuisances that do not use that data point[3], in what follows we introduce a pragmatic representation learning approach for $(\hat{g}, \hat{h})$ that does not use cross-fitting[4] while still being able to beat competitor methods in our experimental evaluation. We call this approach *Graph Intervention Networks* (GIN).

4.3 Algorithm

We introduce a two-stage algorithm for training. To allow for both flexibility in representation learning of complex causal relationships as well as scalability to large and high-dimensional data-sets, we instantiate our method using neural networks. In principle however, any model trainable via gradient descent can be used. We learn surrogate models for the mean outcome and propensity features $\widehat{m}_{\boldsymbol{\theta}}(\mathbf{X})$ and $\widehat{e}^h_{\boldsymbol{\eta}}(\mathbf{X})$ with parameters $\boldsymbol{\theta} \in \mathbb{R}^{d_{\boldsymbol{\theta}}}, \boldsymbol{\eta} \in \mathbb{R}^{d_{\boldsymbol{\eta}}}$, as well as feature maps for covariates and treatments $\widehat{g}_{\boldsymbol{\psi}}(\mathbf{X}), \widehat{h}_{\boldsymbol{\phi}}(\mathbf{T})$, parameterized by $\boldsymbol{\psi} \in \mathbb{R}^{d_{\boldsymbol{\psi}}}, \boldsymbol{\phi} \in \mathbb{R}^{d_{\boldsymbol{\phi}}}$. We denote regularizers by $\Lambda(\cdot)$. Figure 2 summarizes the algorithm. As the mean outcome model $\widehat{m}_{\boldsymbol{\theta}}(\mathbf{X})$ does not depend on the other components, we learn it separately in Stage 1. In Stage 2, we alternate between learning $\boldsymbol{\psi}, \boldsymbol{\phi}, \boldsymbol{\eta}$.

[3] For instance, $\hat{m}(\cdot)$ can vary from data point to data point, which can be denoted as $\hat{m}_i(\cdot)$. Under cross-fitting, the requirement is that $\hat{m}_i$ is not a function of $(\mathbf{X}_i, \mathbf{T}_i, Y_i)$.

[4] We could in principle use cross-fitting for $\hat{e}^h$, although the loop between fitting $\hat{h}$ alternating with $\hat{e}^h$ would break the overall independence between $\hat{e}^h_i(\cdot)$ and data point i. While it is possible that cross-fitting for $\hat{e}^h$ is still beneficial in this case, for simplicity and for computational savings, we did not implement it.

a GIN Training.

Input: Stage 1 data $\mathcal{D}_1 := \{(\mathbf{x}_i, y_i)\}_{i=1}^m$, Stage 2 data $\mathcal{D}_2 := \{(\mathbf{x}_i, \mathbf{t}_i, y_i)\}_{i=1}^n$ Step sizes $\lambda_{\boldsymbol{\theta}}, \lambda_{\boldsymbol{\eta}}, \lambda_{\boldsymbol{\psi}}, \lambda_{\boldsymbol{\phi}}$. Number of update steps K. Mini-batch sizes B_1, B_2.

Initialize parameters: $\boldsymbol{\theta}, \boldsymbol{\eta}, \boldsymbol{\psi}, \boldsymbol{\phi}$
while not converged **do** ▷ *Stage 1*
  Sample mini-batch $\{(\mathbf{x}_b, y_b)\}_{b=1}^{m_{B_1}}$
  Evaluate $J_m(\boldsymbol{\theta})$
  Update $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \lambda_{\boldsymbol{\theta}} \widehat{\nabla}_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$
end while
while not converged **do** ▷ *Stage 2*
  Sample mini-batch $\{(\mathbf{x}_b, \mathbf{t}_b, y_b)\}_{b=1}^{m_{B_2}}$
  Evaluate $J_{g,h}(\boldsymbol{\psi}, \boldsymbol{\phi}), J_{e^h}(\boldsymbol{\eta})$
  for $k = 1$ to K **do**
    Update $\boldsymbol{\phi} \leftarrow \boldsymbol{\phi} - \lambda_{\boldsymbol{\phi}} \widehat{\nabla}_{\boldsymbol{\phi}} J_{g,h}(\boldsymbol{\psi}, \boldsymbol{\phi})$
    Update $\boldsymbol{\psi} \leftarrow \boldsymbol{\psi} - \lambda_{\boldsymbol{\psi}} \widehat{\nabla}_{\boldsymbol{\psi}} J_{g,h}(\boldsymbol{\psi}, \boldsymbol{\phi})$
  end for
  Update $\boldsymbol{\eta} \leftarrow \boldsymbol{\eta} - \lambda_{\boldsymbol{\eta}} \widehat{\nabla}_{\boldsymbol{\eta}} J_{e^h}(\boldsymbol{\eta})$
end while

b Pseudocode in a PyTorch-like style.

```
# Initialize submodels and optimizers
m, e, g, h = MLP(...), MLP(...), MLP(...),
     GNN(...)
m_opt, e_opt, g_opt, h_opt = Adam(m.params(),
      m_lr), Adam(e.params(), e_lr), ...

# Stage 1
for batch in train_loader:
    X, Y = batch.X, batch.Y
    m_opt.zero_grad()
    F.mse_loss(m(X), Y).backward()
    m_opt.step()

# Stage 2
for batch in train_loader:
    X, T, Y = batch.X, batch.T, batch.Y
    for _ in range(num_update_steps):
        g_opt.zero_grad()
        h_opt.zero_grad()
        F.mse_loss((g(X)*(h(T) - e(X))).sum
             (-1), (Y-m(X))).backward()
        g_opt.step()
        h_opt.step()
    e_opt.zero_grad()
    F.mse_loss(e(X), h(T)).backward()
    e_opt.step()
```

Figure 2: The two-stage algorithm for training GIN.

Stage 1: Learn parameters $\boldsymbol{\theta}$ of the mean outcome model $\widehat{m}_{\boldsymbol{\theta}}(\mathbf{X})$ based on the objective

$$J_m(\boldsymbol{\theta}) = \sum_{i=1}^{m} \left\| y_i - \widehat{m}_{\boldsymbol{\theta}}(\mathbf{x}_i) \right\|_2^2 + \Lambda(\boldsymbol{\theta}), \tag{13}$$

which relies only on covariates and outcome data $\mathcal{D}_1 := \left\{(\mathbf{x}_i, y_i)\right\}_{i=1}^m$.

Stage 2: Learn parameters $\boldsymbol{\psi}, \boldsymbol{\phi}$ for the covariates and treatments feature maps $\widehat{g}_{\boldsymbol{\psi}}(\mathbf{X}), \widehat{h}_{\boldsymbol{\phi}}(\mathbf{T})$, as well as parameters $\boldsymbol{\eta}$ for the propensity features $\widehat{e}^h_{\boldsymbol{\eta}}(\mathbf{X})$.

$$J_{g,h}(\boldsymbol{\phi}, \boldsymbol{\psi}) = \sum_{i=1}^{n} \left\| y_i - \left\{ \widehat{m}_{\boldsymbol{\theta}}(\mathbf{x}_i) + \widehat{g}_{\boldsymbol{\psi}}(\mathbf{x}_i)^\top \left(\widehat{h}_{\boldsymbol{\phi}}(\mathbf{t}_i) - \widehat{e}^h_{\boldsymbol{\eta}}(\mathbf{x}_i) \right) \right\} \right\|_2^2 + \Lambda(\boldsymbol{\psi}) + \Lambda(\boldsymbol{\phi}). \tag{14}$$

This loss hinges on $\widehat{e}^h_{\boldsymbol{\eta}}(\mathbf{X})$, which needs to be learned by

$$J_{e^h}(\boldsymbol{\eta}) = \sum_{i=1}^{n} \left\| \widehat{h}_{\boldsymbol{\phi}}(\mathbf{t}_i) - \widehat{e}^h_{\boldsymbol{\eta}}(\mathbf{x}_i) \right\|_2^2 + \Lambda(\boldsymbol{\eta}), \tag{15}$$

and in turn depends on $\widehat{h}_{\boldsymbol{\phi}}(\mathbf{T})$. While it may be tempting to learn $\boldsymbol{\psi}, \boldsymbol{\phi}$ and $\boldsymbol{\eta}$ jointly, they have fundamentally different objectives ($\widehat{e}^h_{\boldsymbol{\eta}}(\mathbf{X})$ is defined as an estimate of the expectation $\mathbb{E}\left[h(\mathbf{T}) \mid \mathbf{x}\right]$). Therefore, we employ an alternating optimization procedure, where we take $k \in \{1, \ldots, K\}$ optimization steps for $\boldsymbol{\psi}, \boldsymbol{\phi}$ towards $J_{g,h}(\boldsymbol{\psi}, \boldsymbol{\phi})$ and one step for learning $\boldsymbol{\eta}$. We observe that setting $K > 1$, i.e. updating $\boldsymbol{\psi}, \boldsymbol{\phi}$ more frequently than $\boldsymbol{\eta}$, stabilizes the training process.

4.4 Advantages of GINs

We conclude by describing the beneficial properties of GIN, particularly in finite-sample regimes:

1. **Targeted regularization**: Regularizing $\widehat{g}_{\boldsymbol{\psi}}(\mathbf{X}), \widehat{h}_{\boldsymbol{\phi}}(\mathbf{T})$ in eq. (14) after partialing out confounding allows for a type of targeted regularization of the isolated causal effect. We will show that this leads to improved estimation. In contrast, outcome estimation methods can suffer from

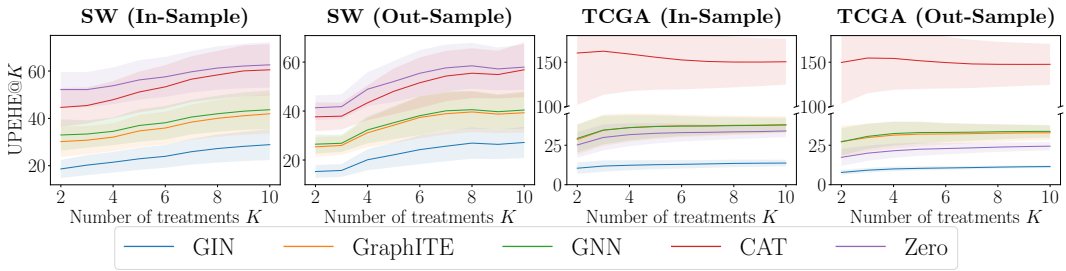

Figure 3: UPEHE@K for $K \in \{2, \ldots, 10\}$.

regularization-induced confounding, i.e., inadvertently regularizing the causal effect estimate away from zero in the service of trying to improve predictive performance [9, 11, 17, 28, 30, 40].

2. **Propensity features**: There are often parts of $\mathbf{X}$ that cause the treatment but not the outcome; to avoid modeling them is one of the attractions of regression-based causal inference [21]. Likewise, we would like to dispose of unnecessary components of a high-dimensional $\mathbf{T}$, but there is little in the literature about what to do when a criterion such as Robinson decomposition is used to define a method. Propensity features are a way to avoid using $p\left(\mathbf{t} \mid \mathbf{x}\right)$ as a nuisance parameter.
3. **Data-efficiency**: In contrast to methods that split the data into disjoint models for each treatment group (known as *T-learners* for binary treatments [9, 11]), sharing causal effect parameters between all covariates regardless of their assigned treatment increases data-efficiency, especially if certain graph interventions are only observed a few times and are supported on few covariates.
4. **Partial data**: In settings without access to both the treatment assignment and the outcome but only access to one of them, one can leverage that data to improve our estimator further. A typical scenario is significant lags between applying a treatment and observing the outcome, e.g., a patients' recovery in one year after a drug was administered [34]. There is data about the treatment in such scenarios that may refine the estimate of $\widehat{e}^h_{\boldsymbol{\eta}}\left(\mathbf{X}\right)$. Analogously, if only treatment data is missing, one can use covariates and outcome data to improve $\widehat{m}_{\boldsymbol{\theta}}\left(\mathbf{X}\right)$.

5 Experiments

The goal of our experiments is to understand how deconfounding through the generalized Robinson decomposition compares with prior methods in graph intervention settings.

5.1 Experimental Setup

Datasets. To be able to compute CATE estimation error w.r.t. a ground truth, we design two causal models: a simpler synthetic model with small-world graph treatments and a more complex model with real-world molecular graph treatments and gene expression covariates. The Small-World (SW) simulation contains 1,000 uniformly sampled covariates and 200 randomly generated Watts–Strogatz small-world graphs [58] as treatments. *The Cancer Genomic Atlas* (TCGA) simulation uses 9,659 gene expression measurements of cancer patients for covariates [59] and 10,000 sampled molecules from the QM9 dataset [43] as treatments. Appendix B details the data-generating schemes.

Baselines. We compare our method to (1) **Zero**, a sanity-check baseline that consistently predicts zero treatment effect and equals the mean squared treatment effect (poorly regularized models may perform worse than that due to confounding), (2) **CAT**, a categorical treatment variable model using one-hot encoded treatment indicator vectors, (3) **GNN**, a model that first encodes treatments with a GNN and then concatenates treatment and individual features for regression, (4) **GraphITE** [18], a CATE estimation method designed for graph treatments (more details in Section 2). GNN and CAT reflect the performance of standard regression models. The contrast between these two provides insight into whether the additional graph structure of the treatment improves CATE estimation. To deal with unseen treatments during CAT's evaluation, we map such to the most similar ones seen during training based on their Euclidean distance in the embedding space of the GNN baseline.

Graph models. For small-world networks, we use *k-dimensional GNNs* [37], as to distinguish graphs they take higher-order structures into account. To model molecular graphs, we use *Relational*

Table 1: Error of CATE estimation for all methods, measured by WPEHE@6. Results are averaged over 10 trials, $\pm$ denotes std. error (each trial samples treatment assignment matrix $\mathbf{W}$).

Method	**SW**		**TCGA**	
	In-sample	Out-sample	In-sample	Out-sample
Zero	56.26 ± 8.12	53.77 ± 8.93	26.63 ± 7.55	17.94 ± 4.86
CAT	51.75 ± 8.85	49.76 ± 9.73	155.88 ± 52.82	146.62 ± 42.32
GNN	37.10 ± 6.84	36.74 ± 7.42	30.67 ± 8.29	27.57 ± 7.95
GraphITE	34.81 ± 6.70	35.94 ± 8.07	30.31 ± 8.96	27.48 ± 8.95
GIN	$\mathbf{23.00 \pm 4.56}$	$\mathbf{23.19 \pm 5.56}$	$\mathbf{10.98 \pm 3.45}$	$\mathbf{8.15 \pm 1.46}$

Graph Convolutional Networks [47], where the nodes are atoms and each edge type corresponds to a specific bond type. We use the implementations of PyTorch Geometric [12].

Evaluation metrics. We extend the *expected Precision in Estimation of Heterogeneous Effect* (PEHE) commonly used in binary treatment settings [21] to arbitrary pairs of treatments $(\mathbf{t}, \mathbf{t}')$ as follows. We denote the *Unweighted PEHE* (UPEHE) and the *Weighted PEHE* (WPEHE) as

$$\epsilon_{\text{UPEHE(WPEHE)}} \triangleq \int_{\mathcal{X}} \left(\widehat{\tau}\left(\mathbf{t}', \mathbf{t}, \mathbf{x}\right) - \tau\left(\mathbf{t}', \mathbf{t}, \mathbf{x}\right)\right)^2 p\left(\mathbf{t} \mid \mathbf{x}\right) p\left(\mathbf{t}' \mid \mathbf{x}\right) p\left(\mathbf{x}\right) \mathrm{d}\mathbf{x}, \tag{16}$$

where the weighted version gives less importance to treatment pairs that are less likely; to account for the fact that such pairs will have higher estimation errors. In fact, as the reliability of estimated effects decreases by how likely they are in the observational study, we evaluate all methods on U/WPEHE truncated to the top K treatments, which we call U/WPEHE@K. To compute this, for each $\mathbf{x}$, we rank all treatments by their propensity $p\left(\mathbf{t} \mid \mathbf{x}\right)$ (given by the causal model) in descending order. We take the top K treatments and compute the U/WPEHE for all $\binom{K}{2}$ treatment pairs.

In-sample vs. out-sample. A common benchmark for causal inference methods is the *in-sample* task, which we include here for completeness: estimating CATEs for covariate values $\mathbf{x}$ found in the training set. This task is still non-trivial, as the outcome of only one treatment is observed during training [5]. In contrast, and arguably of more relevance to decision making, the goal of the *out-sample* task is to estimate CATEs for completely unseen covariate realizations $\mathbf{x}'$.

Hyper-parameter tuning. To ensure a fair comparison, we perform hyper-parameter optimization with random search for all models on held-out data and select the best hyper-parameters over 10 runs.

Propensity. We define the propensity (or *treatment selection bias*) as $p\left(\mathbf{T} \mid \mathbf{x}\right) = \mathrm{softmax}\left(\kappa \mathbf{W}^\top \mathbf{X}\right)$, where $\mathbf{W} \in \mathbb{R}^{|\mathcal{T}| \times d}, \forall i, j : W_{ij} \sim \mathcal{U}\left[0, 1\right]$ is a random matrix (sampled then fixed for each run). Recall $|\mathcal{T}|$ is the number of available treatments and let d be the dimensionality of the covariates. Here the *bias strength* κ is a temperature parameter that determines the flatness of the propensity (the lower the flatter, i.e., $\kappa = 0$ corresponds to the uniform distribution).

5.2 Comparison of Performances on different K Treatments

Figure 3 shows the UPEHE@K of all methods for $K \in \{2, \ldots, 10\}$. We also report the WPEHE@6 of all methods in Table 1. Unless stated otherwise, we report results for bias strengths $\kappa = 10$ and $\kappa = 0.1$ in the SW and TCGA datasets, respectively across 10 random trials.

The results indicate that the relative performance of each method, for both the in-sample and out-sample estimation tasks, is consistent. Further, they suggest that, overall, the performance of GIN is best due to a better isolation of the causal effect from the observed data compared to other methods. The performance difference between CAT and GNN across all results indicate that accounting for graph information significantly improves the estimates. We observe from the SW experiments that GraphITE [18] performs slightly better than GNN, while it is nearly the same as GNN on TCGA.

[5] The original motivation comes from Fisherian designs where the only source of randomness is on the treatment assignment [22]. Our motivation is simpler: rule out the extra variability from different covariates, highlighting the difference between methods due to different loss functions and less due to smoothing abilities.

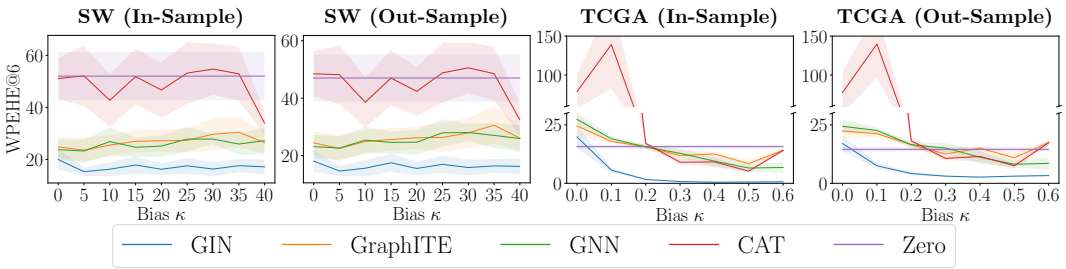

Figure 4: WPEHE@6 over increasing bias strength κ.

Surprisingly, the results of the TCGA experiments with low bias strength $\kappa = 0.1$ expose that all models but GIN fail to isolate causal effects better than the Zero baseline. These results confirm that confounding effects of $\mathbf{X}$ on Y combined with moderate causal effects can cause severe regularization bias for black-box regression models, while GIN partials these out from the outcome by $\widehat{m}_{\boldsymbol{\theta}}(\mathbf{X})$. We include additional results on convergence and larger values of K in the Appendix.

5.3 Comparison of Robustness to different Bias Strengths κ

A strong selection bias (i.e. large κ) in the observed data makes CATE estimation more difficult, as it becomes unlikely to see certain treatments $\mathbf{t} \in \mathcal{T}$ for particular covariates $\mathbf{x}$. Here, we assess each model's robustness to varying levels of selection bias, determined by κ, across 5 random seeds. In Figure 4, we see that GIN outperforms the baselines across the entire range of considered biases. Interestingly, GIN performs competitively even in a case with no selection bias ($\kappa = 0$, which corresponds to a randomized experiment). Importantly, all performances seem to either stagnate (SW) or to increase (TCGA) with increasing biases. Notably, the poor performance of CAT suddenly improves on datasets with high bias. We believe this is because, in high bias regimes, we see fewer distinct treatments overall, which allows the CAT model to approach the performance of GNN.

6 Limitations, Future Work and Potential Negative Societal Impacts

Limitations and future work. One limitation of the current work is the lack of a convergence guarantee for estimating $\widehat{g}_{\boldsymbol{\psi}}(\mathbf{X}), \widehat{h}_{\boldsymbol{\phi}}(\mathbf{T})$, as is done in the R-Learner [40]. Our first direction of future work is thus adapting the quasi-oracle guarantee of Nie & Wager [40] to this setting. This would help to theoretically characterize the extent to which our objective mitigates regularization bias present in traditional regression methods. Second, in some real-life domains, Assumption 1 (Unconfoundedness) can be too strong, as there may exist *hidden confounders*. There are two common strategies to deal with them: utilizing *instrumental variables* [19, 54, 60] or *proxy variables* [35, 36, 55]. Developing new approaches for graph interventions in such settings is a promising future direction. Finally, neural network initialization can impact final estimates. To obtain consistency guarantees, kernel methods can be used [35, 54]. Investigating a kernel formulation of our objective is thus an interesting direction for future work.

Potential negative societal impacts. Because causal inference methods make recommendations about interventions to apply in real-world settings, misapplying them can have a negative real-world impact. This issue applies to all research in causal effect estimation. It is crucial to thoroughly test these methods on realistic simulations and alter aspects of them to understand how violations of assumptions impact estimation. There is a significant sub-field of causal inference devoted to such sensitivity analyses. We have aimed to provide a comprehensive evaluation of graph treatment methods by showing how estimation degrades as less likely treatments are considered (Figure 3) and as treatment bias increases (Figure 4). When applying this method in practice, there are additional sensitivity analysis tools that should be leveraged [15, 56] to mitigate negative impacts.

7 Conclusion

This paper introduced GIN, a method for CATE estimation with graph interventions. GIN generalizes the Robinson decomposition to isolate the causal estimand from confounding associations. Our plug-in estimator supports any supervised learning method. In experiments, GIN outperforms baseline CATE estimators for graph treatments, across multiple treatment pairs and bias levels.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] For clarity, we added a bullet-point for each of our contributions, roughly matching the structure of Section 4.
   (b) Did you describe the limitations of your work? [Yes] See Section 6.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Section 6.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes] We provide proofs in the Appendix.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] The code to reproduce the results will be provided in the supplemental material and here: https://github.com/GINAuthors/GraphInterventionNetworks.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Additional details are in the Appendix.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] We will provide these details in the Appendix.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes] We will provide these details in the Appendix.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

References

[1] Alaa, A. and van der Schaar, M. Limits of estimating heterogeneous treatment effects: Guidelines for practical algorithm design. In Dy, J. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 129–138, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR.

[2] Alaa, A. M. and van der Schaar, M. Bayesian inference of individualized treatment effects using multi-task gaussian processes. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[3] Arbour, D., Dimmery, D., and Sondhi, A. Permutation weighting, 2020.

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

[5] Athey, S. and Wager, S. Estimating treatment effects with causal forests: An application. *arXiv preprint arXiv:1902.07409*, 2019.

[6] Athey, S., Tibshirani, J., and Wager, S. Generalized random forests. *The Annals of Statistics*, 47 (2):1148 – 1178, 2019. doi: 10.1214/18-AOS1709.

[7] Bica, I., Jordon, J., and van der Schaar, M. Estimating the effects of continuous-valued interventions using generative adversarial networks. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[8] Bronstein, M. M., Bruna, J., Cohen, T., and Veličković, P. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges, 2021.

[9] Caron, A., Manolopoulou, I., and Baio, G. Estimating individual treatment effects using non-parametric regression models: a review, 2020.

[10] Chernozhukov, V., Chetverikov, D., Demirer, M., Duflo, E., Hansen, C., Newey, W., and Robins, J. Double/debiased machine learning for treatment and structural parameters. *The Econometrics Journal*, 21(1):C1–C68, 01 2018. ISSN 1368-4221.

[11] Curth, A. and van der Schaar, M. Nonparametric estimation of heterogeneous treatment effects: From theory to learning algorithms. In Banerjee, A. and Fukumizu, K. (eds.), *The 24th International Conference on Artificial Intelligence and Statistics, AISTATS 2021, April 13-15, 2021, Virtual Event*, volume 130 of *Proceedings of Machine Learning Research*, pp. 1810–1818. PMLR, 2021.

[12] Fey, M. and Lenssen, J. E. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

[13] Flaxman, S., Mishra, S., Gandy, A., Unwin, H. J. T., Mellan, T. A., Coupland, H., Whittaker, C., Zhu, H., Berah, T., Eaton, J. W., et al. Estimating the effects of non-pharmaceutical interventions on covid-19 in europe. *Nature*, 584(7820):257–261, 2020.

[14] for Health Statistics, N. C. et al. 2007–2008 national health and nutrition examination survey (nhanes). *US Department of Health and Human Services, Centers for Disease Control and Prevention: Hyattsville, MD, USA*, 2008.

[15] Franks, A., D'Amour, A., and Feller, A. Flexible sensitivity analysis for observational studies without observable implications. *Journal of the American Statistical Association*, 2019.

[16] Gretton, A., Fukumizu, K., Teo, C. H., Song, L., Schölkopf, B., and Smola, A. J. A kernel statistical test of independence. In Platt, J. C., Koller, D., Singer, Y., and Roweis, S. T. (eds.), *Advances in Neural Information Processing Systems 20, Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 3-6, 2007*, pp. 585–592. Curran Associates, Inc., 2007.

[17] Hahn, P. R., Murray, J. S., Carvalho, C. M., et al. Bayesian regression tree models for causal inference: Regularization, confounding, and heterogeneous effects (with discussion). *Bayesian Analysis*, 15(3):965–1056, 2020.

[18] Harada, S. and Kashima, H. Graphite: Estimating individual effects of graph-structured treatments. *arXiv preprint arXiv:2009.14061*, 2020.

[19] Hartford, J., Lewis, G., Leyton-Brown, K., and Taddy, M. Deep IV: A flexible approach for counterfactual prediction. In Precup, D. and Teh, Y. W. (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1414–1423. PMLR, 06–11 Aug 2017.

[20] Hatt, T. and Feuerriegel, S. Estimating average treatment effects via orthogonal regularization. *arXiv preprint arXiv:2101.08490*, 2021.

[21] Hill, J. L. Bayesian nonparametric modeling for causal inference. *Journal of Computational and Graphical Statistics*, 20(1):217–240, 2011. doi: 10.1198/jcgs.2010.08162.

[22] Imbens, G. and Rubin, D. *Causal Inference for Statistics, Social, and Biomedical Sciences*. Cambridge University Press, 2015.

[23] Imbens, G. W. Nonparametric estimation of average treatment effects under exogeneity: A review. *The Review of Economics and Statistics*, 86(1):4–29, 2004. doi: 10.1162/003465304323023651.

[24] Jesson, A., Mindermann, S., Shalit, U., and Gal, Y. Identifying causal-effect inference failure with uncertainty-aware models. *Advances in Neural Information Processing Systems*, 33, 2020.

[25] Jesson, A., Mindermann, S., Gal, Y., and Shalit, U. Quantifying ignorance in individual-level causal-effect estimates under hidden confounding. In *International Conference on Machine Learning*. PMLR, 2021.

[26] Johansson, F., Shalit, U., and Sontag, D. Learning representations for counterfactual inference. In *International conference on machine learning*, pp. 3020–3029. PMLR, 2016.

[27] Kallus, N. DeepMatch: Balancing deep covariate representations for causal inference using adversarial training. In III, H. D. and Singh, A. (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 5067–5077. PMLR, 13–18 Jul 2020.

[28] Kennedy, E. H. Optimal doubly robust estimation of heterogeneous causal effects. *arXiv preprint arXiv:2004.14497*, 2020.

[29] Kennedy, E. H., Ma, Z., McHugh, M. D., and Small, D. S. Nonparametric methods for doubly robust estimation of continuous treatment effects. *Journal of the Royal Statistical Society. Series B, Statistical Methodology*, 79(4):1229, 2017.

[30] Künzel, S. R., Sekhon, J. S., Bickel, P. J., and Yu, B. Metalearners for estimating heterogeneous treatment effects using machine learning. *Proceedings of the National Academy of Sciences*, 116(10):4156–4165, 2019. ISSN 0027-8424. doi: 10.1073/pnas.1804597116.

[31] Li, F. et al. Propensity score weighting for causal inference with multiple treatments. *Annals of Applied Statistics*, 13(4):2389–2415, 2019.

[32] Lopez, M. J. and Gutman, R. Estimation of causal effects with multiple treatments: a review and new ideas. *Statistical Science*, pp. 432–454, 2017.

[33] Louizos, C., Shalit, U., Mooij, J. M., Sontag, D., Zemel, R., and Welling, M. Causal effect inference with deep latent-variable models. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[34] Lu, D., Tao, C., Chen, J., Li, F., Guo, F., and Carin, L. Reconsidering generative objectives for counterfactual reasoning. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 21539–21553. Curran Associates, Inc., 2020.

[35] Mastouri, A., Zhu, Y., Gultchin, L., Korba, A., Silva, R., Kusner, M. J., Gretton, A., and Muandet, K. Proximal causal learning with kernels: Two-stage estimation and moment restriction. In *International Conference on Machine Learning*. PMLR, 2021.

[36] Miao, W., Geng, Z., and Tchetgen, E. T. Identifying causal effects with proxy variables of an unmeasured confounder, 2018.

[37] Morris, C., Ritzert, M., Fey, M., Hamilton, W. L., Lenssen, J. E., Rattan, G., and Grohe, M. Weisfeiler and leman go neural: Higher-order graph neural networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):4602–4609, Jul. 2019. doi: 10.1609/aaai.v33i01.33014602.

[38] Nabi, R., McNutt, T., and Shpitser, I. Semiparametric causal sufficient dimension reduction of high dimensional treatments, 2020.

[39] Nie, L., Ye, M., qiang liu, and Nicolae, D. {VCN}et and functional targeted regularization for learning causal effects of continuous treatments. In *International Conference on Learning Representations*, 2021.

[40] Nie, X. and Wager, S. Quasi-oracle estimation of heterogeneous treatment effects. *Biometrika*, 09 2020. ISSN 0006-3444. doi: 10.1093/biomet/asaa076.

[41] Pearl, J. *Causality: models, reasoning, and inference*. Cambridge University Press, 2000.

[42] Pollmann, M. Causal inference for spatial treatments, 2020.

[43] Ramakrishnan, R., Dral, P. O., Rupp, M., and von Lilienfeld, O. A. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific Data*, 1, 2014.

[44] Robinson, P. M. Root-n-consistent semiparametric regression. *Econometrica*, 56(4):931–954, 1988. ISSN 00129682, 14680262.

[45] ROSENBAUM, P. R. and RUBIN, D. B. The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1):41–55, 04 1983. ISSN 0006-3444. doi: 10.1093/biomet/70.1.41.

[46] Ruddigkeit, L., van Deursen, R., Blum, L. C., and Reymond, J.-L. Enumeration of 166 billion organic small molecules in the chemical universe database gdb-17. *Journal of Chemical Information and Modeling*, 52(11):2864–2875, 2012. doi: 10.1021/ci300415d. PMID: 23088335.

[47] Schlichtkrull, M. S., Kipf, T. N., Bloem, P., van den Berg, R., Titov, I., and Welling, M. Modeling relational data with graph convolutional networks. In Gangemi, A., Navigli, R., Vidal, M., Hitzler, P., Troncy, R., Hollink, L., Tordai, A., and Alam, M. (eds.), *The Semantic Web - 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3-7, 2018, Proceedings*, volume 10843 of *Lecture Notes in Computer Science*, pp. 593–607. Springer, 2018.

[48] Schwab, P., Linhardt, L., and Karlen, W. Perfect match: A simple method for learning representations for counterfactual inference with neural networks, 2019.

[49] Schwab, P., Linhardt, L., Bauer, S., Buhmann, J. M., and Karlen, W. Learning Counterfactual Representations for Estimating Individual Dose-Response Curves. In *AAAI Conference on Artificial Intelligence*, 2020.

[50] Shalit, U., Johansson, F. D., and Sontag, D. Estimating individual treatment effect: generalization bounds and algorithms. In *International Conference on Machine Learning*, pp. 3076–3085. PMLR, 2017.

[51] Shi, C., Blei, D., and Veitch, V. Adapting neural networks for the estimation of treatment effects. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[52] Shi, C., Veitch, V., and Blei, D. Invariant representation learning for treatment effect estimation, 2020.

[53] Silva, R. Observational-interventional priors for dose-response learning. In Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

[54] Singh, R., Sahani, M., and Gretton, A. Kernel instrumental variable regression. In Wallach, H. M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E. B., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 4595–4607, 2019.

[55] Tchetgen, E. J. T., Ying, A., Cui, Y., Shi, X., and Miao, W. An introduction to proximal causal learning. *arXiv preprint arXiv:2009.10982*, 2020.

[56] VanderWeele, T. J. and Ding, P. Sensitivity analysis in observational research: introducing the e-value. *Annals of internal medicine*, 167(4):268–274, 2017.

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

[58] Watts, D. J. and Strogatz, S. H. Collective dynamics of 'small-world'networks. *nature*, 393 (6684):440–442, 1998.

[59] Weinstein, J. N., Collisson, E. A., Mills, G. B., Shaw, K. R. M., Ozenberger, B. A., Ellrott, K., Shmulevich, I., Sander, C., and Stuart, J. M. The cancer genome atlas pan-cancer analysis project. *Nature genetics*, 45(10):1113–1120, 2013.

[60] Xu, L., Chen, Y., Srinivasan, S., de Freitas, N., Doucet, A., and Gretton, A. Learning deep features in instrumental variable regression. In *International Conference on Learning Representations*, 2021.

[61] Yao, L., Li, S., Li, Y., Huai, M., Gao, J., and Zhang, A. Representation learning for treatment effect estimation from observational data. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[62] Zhang, D. W., Burghouts, G. J., and Snoek, C. G. M. Set prediction without imposing structure as conditional density estimation. In *International Conference on Learning Representations*, 2021.

[63] Zhou, Z., Athey, S., and Wager, S. Offline multi-action policy learning: Generalization and optimization, 2018.

[64] Zhu, Y., Coffman, D. L., and Ghosh, D. A boosting algorithm for estimating generalized propensity scores with continuous treatments. *Journal of Causal Inference*, 3(1):25–40, 2015. doi: doi:10.1515/jci-2014-0022.