DISSERTATION

submitted

to the

Combined Faculty of Natural Sciences and Mathematics

of

Heidelberg University, Germany

for the degree of

Doctor of Natural Sciences

Put forward by

Andrei Sirazitdinov, M.Sc.

Born in: Irkutsk, Russia

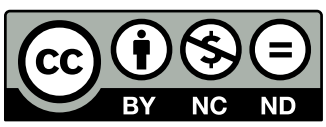
Oral examination:

Graph Neural Networks for Individual Treatment Effect Estimation

Advisor: Prof. Dr. Jürgen Hesser

Prof. Dr. Vincent Heuveline

This work is licensed under a Creative Commons "Attribution-NonCommercial-NoDerivs 3.0 Unported" license.



Abstract

This dissertation advances the field of causal inference by developing and evaluating Graph Neural Network (GNN)-based methods for estimating Individual Treatment Effects (ITE), leveraging causal graph structures to improve predictive accuracy. Traditional ITE estimation approaches often fail to account for dependencies among covariates, limiting their performance, particularly in data-scarce scenarios. To address this, we propose two novel architectures, GNN-TARnet and GAT-TARnet, which integrate structural causal models with GNNs to explicitly model these dependencies. We evaluate the proposed methods on synthetic datasets with known causal structures, established benchmarks such as IHDP and JOBS, and real-world randomized controlled trial data from the PerPAIN consortium. PerPAIN is a German research initiative focused on developing personalized treatment strategies for chronic musculoskeletal pain. Our models consistently outperform non-structural baselines, achieving lower $\sqrt{\varepsilon_{\text{PEHE}}}$ in low-data settings while remaining competitive with state-of-the-art approaches when data is abundant. The practical application to the PerPAIN trial, which tests tailored psychological interventions based on patient pain profiles, highlights the utility of GNN-based ITE estimation in real-world treatment allocation and demonstrates superior performance compared to clustering-based strategies. Key contributions of this work include a peer-reviewed publication, open-source software, and a web application for patient stratification, bridging theoretical innovation with practical tools for personalized decision-making. Future extensions of this work include incorporating time-series modeling to broaden applicability across dynamic domains and reducing the computational burden to enhance scalability. Another important direction is the creation of publicly available datasets in which features are causally dependent on each other and on the outcome. Such datasets can support more rigorous validation and foster the development of improved GNN-based methods for causal effect estimation.

Zusammenfassung

Diese Dissertation treibt das Gebiet der kausalen Inferenz voran, indem sie auf Graph Neural Network (GNN) basierende Methoden zur Schätzung individueller Behandlungseffekte (ITE) entwickelt und bewertet und dabei kausale Graphstrukturen nutzt, um die Vorhersagegenauigkeit zu verbessern. Herkömmliche ITE-Ansätze vernachlässigen oft Abhängigkeiten zwischen Kovariaten, was insbesondere bei begrenzter Datenverfügbarkeit die Leistung mindert. Zur Lösung dieses Problems schlagen wir zwei neue Architekturen vor: GNN-TARnet und GAT-TARnet. Diese kombinieren strukturelle Kausalmodelle mit GNNs, um Kovariatenabhängigkeiten explizit zu modellieren. Die Methoden werden auf synthetischen Datensätzen mit bekannten Kausalstrukturen, auf etablierten Benchmarks (IHDP, JOBS) sowie auf realen Daten aus einer randomisierten kontrollierten Studie des PerPAIN Konsortiums evaluiert. PerPAIN ist eine deutsche Forschungsinitiative zur Entwicklung personalisierter Therapien für chronische muskuloskelettale Schmerzen. Unsere Modelle übertreffen nicht-strukturelle Baselines und erreichen niedrigere $\sqrt{\varepsilon_{\text{PEHE}}}$ in datensparsamen Szenarien. Bei größeren Datensätzen bleiben sie mit aktuellen Verfahren konkurrenzfähig. Die Anwendung auf PerPAIN, in der psychologische Interventionen anhand individueller Schmerzprofile getestet werden, zeigt den praktischen Nutzen unserer Ansätze und übertrifft Clustering-basierte Zuweisungsmethoden. Wesentliche Beiträge dieser Arbeit sind eine begutachtete Publikation, Open-Source-Software sowie eine Webanwendung zur Patientenstratifizierung. Diese verbinden theoretische Innovation mit praxisnahen Werkzeugen für personalisierte Entscheidungen. Zukünftige Arbeiten sollten Zeitreihenmodelle einbeziehen, um dynamische Anwendungen zu unterstützen, und die Rechenkosten senken, um die Skalierbarkeit zu verbessern. Zudem ist die Entwicklung öffentlich zugänglicher Datensätze mit kausalen Abhängigkeiten zwischen Merkmalen und Ergebnissen ein wichtiger Schritt zur besseren Validierung und Weiterentwicklung GNN-basierter ITE-Methoden.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Observational Studies and RCTs

Causal inference is essential in fields such as social science, medicine, and epidemiology for understanding cause-and-effect relationships [4, 5]. Estimating the *individual treatment effect* (ITE) is critical for personalizing treatment decisions and improving outcomes. ITE measures the difference in outcomes for a given individual under treatment versus control conditions, offering insights beyond population-level averages [6]. This can be estimated using data from *randomized controlled trials* (RCTs) [7] or *observational studies* [8]. RCTs assign participants to treatment or control groups at random, ensuring covariate balance and minimizing confounding [9]. More complex designs, like that of the PerPain consortium [10], combine randomization with stratification algorithms to personalize treatment. While this approach aims to improve targeting, it introduces risks such as misclassification or insufficient capture of relevant patient characteristics [11].

In contrast, observational studies lack randomization, which introduces potential biases like confounding and selection bias [12]. To address this, causal inference methods such as propensity score matching, inverse probability weighting, and covariate adjustment have been developed to approximate RCT conditions [13, 14]. Modern methods go further by modeling non-linear covariate-outcome relationships using machine learning techniques, thus enhancing inference robustness from observational data [15]. While RCTs remain the gold standard, they face limitations in cost, ethics, and scalability [16, 17]. Non-linearities and small sample sizes further complicate analysis [18]. Observational studies, despite their imperfections, offer a complementary approach when paired with advanced techniques to recover individualized effects [19, 20].

1.2 Importance of ITE Estimation

ITE estimation enables personalized decision-making by identifying how specific individuals respond to interventions. Unlike average treatment effect (ATE) estimates, which describe population-level trends, ITE highlights outcome heterogeneity, crucial in domains like precision medicine, education, and policy design [6]. For instance, ITE can help determine which patients will benefit most from a therapy or which students will respond best to a specific educational intervention.

Accurate ITE estimation is vital: errors can lead to ineffective treatments or wasted resources. While RCTs establish causality through randomization, they often overlook individual variation and face practical constraints [16]. Conversely, real-world observational data are abundant but require sophisticated techniques to account for confounding and bias [13]. Recent advances in machine learning have addressed these challenges, though traditional models often struggle with complex covariate dependencies. This motivates the need for methods that can incorporate structural knowledge.

1.3 GNNs for ITE Estimation

Graph Neural Networks (GNNs) provide a compelling framework for leveraging relational structures in data. Unlike standard neural networks, which treat inputs as independent, GNNs process data represented as graphs, where nodes correspond to entities (e.g., covariates) and edges reflect relationships among them [21,22]. This makes them particularly well-suited for modeling the intricate dependencies present in causal systems [23]. For ITE estimation, GNNs can incorporate structural information from causal graphs, allowing models to explicitly encode the influence pathways among variables. This capacity to aggregate and propagate information across nodes offers a substantial advantage in settings where treatment effects depend on complex interactions, such as in healthcare or policy applications.

1.4 Objectives and Scope of the Thesis

This thesis advances ITE estimation by proposing GNN-based models that leverage causal graph structures. Specifically, we introduce two novel architectures, GNN-TARnet and GAT-TARnet, that extend the TARnet framework [24] by incorporating graph-based reasoning. These models aim to improve prediction accuracy, especially in data-scarce environments, by capturing relational dependencies ignored by traditional methods [25]. This work builds on our prior research published in *IEEE Access* [1], where we introduced GNN-

TARnet and demonstrated its efficacy on benchmark and synthetic datasets. It also extends the preprint [3], which surveys deep learning approaches for ITE estimation and proposes automatic hyperparameter optimization strategies. These foundations are now contextualized with theoretical analysis and real-world validation.

We investigate how GNNs embed causal relationships and support counterfactual reasoning, bridging structural causal models (SCMs) [26] with deep learning. Empirically, we evaluate our models across benchmark datasets (IHDP, JOBS), synthetic data (SUM), and real-world RCT data from the PerPain consortium [10]. Though focused on personalized treatment, our findings generalize to broader applications requiring individualized decision-making.

1.5 Hypothesis and Research Question

Hypothesis: Incorporating causal graph structures into GNN-based ITE estimation improves predictive accuracy compared to models that do not leverage such structures, especially when training data is limited. GNNs reduce model complexity by explicitly modeling interactions among covariates [27, 28].

Research Question: How does the integration of causal graph structures with GNNs affect the accuracy of ITE estimation across datasets with varying data availability, compared to traditional methods that ignore structural relationships? This question seeks to assess whether our approach offers consistent benefits in both data-rich and data-scarce settings and how it compares to clustering-based strategies in personalized treatment allocation [10].

1.6 Contributions

This thesis contributes to the field of ITE estimation through methodological innovation, empirical validation, and practical application:

1. **GNN-Based Methods:** We propose GNN-TARnet and GAT-TARnet, integrating graph neural networks into TARnet to exploit relational dependencies among covariates for more accurate ITE estimation.

2. **Theoretical and Empirical Validation:** We provide theoretical motivation within the potential outcomes framework and validate our models on:

   - Benchmark datasets (IHDP, JOBS),
   - A synthetic dataset (SUM) with known graph structure,
   - Real-world RCT data from the PerPain consortium [10].

3. **Real-World Application:** In collaboration with PerPain, we processed RCT data, designed a clustering algorithm for treatment stratification, and compared treatment allocation based on GNN-predicted ITEs versus clustering-based approaches.

4. **Open-Source Tools:** We released code and tools in a public CodeOcean capsule[1], including:

   - Model implementations and evaluation scripts,
   - Synthetic data generators with graph structures,
   - A web application for patient clustering.

5. **Published Work:** Our research appears in a joural paper and a preprint:

   - A. Sirazitdinov, M. Buchwald, V. Heuveline and J. Hesser, "Graph Neural Networks for Individual Treatment Effect Estimation," in *IEEE Access*, vol. 12, pp. 106884–106894, 2024, doi: 10.1109/ACCESS.2024.3437665.
   - A. Sirazitdinov, M. Buchwald, J. Hesser, and V. Heuveline, "Review of Deep Learning Methods for Individual Treatment Effect Estimation with Automatic Hyperparameter Optimization," December 2022.

1.7 Thesis Outline

This thesis is organized into several chapters to systematically present the development, validation, and implications of our GNN-based method for ITE estimation. Below is an outline of the structure:

Chapter 2: Background and Literature Review provides the foundational context for this thesis. It begins by distinguishing correlation from causation and introduces core concepts in causal inference, including ITE estimation and its associated challenges. The chapter then discusses the identifiability assumptions necessary for causal inference and explores classical and modern machine learning techniques, including DNNs, VAEs, GANs, and Transformers, used for ITE estimation. Structural tools such as BNs, SCMs, and GNNs are introduced, highlighting their relevance in encoding and utilizing causal structures. The role of the do-operator in modeling interventions is explained, followed by a discussion of causal discovery algorithms for inferring DAGs from data. The chapter concludes with a comprehensive review of existing methods for ITE estimation, including classical statistical approaches, deep learning models, and meta-learners.

[1] https://codeocean.com/capsule/4645832/tree

Chapter 3: Materials and Methods details the methodology proposed in this thesis for ITE estimation using GNNs and Graph Attention Networks (GATs). The chapter begins by formally defining the problem of ITE estimation in observational settings, introducing the role of DAGs in modeling structural dependencies. It then elaborates on the architecture and message-passing mechanisms of GNNs and GATs, including intervention modeling aligned with the do-operator from structural causal models. The chapter introduces two novel architectures, GNN-TARnet and GAT-TARnet, which integrate graph-based reasoning into treatment-agnostic representation learning. To support implementation and experimentation, a custom Python library is presented, enabling training, evaluation, and hyperparameter tuning. The chapter further describes four datasets: JOBS, the synthetic SUM dataset, and the real-world PerPain dataset used for evaluation, detailing data generation, structure, and relevance to causal inference. Finally, hyperparameter optimization strategies, including Random Search, Hyperband, and Bayesian optimization, are explained and applied to both the proposed models and thirteen baseline methods to ensure a fair performance comparison.

Chapter 4: Results evaluates GNN-TARnet and GAT-TARnet across benchmark (IHDP, JOBS), synthetic (SUM), and real-world (PerPain) datasets. It outlines implementation details and compares model performance to baseline methods using standard causal inference metrics.

Chapter 5: Discussion analyzes the results, highlighting our contributions, such as improved ITE estimation in data-scarce scenarios, and comparing our GNN-based method to state-of-the-art approaches. We discuss the potential of our methods to enhance personalized treatment allocation, as well as their limitations, and offer directions for future research.

Chapter 6: Summary and Conclusion summarizes our work, reiterating the significance of integrating GNNs with causal inference for ITE estimation. It recaps key findings and contributions and concludes with reflections on the broader impact of this research in healthcare and causal analysis.

Chapter 2

Background and Literature Review

2.1 Background

This section introduces the foundations of causal inference with a focus on ITE estimation. It also discusses core assumptions, key challenges, and modern machine learning methods such as deep neural networks, variational autoencoders, GANs, and Transformers. Structural approaches such as Bayesian networks, structural causal models, and graph neural networks are also discussed, along with causal discovery techniques and intervention modeling, laying the groundwork for the thesis's GNN-based approach.

2.1.1 Correlation vs. Causation

The distinction between correlation and causation is fundamental in understanding relationships between variables [29]. Correlation refers to a statistical association in which changes in one variable are linked to changes in another [30]. However, this does not imply a cause-and-effect relationship. In contrast, causation describes a direct connection where one variable actively influences another, meaning that changes in the cause directly result in changes in the effect [31]. Understanding this difference is critical for drawing accurate conclusions and designing effective interventions [6].

The nature of the relationship between variables further distinguishes correlation from causation. Correlation describes the strength and direction of a relationship between two variables, which can be positive, negative, or nonexistent [32]. For example, taller people tend to weigh more, illustrating a positive correlation. In contrast, causation implies a directional influence, where one variable directly affects another [29]. For instance, administering a vaccine reduces the risk of disease, demonstrating a causal relationship [33]. Beyond simple examples, consider education and income: higher education correlates with higher income, but causation requires isolating education's effect from confounders like so-

cioeconomic background [34]. This distinction underscores the importance of discerning whether a relationship is merely associative or truly causal.

The implications of correlation and causation differ significantly. Correlation indicates a pattern or association between variables but does not provide evidence of an underlying mechanism or explain why the relationship exists [30]. On the other hand, causation offers actionable insights by identifying the mechanism through which one variable directly influences another [7]. This distinction is crucial for guiding decisions and developing effective interventions based on a clear understanding of cause-and-effect relationships [35]. In policy, mistaking correlation (e.g., urban density and crime rates) for causation could lead to misguided urban planning, whereas causal evidence supports targeted interventions [36].

One of the most frequent errors in interpreting data is assuming that correlation implies causation [37]. Such assumptions can lead to erroneous conclusions and poor decision-making. For example, consider the observation that ice cream sales and drowning incidents both increase during the summer. While these two events are correlated, it would be incorrect to conclude that eating ice cream causes drowning. Instead, a third variable, hot weather, influences both and drives the observed relationship [12]. Similarly, a study might find that regions with more hospitals have higher cancer rates, but this reflects population density, not hospitals causing cancer [38]. These examples highlight the importance of considering external factors that might explain a correlation, avoiding ineffective or harmful actions [39].

Confounders are variables that influence both the independent and dependent variables, creating spurious associations [12]. For example, studies may observe that coffee drinkers tend to live longer. While this correlation is evident, it may be driven by confounding factors such as socioeconomic status or health consciousness, which are related to both coffee consumption and longevity [38]. To establish a causal relationship, it is essential to isolate the effect of the variable of interest from these confounders. This can be achieved through experimental designs such as RCTs [9], or through statistical techniques like propensity score matching [13] and instrumental variables [40] in observational studies, ensuring robust causal inference [6].

Researchers use a variety of methods to distinguish correlation from causation, ranging from experimental designs to advanced observational techniques [6]. Correlation is typically measured using statistical tools such as Pearson's correlation coefficient or Spearman's rank correlation [32], quantifying the strength and direction of relationships but not causality [30]. Causal methods, like regression discontinuity [41] or difference-in-differences [42], leverage natural experiments to infer causation, complementing RCTs and offering practical alternatives when randomization is infeasible.

2.1.2 Challenges in ITE Estimation

ITE estimation is a fundamental task in causal inference, aiming to determine the difference in outcomes for a given individual under different treatment conditions [7]. The goal is to compute the potential outcomes as:

$$ITE = E[Y(1) - Y(0)|X], \tag{2.1}$$

where $Y(1)$ and $Y(0)$ are potential outcomes under treatment and control, and X denotes covariates. Despite advancements [29], ITE estimation faces challenges in data quality, model specification, bias, and interpretability.

The fundamental problem of causal inference is the inability to observe both $Y(1)$ and $Y(0)$ for any individual [31], requiring statistical inference of counterfactuals with inherent uncertainty. RCTs mitigate this via randomization, but observational studies face selection bias and confounding [6]. Selection bias arises when treatment assignment correlates with outcome-affecting covariates, while confounding occurs with unmeasured factors influencing both treatment and outcome [12]. Methods like propensity score matching [13], inverse probability weighting [14], and deep latent confounder models [43] address these, but unmeasured confounding persists as a challenge, necessitating sensitivity analyses [39].

Data quality issues such as missing data, measurement errors, and small sample sizes hinder ITE estimation [20]. High-dimensional covariate spaces exacerbate overfitting and sparsity [18], while covariate shift misaligns training and test distributions [44]. Modern solutions include domain adaptation [45] and transfer learning [46], enhancing generalizability. The focus of ITE on heterogeneity contrasts with the uniformity of ATE [47], defined as:

$$ATE = E[Y(1) - Y(0)], \tag{2.2}$$

requiring subgroup analysis [25], though overfitting risks remain [11].

Evaluation of ITE results is complex without ground truth counterfactuals [48]. Synthetic datasets [19], semi-synthetic approaches, and RCT benchmarks assess performance via metrics like PEHE [49]. Model misspecification in traditional methods (e.g., linear regression [50]) biases estimates, while deep learning offers flexibility [24] but demands computational resources and tuning [51]. Interpretability, crucial in high-stakes fields like healthcare, is limited in complex models, prompting advances in explainable artificial intelligence (AI) [52, 53]. Fairness in ITE estimation, critical for equitable outcomes, requires bias mitigation techniques [54], yet defining fairness remains context-dependent [55].

2.1.3 Identifiability Assumptions

The potential outcomes framework, as articulated by Rubin [56], provides a foundational structure for causal inference by defining theoretical outcomes for an individual under distinct scenarios: one where the individual receives a specific treatment and another where they do not. This framework delineates three key concepts: actual outcomes, which are the observed results for an individual given the treatment they received; potential outcomes, which refer to the hypothetical results under each possible treatment; and counterfactual outcomes, the unobserved results corresponding to the treatment not received, meaning the opposite of the actual scenario. This distinction is critical because it underscores the inherent challenge in causal inference: we can observe only one outcome for any given individual, leaving the counterfactual forever hypothetical.

To enable unbiased and generalizable estimation of ITE from observational studies, rather than randomized experiments, a set of critical identifiability assumptions must be satisfied [57]. These assumptions, consistency, exchangeability, positivity, the Stable Unit Treatment Value Assumption (SUTVA), and unconfoundness bridge the gap between theoretical causal effects and empirical estimation, allowing researchers to infer causal relationships from non-experimental data. Each assumption addresses a specific aspect of the data-generating process and imposes constraints that, if violated, could undermine the validity of ITE estimates.

The first assumption, *consistency* [57], posits that the potential outcome for an individual under the treatment they actually received aligns exactly with their observed outcome. In other words, if a patient receives a particular treatment, the outcome observed for that patient is precisely the potential outcome associated with that treatment. This assumption ensures that the treatment effect is well-defined and eliminates ambiguity arising from variations in how a treatment might be applied or interpreted. For instance, consistency would be violated if the same treatment label (e.g., "surgery") encompassed meaningfully different procedures across individuals, leading to discrepancies between potential and observed outcomes.

The second assumption, *exchangeability*, asserts that the treatment is sufficiently well-defined and uniform such that its application does not introduce hidden variation across individuals or contexts. Often interpreted as a form of conditional independence, exchangeability implies that, conditional on observed covariates, the distribution of potential outcomes is the same across treated and untreated groups. This assumption is essential for ruling out unobserved confounding factors that could systematically differ between treatment groups, thereby ensuring that comparisons between treated and untreated individuals are fair and meaningful.

Next, the *positivity* assumption requires that every individual in the population of interest has a non-zero probability of receiving each possible treatment under consideration [58].

This condition ensures that there are no subgroups defined by observed covariates for which treatment assignment is deterministic (i.e., always or never treated). For example, if a certain demographic group is systematically excluded from receiving a treatment due to policy or practice, positivity would be violated, rendering ITE estimation impossible for that group. Positivity thus supports the generalizability of causal estimates across the population and prevents extrapolation beyond the support of the data.

The *SUTVA* combines two interrelated principles: consistency (as described above) and the absence of interference between units [59]. The "no interference" component stipulates that the potential outcomes for one individual are unaffected by the treatment assignments of others. For instance, in a study of a contagious disease treatment, if the treatment status of one patient (e.g., vaccination) influences the infection risk of another, SUTVA would be violated. Similarly, in social interventions, spillover effects, where the treatment of one person indirectly affects the outcome of another, can invalidate this assumption. By assuming both consistency and no interference, SUTVA ensures that the treatment effect for an individual can be isolated and attributed solely to their own treatment status.

Finally, the assumption of *unconfoundness*, particularly crucial in observational data settings, asserts that there are no unobserved variables that simultaneously influence both the treatment assignment and the outcome, beyond those already accounted for in the observed covariates. Also known as ignorability, this assumption allows researchers to treat the observed data as if it were generated from a randomized experiment, conditional on the covariates. For example, if an unobserved factor like socioeconomic status affects both a patient's likelihood of receiving a treatment and their health outcome, and this factor is not included in the model, unconfoundness would fail, leading to biased ITE estimates. Achieving unconfoundness in practice often requires careful selection of covariates based on domain knowledge and causal diagrams, such as those proposed by Pearl [57].

Together, these assumptions form the theoretical backbone of ITE estimation from observational data. While they enable causal inference in the absence of randomization, their validity is not directly testable and must be justified based on the study design, data collection process, and substantive knowledge of the domain. Violations of any one assumption, whether due to inconsistent treatment definitions, hidden confounders, or interference, can introduce bias or limit the scope of the conclusions drawn from the analysis. As such, sensitivity analyses and robustness checks are often employed to assess the plausibility of these assumptions in a given context, ensuring that ITE estimates remain credible and interpretable.

2.1.4 Multilayer Perceptrons

Multilayer Perceptrons (MLPs) are multi-layer architectures that learn hierarchical feature representations from raw data, forming a cornerstone of modern machine learning [51]. An example of MLP with three hidden layers is illustrated in Figure 2.1. The MLP processes data through interconnected nodes, where each layer applies a linear transformation followed by a non-linear activation σ, such as ReLU

$$\sigma(x) = \max(0, x). \tag{2.3}$$

Mathematically, for an input $x \in \mathbb{R}^d$ at the input layer, a hidden layer $l + 1$ computes

$$h^{(l+1)} = \sigma(W^{(l)} h^{(l)} + b^{(l)}), \tag{2.4}$$

where $h^{(l)}$ is the output of the previous layer, $W^{(l)}$ and $b^{(l)}$ are learnable weights and biases, and σ is the activation function. This process propagates through h_1, h_2, and h_3 to produce the final outputs at the output layer. Training optimizes a loss function, such as the Mean Squared Error (MSE):

$$\mathcal{L} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2, \tag{2.5}$$

using backpropagation and gradient descent [60], where y_i is the real outcome and $\hat{y}_i$ is the predicted outcome.

In causal inference, MLPs excel at modeling complex, non-linear relationships between covariates X, treatments T, and outcomes Y. Methods like TARnet [24] leverage MLPs with two branches to estimate potential outcomes ($\mathbb{E}[Y^1|X]$ and $\mathbb{E}[Y^0|X]$) for ITE estimation.

2.1.5 Generative Adversarial Networks and Variational Autoencoders

Generative Adversarial Networks (GANs), introduced by Goodfellow *et al.* [61], are a class of generative models that learn to produce realistic data through a competitive framework. GANs consist of two neural networks: a generator G, which synthesizes data from random noise $z \sim p(z)$, and a discriminator D, which distinguishes real data $x \sim p_{\text{data}}(x)$ from generated samples $G(z)$. These networks are trained adversarially via a minimax objective:

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{\text{data}}}[\log D(x)] + \mathbb{E}_{z \sim p(z)}[\log(1 - D(G(z)))]. \tag{2.6}$$

The discriminator maximizes its classification accuracy, while the generator minimizes the discriminator's ability to detect fakes, converging when G approximates the true data distribution.

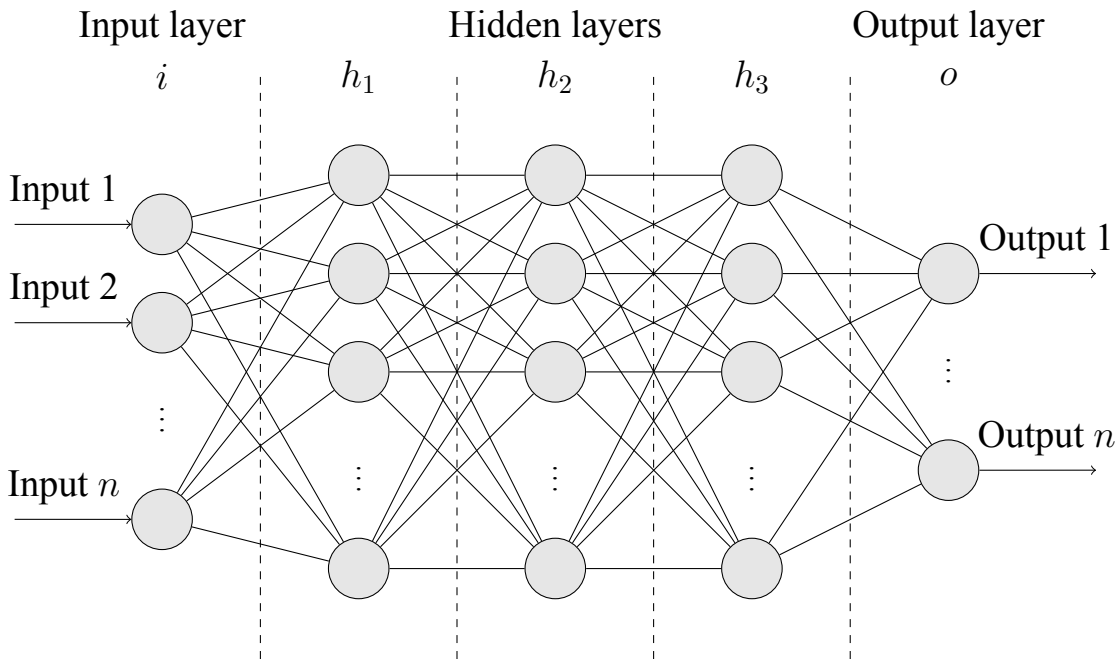


Figure 2.1. The example MLP comprises an input layer i with inputs labeled "Input 1", "Input 2", …, "Input n", followed by three hidden layers h_1, h_2, and h_3, each containing a predefined number of nodes, and an output layer o with outputs labeled "Output 1", …, "Output n". The dots between the hidden units indicate the potential for additional hidden units.

Variational Autoencoders (VAEs), proposed by Kingma and Welling [62], offer an alternative generative approach by learning a probabilistic latent space. VAEs consist of an encoder, which maps input x to a latent distribution $q(z|x)$ (typically Gaussian, parameterized by mean μ and variance σ^2), and a decoder, which reconstructs x from samples $z \sim q(z|x)$. Training optimizes a variational lower bound:

$$\mathcal{L} = \mathbb{E}_{z\sim q(z|x)}[\log p(x|z)] - D_{\mathrm{KL}}(q(z|x)||p(z)), \tag{2.7}$$

balancing reconstruction accuracy and regularization via the KL-divergence to a prior $p(z)$ (e.g., $\mathcal{N}(0,1)$).Unlike GANs, VAEs provide a structured and interpretable latent space, making them particularly useful for tasks like data imputation and distribution modeling. In causal inference, VAEs have been extended to models such as CEVAE [43], which uses the VAE framework to infer latent confounders and thereby improve estimates of individual treatment effects in the presence of unobserved confounding.

2.1.6 Transformers

Transformers, introduced by Vaswani *et al.* [63], are attention-based architectures designed to model long-range dependencies, initially for natural language processing (NLP). Un-

like recurrent neural networks (RNNs), Transformers process inputs in parallel, using self-attention to weigh element importance. Given an input matrix $X \in \mathbb{R}^{n \times d}$, where n is the sequence length and d is the feature dimension, the model computes queries $Q = XW^Q$, keys $K = XW^K$, and values $V = XW^V$ through learned linear projections. Self-attention is then computed as

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^\top}{\sqrt{d_k}})V, \tag{2.8}$$

where d_k is the key dimension. Multi-head attention enhances this by applying attention across multiple subspaces, improving expressiveness and scalability.

Beyond NLP, the ability of Transformers to capture contextual relationships has influenced graph learning, notably through Graph Attention Networks (GATs) [64], which adapt self-attention to weight node neighbors dynamically. In causal inference, Transformers hold potential for modeling temporal or relational confounders, though their direct use remains limited compared to MLPs. Our work draws inspiration from this paradigm: the extension of GNN-TARnet, GAT-TARnet (Section 3.1.5), incorporates GAT layers to enhance ITE estimation by adaptively weighting features within causal DAGs. While Transformers themselves are not applied here, their attention mechanism informs our approach, bridging MLPs' flexibility with graph structure awareness.

2.1.7 Integral Probability Metrics

Integral Probability Metrics (IPMs) quantify differences between probability distributions and have become valuable in causal inference for aligning treated and untreated groups, effectively mimicking conditions of RCTs. Commonly used IPMs include Maximum Mean Discrepancy Squared (MMDSQ) and Wasserstein distance.

The MMDSQ measures the squared difference between expectations of distributions p and q in a reproducing kernel Hilbert space (RKHS):

$$\text{MMDSQ}(p, q) = \|\mathbb{E}_{x \sim p}[\phi(x)] - \mathbb{E}_{y \sim q}[\phi(y)]\|^2_{\mathcal{H}}\,, \tag{2.9}$$

where ϕ is a feature mapping into the RKHS, associated with a kernel function such as the Gaussian kernel [65].

The Wasserstein distance, also known as the Earth Mover's Distance, quantifies the minimal cost required to transform one probability distribution into another by optimally transporting mass between them. In optimal transport theory [66], the first-order Wasserstein distance between probability measures p and q, defined on a metric space $\mathcal{X} \subseteq \mathbb{R}^d$, is formally defined as:

$$W(p, q) = \inf_{\gamma \in \Pi(p,q)} \mathbb{E}_{(x,y) \sim \gamma}\big[\|x - y\|\big], \tag{2.10}$$

where $\Pi(p, q)$ denotes the set of all joint distributions (couplings) with marginals p and q, and $\|\cdot\|$ typically refers to the Euclidean norm. The marginals p and q are obtained by integrating the joint distribution $\gamma(x, y)$:

$$p(x) = \int \gamma(x, y)\, dy, \quad q(y) = \int \gamma(x, y)\, dx. \tag{2.11}$$

2.1.8 Bayesian Networks

Bayesian Networks (BNs) are probabilistic graphical models that encode the joint probability distribution of a set of variables using a directed acyclic graph (DAG). They represent statistical relationships, whether causal or associative, through directed edges and enable efficient probabilistic inference by factorizing the joint distribution into local conditional probability distributions (CPDs). Widely applied in domains such as medical diagnosis, fraud detection, and decision support systems, BNs offer a structured approach to reasoning under uncertainty [67]. Formally, a Bayesian Network is defined by a set of random variables $X_1, X_2, ..., X_n$, each represented as a node in a DAG $G = (V, E)$, where edges denote dependency relations. Each node X_i is associated with a conditional probability function $P(X_i|\text{pa}_i)$, where pa_i are its parent nodes. This factorization allows BNs to model complex probability distributions efficiently, avoiding the need to store an exponential number of parameters. A key strength of BNs is their ability to perform probabilistic reasoning based on partial observations. Using Bayes' theorem:

$$P(H|E) = \frac{P(E|H)P(H)}{P(E)},$$

where H is the hypothesis and E is the evidence, posterior probabilities can be computed given observed data. For small or tree-structured BNs, exact inference methods like variable elimination or belief propagation are feasible [67]. However, inference in general BNs is NP-hard, especially for large, densely connected graphs, necessitating approximate methods such as Monte Carlo techniques, Markov Chain Monte Carlo (MCMC), or variational inference for scalability in high-dimensional settings [68].

BNs excel at answering associational queries, such as predicting the probability of a disease given symptoms, but they do not inherently distinguish between correlation and causation. This limits their ability to address causal queries, such as estimating treatment effects, without additional frameworks like those introduced in SCMs, which are discussed in the next subsection. Learning BN structures from observational data poses challenges: multiple DAGs can represent the same probabilistic relationships, latent confounders may introduce biases, and the combinatorial search space grows rapidly with high-dimensional data. Structure learning employs constraint-based methods (e.g., the PC Algorithm [69]), score-based methods (e.g., Bayesian Information Criterion, BIC [70]), or hybrid approaches.

2.1.9 Structural Causal Models

Structural Causal Models (SCMs), introduced by Judea Pearl [29], provide a mathematical framework for representing and reasoning about causal relationships, building on probabilistic foundations like those of BNs introduced in the previous section. Unlike BNs, which prioritize probabilistic dependencies [68], SCMs explicitly model causal mechanisms. An SCM comprises endogenous variables $X = \{X_1, X_2, \ldots, X_n\}$ (observed factors), exogenous variables U (latent sources of variation), and structural equations:

$$X_i = f_i(\text{pa}_i, U_i), \tag{2.12}$$

where pa_i are the parent nodes of X_i in a causal DAG, and f_i is a function encoding causal mechanisms [29]. A probability distribution $P(U)$ over exogenous variables introduces uncertainty, aligning with BNs' probabilistic factorization but emphasizing causality. The DAG structure formalizes conditional independence assumptions, enabling precise causal analysis [69].

SCMs face practical challenges: specifying the causal graph requires domain knowledge and is error-prone [35], unmeasured confounders can bias estimates [12], and modeling complex systems is computationally intensive [71]. Unlike data-driven methods, traditional SCMs rely on predefined relationships, limiting adaptability to high-dimensional data [24]. However, their ability to handle interventions (e.g., via the $do(\cdot)$ operator) distinguishes them from BNs, offering a framework for causal queries central to our ITE estimation. Recent research integrates SCMs with GNNs [28], enabling data-driven causal discovery and scalable inference via message-passing [27, 72]. This hybrid approach enhances SCMs' applicability in healthcare (e.g., personalized treatment), economics, and social sciences [73], directly informing our method's causal structure.

2.1.10 Graph Neural Networks

GNNs have emerged as powerful tools for analyzing graph-structured data, evolving from foundational models to advanced applications, as reviewed by Wu *et al.* [23]. Early GNNs, such as recurrent models by Scarselli *et al.* [21], paved the way for tasks like traffic prediction [74, 75] and scene graph generation [76], while spatial methods like GCN by Kipf and Welling [72] scaled applications to online shopping recommendations [77] and drug discovery [78]. Modern variants, including GATs [64] and GraphSAGE [79], handle diverse graphs [23, 80]. Unlike BNs, which focus on probabilistic inference, or SCMs, which model causality, GNNs excel at learning representations from graph data, often prioritizing correlations over causality [81], except in works like Zhai *et al.*'s causal advertising model [82].

Efforts to infer causal effects with structural information bridge SCMs and GNNs. Zečević *et al.* [28] showed that interventions on GNNs align with interventions on SCMs, a

principle we adapt for ITE estimation, enhancing data representation over BN-style associational models. Wein *et al.* [83] predicted brain dynamics, and Chu *et al.* [84] estimated treatment effects, both leveraging subject-related graphs, unlike our intra-subject focus. Non-convolutional method by Parafita *et al.* [85] contrasts with our convolutional GNN approach.

2.1.11 The Do-Operator and Causal Interventions

Causal inference, as we explore in this thesis, seeks to reveal cause-and-effect relationships beyond statistical patterns, a goal formalized by **do-operator** introduced by Judea Pearl within SCMs [29]. The do-operator, written as $do(X = x)$, represents an intervention that fixes a variable X to a value x, breaking its natural dependencies in the causal graph. This lets us estimate the causal effect of X on an outcome Y, distinct from observational conditioning. While $P(Y|X = x)$ reflects correlations, $P(Y|do(X = x))$ captures the effect of deliberately setting X, a key step in ITE estimation.

The strength of the do-operator is its handling of confounding, a major hurdle in causal inference. Imagine a DAG where a confounder Z influences both treatment T and outcome Y (i.e., $Z \to T$ and $Z \to Y$), alongside $T \to Y$. The observational $P(Y|T = t)$ blends T's causal effect with Z's confounding impact. Applying the do-operator, $P(Y|do(T = t))$, cuts the $Z \to T$ edge, isolating T's effect on Y. This is formalized by the backdoor adjustment:

$$P(Y|do(T = t)) = \sum_z P(Y|T = t, Z = z)P(Z = z), \tag{2.13}$$

where summing over Z corrects for confounding, assuming Z meets the backdoor criterion [29]. This ensures causal effects are identifiable, vital for accurate ITE estimation [6].

In ITE estimation, the do-operator supports the potential outcomes framework, central to our work [56]. The do-operator helps by modeling interventions on T, letting us estimate $P(Y|do(T = 1))$ and $P(Y|do(T = 0))$ from the graph. In our GNN approach, this means altering the graph structure. for example removing edges into T during node updates, to predict these distributions. The do-operator shines in both randomized controlled trials and observational data. In RCTs, randomization mimics $do(T = t)$ by eliminating confounding, making $P(Y|T = t) = P(Y|do(T = t))$ [9]. In observational studies, without such control, it guides methods like propensity score matching [13] or our GNN framework. Its role in GNNs, explored in Section 3.1.4.

2.1.12 Causal Discovery Techniques

Bayesian Networks, SCMs, and GNNs all rely on connectivity information to model relationships between variables, making the acquisition of such structures a critical step in causal

inference. This process, known as causal discovery, seeks to infer the underlying DAG that represents causal relationships from observational data. Several widely used algorithms have been developed to address this challenge, each with distinct assumptions and methodologies. Notable among these are the Linear Non-Gaussian Acyclic Model (LiNGAM) [86], the Peter-Clark (PC) algorithm [69], and the Greedy Equivalence Search (GES) [87]. These techniques form the backbone of many causal inference frameworks, including those employed in our GNN-based ITE estimation approach. For a comprehensive review of these and other methods, see Glymour *et al.* [73]. Below, we detail each method, highlighting their mechanisms, assumptions, and relevance to our work.

LiNGAM [86] is a constraint-based approach that leverages the non-Gaussianity of noise distributions to identify causal structures. LiNGAM assumes that the data-generating process follows a linear model, where each variable X_i is a linear combination of its parent variables plus an independent, non-Gaussian error term:

$$X_i = \sum_{j \in \text{pa}_i} a_{ij} X_j + \epsilon_i, \tag{2.14}$$

where pa_i denotes the parents of X_i, a_{ij} are coefficients, and ϵ_i is a non-Gaussian noise term with zero mean. Unlike traditional Gaussian models, where causal directionality is unidentifiable due to symmetry, LiNGAM exploits the asymmetry of non-Gaussian distributions (e.g., via higher-order moments like skewness or kurtosis) to determine the direction of edges. The algorithm employs Independent Component Analysis to separate the observed variables into independent sources, reconstructing the DAG by estimating the mixing matrix A. LiNGAM's strength lies in its ability to provide a unique DAG under the non-Gaussian assumption, making it particularly useful for datasets where noise deviates from normality. However, its reliance on linearity and non-Gaussianity limits its applicability to nonlinear systems or datasets with Gaussian noise, necessitating validation against alternative methods in practice.

PC algorithm [69], named after its developers Peter Spirtes and Clark Glymour, is another constraint-based method that infers causal structures by testing conditional independence among variables. It begins with a fully connected undirected graph and iteratively removes edges based on statistical tests (e.g., Fisher's z-test for continuous data or chi-squared tests for discrete data) that identify when two variables X_i and X_j are independent given a conditioning set S. Formally, if $X_i \perp X_j | S$, the edge between X_i and X_j is removed. Once the skeleton (undirected graph) is established, the algorithm orients edges using rules such as the collider rule: if $X_i \rightarrow X_k \leftarrow X_j$ and X_i and X_j are not adjacent, X_k is a collider, fixing the directionality. The PC algorithm assumes causal faithfulness, meaning all independencies in the data reflect the DAG structure, and causal sufficiency, meaning no unmeasured confounders exist. Its computational efficiency and flexibility with both

continuous and categorical data make it a cornerstone of causal discovery. However, its sensitivity to sample size and test accuracy can lead to spurious edges or incomplete graphs, particularly in small or noisy datasets, requiring stability techniques like multiple runs or bootstrapping [73].

GES [87] adopts a score-based approach, optimizing a goodness-of-fit score to search for the DAG that best explains the data. GES uses a two-phase greedy search: the forward phase adds edges to maximize a score, e.g., Bayesian Information Criterion (BIC), starting from an empty graph, while the backward phase removes edges to refine the structure, converging to a local optimum within the Markov equivalence class of DAGs. The BIC, for instance, balances model fit and complexity:

$$\text{BIC} = -2\ln(\mathcal{L}) + k\ln(n), \tag{2.15}$$

where $\mathcal{L}$ is the likelihood of the data given the model, k is the number of parameters, and n is the sample size. Unlike constraint-based methods, GES does not rely on conditional independence tests, making it robust to statistical errors in small samples. It assumes the data is generated by a DAG and that the scoring function is consistent, meaning it identifies the true structure as $n \to \infty$. GES excels in settings where a probabilistic model, for example, BN, is specified, offering a globally optimized structure rather than the locally constrained output of PC or LiNGAM. However, its computational complexity grows with the number of variables, and it may struggle with high-dimensional datasets unless paired with dimensionality reduction techniques.

These causal discovery techniques underpin the connectivity information used in BNs, SCMs, and GNNs. BNs model joint probability distributions over variables, with edges representing conditional dependencies inferred via methods like PC or GES [29]. SCMs extend this by incorporating causal mechanisms, often requiring DAGs from discovery to simulate interventions like the do-operator, critical for ITE estimation [73]. GNNs, as employed in our GNN-TARnet and GAT-TARnet, leverage these DAGs to propagate information across nodes, enhancing predictive power by encoding structural relationships [23]. In our work, we use all tree methods to discover the causal graphs if they are missing.

2.2 Review of ITE Estimation Methods

Papers were sourced via Google Scholar using keywords "Causal Inference" and "Individual Treatment Effect," supplemented by Li *et al.*'s review [15].

2.2.1 Classical Methods for Causal Effect Estimation

Estimating causal effects, particularly the ITE for individual i defined as

$$ITE_i = Y_i(1) - Y_i(0), \tag{2.16}$$

where $Y_i(1)$ and $Y_i(0)$ are potential outcomes under treatment and controll, is a cornerstone of causal inference, especially in observational studies where RCT data are unavailable [7]. Unlike RCTs, which ensure random treatment assignment to eliminate confounding, observational studies require methods to adjust for biases introduced by non-random treatment allocation. Classical approaches, rooted in the potential outcomes framework, include regression adjustment (notably linear regression), matching, propensity score techniques, and forest-based methods like Random Forests and Causal Forests. These methods operate on observed data (X_i, T_i, Y_i), where X_i are covariates, T_i is the binary treatment (1 for treated, 0 for control), and Y_i is the observed outcome, aiming to balance covariates and estimate ITE [6].

Regression adjustment, with linear regression as its simplest form, models the outcome Y as a function of treatment and covariates. The linear model is typically:

$$Y = \beta_0 + \beta_1 T + \beta_2 X_1 + \cdots + \beta_{p+1} X_p + \epsilon,$$

where β_0 is the intercept, β_1 estimates the average treatment effect (ATE), $\beta_2, \ldots, \beta_{p+1}$ adjust for confounding covariates, and $\epsilon \sim N(0, \sigma^2)$ is random error [88]. ITE is derived as $\hat{ITE}_i = \hat{Y}_i(1) - \hat{Y}_i(0) = \hat{\beta}_1$, with counterfactuals $\hat{Y}_i(1) = \hat{\beta}_0 + \hat{\beta}_1 + \hat{\beta}_2 X_{i1} + \cdots$ and $\hat{Y}_i(0) = \hat{\beta}_0 + \hat{\beta}_2 X_{i1} + \cdots$, assuming a constant effect across individuals [6]. This method excels in RCTs or settings with fully observed confounders and linear relationships, offering interpretability and computational efficiency. However, its reliance on correct model specification (linearity, homoscedasticity) and the absence of unmeasured confounding $(Y(0), Y(1) \perp T|X)$ often fails in observational data with nonlinear effects, interactions, or high-dimensional covariates, risking bias and overfitting [12, 18].

Matching methods counter confounding by pairing each treated individual ($T_i = 1$) with an untreated counterpart ($T_j = 0$) based on covariate similarity, using metrics like Euclidean or Mahalanobis distance on X [89]. ITE is estimated as $\hat{ITE}_i = Y_i - Y_j$ for matched pairs, averaging over multiple matches if applicable. By approximating RCT-like balance, matching reduces bias but requires sufficient overlap (positivity: $0 < P(T = 1|X) < 1$) and precise covariate measurement. Variants like coarsened exact matching (CEM) bin covariates for robustness [90]. Limitations arise when matches are scarce or unmeasured confounders persist, undermining causal validity [8].

Propensity score methods estimate $g(X) = P(T = 1|X)$, often via logistic regression, and use it for matching, stratification, or inverse probability of treatment weighting (IPTW)

to balance groups [13]. IPTW estimates ITE as:

$$\hat{ITE}_i = \frac{T_i Y_i}{g(X_i)} - \frac{(1-T_i)Y_i}{1-g(X_i)},$$

assuming exchangeability, positivity, and consistency [6]. This approach simplifies high-dimensional covariate adjustment but shares matching's vulnerabilities to unmeasured confounding and positivity violations, where extreme $g(X)$ values destabilize weights [91].

To overcome rigidity of the linear regression, forest-based methods employ ensemble learning for flexibility. Random Forests (RF), developed by Breiman [92], build multiple decision trees on bootstrapped samples, splitting on random feature subsets. For ITE, RF fits separate models for treated and control groups (T-Learner [93]), predicting $\hat{Y}_i(1)$ and $\hat{Y}_i(0)$ to compute $\hat{ITE}_i = \hat{Y}_i(1) - \hat{Y}_i(0)$. RF captures nonlinearities (e.g., $Y = X_1^2 T + X_2$) and interactions, reducing overfitting via averaging [18]. However, it lacks native causal adjustment, relying on external balancing (e.g., propensity scores), which may falter with strong confounding [13].

Causal Forests (CF), advanced by Wager and Athey [25], refine RF for heterogeneous treatment effects. CF splits trees to maximize treatment effect variance within leaves, estimating $\hat{ITE}_i$ as:

$$\hat{ITE}_i = \frac{1}{|L_T|} \sum_{j \in L_T} Y_j - \frac{1}{|L_C|} \sum_{j \in L_C} Y_j,$$

where L_T and L_C are treated and control units in leaf L containing i. "Honest" splitting, which separates tree-building and estimation samples, reduces bias, and CF offers confidence intervals via asymptotic normality [25]. CF excels at ITE heterogeneity and high-dimensional data but assumes unconfoundedness and positivity, with computational complexity rising in sparse or noisy settings [94]. Recent advances, like Bayesian Additive Regression Trees (BART) [95], enhance flexibility and uncertainty quantification [19].

hese classical methods, including regression adjustment, matching, propensity scores, and forest-based approaches, provide robust baselines for causal effect estimation in controlled, low-dimensional settings. Yet, they can struggle with structural covariate dependencies (e.g., $X_1 \to X_2 \to Y$), limitations our GNN-based approach addresses by encoding DAG relationships.

2.2.2 Deep Learning Methods for ITE Estimation

Most of the deep learning methods for ITE estimation can be categorized into three primary classes, as outlined in the comprehensive review by Li *et al.* [15]. These classes include representation-based methods, covariate-confounding learning methods, and GAN-based methods for counterfactual generation. In addition to these, we categorize meta-learners separately due to their distinct approach and methodological differences, which diverge from

the specialized frameworks of the other three classes. Each category reflects a unique strategy for addressing the core challenge of ITE estimation, namely, inferring causal effects at the individual level from observational or experimental data, while addressing issues such as confounding, distributional imbalance, and counterfactual prediction.

Representation-based methods are among the most widely used approaches for ITE estimation, making this a dynamic and active research area with frequent advancements and a steady stream of new publications. These methods are termed "representation-base" because they transform the original covariates, which describe individuals or units in the dataset, into a hidden representation space, effectively re-encoding the information into a new, often lower-dimensional form. This transformation aims to distill the most causally relevant aspects of the data while filtering out noise or irrelevant variation. The distinguishing feature of these methods lies in how they manipulate features within this hidden representation space to enable accurate and robust treatment effect estimation.

Some representation-based methods, such as TARNet [24], adopt a straightforward approach by constructing a shared neural network architecture for both treated and untreated groups without explicitly altering the hidden representation beyond this structural design. In contrast, other methods prioritize minimizing the discrepancy between the distributions of treated and untreated groups within the representation space, thereby approximating the balance achieved in a RCT. This balancing act reduces bias due to covariate imbalance and enhances the reliability of causal inferences. Notable examples of such methods include BNN (Balancing Neural Network) [48], which uses balanced representations to adjust for confounding; DCN-PD [96], a deep counterfactual network with propensity dropout; CFR (Counterfactual Regression) [24], which enforces domain-invariant representations; SITE [97], focusing on sufficient representation learning; ACE [98], an adaptive causal estimator; DragonNet [99], which integrates treatment propensity into the architecture; BWCFR [100], a balanced Wasserstein CFR variant; CFR-Weight [101], which emphasizes generalization across domains; and CITE [102], leveraging contrastive learning for improved representations. Alternatively, methods like DKLITE [49] take a different tack by clustering factual data around counterfactual representations, using kernel-based techniques to enhance the estimation of causal effects. This diversity within the representation-based class underscores its flexibility and ongoing evolution as researchers refine techniques to address specific challenges in ITE estimation.

GAN-based methods, as their name suggests, harness the power of GANs to either generate counterfactual outcomes or balance distributions within the latent space, offering a data-driven solution to the problem of unobserved counterfactuals. In this framework, a generator learns to produce realistic counterfactuals, representing what would have happened under an alternative treatment, while a discriminator evaluates their plausibility, creating an

adversarial training dynamic that refines the model's predictions. This approach is particularly advantageous in settings with complex or high-dimensional data, where traditional methods may struggle to model counterfactuals effectively. Prominent examples include GANITE [103], which generates individualized treatment effects directly; ABCEI [104], an adversarial balancing method for causal effect inference; CETransformer [105], which integrates transformer architectures with GANs for enhanced counterfactual modeling; and CBRE [106], a cycle-consistent approach that ensures coherence between factual and counterfactual predictions. By simulating unobserved scenarios, GAN-based methods provide a powerful tool for ITE estimation, though their reliance on adversarial training can introduce challenges related to stability and computational complexity.

Covariate-confounding learning methods focus on explicitly addressing the role of confounding variables, which are covariates that influence both treatment assignment and outcomes, within observational data. These methods aim to disentangle true causal effects from spurious correlations by modeling latent variables or hidden confounders that might otherwise bias the estimation process. For instance, CEVAE (Causal Effect Variational Autoencoder) [43] employs a variational approach to infer latent confounders from proxy variables, while TEDVAE [107] extends this idea to treatment effect estimation with disentangled representations. Other methods, such as DerCFR [108], distinguish confounders from non-confounders by learning deconfounded representations tailored to causal inference. Alternatively, approaches like DONUT (Deconfounded Neural Treatment) [109] enforce orthogonality constraints between the outcomes and treatment assignment, ensuring that the model isolates the treatment's direct effect. This class of methods is particularly critical in real-world settings where randomized experiments are infeasible, and confounding poses a persistent threat to valid inference.

Meta-learners [93] constitute a general framework for estimating potential outcomes, that is, what would happen under different treatment conditions, using data associated with a treatment. The term "general" reflects their flexibility, as they can incorporate virtually any machine learning algorithm as a base learner at their core, adapting these models to the task of ITE estimation. This adaptability distinguishes meta-learners from the more specialized approaches of the other classes. Key examples include the S-learner [93], which uses a single model to predict outcomes for all treatment conditions; the T-learner [93], which trains separate models for treated and untreated groups; the X-learner [93], which builds on the T-learner by incorporating propensity scores and refining estimates in a two-stage process; and the R-learner [110], which reframes ITE estimation as a residual learning problem to improve robustness. By leveraging pre-existing algorithms and tailoring them to causal inference, meta-learners offer a practical and widely applicable solution, though their performance often depends heavily on the choice and quality of the base learners.

The following sections detail the key features and drawbacks of these methods, organized by their category and year of development. This structure provides a chronological perspective on advancements in the field, tracing the evolution of ITE estimation techniques from early foundational work to cutting-edge innovations. By highlighting methodological breakthroughs alongside inherent limitations, such as computational demands, assumptions about data distributions, or sensitivity to model specification, this organization aims to offer a balanced and comprehensive overview of the state of the art in ITE estimation.

2.2.3 Representation-based Methods

One of the earliest methods leveraging deep learning for representation learning in counterfactual inference was BNN, introduced in June 2016 by Johansson *et al.* [48]. In this work, the authors conceptualized the causal inference problem as a domain adaptation task, aiming to minimize the distributional discrepancy between treated and untreated groups to achieve balanced representations. The proposed model employed a fully connected neural network where the initial layers transformed input data into a latent representation space. These representations, concatenated with the treatment assignment, were subsequently used to predict the factual outcomes. The loss function comprised two components: a term quantifying the discrepancy between the treated and untreated distributions and the MSE between the observed and predicted outcomes. A notable strength of this approach was its ability to jointly optimize representation learning and outcome prediction. However, a limitation arose from the model design, which used a single branch for predicting outcomes for both treated and untreated groups. Since the treatment assignment is merely concatenated with the latent representation, there existed a risk that the model might not effectively incorporate the treatment assignment during outcome prediction.

In June 2017, Alaa *et al.* [96] proposed the Deep Counterfactual Network with Propensity Dropout (DCN-PD) method. In this approach, data first passes through fully connected layers, referred to as representation layers, and subsequently through multitask learning branches. These branches treat the estimation of potential outcomes as two separate but related learning tasks. To address the issue of selection bias in observational studies, the authors introduced propensity score learning layers. These layers compute the propensity score, which is then used to determine the probability of dropout. Specifically, as the score deviates further from 0.5, the dropout probability increases. This mechanism ensures greater focus on overlapping regions of the data distribution, thereby mitigating the influence of outliers on the estimated outcomes. However, we identify the complexity of the dropout scheme as a limiting factor, posing challenges in its practical implementation and potentially affecting the scalability of the approach.

In August 2017, architectures utilizing representation layers followed by treatment-specific branches garnered significant attention and were adopted in several methodologies, notably Counterfactual Regression (CFR) and TARnet (Treatment Agnostic Representation Network), as introduced by Shalit *et al.* [24]. These approaches share a common structure but differ in their handling of hidden representations. CFR networks focus on minimizing the distributional discrepancy between treated and untreated populations using metrics such as the Wasserstein distance (CFR-Wass) and Maximum Mean Discrepancy (CFR-MMD). By doing so, these methods effectively reduce covariate and distributional imbalances, improving the quality of ITE estimation. However, CFR-Wass can become computationally expensive on large datasets, while CFR-MMD may struggle with extremely unbalanced treatment and control groups, limiting its robustness under such conditions. In contrast, TARnet does not explicitly address distributional discrepancies in its design, making it less robust in addressing covariate imbalance. Nevertheless, its simpler architecture results in improved computational efficiency, making it a practical choice for scenarios with fewer concerns about covariate imbalance.

In December 2018, Yao *et al.* [97] introduced the SITE (Similarity-preserved Individualized Treatment Effect) method, a novel approach designed to enhance ITE estimation. The key strength of the method lies in its ability to preserve local similarity within the data while simultaneously balancing distributions. Additionally, SITE incorporates a mechanism to focus on extreme case samples within each mini-batch, which can improve performance in scenarios with limited overlap or challenging data distributions. However, a notable drawback of SITE is its tendency to underperform when the data exhibits strong overlap between treated and untreated groups. In such cases, the emphasis of the method on extreme cases may become less relevant, leading to suboptimal results.

In October 2019, Shi *et al.* [99] proposed an extension of the TARNet architecture by introducing a separate branch dedicated to propensity score computation, resulting in a model named DragonNet. This approach is based on the assumption that the treatment effect is independent of covariates, where covariates are used exclusively for outcome prediction rather than treatment assignment. The primary drawback of DragonNet compared to TARNet is its increased model complexity, which may pose challenges in terms of computational efficiency and implementation.

In November 2019, Yao *et al.* [98] introduced the ACE (Adaptively Similarity-Preserved Representation Learning) method, representing a significant improvement over the SITE approach for causal effect estimation. ACE enhances representation learning by incorporating similarity information from the original feature space. It achieves this by minimizing the distance between treated and untreated groups while simultaneously optimizing a similarity loss. This similarity loss ensures that the structural fidelity of the learned representation

space aligns with the original feature space, which is critical for robust causal inference. The primary drawback of the ACE method is the added computational overhead, which may limit its efficiency in large-scale applications.

In August 2020, Zhang *et al.* [49] proposed DKLITE (Deep Kernel Learning for Individual Treatment Effect estimation), a novel method aimed at minimizing counterfactual variance in treatment effect estimation. This approach uses clustering to align counterfactual data with factual representations, effectively addressing covariate shift. Notably, DKLITE avoids relying on IPM for balancing representations between treatment and control groups, which is a common strategy in comparable methods. A key drawback of DKLITE is its sensitivity to kernel choice and parameter settings, which can significantly impact its performance and robustness.

In the same month, Johansson *et al.* [101] introduced CFR-Weight, a method that employs the MMDSQ metric to balance the distributions of treated and untreated groups. This balancing is weighted by the propensity score, building on a mechanism previously implemented in DragonNet. The authors assert that this weighting approach yields more reliable results, particularly in settings with significant treatment assignment imbalance. However, the effectiveness of the method is sensitive to errors in propensity score estimation, which can adversely affect its performance and reliability.

In April 2022, Li and Yao [102] proposed CITE (Contrastive Individual Treatment Effects Estimation), a contrastive learning framework specifically designed for causal inference. This method utilizes self-supervision within the data to create balanced and predictive representations while integrating causal prior knowledge to improve the reliability of causal inference. The primary drawbacks of CITE are its architectural complexity and its reliance on the estimated propensity score, which can introduce vulnerabilities if the propensity scores are inaccurately estimated.

In March 2023, Tesei *et al.* [111] introduced BCAUSS (Balancing Covariates Automatically Using Self-Supervision), an improved version of Dragonnet designed to perform effectively under challenging conditions, such as violations of the positivity assumption. This enhancement aims to address scenarios where treatment assignment probabilities deviate significantly from uniformity, ensuring robust causal effect estimation in the presence of such complexities. The primary drawback of BCAUSS is its increased computational cost. This arises from the need to load the entire training set to compute the auto-balancing self-supervised term, which can be a significant limitation for large-scale datasets.

A concise summary of the strengths and limitations of the representation-based methods is provided in Table 2.3 offering a clear and comparative overview, aiding in understanding the trade-offs associated with each approach as well as the code availability.

2.2.4 GAN-based Methods

In 2018, significant progress was made in leveraging Generative Adversarial Networks (GANs) [112] for ITE estimation. In April, Yoon *et al.* [103] introduced GANITE (Generative Adversarial Network for Inference of Individualized Treatment Effects), a novel method that uses an adversarial training framework to model potential outcomes under alternative treatments. GANITE explicitly generates proxies for counterfactual outcomes, which are then used to train its ITE estimator. However, a key drawback of GANITE is its complex training process, which arises from the inherent challenges of training adversarial networks.

In the same month, Du *et al.* [104] proposed ABCEI (Adversarial Balancing-based representation learning for Causal Effect Inference), a method leveraging adversarial learning to balance covariate distributions in the latent space. Unlike traditional methods, ABCEI does not rely on specific assumptions regarding the treatment assignment mechanism. To mitigate potential information loss during representation learning, the approach incorporates a mutual information estimator designed to preserve essential predictive information from the original covariates. This integration enhances robustness of the method to treatment selection bias and improves its overall efficacy. However, the complexity of the method is a notable drawback.

In July 2021, Guo *et al.* [105] introduced CETransformer (Casual Effect estimation model via Transformer based representation learning), a transformer-based model designed to address two critical challenges in treatment effect estimation: selection bias and the absence of counterfactual outcomes. This method employs transformer-based representation learning, utilizing a self-supervised transformer to capture intricate correlations among covariates through self-attention mechanisms. Furthermore, an adversarial network is incorporated to balance the distributions of treated and control groups within the representation space, mitigating biases and enhancing the robustness of the model.

In April 2022, Zhou *et al.* [106] proposed the CBRE (Cycle-Balanced REpresentation learning for counterfactual inference) method, which employs adversarial training to balance the representations of treatment and control groups, thereby reducing confounding bias. The authors assert that their approach preserves data integrity through an "information loop," minimizing information loss and enabling robust causal inference. However, a potential concern with CBRE pertains to its application in scenarios characterized by nonlinear dependencies between covariates and the outcome. Specifically, the tendency of the method to map treated and untreated units closely in the hidden space can result in a loss of outcome-discriminative information, which, in turn, may obscure critical distinctions necessary for accurate treatment effect estimation. We highlight a similar concern in the context of the CETransformer approach. Summary of the methods is presented in the Table 2.1.

2.2.5 Covariate-Confounding Learning Methods

In December 2017, Louizos *et al.* [43] introduced the Causal Effect Variational Autoencoder (CEVAE), representing a pioneering effort to integrate Variational Autoencoders (VAEs) into the domain of causal inference. This method adapts VAEs to DAGs and defines a generative process underlying the observed data. Specifically, CEVAE models the joint distribution of covariates, treatment, and outcomes conditioned on latent variables, offering a robust framework to address causal effect estimation. By leveraging the expressiveness of VAEs, the approach facilitates the modeling of complex distributions and provides a principled solution for handling hidden confounding variables in causal inference. The method is, however, sensitive to the choice of the hyperparameters.

TEDVAE [107] was an improvement over CEVAE introduced in May 2021. Unlike CEVAE it disentangles latent confounders into independent components based on their type, namely binary or continuous which in the end improves their interpretability. The biggest drawback of the method is an additional computational overhead due to this disentanglement process.

In October 2021, Hatt *et al.* [109] proposed the DONUT (Deep Orthogonal Networks for Unconfounded Treatments) method, which seeks to improve average treatment effect estimation by formalizing unconfoundedness as an orthogonality constraint between outcomes and treatment assignment. This constraint is implemented as a regularization term within the loss function, guiding the model to predict outcomes that are orthogonal to the treatment assignment, thereby improving bias reduction in observational data. The limiting factor of this method might be scenarios with high noise or weak confounding. The authors also avoided specifying the importance of the orthogonality constraint to the ITE estimation.

In April 2020, Hassanpour and Russel [113] proposed DRCFR, an algorithm aimed at enhancing treatment effect estimation from observational data. DRCFR disentangles the factors influencing treatment selection from those driving outcome determination. By first identifying representations of these distinct underlying sources, the algorithm leverages this disentangled knowledge to mitigate the impacts of selection bias. A disadvantage of this methods is that correct disentanglement was not guaranteed.

In February 2022, Wu *et al.* [108] proposed the DerCFR method, which advances treatment effect estimation by disentangling confounders from non-confounders in observational data guaranteeing correct disentanglement. By selectively balancing confounders, the method reduces biases introduced by irrelevant features. The primary disadvantage of this approach is its reliance on the quality of the disentanglement process, as any inaccuracies in separating confounders from non-confounders can negatively impact the results. Summary of the methods is presented in the Table 2.2.

2.2.6 Meta-learners

In March 2019, Künzel *et al.* [93] introduced the S-Learner, T-Learner, and X-Learner frameworks for ITE estimation. These meta-learning algorithms are highly versatile, capable of utilizing any predictive algorithm, including neural networks, as their foundational models. Subsequently, in September 2020, Nie and Wager presented the R-Learner, a meta-learner framework designed to provide a robust approach to ITE estimation. The main strength of meta-learners is their flexibility, allowing any machine learning model as a base learner to fit different datasets. The S-Learner is simple and efficient for small data, the T-Learner captures differences between groups, the X-Learner improves accuracy in uneven datasets, and the R-Learner handles confounding well, especially in observational studies.

However, meta-learners have limitations. The S-Learner struggles when treatment effects vary a lot or groups differ greatly, risking biased results. The T-Learner splits data, which can lead to overfitting if one group is small, and assumes the models are independent, which may not hold. The X-Learner shares the small-sample issues of the T-Learner and relies on tricky weighting that can fail if off. The R-Learner depends on accurate initial steps, and mistakes there can weaken results, plus tuning it can be hard. All meta-learners rely on a good base model, and they are not handling networked data well without extra help. Summary of the methods is presented in the Table 2.4.

Table 2.1. GAN-Based Methods

Method	Key Features	Strengths	Limitations	Year	Framework
GANITE [103]	Generates counterfactual outcomes using a generative adversarial network.	Strong performance in counterfactual generation.	Complex training process with adversarial networks.	2018	Tensorflow 1.15
ABCEI [104]	Incorporates mutual information estimation in adversarial training.	Retains essential predictive information from covariates.	Relies on robust mutual information estimators.	2020	Tensorflow 1.4
CETransformer [105]	Captures complex correlations using self-attention mechanisms.	Scalable to high-dimensional data.	Prone to loss of information critical for outcome prediction.	2021	
CBRE [106]	Uses adversarial training for balanced representations.	Effective in addressing confounding bias.	Prone to loss of outcome-discriminative information.	2022	Tensorflow 1.4

Table 2.2. Covariate-Confounding Learning Methods

Method	Key Features	Strengths	Limitations	Year	Framework
CEVAE [43]	Adapts Variational Autoencoders to causal DAGs; models joint distribution conditioned on latent variables.	Handles hidden confounding; models complex distributions.	Sensitive to hyperparameter selection.	2017	Pytorch
TEDVAE [107]	Disentangles latent confounders into binary and continuous types.	Improved interpretability through structured latent space.	Computational overhead from disentanglement.	2021	PyTorch
DONUT [109]	Enforces orthogonality between outcome and treatment via regularization.	Reduces bias in ATE estimation under uncon-foundedness.	May underperform with high noise or weak confounding.	2021	PyTorch
DRCFR [113]	Disentangles treatment-related and outcome-related latent factors.	Mitigates selection bias through separate representation.	No guarantee of correct disentanglement.	2020	TensorFlow 1.13
DerCFR [108]	Selectively balances confounders after disentangling from non-confounders.	Guarantees correct confounder disentanglement.	Depends on disentanglement quality.	2022	TensorFlow 1.15

Table 2.3. Representation-Based Methods

Method	Key Features	Strengths	Limitations	Year	Framework
BNN [48]	Balances treated/untreated groups using a neural network.	Effective in high-dimensional data.	Limited flexibility in model architecture.	2016	Unavailable
DCN-PD [96]	Disentangled representation learning for treatment estimation.	Reduces confounding effects.	Complex training process.	2017	Pytorch
TARNet [24]	Treatment-agnostic representation network.	Simple architecture, computationally efficient.	Less robust in addressing covariate imbalance.	2017	Tensorflow 0.12
CFR-Wass [24]	Uses Wasserstein distance for balancing distributions.	Reduces covariate imbalance effectively.	Computationally expensive for large datasets.	2017	Tensorflow 0.12
CFR-MMD [24]	Uses Maximum Mean Discrepancy for distribution alignment.	Robust distributional balancing.	May struggle with extreme treatment/control imbalances.	2017	Tensorflow 0.12
SITE [97]	Preserves local similarity while balancing distributions.	Focuses on extreme cases in data.	May underperform with well-overlapping data.	2018	Tensorflow 1.7
DragonNet [99]	Uses a propensity score branch to enhance balancing.	Explicitly addresses selection bias.	Increased model complexity.	2019	Tensorflow 1.13
ACE [98]	Combines similarity preservation with structural fidelity.	Retains original feature space structure.	Computational overhead.	2019	
DKLITE [49]	Minimizes counterfactual variance using kernel-based representations.	Effective for covariate shift handling.	Sensitive to kernel choice and parameter settings.	2020	Tensorflow 1.14
CFR-Weight [101]	Incorporates propensity score weighting in balancing.	Improved balance for imbalanced treatment scenarios.	Sensitive to errors in propensity score estimation.	2022	
DerCFR [108]	Disentangles factors influencing treatment/outcome.	Reduces selection bias.	Sensitive to disentanglement quality.	2022	Tensorflow 1.15
CITE [102]	Contrastive representation learning for treatment effect estimation.	Improves representation learning.	Sensitive to errors in propensity score estimation.	2022	TensorFlow 1.0
BCAUSS [111]	Addresses positivity assumption violations.	Robust under extreme treatment imbalances.	Additional computational cost.	2023	Tensorflow 2

Table 2.4. Meta-Learners

Method	Key Features	Strengths	Limitations	Year
S-Learner [93]	Uses a single model for outcome prediction, combining covariates and treatment.	Simple to implement and interpretable.	Limited adaptability in highly heterogeneous data.	2019
T-Learner [93]	Fits separate models for treated and untreated groups.	Flexibility in estimating treatment-specific outcomes.	May overfit when the sample size is small in either group.	2019
X-Learner [93]	Imputes missing outcomes to refine treatment effect estimates.	Handles treatment effect heterogeneity well.	Requires an additional step for imputing counterfactuals.	2020
R-Learner [110]	Uses Robinson decomposition to decouple treatment and outcome modeling.	Robust handling of covariates affecting both treatment and outcomes.	Requires careful model specification for reliable results.	2020

Chapter 3

Materials and Methods

3.1 Method

In this section, we present our proposed method for estimating ITE using GNNs. We first provide a detailed description of our methodology, including the architectural design and theoretical motivation behind employing GNNs for ITE estimation. Following this, we introduce the datasets used for training and evaluation, highlighting their key characteristics and relevance to our study.

3.1.1 Problem Formulation

Consider a dataset derived from observational studies, denoted as $D = \{[x_i, y_i, t_i]\}_{i=1}^{N}$, where each tuple corresponds to an individual observation. Here, $x_i \in X \in \mathbb{R}^M$ represents a vector of M-dimensional covariates capturing features such as demographic information, clinical measurements, or socioeconomic factors. The binary treatment indicator $t_i \in T \in \{0, 1\}$ denotes whether the i-th individual received the treatment ($t_i = 1$) or was assigned to the control group ($t_i = 0$). The outcome $y_i \in Y$ can be either discrete (e.g., binary indicators like survival or disease remission) or continuous (e.g., blood pressure or test scores), reflecting the response variable of interest. This dataset is typical of real-world observational studies where treatment assignment is not randomized, introducing potential confounding and necessitating causal inference techniques to estimate treatment effects accurately.

Additionally, assume the data generation process is described by a weighted DAG $G = (V, E)$, which provides a structural representation of the relationships among variables. The vertices $V \in \mathbb{R}^M$ correspond to the covariates, treatment, and outcome variables—collectively representing the M features plus t_i and y_i. The directed edges $E \in \mathbb{R}^K$ capture causal or associative dependencies between these variables, with K being the number of

edges. The structure of the DAG is encoded by an adjacency matrix $A \in \mathbb{R}^{M\times M}$, where each element a_{ij} indicates the presence (non-zero) or absence (zero) of a directed edge from node v_i to node v_j. Corresponding edge weights are stored in a weight matrix $W \in \mathbb{R}^{M\times M}$, where w_{ij} quantifies the strength or influence of the relationship if $a_{ij} \neq 0$. Since G is a DAG, A is constrained to be an upper triangular matrix under a suitable ordering of nodes, ensuring acyclicity: for each edge $v_i \to v_j$, $i < j$ [29]. This structure reflects a hierarchical data generation process, where v_i is a parent of v_j if $w_{ij} \neq 0$, implying that v_i directly influences v_j. In practice, G might be partially known from domain expertise or learned from data, and its incorporation aims to model dependencies that affect treatment assignment and outcomes.

The primary objective is to estimate the ITE, specifically the Conditional Average Treatment Effect (CATE), for each individual based on their covariates. The CATE is formally defined as:

$$\tau(x_i) = \mathbb{E}[Y^1 - Y^0 | X = x_i], \tag{3.1}$$

i where Y^1 and Y^0 are the potential outcomes, representing what y_i would be if the i-th individual were treated or untreated, respectively, conditioned on their covariate vector x_i [7]. These potential outcomes are counterfactual in nature: only one is observed for each individual ($y_i = t_i Y^1 + (1 - t_i)Y^0$), and the other must be inferred. To enhance this estimation by accounting for relational dependencies within the data, we extend the definition to incorporate the graph structure via the adjacency matrix A:

$$\tau(x_i) = \mathbb{E}[Y^1 - Y^0 | X = x_i, A]. \tag{3.2}$$

This formulation posits that the treatment effect depends not only on covariates of an individual but also on the network of dependencies encoded in A [27]. For example, if A indicates that certain covariates influence treatment assignment or mediate the outcome, including A provides additional context that may reduce bias and improve estimation accuracy. This is particularly relevant in settings like healthcare systems, where interactions between variables can play a significant role.

To evaluate the performance of a model estimating $\tau(x_i)$, we employ several metrics. First, the Precision in Estimating Heterogeneous Effect (PEHE) quantifies the accuracy of individual-level predictions:

$$\epsilon_{\text{PEHE}} = \frac{1}{N}\sum_{i=1}^{N}(\tau(x_i) - \hat{\tau}(x_i))^2, \tag{3.3}$$

where $\hat{\tau}(x_i)$ is the estimated ITE of the model for an individual i [19]. This metric computes the mean squared error between true and estimated treatment effects, emphasizing the model's ability to capture heterogeneity across individuals. A lower ϵ_{PEHE} indicates better

precision in estimating variable treatment effects, which is critical for personalized decision-making.

Second, we assess the Absolute Error of the Average Treatment Effect (ϵ_{ATE}), which measures the accuracy of the aggregated treatment effect across the population:

$$\epsilon_{\text{ATE}} = \left| \frac{1}{N} \sum_{i=1}^{N} (\tau(x_i) - \hat{\tau}(x_i)) \right|. \tag{3.4}$$

Unlike PEHE, which focuses on individual errors, ϵ_{ATE} evaluates the bias in the average effect, providing a single scalar that reflects how well the model estimates the overall treatment impact. A small ϵ_{ATE} suggests that the model's estimates are unbiased on average, even if individual predictions vary in accuracy.

Third, we compute the policy risk ($\mathcal{R}_{\text{pol}}$), which assesses the expected loss when treatment decisions are made based on the model's ITE predictions. Following [49], it is defined as:

$$\mathcal{R}_{\text{pol}} = 1 - \mathbb{E}[Y^1|\pi(X) = 1]P(\pi(X) = 1) + \mathbb{E}[Y^0|\pi(X) = 0]P(\pi(X) = 0), \tag{3.5}$$

where the treatment policy $\pi(X)$ is:

$$\pi(X) = \begin{cases} 1 & \text{if } Y^1_{\text{RCT}} - Y^0_{\text{RCT}} > 0, \\ 0 & \text{otherwise.} \end{cases}$$

Here, $Y^1_{\text{RCT}} - Y^0_{\text{RCT}}$ represents the true treatment effect from a RCT setting, used as a benchmark to define an optimal policy. $\mathcal{R}_{\text{pol}}$ measures the regret of assigning treatments based on $\hat{\tau}(x_i)$, penalizing incorrect decisions (e.g., treating when the effect is negative). A lower policy risk indicates better alignment between the model's recommendations and optimal outcomes.

Finally, we evaluate the error in the Average Treatment Effect on the Treated (ATT), as outlined by [4]:

$$\epsilon_{\text{ATT}} = \left| \text{ATT} - \frac{1}{|T|} \sum_{i \in T} \tau(x_i) \right|, \tag{3.6}$$

where $\text{ATT} = |T|^{-1} \sum_{i \in T} y_i - |C \cap E|^{-1} \sum_{i \in C \cap E} y_i$ is the true ATT, with T as the treated group, C as the control group, and E as a randomized subset if available. This metric focuses on the treated population, comparing the true average effect among those who received treatment to the estimate of a model, providing insight into accuracy for this subgroup.

Together, these metrics, including PEHE, ϵ_{ATE}, policy risk, and ϵ_{ATT}, offer a comprehensive evaluation framework, balancing individual precision, population-level accuracy, decision-making utility, and subgroup performance.

3.1.2 Graph Neural Networks

GNNs [23] are neural networks designed to process and learn from graph-structured data. They consist of multiple graph convolutional layers [114], which take as input the embeddings of node values, the adjacency matrix A, and, in weighted graphs, the weight matrix W. These components collectively define the structure of the graph and the relationships between nodes. GNNs iteratively update the state of each node by aggregating information from its neighbors, making them powerful tools for tasks involving relational data, such as social networks, molecular structures, or causal graphs [72]. Unlike traditional neural networks that assume independent data points, GNNs leverage the inherent connectivity of graphs, enabling them to capture dependencies and patterns that span across nodes.

GNNs are versatile and can operate on both directed and undirected graphs, adapting their aggregation mechanisms to the topology of the graph [79]. However, in the context of this work, we focus on DAGs, which impose a specific constraint: the absence of cycles ensures a hierarchical structure where information flows in a single direction, from parent nodes to child nodes [69]. Consequently, we assume that update of each node relies solely on information from its parent nodes, reflecting the causal or dependency relationships encoded in the DAG. This restriction aligns with applications like causal inference, where the directionality of influence is critical, and distinguishes GNNs on DAGs from their use on undirected or cyclic graphs, where bidirectional or recursive information flow might occur.

Node Update Mechanism

The forward pass for node updates in GNNs involves three primary steps: message preparation, aggregation, and update, as outlined by Gilmer *et al.* [115]. These steps form the backbone of how GNNs process and propagate information across the graph. In the **message preparation** step, information from parent nodes is refined to create meaningful signals for the target node. This process begins by taking the embedding of each parent node, typically a vector representation capturing its features, and passing it through a learnable function such as a linear transformation or a neural network layer. The output is then multiplied by the corresponding edge weights from the weight matrix W, which modulate the influence of each parent based on the strength or relevance of their connection. For instance, in a causal DAG, a higher weight might indicate a stronger causal link, allowing the model to prioritize more impactful relationships.

Next, in the **aggregation** step, the prepared messages from all parent nodes are combined into a single representation. Common aggregation methods include summation, which adds the weighted messages; averaging, which normalizes their contribution; or selecting the maximum value, which emphasizes the most prominent signal [116]. The choice of ag-

gregation depends on the task. Summation might suit scenarios where cumulative effects matter, while maximum aggregation could highlight dominant influences. For DAGs, this step ensures that the node integrates information only from its upstream dependencies, respecting the acyclic structure and avoiding feedback loops.

In the **update** step, the aggregated messages are merged with the existing representation of the node to produce its new state. This integration can take various forms, such as element-wise multiplication or addition, followed by processing through several fully connected layers to capture complex interactions and refine the embedding of the node [64]. Alternatively, more sophisticated approaches concatenate the message representations and pass them through a Gated Recurrent Unit (GRU), as explored by Chung *et al.* [117], Li *et al.* [118], and Battaglia *et al.* [119]. GRUs introduce a memory-like mechanism, allowing the node to selectively retain or discard information from its parents, which are particularly useful for modeling sequential or hierarchical dependencies in DAGs. Assuming a new node value $h_i \in \mathbb{R}^F$, where F is the dimensionality of the input features, depends on the values calculated in the previous step, the update rule for node embedding d_i can be written as [120]:

$$h_i = \phi\left(d_i, \bigoplus_{j \in \mathcal{N}_i} w_{ij}\psi(d_j)\right), \tag{3.7}$$

where $\mathcal{N}_i$ is the set of parents of node i, ψ and ϕ are learnable functions (e.g., neural networks), w_{ij} is the edge weight between nodes i and j, and $\bigoplus$ denotes the aggregation operation (e.g., sum, mean, or max). This formulation encapsulates the iterative nature of GNNs, where each layer refines node representations by incorporating neighborhood information, tailored to the directed structure of the DAG.

The power of GNNs lies in their ability to learn rich, context-aware representations by iteratively refining node states across multiple layers [22]. In the context of DAGs, this process mirrors the propagation of causal effects, making GNNs well-suited for tasks like causal inference or treatment effect estimation [27]. However, their effectiveness depends on the quality of the input graph structure and the choice of aggregation and update mechanisms, which must be carefully tuned to the specific problem domain.

3.1.3 Graph Attention Networks

Graph Attention Networks (GATs) [64] are a specialized variant of GNNs that leverage self-attention mechanisms to assign varying levels of importance to neighboring nodes. Unlike traditional GNNs that treat all neighbors equally or rely on fixed aggregation rules [72], GATs dynamically weigh the contributions of neighbors based on their relevance to the target node. This adaptability makes GATs particularly effective for tasks where the influence of neighboring nodes varies significantly, such as social network analysis, citation networks, or

biological graphs [23]. GATs operate on a graph $G = (V, E)$, where V is the set of nodes and E is the set of edges, with each node $i \in V$ associated with a feature vector $d_i \in \mathbb{R}^F$, where F represents the dimensionality of the input features. The model updates node representations by iteratively processing these features through graph attentional layers, refining them layer by layer to capture complex relational patterns.

Node Update Mechanism

Each graph attentional layer begins by applying a shared linear transformation to the input features of every node, a step that prepares the data for subsequent attention computations. This transformation is defined by a learnable weight matrix $W \in \mathbb{R}^{F' \times F}$, which projects the original feature vectors from dimension F into a new feature space of dimension F':

$$d'_i = W d_i. \tag{3.8}$$

This projection allows GATs to emphasize or suppress certain aspects of the input features, tailoring them to the task at hand [79]. The next step involves computing the importance of each neighboring node j to a given node i using a shared attention mechanism, inspired by transformer models [63]. The raw attention score e_{ij} quantifies this importance and is calculated as:

$$e_{ij} = \text{LeakyReLU}\big(a^\top [d'_i \| d'_j]\big), \tag{3.9}$$

where $a \in \mathbb{R}^{2F'}$ is a learnable parameter vector, $\|$ denotes concatenation of the transformed feature vectors d'_i and d'_j, and LeakyReLU introduces non-linearity with a small negative slope to retain gradient flow for negative inputs [121]. To incorporate the graph structure, the attention mechanism is masked, meaning attention scores are computed only for $j \in \mathcal{N}_i$, the neighborhood of node i (including i itself in self-attention settings).

These raw attention scores are then normalized across the neighbors using the softmax function to produce attention coefficients:

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k \in \mathcal{N}_i} \exp(e_{ik})}. \tag{3.10}$$

The normalized coefficients α_{ij} represent the relative importance of node j to node i, summing to 1 over the neighborhood $\mathcal{N}_i$ [64]. Using these coefficients, the features of neighboring nodes are aggregated to compute the updated feature vector for node i. To stabilize training and enhance the representational capacity of the model, GATs employ multi-head attention, a technique borrowed from transformers [63]. In this approach, K independent attention mechanisms (heads) are applied, each with its own weight matrix W^k and attention coefficients α^k_{ij}. For intermediate layers, the outputs of these heads are concatenated to

form the final representation:

$$h_i = \|_{k=1}^{K} \sigma \left(\sum_{j \in \mathcal{N}_i} \alpha_{ij}^k W^k d_j \right), \tag{3.11}$$

where σ is a non-linear activation function, such as the Exponential Linear Unit (ELU) [122], α_{ij}^k are the attention coefficients computed by the k-th attention head, and W^k is the corresponding weight matrix. This multi-head mechanism allows GATs to model diverse relationships within the graph, improving both stability and expressiveness [116], though it increases computational complexity for large graphs.

3.1.4 Intervention on GNN and GAT

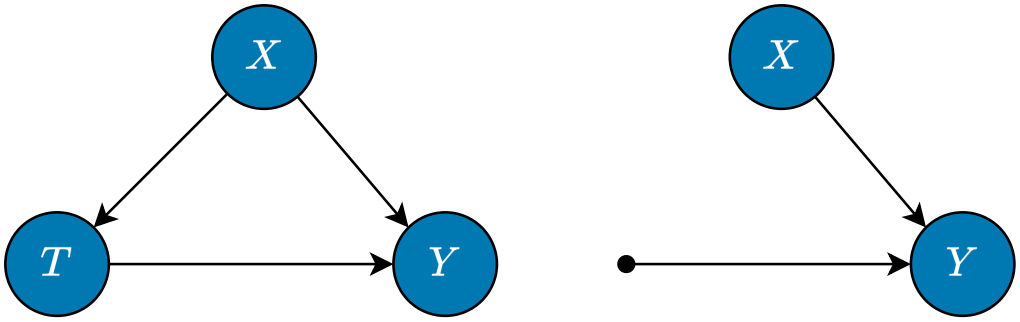


Figure 3.1. Causal graphs before (left) and after (right) intervention on node T [1].

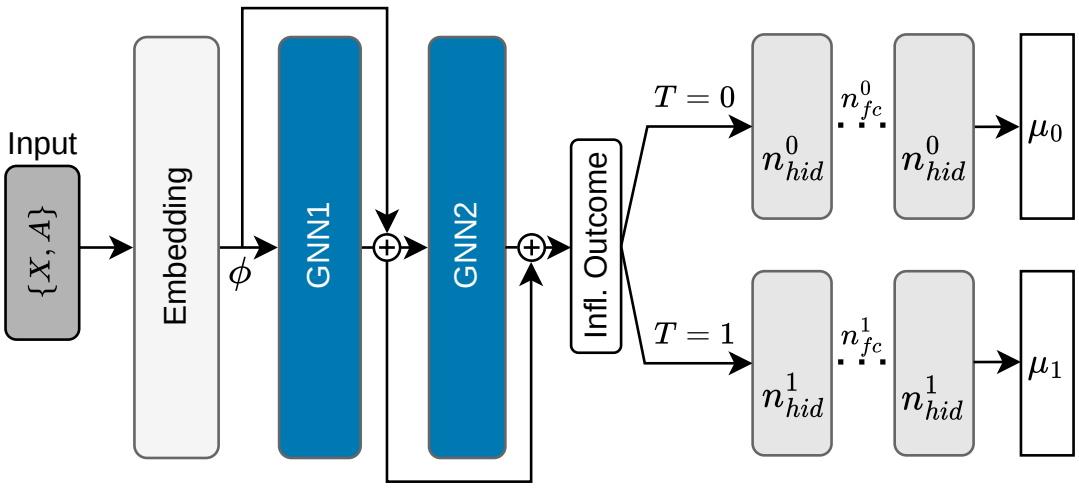


Figure 3.2. The GNN-TARnet architecture [1].

GNNs and GATs are powerful frameworks for modeling relational data, using message-passing mechanisms to capture dependencies within graph structures [23]. Applying these models to causal inference, especially for estimating ITE, requires specific modifications to

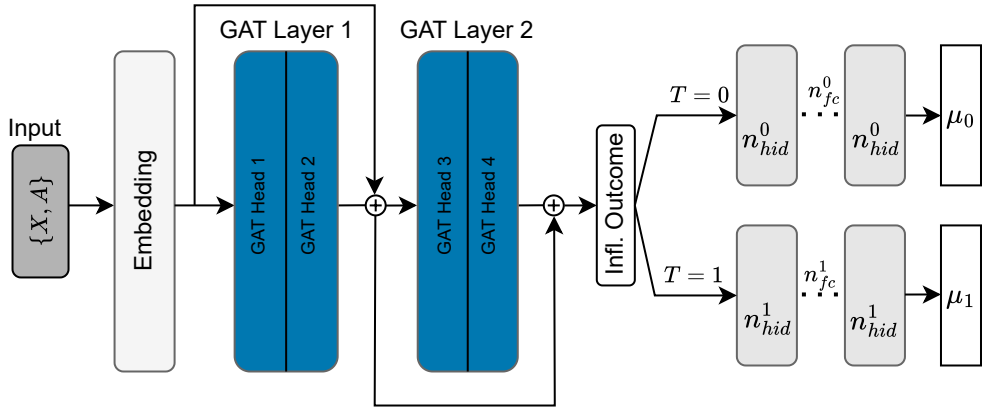


Figure 3.3. The GAT-TARnet architecture with two GAT layers each consisting of two GAT heads.

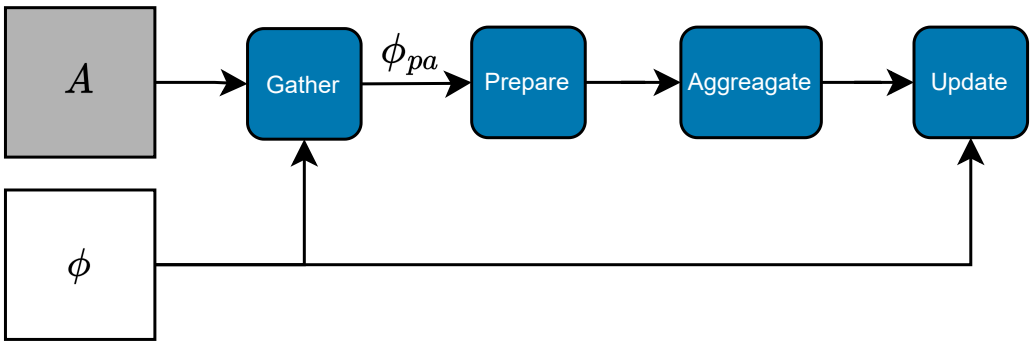


Figure 3.4. GNN block structure [1].

account for interventions, similar to those described in SCMs [29]. In typical GNN architectures, node features h_i are updated through aggregation of information from neighboring nodes. However, causal interventions, such as setting a treatment variable exogenously, demand a different approach: the update rules must reflect the effect of the "do" operator, which breaks the link between a node and its natural causes [73]. Zečević *et al.* [28] introduced a strategy to achieve this by intervening on a node d_k, removing all of its incoming edges. This operation simulates the SCM intervention $do(d_k)$, effectively fixing the value of d_k independent of its parents. Such a modification ensures that downstream effects of the intervened node are propagated without confounding from prior dependencies, which is essential for accurate treatment effect estimation [27].

Under an intervention on node d_k, Zečević *et al.* redefine the neighbor set as

$$\mathcal{M}_i = \{j \mid j \in \mathcal{N}_i,\ j \notin \text{pa}_i \text{ if } j = k\}, \tag{3.12}$$

where pa_i denotes the set of parents of node i. When $i = k$ (the intervened node), the

set becomes $\mathcal{M}_k = \emptyset$, effectively removing all incoming edges. The node representation is then set to $h_k = \phi(d_k, 0)$, reflecting the isolation induced by the intervention. For any downstream node ($i \neq k$), $\mathcal{M}_i$ excludes the original parents of d_k if $d_k \in \mathcal{N}_i$, ensuring that information propagates without confounding. The modified message-passing update rule is defined as:

$$h_i = \phi\left(d_i, \bigoplus_{j \in \mathcal{M}_i} w_{ij}\psi(d_j)\right), \tag{3.13}$$

where $\mathcal{M}_i$ is dynamically adjusted based on the intervention target. This approach enforces causal consistency in the data-generating process represented by the graph, aligning the GNN framework with the principles of SCMs [69].

For GATs, interventions adapt the multi-head attention mechanism, which weighs neighbor contributions dynamically [64]. Under intervention on d_k, the neighbor set shifts to $\mathcal{M}_i$, excluding d_k's parents: if $i = k$, $\mathcal{M}_k = \emptyset$, and $h_k = \|_{k=1}^{K}\sigma(0)$; otherwise, $\mathcal{M}_i$ omits d_k's confounded inputs. The intervened rule is:

$$h_i = \|_{k=1}^{K}\sigma\left(\sum_{j \in \mathcal{M}_i} \alpha_{ij}^{k} W^k d_j\right). \tag{3.14}$$

This formulation preserves the ability of the attention mechanism in GATs to prioritize relevant neighbors while enforcing the semantics of causal interventions, providing greater flexibility than fixed-weight graph neural network architectures [23].

The intervention mechanism operationalizes the $do(d_k)$ operator introduced by Pearl [29], which is critical for estimating individual treatment effects. This is achieved by constructing a graph that includes covariates and outcomes, and by severing all incoming edges to the treatment node T (see Fig. 3.1). This assumption implies that T directly influences the outcome Y without mediation by its original parents (such as confounders), aligning with the potential outcomes framework proposed by Rubin [7]. For example, in a covariate graph defined by $X \to T \to Y$ and a confounder structure $Z \to T$ and $Z \to Y$, performing an intervention on T removes the edge $Z \to T$, thereby isolating the effect of T on Y. Figure 3.1 illustrates the structure of the graph before and after intervention, highlighting edge removal and the propagation of unconfounded effects.

In practice, these intervention-based rules allow graph-based models to simulate counterfactual "what-if" scenarios. This functionality underpins the methods introduced in Section 3.1.5, where the proposed architectures based on graph neural networks and graph attention mechanisms estimate treatment effects. By redefining the neighbor set $\mathcal{M}_i$ under intervention, these models achieve consistency with causal DAGs [69], enabling isolation of the direct effect of a treatment on an outcome while blocking confounding paths. This approach mirrors the counterfactual reasoning used in SCMs, where the intervention $do(T = t)$ induces a modified model to predict Y under hypothetical treatment values [29].

3.1.5 GNN-TARnet and GAT-TARnet

With the above assumptions, we introduce methods called GNN-TARnet (Graph Neural Network Treatment Agnostic Representation Network) and GAT-TARnet (Graph Attention Treatment Agnostic Representation Network) which integrate information about causal relationships among covariates to predict ITE. These methods follow a design similar to TARnet, proposed by Shalit *et al.* [24]. We chose TARnet as the foundation for our method due to its simplicity and robust performance compared to other models on existing datasets [3]. Notably, we did not base our approach on architectures that reduce distributional discrepancy between treated and untreated groups, such as CFR-Wass [24], as we aimed to isolate the effect of incorporating DAGs into ITE estimation without introducing additional complexity. Although our approach could be extended to include discrepancy reduction techniques, this is beyond the scope of this thesis.

The primary distinction of GNN-TARnet and GAT-TARnet from TARnet lies in the computation of the hidden representation prior to branching (see Fig. 3.2). In GNN-TARnet, we employ graph convolutional layers [114] instead of fully connected layers, and in GAT-TARnet we use the graph attention layers consisting of several concatenated graph attention heads (see Fig. 3.3).

To train the networks, we minimize the following loss function:

$$\mathcal{L} = \mathbb{E}\left[(1-T)(\mu_0(X,A)-Y)^2 + T(\mu_1(X,A)-Y)^2\right]. \tag{3.15}$$

The loss function $\mathcal{L}$ represents the expected value $\mathbb{E}$, calculated as the average squared differences between the observed outcome Y and the estimated outcomes μ_0 and μ_1 under control and treatment conditions, respectively. These differences are weighted by the probabilities of not receiving treatment, $(1-T)$, and receiving treatment, T. The network takes as input the covariates X and the adjacency matrix A. Since edge weights W for GNN-TARnet are rarely available, we set them to a default value of one. Initially, the covariates, which also serve as node values in the graph, are passed through embedding layers. In this step, the original covariate information is combined with the trained neural network weights and transformed to match the embedding dimension required by the GNN or GAT layers. Each covariate is assigned an independent embedding, achieved by reshaping the covariates to add an extra dimension before embedding. The resulting embeddings and edges are then processed through a GNN block (see Fig. 3.4) or through the GAT layers.

The GNN block extracts node indices and identifies causal parents from the adjacency matrix. Parent representations ϕ_{pa} are gathered from the node representations using these parent indices. In the next step, parent messages are prepared by passing the parent representations through fully connected layers without bias [115]. Messages from the parent nodes are then aggregated [115]; in our case, they are summed for each node. Finally, the

node representations are updated with the aggregated messages from the parents by adding or multiplying them and then passing them through fully connected layers [119]. Each GAT layer consists of multiple GAT heads [64] that can be stacked together. Within each GAT head, the input features are first linearly transformed using a learnable weight matrix. Attention scores are then computed that capture the importance of features from neighboring nodes. These attention scores are used to perform a weighted aggregation of the neighboring node features. Additionally, we incorporated skip connections [123] after each GNN block and GAT layers, as these, in our observations, can improve the performance of the models.

Following these steps, we obtain a vector representing the updated node embeddings. If the adjacency matrix between X and Y is known, we condition on the nodes that directly influence Y, flatten the hidden representations, and pass them through treatment-specific branches. In cases where only an identity graph is available, we assume an equally weighted influence of all nodes on the outcome variable and condition on all available nodes. Importantly, interventions based on the do-operator are applied only during inference. During training, the model uses the covariates X and their graph structure A, without including the treatment T in the graph. At inference time, we simulate counterfactual outcomes by exogenously setting T and blocking any upstream influences, in line with the semantics of $do(T = t)$. This allows the model to estimate potential outcomes in a way that reflects structural causal assumptions. Finally, the loss is computed, and the weights of the entire system are updated through backpropagation.

3.1.6 Software Library for Causal Inference

To operationalize our methodology for ITE estimation using GNNs, as detailed in the previous sections, we developed a custom Python library encapsulated in the `CausalModel` class. This library serves as a flexible and extensible framework for training, evaluating, and analyzing causal inference models across diverse datasets, with a particular emphasis on supporting our GNN-TARnet and GAT-TARnet architectures [1]. Below, we outline its core functionalities, design principles, and its role in facilitating the experiments presented in Chapter 4.

The `CausalModel` class is initialized with a parameter dictionary specifying the dataset (e.g., IHDP, JOBS, SUM), model type, number of trials, and outcome type (binary or continuous). To ensure reproducibility, a critical aspect of causal inference research, it includes a static method, `setSeed`, which configures deterministic behavior across TensorFlow, NumPy, and the random number generators of Python by setting seeds and enforcing single-threaded execution.

The library features robust data handling capabilities, implemented through special-

ized loaders (e.g., `load_ihdp_data`, `load_jobs_data`, `load_sum_data`). These methods retrieve datasets from predefined file paths, preprocess features using standardization via `StandardScaler` from the scikit-learn library, and structure the data into dictionaries containing covariates (X), treatments (t), observed outcomes (y), and potential outcomes (μ_0, μ_1). For instance, the SUM dataset loader supports variable training sizes, enabling ablation studies on sample size effects (Section 3.2.2), while the GNN-specific loader accommodates folder-based data structures reflecting varying edge counts. This preprocessing ensures consistency across datasets, addressing challenges like scale differences and missing values inherent in observational data [56].

The training and evaluation pipeline of the library is centered around the method called `train_and_evaluate`, designed to be overridden by model-specific subclasses, which computes performance metrics such as PEHE and ATE error. Dataset-specific evaluation methods iterate over trials or folders, storing results in NumPy arrays or CSV files for persistence and subsequent analysis. Beyond standard metrics, the library supports advanced evaluation through policy risk and cumulative gain analysis. The `find_policy_risk` method computes policy value and risk. The `define_tuner` method integrates with Keras Tuner [124] to optimize hyperparameters using strategies like Hyperband or Random Search, as detailed in Section 3.3. In summary, the `CausalModel` library provides a comprehensive toolkit for causal inference, tailored to our GNN-based ITE estimation framework. Its modular design supports experimentation across synthetic (e.g., SUM) and real-world (e.g., IHDP, PerPain) datasets, ensuring robust and reproducible analysis. The outputs, including the library's metrics and plots, inform the results and discussions in Chapter 4, demonstrating its pivotal role in this dissertation.

3.2 Datasets

Our primary objective is to assess the performance of our GNN-based ITE estimation strategy, specifically GNN-TARnet and GAT-TARnet (Section 3.1.5), across a variety of scenarios and datasets, testing its robustness and efficacy under diverse conditions. This section outlines the datasets employed and the methodology for applying our approach, spanning theoretical datasets with explicit feature relationships and real-world datasets with predefined covariate interactions. We investigate two central hypotheses: (1) that a GNN-based model, leveraging a DAG to encode structural causal relationships, performs comparably to state-of-the-art ITE estimation methods (e.g., TARnet [24]) on datasets with known or inferred causal structures; and (2) that incorporating such structural information enhances ITE estimation accuracy in real-world applications, particularly where traditional methods falter due the low amount of data, as exemplified by the PerPain consortium data [10].

A core challenge in validating causal inference models is the unobservability of counterfactual outcomes ($Y_i(1)$ and $Y_i(0)$) in real-world data, as only one outcome is observed per individual [7]. To address this, we evaluate our algorithms on publicly available benchmark datasets: the Infant Health and Development Program (IHDP) [19,24,125] and JOBS [126], detailed in Section 3.2.1. However, these datasets lack accompanying causal graphs, limiting their ability to test structural modeling directly. To fill this gap and rigorously assess our first hypothesis, we designed an artificial dataset, SUMmation (SUM), with a layered DAG structure where covariate relationships are explicitly defined (Section 3.2.2). SUM computes covariate values as sums of parent nodes and generates potential outcomes under controlled conditions, mimicking real-world causal dynamics for precise evaluation. Additionally, we apply our method to real data from the PerPain consortium (Section 3.2.3), targeting our second hypothesis by refining treatment assignments through ITE estimation, leveraging inferred causal structures to enhance personalization in chronic pain management [10].

3.2.1 Existing Datasets

In this section we describe dataset often used in the literature for comparison of the methods. These are often semi-artificial or real datasets with available potential outcomes.

IHDP

The IHDP dataset stems from a randomized controlled trial under the Infant Health and Development Program [125], comprising 747 preterm infants with very low birth weights. It includes 25 covariates, such as parental demographics (e.g., maternal age, education), socioeconomic factors (e.g., income), and infant characteristics (e.g., birth weight, gestational age). The treatment variable denotes participation in an intensive childcare program, involving regular home visits by healthcare professionals over three years, aimed at improving cognitive development. Counterfactual outcomes, in this case cognitive test scores, are synthetically generated using probabilistic models, following Hill's methodology [19]. In "Setting A" (IHDP_A), outcomes are linear functions of covariates, e.g.,

$$Y(t) = \beta_0 + \beta_1 X_1 + \cdots + \beta_{25} X_{25} + \gamma t + \epsilon, \tag{3.16}$$

with $\epsilon \sim N(0, 1)$. In "Setting B" (IHDP_B), the control outcome ($Y(0)$) incorporates nonlinearity via an exponential function, e.g., $Y(0) = \exp(\beta X) + \epsilon$, while $Y(1)$ remains linear, introducing complexity in treatment effect heterogeneity. Both settings feature a class imbalance, with approximately 18% of samples (135 instances) in the treatment group. We adopt 100 simulated IHDP datasets per setting, each split into training (60%), validation

(20%), and test (20%) sets during hyperparameter tuning, deviating from Shalit *et al.*'s 63/27/10 split [24]. Post-tuning, we retrain on the full training set (80%) to optimize test performance, a modification our experiments show improves generalization. We also scale the outcome variable with the standard scaler as we noticed that it can improve performance of the models. For the dataset we report the mean and standard error of $\sqrt{\epsilon_{\text{PEHE}}}$ and ϵ_{ATE} for the training as well as the test sets.

JOBS

The JOBS dataset, introduced by Dehejia and Wahba [126], combines data from the National Supported Work Program RCT and an observational study, totaling 3,212 instances. It features 8 covariates, including demographic details (e.g., age, race) and financial metrics (e.g., 1974–1975 earnings), with treatment defined as participation in a professional job training program and employment status as the outcome. The RCT subset (approximately 480 treated instances) provides a "ground truth" for causal effects, while the observational component introduces real-world confounding. With only 10% of samples (321 instances) in the treatment group, JOBS exhibits significant imbalance, challenging model robustness. Following Shalit *et al.* [24], we generate 100 random train/test splits (80/20) to mitigate this imbalance and ensure reliable performance assessment, averaging results across splits to reduce variance in ITE estimates. For the dataset we report the policy risk $\mathcal{R}_{\text{pol}}$ and ϵ_{ATT}.

To uncover the causal structure of IHDP and JOBS, we apply causal discovery methods, highlighted in section 2.1.12, namely LiNGAM, GES, and PC. To find the edges between covariates X and the observed outcomes Y we combine them into a dataset $\{X, Y\}$. We designate Y as a sink node (no outgoing edges), reflecting its role as an effect rather than a cause, and run the algorithms to generate an adjacency matrix A. Non-zero elements in A define directed edges, stored as tuples (parent, child), with edges to Y removed to enforce its sink status. Indices of nodes influencing Y are recorded separately for conditioning in GNN-TARnet/GAT-TARnet. If discovery fails (e.g., due to insufficient sample size or independence), we default to an *identity* graph ($A = I$), assuming no inter-covariate relationships, testing our method's robustness to minimal structural information [1].

3.2.2 Artificial Dataset

The SUMmation (SUM) dataset is a synthetic dataset engineered to assess our GNN-based ITE estimation strategy, specifically GNN-TARnet and GAT-TARnet, by leveraging a predefined DAG to test our hypothesis: that structural modeling enhances performance comparable to state-of-the-art methods (e.g., TARnet [24]). Its graph structure is inspired by Bayesian networks in the *bnlearn* repository [2], which feature layered architectures with

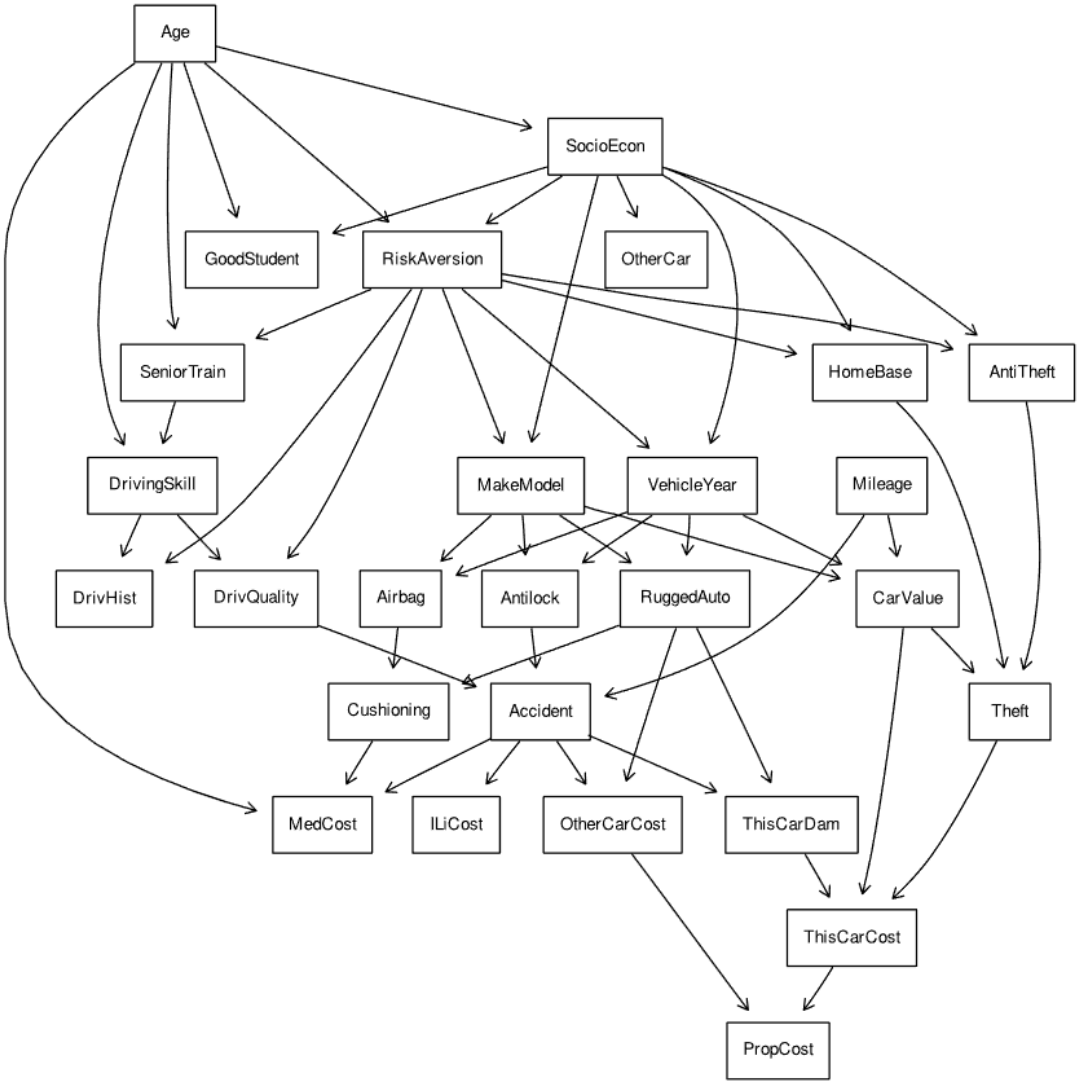


Figure 3.5. An example of the Bayesian model "Insurance" [2].

no intra-layer connections. In these networks, nodes in one layer serve as the sole parents of nodes in the next, ensuring a unidirectional causal flow. Typically, the number of root nodes (first layer) equals or exceeds the child nodes per root in subsequent layers, and the layer preceding the output (leaf) nodes is substantially smaller, often comprising 20-30% of the total node count, thereby concentrating the flow of causal influence through a smaller subset of nodes. The *bnlearn* graphs exhibit average node degrees of 2 in smaller networks (e.g., 10-20 nodes) and approximately 3 in larger ones (e.g., 50-100 nodes), with a maximum in-degree of 13, as seen in complex models like "insurance" (Fig. 3.5) [127]. SUM adopts these characteristics, constructing a layered DAG to simulate realistic causal relationships under controlled conditions. The dataset is built in two phases: generating the graph structure and assigning node values, detailed in the following subsubsections.

Creating Layered Graph

Suppose we are given a number of layers l. To create the first layer of the graph G, we generate a range of length $r + 1$, which contains consecutive integer values from zero to r. Thus, $r + 1$ defines the total number of root nodes. We then store this array as a value in a dictionary L, where the layer number serves as the key. Each subsequent dictionary entry stores ranges of length k, starting with the values immediately following the last element of the array in the previous layer. For this dataset, k is a random number between three and eight, except in the final layer, where the number of nodes m is set to 0.3 times the total number of nodes in all previous layers combined.

The nodes in each new layer serve as potential children of the parent nodes in the previous layer. The pseudocode for generating the dictionary L, which stores the graph layers, is provided in Algorithm 1. To complete the construction of graph G, we create edges between individual parent nodes and k children nodes, where k is uniformly selected from a range of zero to the previously defined k. Finally, all nodes in the output layer are connected to every node in the preceding layer, ensuring that each internal graph node has both a parent and a child.

The scalar m was selected so that the number of output nodes is always greater than two but significantly less than the total number of nodes in previous layers. We set r as a random value between ten and 20 to ensure that the number of root nodes always exceeds the number of k nodes in the next layer. The parameter k was chosen to keep the average node degree within a range similar to that in the *bnlearn* repository. By selecting parameters k, r, and m in this manner, we achieve graphs with structural properties resembling those in the *bnlearn* repository.

Generating Node Values

After constructing the graph, we generated node values as follows. Values for nodes in the first layer were sampled uniformly at random between 0 and 1. For nodes in subsequent layers, values were computed as the sum of their parent nodes. The values of nodes in the final layer influenced the outcome generation. To make the data resemble observational studies, we assigned treatment based on outcome node values. First, we calculated the average value of all outcome nodes across all subjects. For each subject, if the mean of its outcome nodes exceeded this overall average, the treatment was set to one; otherwise, it was set to zero. Formally, for the k-th subject x_k in a dataset with N subjects and a set O of all node indices influencing the outcome node o, treatment t_k is assigned as:

Algorithm 1 Generate Layers

Given: number of layers l
$m \leftarrow 0.3$
Initialize an empty dictionary L
$r \leftarrow$ random number between 10 and 20
$L[0] \leftarrow$ range of length $r + 1$ from 0 to r
for $i \leftarrow 1$ to l **do**
    if $i = l$ **then**
        $k \leftarrow \lfloor m \times (\sum_{j=0}^{l-1} \text{length}(L[j])) \rfloor$
    else
        $k \leftarrow$ random number between 3 and 8
    end if
    $L[i] \leftarrow$ range of length k starting from the last element of $L[i-1] + 1$
end for

$$t_k = \begin{cases} 1, & \text{if } \frac{1}{n} \sum_{j \in O} x_{k,j} > \frac{1}{n \cdot N} \sum_{i=1}^{N} \sum_{j \in O} x_{i,j} \\ 0, & \text{otherwise,} \end{cases} \tag{3.17}$$

where $O = \{j \mid j \in \text{pa}_o,\ j = 1, \ldots, M\}$. The outcomes were generated as the sum of the output node values in the case of treatment and the mean of the output node values in the case of no treatment. Figure 3.6 shows an example of a graph from the SUM dataset with two layers.

The SUM dataset was used to examine the importance of using a causal graph versus an identity graph. We ran experiments with training sets containing 16, 32, 64, and 128 subjects, with up to four graph layers. These values were chosen to demonstrate model performance across different scenarios and complexities. Increasing the numbers beyond these values does not significantly change overall model performance. The test set was fixed and contained 120 data points. Additionally, we report the performance of TARnet on this dataset.

Node Masking

By design, GNN-TARnet relies on nodes that influence the outcomes to make predictions, where these nodes correspond to the embedding values of input variables processed through GNN layers. Theoretically, if we know all nodes influencing the outcome and no other nodes influence them, training can be significantly simplified. To assess the accuracy of causal discovery methods in identifying such influential nodes in existing datasets, as well as to investigate any underlying structure, we mask the values of covariates identified as

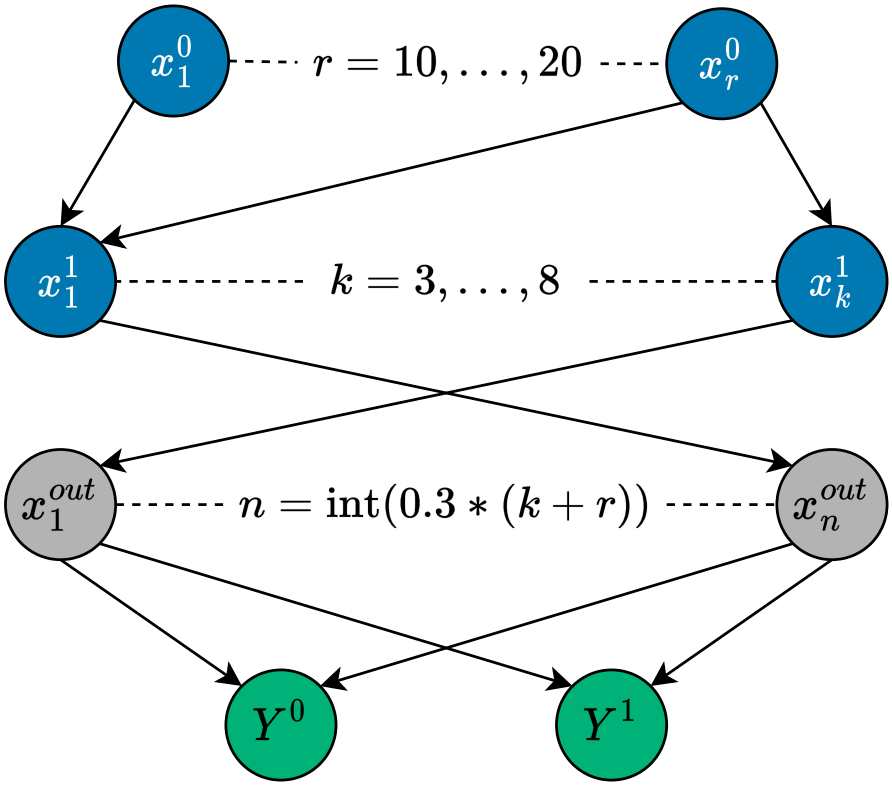


Figure 3.6. DAG for SUM dataset with two layers. Nodes in the output layer are marked as gray, potential outcomes as green, and the nodes of layers zero and one are blue. Not all edges and nodes are presented [1].

influencing the outcome by replacing them with zeros.

Our assumption is that if the causal discovery method correctly identifies all nodes influencing the outcome, and no additional nodes influence the outcome, the prediction result should be equivalent to replacing all values in the dataset with zeros. We report the performance of GNN-TARnet (LiNGAM) on datasets where either all covariates are masked with zeros or only the covariates influencing the outcome are replaced by zeros.

In the SUM dataset, we mask all covariate values corresponding to nodes in the last layer of graph G by replacing them with zeros, while preserving the original graph structure. Note that masked output nodes are not counted as a layer; this increases the difficulty of inference, as otherwise, one could simply train a model on variables influencing the outcome and disregard all other variables to obtain optimal predictions.

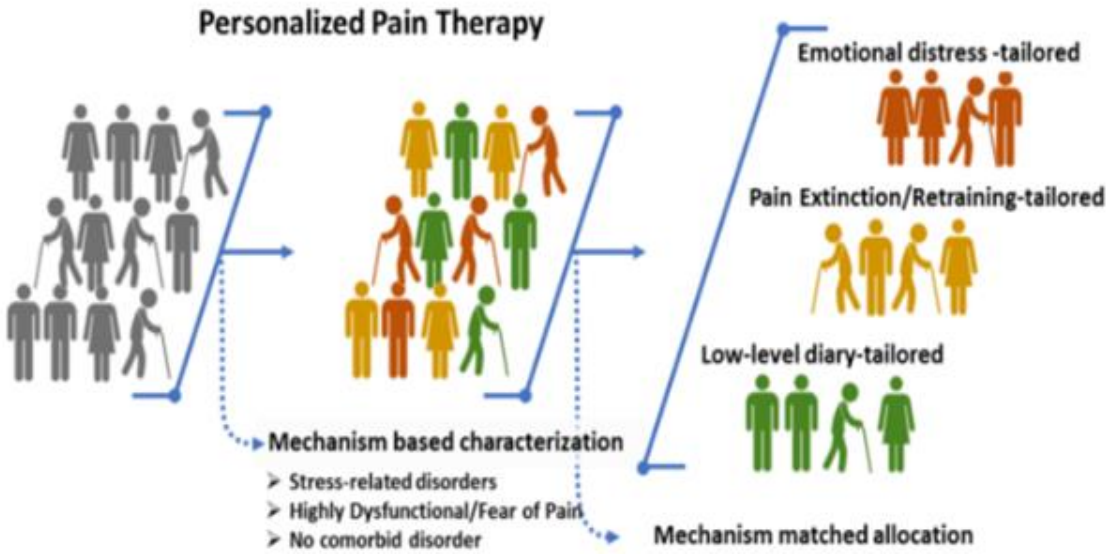


Figure 3.7. Personalized Pain Therapy aims to treat patients based on their individual characteristics.

3.2.3 Disclosed Dataset

Overview of the PerPain Trial

Chronic musculoskeletal pain (CMSP) frequently persists despite adequate medical treatment of underlying musculoskeletal disorders, largely due to the presence of psychological comorbidities and complex psychobiological mechanisms. These factors can independently maintain pain beyond its initial physical cause, significantly impairing overall well-being and quality of life of a patient. Recognizing this multifaceted nature, the PerPain consortium was established with the primary goal of enhancing therapeutic outcomes for CMSP patients by identifying distinct subgroups characterized by specific pain-maintenance mechanisms and subsequently tailoring individualized therapeutic interventions accordingly.

At the beginning of the PerPain trial, patients with CMSP underwent an extensive initial screening process based on predefined eligibility criteria. Patients who qualified proceeded to a comprehensive baseline assessment involving multiple standardized questionnaires designed to capture critical psychological, behavioral, and functional dimensions related to chronic pain. After data collection, rigorous preprocessing was performed, including careful identification and removal of statistical outliers to ensure high-quality data. Subsequently, baseline scores for each patient were computed as either the sum or average of questionnaire responses. The computed baseline outcome measure for the i-th questionnaire is denoted as Y_{base_i}.

Following baseline assessments, eligible participants were randomized into one of two study arms: a personalized treatment allocation group, where assignments were guided by

a decision-making algorithm integrating unsupervised clustering with XGBoost classification, and a random allocation group, which served as a control condition. In the random allocation group, algorithm recommendations were deliberately ignored, and treatments were instead randomly assigned. This approach allowed rigorous evaluation of the effectiveness and added value of personalized therapeutic assignments. After several weeks of intervention, patients underwent a follow-up assessment, yielding end-of-treatment outcome data that corresponded directly to baseline measures. These end-of-treatment scores for the i-th questionnaire are represented as Y_{EoT_i}. The comparison between personalized and random allocations provided essential insights into the benefits of tailored interventions for CMSP.

The following subsections offer detailed descriptions of the interventions and explain the allocation algorithms used to guide personalized treatment assignments.

PerPain Trial Interventions

The PerPain Trial involved three tailored therapeutic programs, each specifically adapted to patient profiles identified during baseline assessment. These interventions targeted essential psychological and behavioral dimensions implicated in chronic musculoskeletal pain, differing clearly in their approach, intensity, and mode of delivery. Comprehensive intervention descriptions, therapist training protocols, and session contents have been thoroughly documented in prior literature [10]. Participants received one of three treatments based on their identified characteristics: Emotional Distress Tailored Therapy (EDTT), Pain Extinction and Retraining Therapy (PERT), or Ecological Momentary Diary Intervention (EMDI).

EDTT provided individualized face-to-face psychotherapy designed for participants experiencing significant emotional distress or trauma-related symptoms. EDTT integrated core principles from Eye Movement Desensitization and Reprocessing (EMDR) along with trauma-focused cognitive-behavioral therapy (CBT) methods, bilateral sensory stimulation (e.g., guided eye movements), and dual-attention tasks [128]. Its primary objective was facilitating the processing of distressing memories, reducing emotional reactivity, and alleviating emotional pain. Therapy sessions, guided by a structured manual, lasted approximately 100 minutes each, delivered weekly over 12 weeks. The effectiveness of EDTT in reducing emotional distress and improving pain-related outcomes has been validated by previous studies [129, 130].

PERT was developed as a group-based intervention specifically targeting patients exhibiting dysfunctional pain behaviors or maladaptive coping responses. Based on CBT principles, PERT aimed to retrain maladaptive behaviors, encourage greater activity engagement, and foster improved social interactions. Structured behavioral exercises, role-playing, and video feedback were integral components of each session. Each therapy group consisted of approximately five to six patients, with sessions held weekly for 100 minutes

over 12 weeks. Additionally, two individual sessions were provided, and spouse or partner involvement was actively encouraged. Prior evidence has demonstrated the efficacy of PERT in improving patient outcomes and influencing brain activation patterns among highly dysfunctional patients [131].

EMDI utilized a smartphone-based platform specifically for patients with lower psychological distress and higher functional abilities. Participants engaged in daily logging of positive activities using a dedicated smartphone application, supported by frequent notifications to encourage sustained engagement. This intervention aimed to shift patient attention away from pain and toward meaningful, rewarding daily activities, thereby promoting behavioral activation and emotional self-regulation. Diary entries and responses to smartphone prompts were collected daily throughout the 12-week intervention period. Previous research has supported the efficacy of EMDI in reducing pain and stress levels while enhancing mood and overall engagement in meaningful activities [132].

All three interventions were rigorously standardized to ensure consistency and comparability across treatment groups. Therapists delivering EDTT and PERT were trained psychologists who underwent specialized training, and adherence to manualized treatment protocols was closely supervised. This careful matching of interventions to patient characteristics was central to achieving optimal therapeutic effectiveness and generating actionable insights for future personalized pain management strategies.

Allocation Algorithm

For personalized treatment allocation within the randomized controlled trial, we selected key outcome variables derived from the West Haven-Yale Multidimensional Pain Inventory (MPI-D) questionnaire [133], following the methodology described by Rudy and Turk [134]. In addition to the psychological and behavioral constructs captured by the MPI-D, demographic covariates, specifically age and gender, were also included. The selected variables represent key dimensions of pain experience and psychosocial functioning: Pain Severity (PS), Interference (I), Life Control (LC), Affective Distress (AD), Social Support (S), Punishing Response (PR), Solicitous Response (SR), Distracting Response (DR), and General Activity (GA). Each of these variables is measured on a numeric scale ranging from 0 to 6, representing the intensity or frequency of the respective trait. Age was recorded in years, while gender was encoded as a binary variable (0: male, 1: female).

To perform clustering, we used a previously available datasets provided by the consortium partners, which included the MPI variables along with demographic data. Subjects with missing data were excluded from the analysis. The final combined dataset consisted of 461 participants: 189 men (average age: 49 years) and 272 women (average age: 51 years). Before clustering, all variables were standardized using Z-score normalization, and

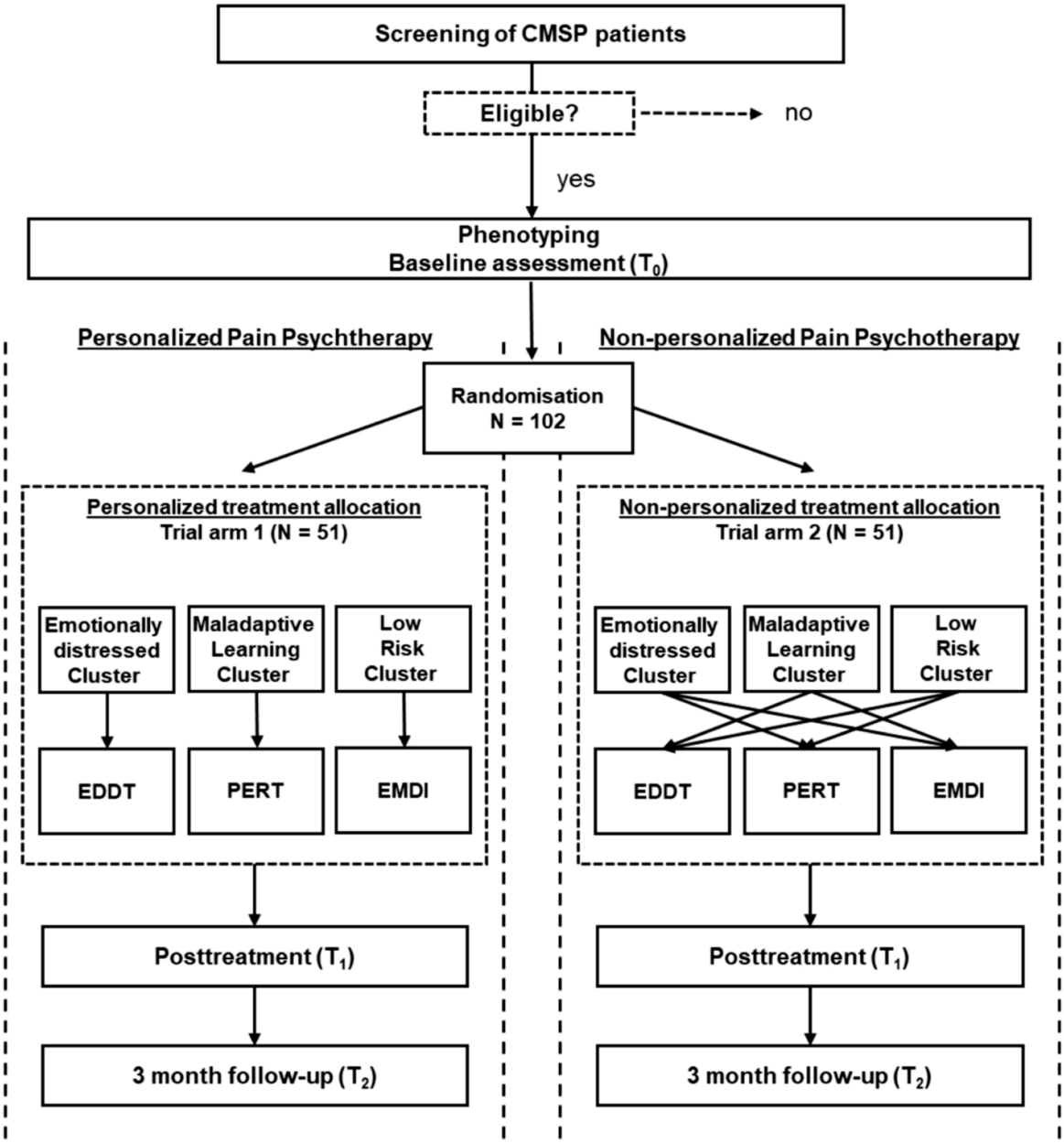


Figure 3.8. Description of the PerPain trial.

the data were randomly permuted to mitigate ordering effects. Clustering was carried out using the *kmeans* algorithm in R, with the number of clusters set to three, consistent with the structure proposed in earlier literature [134]. The clustering procedure was initialized with 16 random centroids. To facilitate interpretation and comparison with previous studies, we computed standardized T-scores (mean = 50, standard deviation = 10) for each variable after clustering.

The analysis yielded three distinct patient profiles: *Dysfunctional*, characterized by high Pain Severity and low General Activity; *Distressed*, marked by high Affective Distress and low Social Support; and *Copers*, defined by low Pain Severity, high Life Control, and high General Activity (see Figure 3.9).

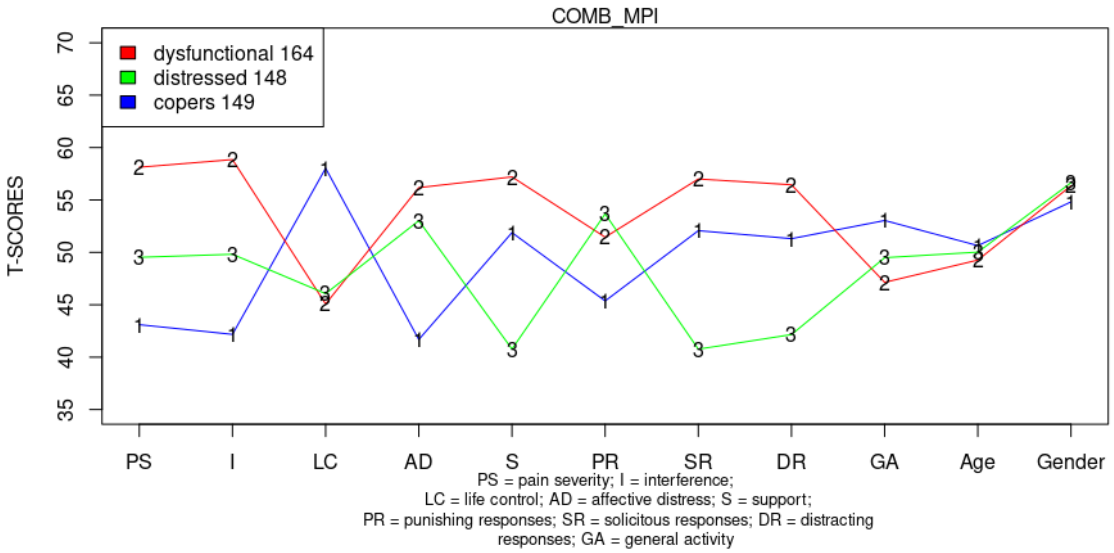


Figure 3.9. Mean T-scores for each cluster and variable.

To assign new patients to one of the identified clusters, we trained a predictive model using Gradient Boosted Decision Trees, implemented via the `xgbTree` method from the `caret` package in R. The model was optimized using 16-fold cross-validation and a grid search for hyperparameter tuning. To evaluate performance, we randomly split the combined dataset into training (75%) and test (25%) subsets. On the test set, the classifier achieved an accuracy of 97%. Following this evaluation, the model was retrained on the full dataset and saved for future use.

To classify a new case, the model requires the outcomes from the MPI-D questionnaire along with age and gender as inputs. It then returns the predicted cluster (i.e., treatment group) along with the associated likelihood score. For interpretability, we also provide a visual comparison of the current subject's profile with the average profile of each cluster from the combined dataset (see Figure 3.10).

New Treatment Assignment using ITE Estimators

Upon completion of the trial, evaluation results indicated no statistically significant difference in Pain Severity (PS), the primary outcome measure, between patients treated according to the personalized algorithm and those assigned randomly (3.19 ± 1.01 for personalized allocation versus 3.26 ± 0.93 for random allocation). This absence of significant findings may primarily be attributed to the relatively small sample size available for the final analysis; initially, 105 participants were enrolled and randomized, but due to attrition during the trial period, only 87 patients remained and provided end-of-treatment (EoT) data.

Given these outcomes, we explored a crucial research question: could alternative treat-

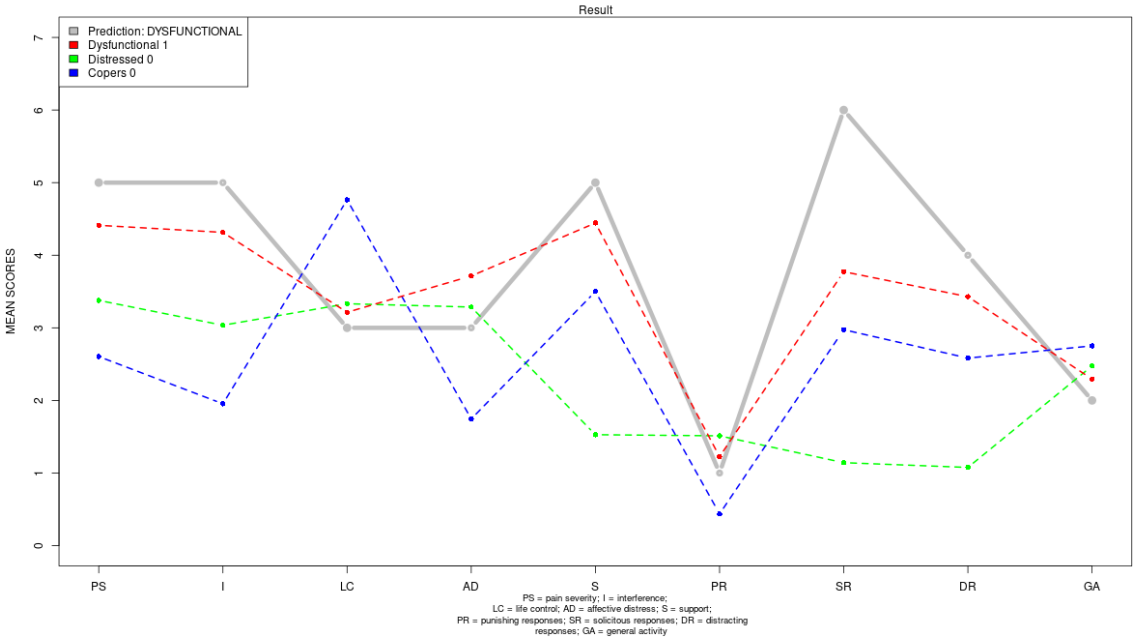


Figure 3.10. Comparison of prediction with a mean subject from each category.

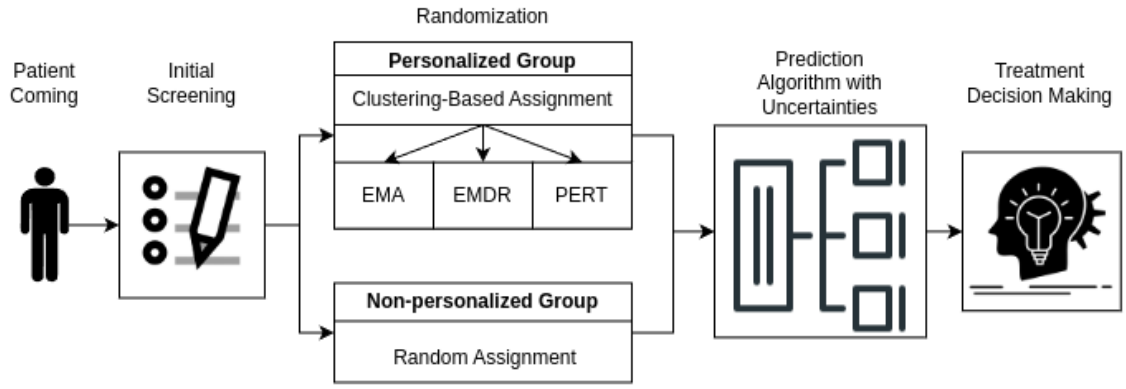


Figure 3.11. How the new treatment assigned to RCT participants based on the trial results.

ment assignments informed by ITE estimation methods have yielded improved therapeutic results compared to the original allocation strategies? To investigate this, we conducted a post-hoc analysis employing ITE estimation techniques. Specifically, we treated the RCT data as observational, allowing us to estimate counterfactual outcomes for individual patients.

Treatment assignments were reassessed based on predicted EoT outcomes, $Y_{EoT_{pred}}$, compared to baseline scores, Y_{base}. The individual treatment effect was explicitly defined as follows:

$$ITE = Y_{EoT_{pred}} - Y_{base}. \tag{3.18}$$

Subsequently, each patient was hypothetically reassigned to the treatment predicted to produce the greatest individual benefit according to these ITE estimations. Figure 3.11 illustrates the complete pipeline from initial patient screening through to the newly generated treatment assignments.

In the following subsections, we detail the methodological procedures used, including the identification and preparation of relevant input variables and the construction of causal graphs essential for implementing advanced graph-based neural network methods, namely GNN-TARnet and GAT-TARnet. Additionally, we included traditional ITE estimation methods, specifically TARnet [24] and T-Learner [93], for comparative analysis. Given that our data originate from a randomized controlled trial, treatment groups were already balanced by design. Thus, more advanced balancing methods such as CFRNet [24] (e.g., using Wasserstein distances to achieve distributional balance) were unnecessary in this context.

Variable Selection

Each questionnaire completed by participants yields a final score, which is calculated as an aggregate of the individual item responses. These aggregate scores, also referred to as composite scores, were used to assess their association with the primary outcome variable, Pain Severity (PS) at the EoT, as measured by the MPI-D questionnaire. Since not all questionnaire scores exhibited equal levels of correlation with the outcome, a variable selection step was necessary to identify the most informative features.

We propose a score-level variable selection approach, in which all individual questionnaire items (subscores) are retained if the corresponding overall score is found to be significantly correlated with the outcome. To quantify the relationship between questionnaire scores and Pain Severity, we computed the Pearson correlation coefficient [135] for each score. The five scores with the highest absolute correlation to the outcome were selected for further analysis.

To promote model interpretability and reduce the risk of overfitting, we excluded any questionnaire scores that were based on more than six individual items. This threshold was chosen as a compromise between preserving predictive power and maintaining a parsimonious feature set. Following this filtering step, we identified the individual questionnaire items (subscores) that contributed to the selected scores. These subscores formed the input feature set for our network.

The final selected variables corresponded to the following subscores:

- *Intensity*, from the Chronic Pain Rating Scale [136],
- *Sexual Abuse*, from the Childhood Trauma Questionnaire (CTQ) [137],
- *Perceived Helplessness*, from the Perceived Stress Scale (PSS) [138],

- *Behavioral*, from the Somatic Symptom Disorder 12 (SSD-12) [139],
- *Belief in Illness*, from the Whiteley-7 scale [140].

In total, the resulting dataset comprised 87 patients and 21 selected features. To ensure robustness, we performed 200 random splits of the data into training and validation sets using different random seeds. For each split, we reported the average MSE on both the training and validation sets.

After identifying the best-performing model, we combined the training and validation sets (66 patients total: 45 in training, 21 in validation) and retrained the model on this combined data. The final evaluation of the treatment assignment strategy was then conducted on the held-out test set, consisting of 21 previously unseen patients.

Extending Models to Handle Multiple Treatments

Let $x_i \in \mathbb{R}^M$ denote the vector of input features for patient i, and let $y_i \in \mathbb{R}$ represent the factual outcome, specifically the end-of-treatment Pain Severity score Y_{EoT}. The treatment assignment is denoted as $T_i \in \{0, 1, 2\}$, corresponding to the three possible interventions: EDTT ($t = 0$), EMDI ($t = 1$), and PERT ($t = 2$). Using the potential outcomes framework [56], and given a dataset $D = \{(x_i, y_i, T_i)\}_{i=1}^N$, our goal is to estimate the potential outcomes $\tilde{y}_{i,0}$, $\tilde{y}_{i,1}$, and $\tilde{y}_{i,2}$ for each patient under all three treatments, including the unobserved counterfactuals. We then assign a new treatment $\tilde{T}_i$ to each patient based on the largest expected gain relative to their baseline value $Y_{\text{base},i}$:

$$\tilde{T}_i = \arg\max_{t \in \{0,1,2\}} \left(\tilde{y}_{i,t} - Y_{\text{base},i}\right). \tag{3.19}$$

Because the data originate from an RCT, we can estimate the conditional expectation of potential outcomes $\tilde{\mu}_t(x_i) = \mathbb{E}[\tilde{y}_{i,t} \mid x_i]$ using supervised learning techniques without requiring distribution balancing. We evaluate prediction quality using the MSE, defined as:

$$\epsilon_{\text{error}} = \frac{1}{3} \sum_{t=0}^{2} \frac{1}{N_t} \sum_{i=1}^{N} w_t(T_i) \left(\tilde{\mu}_t(x_i) - y_i\right)^2, \tag{3.20}$$

where $N_t = \sum_{i=1}^N w_t(T_i)$ is the number of patients who received treatment t, and $w_t(T_i)$ is a binary weighting function (defined below) that selects only patients who received treatment t. Additionally, we apply a two-sample Kolmogorov-Smirnov test [141] to verify that the distributions of predicted and factual outcomes are statistically similar, and to ensure that the new treatment assignment strategy yields different allocation patterns compared to the original randomization.

Uncertainty Computation

Following the approach of Durso-Finley *et al.* [142], we incorporate uncertainty estimation into TARnet, GNN-TARnet, and T-Learner by modifying these models to predict both the mean and variance of a Gaussian distribution for each potential outcome of the treatment. We assume the potential outcome under treatment t is distributed as:

$$\tilde{y}_{i,t} \sim \mathcal{N}(\tilde{\mu}_t(x_i), \tilde{\sigma}_t^2(x_i)). \tag{3.21}$$

The output layer of each model is modified to produce two values per treatment: the predicted mean $\tilde{\mu}_t(x_i)$ and standard deviation $\tilde{\sigma}_t(x_i)$. For a single sample (x_i, y_i) under treatment $T_i = t$, the negative log-likelihood of the observed outcome is:

$$\mathcal{L}_i = -\log p(y_i \mid x_i) = \frac{1}{2}\log(2\pi\tilde{\sigma}_t^2(x_i)) + \frac{(y_i - \tilde{\mu}_t(x_i))^2}{2\tilde{\sigma}_t^2(x_i)}. \tag{3.22}$$

Averaging over all samples, the negative log-likelihood becomes:

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}\mathcal{L}_i. \tag{3.23}$$

To enhance robustness, we combine this probabilistic loss with an MSE loss term $\epsilon_{\text{error},i} = (y_i - \tilde{\mu}_{T_i}(x_i))^2$, resulting in a hybrid loss:

$$L = \frac{1}{N}\sum_{i=1}^{N}\left[\alpha\mathcal{L}_i + (1-\alpha)\epsilon_{\text{error},i}\right], \tag{3.24}$$

where $\alpha \in [0, 1]$ controls the trade-off between uncertainty calibration and point prediction accuracy. To generalize this to the multi-treatment setting ($T_i \in \{0, 1, 2\}$), we define the multi-treatment hybrid loss as:

$$\mathcal{L}_{\text{multi}} = \frac{1}{N}\sum_{t=0}^{2}\sum_{i=1}^{N} w_t(T_i)\left[\alpha\mathcal{L}_{t,i} + (1-\alpha)\epsilon_{t,i}\right], \tag{3.25}$$

where:

- $\mathcal{L}_{t,i}$ is the negative log-likelihood of the outcome for treatment t and patient i,
- $\epsilon_{t,i} = (y_i - \mu_t(x_i))^2$ is the squared error,
- $w_t(T_i) = \delta_{T_i,t}$ is a binary indicator for treatment membership.

For implementation convenience and continuity with prior sections, we define $w_t(T_i)$ using the equivalent polynomial forms:

$$w_0(T_i) = 0.5T_i^2 - 1.5T_i + 1, \tag{3.26}$$

$$w_1(T_i) = -T_i^2 + 2T_i, \tag{3.27}$$

$$w_2(T_i) = 0.5T_i^2 - 0.5T_i. \tag{3.28}$$

This formulation ensures that only the factual treatment contributes to the loss for each sample, while the model still learns parameters for all three treatment arms. This approach allows the network to learn accurate and calibrated estimates of both factual and counterfactual outcomes with quantified uncertainty.

Our implementation, developed in TensorFlow, supports efficient scaling and extends binary-treatment architectures to multi-treatment scenarios. Updated architectures of T-Learner, TARnet, and GNN-TARnet with built-in uncertainty modeling are visualized in Figures 3.12-3.14.

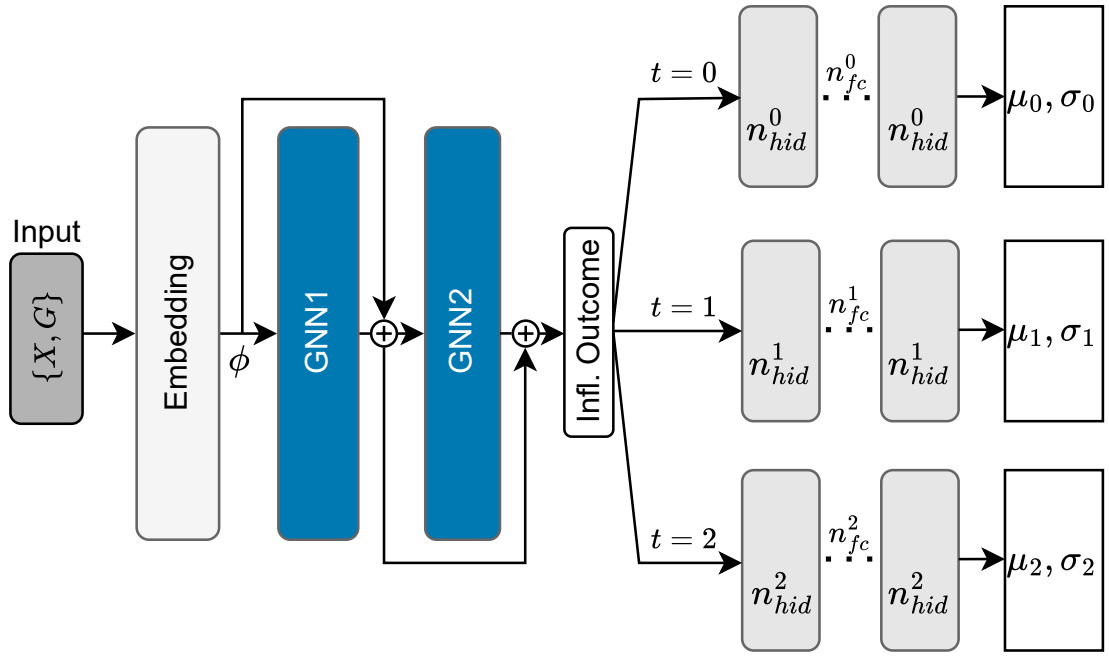


Figure 3.12. The GNN-TARnet architecture with multiple treatments and Uncertainty.

Causal Graph Generation

GNN-TARnet or GAT-TARnet require deterministic causal graph for best performance. Data from PerPain RCT is a compilation of multiple questionnaires that collect information about different aspects of the health of the patient. Each questionnaire has a final score. We propose to create a graph G based on the answers that make up the final score of the questionnaires. First, we concatenate the input features corresponding to the answers to baseline questionnaires with the baseline outcome scores of the same questionnaires. Such outcomes

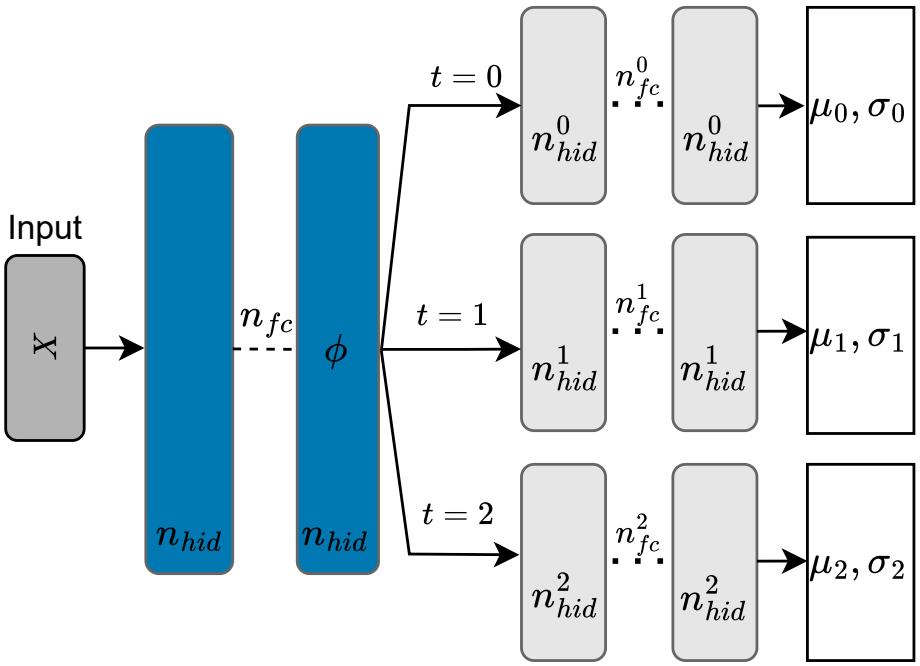


Figure 3.13. The TARnet architecture with multiple treatments and Uncertainy.

are manually computed as summations or averagings of the questionnaire responses, which can be error-prone. To avoid using manually computed outcomes, we replace the values of the baseline scores with zeros. We assume that GNN-TARnet and GAT-TARnet will be able to impute the values. After that, we create an adjacency matrix and store its elements as an array of tuples. The first element of the tuple is the index of an input feature, and the second element is the index of the corresponding baseline score. The tuple corresponds to the parent-child relationship in a causal graph. Figure 3.15 shows an example of a causal graph for the case of $M = 4$ replies to answers coming from two different questionnaires $Q = 2$. We call the nodes containing the answers to the baseline questions the input nodes, and the direct children of the nodes the hidden nodes. We call the nodes hidden because they correspond to zero-valued features and only updated via the GNN blocks during the training.

3.3 Hyperparameter Optimization

This section introduces hyperparameter optimization techniques to determine optimal network configurations for our proposed GNN-TARnet and GAT-TARnet architectures (Section 3.1.5), ensuring robust ITE estimation across diverse datasets (Section 3.2). For a comprehensive comparison with existing methods, we also apply these techniques to thir-

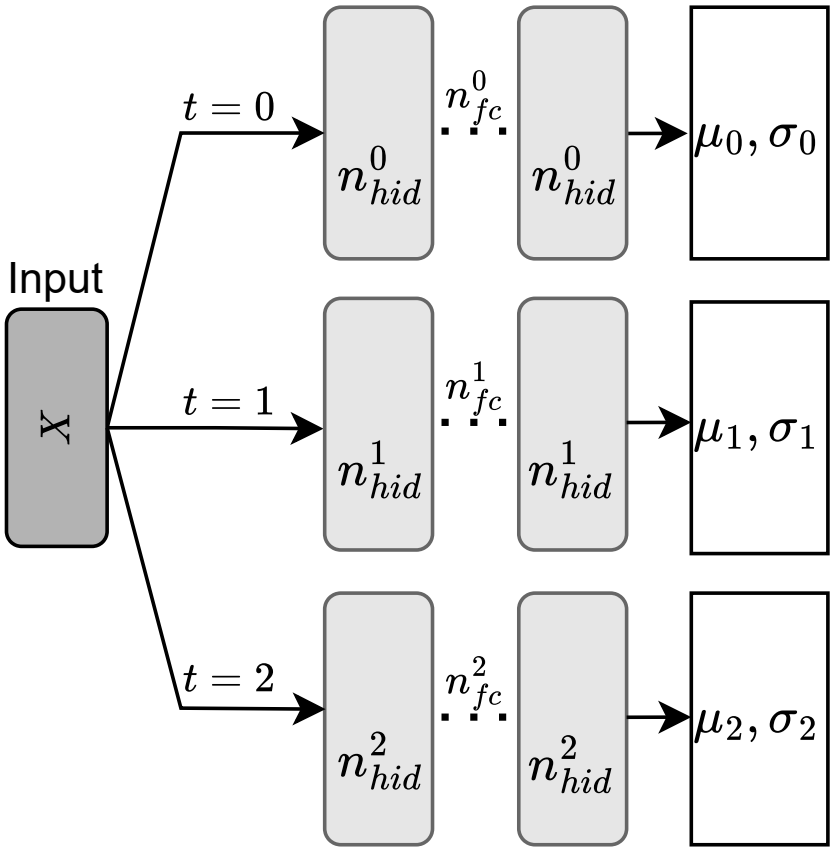


Figure 3.14. The TLearner architecture with three treatments and the Uncertainty.

teen established ITE estimation approaches published before 2022: S-Learner, T-Learner, R-Learner, X-Learner [93], TARNet [24], CEVAE [43], TEDVAE [107], Dragonnet [99], DKLITE [49], GANITE [103], CFR-Weight, CFR-Wass, and CFR-MMDSQ [24] .We focus on these methods due to several strategic considerations. First, they represent foundational paradigms in causal inference, spanning meta-learners, representation-based models, and generative approaches, which are widely validated as benchmarks in the field [3]. Second, their pre-2022 publication ensures a stable, well-documented baseline, avoiding bias toward newest, less-tested innovations. This enables a fair evaluation of whether our GNN-based methods offer genuine advancements or if optimized legacy models can achieve comparable performance. Finally, optimizing these older methods highlights the potential of modern tuning techniques to revitalize established frameworks, providing a transparent and equitable benchmarking framework for GNN-TARnet and GAT-TARnet [1].

Hyperparameter optimization involves training multiple model instances with varied pa-

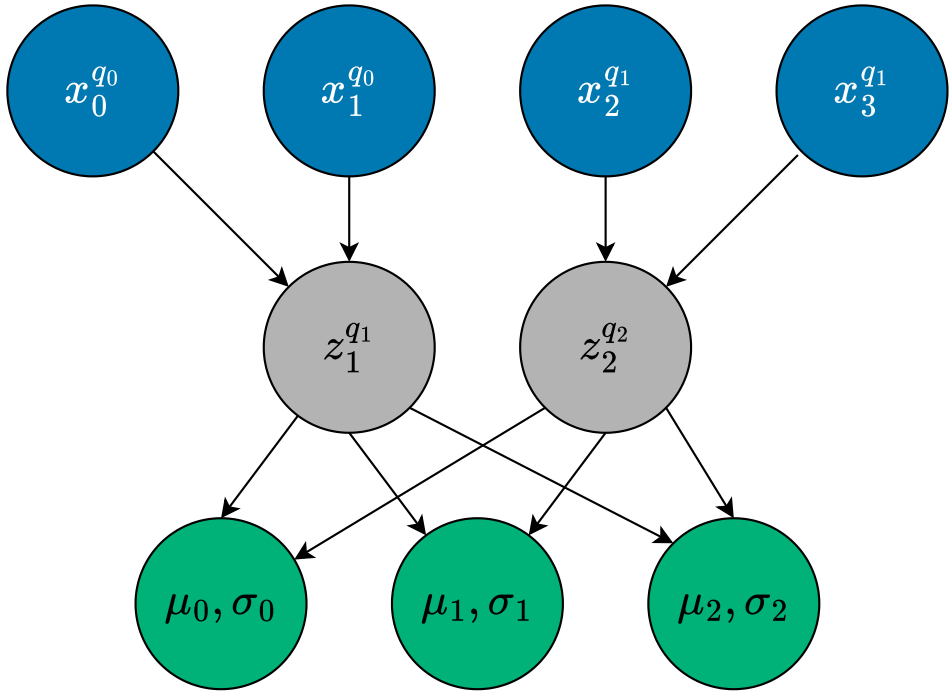


Figure 3.15. An example graph for a dataset consisting of four covariates from two different questionnaires (blue), two hidden nodes representing the questionnaire scores (grey) and three outcome nodes (green).

rameter configurations, evaluating their performance on a validation set, and selecting the configuration with the lowest validation loss, typically MSE for ITE tasks. Numerous optimization strategies exist [143]; we leverage the Keras Tuner library [124] for its flexibility and efficiency, employing three techniques: Random Search, Hyperband [144], and Bayesian Optimization. These methods balance exploration (searching diverse configurations) and exploitation (refining promising ones), optimizing hyperparameters like learning rate, layer depth, and node embedding size for our GNN-based models, while adapting to the architectural specifics of the thirteen baselines.

Random Search

Random Search evaluates a fixed number of hyperparameter configurations sampled randomly from a predefined search space. Users first specify ranges for each parameter , such as learning rate $\in [10^{-4}, 10^{-2}]$, number of fully connected layers $\in [1, 10]$ for TARnet, or hidden dimensionality for treatment specific branches $\in [16, 512]$ for TARnet. The tuner then randomly selects a configuration, trains the model (e.g., for 50 epochs), and assesses validation loss. This process iterates until a budget (e.g., 10 trials) is exhausted, identifying the configuration with the best performance. The strength of Random Search lies in its unbi-

ased exploration, avoiding assumptions about parameter interactions, making it suitable for diverse architectures like S-Learner or CFR-MMDSQ. However, its efficiency diminishes in high-dimensional spaces, as it lacks guided refinement [145].

Hyperband

Hyperband integrates Random Search with early stopping to enhance resource efficiency. It begins by training a large pool of randomly sampled configurations (e.g., 200) for a minimal epoch count (e.g., 5). After evaluating validation loss, it retains the top-performing fraction (e.g., top 50%), training these for additional epochs (e.g., 10), and repeats this halving process until a few configurations (e.g., 1-5) complete full training (e.g., 50 epochs). The best hyperparameters are those of the final top performer. The main advantage of the Hyperband is its ability to discard underperforming models early, exploring more configurations within a fixed computational budget than Random Search, though it assumes early performance predicts final outcomes, which can be a potential limitation for slow-converging models.

Bayesian Optimization

Bayesian Optimization constructs a probabilistic surrogate model, typically a Gaussian Process, of the hyperparameter-performance relationship, guiding the search toward high-performing regions. It starts with a prior distribution over parameters (e.g., uniform for learning rate, discrete for layer count) and evaluates an initial set of configurations (e.g., 10 trials). Using Bayes' rule, it updates the posterior based on validation loss, balancing exploration (untested regions) and exploitation (promising areas) via an acquisition function (e.g., Expected Improvement). New configurations are sampled from the posterior, and the cycle repeats (e.g., 50 iterations) until convergence. Bayesian Optimization can excel in efficiency over Random Search by leveraging prior evaluations, though it requires careful prior specification and can struggle with discrete or high-dimensional spaces [146].

In subsequent subsections, we detail the application of these techniques to thirteen baselines, providing insights into search spaces, optimization outcomes, and comparative performance, illuminating the role of tuning in advancing ITE estimation across methodologies.

Integration with ITE Estimation Methods

S-Learner

S-Learner or single learner uses a single estimation function $\mu(\cdot)$. In our case we use a MLP with n_{fc} consecutive fully-connected layers each having n_{hid} parameters, where n_{fc} is the number of layers and n_{hid} is the number of hidden units in the MLP. As input it takes covariates X concatenated with an observed treatment T. It learns to predict the factual

outcomes, thus we are using a MSE between the predicted values $\bar{Y} = \mu(X, T)$ and the real outcomes as the loss functions:

$$\mathcal{L}_S = \mathbb{E}[(\bar{Y} - Y)^2], \tag{3.29}$$

where $\mathbb{E}$ means averaging over the all subjects in the batch. During inference, the causal effect is calculated as $\tau(x_i) = \mu(x_i, 1) - \mu(x_i, 0)$, where the treatment T is set to 1 or 0, respectively to all subjects. For S-Learner we optimize the number of hidden units n_{hid} and fully connected layers n_{fc} while keeping the batch size bs and learning rate lr constant.

T-Learner

T-Learner estimates treatment response surfaces using MLPs for each unique treatment value. In the binary case, two causal estimators, $\mu_0(x_i) = \mathbb{E}[Y^0|X = x_i, t_i = 0]$ and $\mu_1(x_i) = \mathbb{E}[Y^1|X = x_i, t_i = 1]$, are trained on the covariates and outcomes of subjects receiving the corresponding treatment [93]. The loss function for the T-Learner is defined as:

$$\mathcal{L}_T = \mathbb{E}[(1 - T)(\mu_0(X) - Y))^2 + T(\mu_1(X) - Y)^2]. \tag{3.30}$$

For T-Learner we tune each estimator separately. We are searching for number of hidden units n^0_{hid}, n^1_{hid}, number of fully connected layers in the treated and untreated branches n^0_{fc}, n^1_{fc}, as well as batch sized bs_0, bs_1 and learning rates lr^0, lr^1 independently from each other.

X-Learner

Like the T-Learner, the X-Learner [93] also estimates $\mu_0(x_i)$ and $\mu_1(x_i)$ using MLPs. After that, the imputed treatment effects are computed as: $D^0_i := \mu_1(x^0_i) - y^0_i$ and $D^1_i := y^1_i - \mu_0(x^1_i)$, where x^1_i, x^0_i, and y^1_i, y^0_i are the observed covariates and outcomes for the treated and untreated, respectively. Next, one estimates $\tau_j(x_i) = \mathbb{E}[D^j|X = x_i]$ with $j \in 0, 1$ using again MLPs in our case because of dependence on the factual outcomes. Finally, the CATE is inferred from

$$\tau(X) = g(X)\tau_0(X) + (1 - g(X))\tau_1(X), \tag{3.31}$$

where the propensity score $g(X) = \mathbb{E}[T|X]$ is the probability of receiving treatment T for a given set of covariates X.

For X-Learner we search the number of hidden units $n^{\mu_0}_{hid}$ and $n^{\mu_1}_{hid}$, layers $n^{\mu_0}_{fc}$ and $n^{\mu_1}_{fc}$, and batch sizes bs_{μ_0} and bs_{μ_1} for the response functions μ_0 and μ_1. The number of hidden units, layers, and batch size for the imputed treatment effects d_0 and d_1, and, finally, the number of hidden units n^g_{hid}, layers n^g_{fc}, and batch size bs_g for the propensity score network g.

R-Learner

The R-Learner [110] also learns to estimate $\mu(x_i)$ and $g(x_i)$ with MLPs. The main difference is that the CATE estimator $\tau(x_i)$ is additionally trained using neural networks. The loss is originated from Robinson [147] decomposition and is given by:

$$\mathcal{L}_R = \mathbb{E}\big[\big(\big(Y - \mu(X)\big) - \big(T - g(X)\big)\tau(X)\big)^2\big]. \tag{3.32}$$

The Robinson decomposition reformulates the original problem of estimating the treatment effect as a regression task by removing the influence of covariates on both the treatment and the outcome [148]. For R-Learner we are looking for batch sizes bs^{μ}, bs^{g}, and bs^{τ}, number of fully connected layers n^{μ}_{fc}, n^{g}_{fc}, and n^{τ}_{fc}, and finally number of hidden units n^{μ}_{hid} , n^{g}_{hid}, and n^{τ}_{hid} for each of the networks μ, g, and τ.

Counterfactual Regression

The Counterfactual Regression is a name given by Shalit *et al.* [24] to developed by them representation-based algorithms. As mentioned earlier, the CFR algorithms work by changing the representation of the initial data. In case of the CFR data is transformed into a latent space using MLP, after that an algorithm tries to reduce the distribution shift between the treated and the untreated patients. This is done using distance-based IPM. The IPM can either be a Wasserstein distance, MMDSQ, or no metric at all. The corresponding methods are denoted as CFR-Wass, CFR-MMDSQ, and TARent. After the latent space the information is passed through the treatment specific branches represented by MLPs. In case of a binary treatment the number of MLPs is equal to two. The loss of the general CFR network is presented below:

$$\mathcal{L}_{\text{CFR}} = \mathcal{L}_T + \alpha\,\text{IPM}(\phi(X|T=0), \phi(X|T=1)), \tag{3.33}$$

where $\alpha > 0$ is a regularization term that balances the group distributions. The first part aligns with the loss of the T-Learner (3.30), where $\mu_0(\cdot)$ and $\mu_1(\cdot)$ represent the outcomes of the treatment-specific branches. The second part is an additional loss term that minimizes a distance-based IPM between the two latent distributions $\phi(X|T=0)$ and $\phi(X|T=1)$. A variant of CFR with $\alpha = 0$ corresponds to TAR-Net. For TAR-Net we tune the number of hidden units in layers in the representation network n_{fc}, n_{hid} as well as, unlike the original paper, the number of hidden units in layers in the treatment specific branches n^0_{fc}, n^0_{hid}, and n^1_{fc}, n^1_{hid}. We also tune the dimensionality of the representation space n^{out}_{hid} before branching to see its influence on the results. In case of CFR-Wass and CFR-MMDSQ we are looking for the same parameters as in TAR-Net.

DragonNet

In DragonNet [99], the hidden representation is balanced using the propensity score [13], $g(X)$, which is computed through an additional branch. This approach is designed to minimize error propagation into the outcomes. A weighted Cross-Entropy (CE) loss with a hyperparameter $\alpha > 0$, between $g(X)$ and the actual treatment assignment T, is added to the objective loss $\mathcal{L}_T$ to form the DragonNet model loss $\mathcal{L}_{\text{DN}}$:

$$\mathcal{L}_{\text{DN}} = \mathcal{L}_T + \alpha \, \text{CE}(g(X), T). \tag{3.34}$$

In the case of DragonNet, in addition to the parameters of the TAR-Net, we also optimize the parameters of the propensity score branch, specifically the number of fully connected layers n^g_{fc} and the number of hidden units h^g_{hid}.

Weighted CFR

CFR-Weight [101] uses the MMDSQ metric to balance the distributions of treated and untreated patients, weighted with the propensity score estimated similarly to DragonNet. According to the authors, the weighting mechanism enhances reliability when treatments are assigned with significant imbalance. The algorithm of this method is similar to CFR, initially transforming the covariates into a latent space. Following this, the propensity score $g(X)$ is computed. The treatment-specific representations of the data are then multiplied by their corresponding propensity scores. Subsequently, the MMDSQ distance between the distributions of treated and untreated samples is calculated, with their equality enforced during model training. For CFR-Weight, we optimize the same parameters as in the case of DragonNet.

DKLITE

At first, DKLITE [49] transforms the input via MLP ϕ into a latent space z. After that the resulting vector is passed through a kernel function. Next, the mean and variance of the hidden space distribution are computed and used to calculate the variance and likelihood losses, $\mathcal{L}_{\text{var}}$ and $\mathcal{L}_{\text{like}}$, respectively. A reconstruction loss, $\mathcal{L}_{\text{rec}}$, is computed as the mean squared error (MSE) between the input data X and the network output ϕ^{-1}. The final loss function is expressed as:

$$\mathcal{L}_{\text{DKLITE}} = \mathcal{L}_{\text{like}} + \alpha_1 \, \mathcal{L}_{\text{var}} + \alpha_2 \, \mathcal{L}_{\text{rec}}, \tag{3.35}$$

where $\alpha_1 > 0$ and $\alpha_2 > 0$ are hyperparameters. For DKLITE we are looking for the number of fully-connected layers and hidden units for decoder and encoder: n^{enc}_{fc}, n^{dec}_{fc}, n^{enc}_{hid}, n^{dec}_{hid}. The dimensionality of latent space n^z_{hid} was set to 80 as we found this value to be the best.

GANITE

GANs [112] use competing networks to achieve the desired results in classification or regression by coupling their loss objectives. Typically GANs consist of two networks trained at the same time in a competitive manner by coupling their loss objectives. The first network is called a generator and the second one is a discriminator. The goal of the generator is to create samples similar to the ones from the target distribution. The discriminator learns to distinguish between the generated and the real samples. As the generator improves, the discriminator has more problems in distinguishing the real and generated samples. This principle was used by Yoon *et al.* [103] to generate counterfactual outcomes. Their method called Generative Adversarial Nets for Inference of Individualized Treatment Effects (GANITE) consists of two blocks. In the first block, the generator $G_{CF}(X, Y, T)$ imputes missing counterfactual outcomes $\tilde{Y} = \{\tilde{Y}^0, \tilde{Y}^1\}$ using covariates X, treatment T, and factual outcomes Y as input. At the same time, the discriminator $D_{CF}(X, \bar{Y})$, where $\bar{Y} = \{Y, \tilde{Y}\}$ is a vector of both factual and generated outcomes, is trained to maximize the probability of correctly identifying the factual outcomes $\bar{Y}$. The loss function is presented below:

$$\begin{aligned} \mathcal{L}_D &= -\mathcal{L}(D_{CF}) \\ \mathcal{L}_G &= \text{MSE}(Y, \tilde{Y}) + \alpha \mathcal{L}(D_{CF}), \end{aligned} \tag{3.36}$$

where $\alpha > 0$ is a hyperparameter. The second block called $G_{ITE}(X)$ is then trained to predict potential outcomes using the learned counterfactual outcomes with only covariates X as input. The loss function for the ITE-block is presented below:

$$\mathcal{L}_{ITE} = \text{MSE}(\hat{Y}^1 - \hat{Y}^0, \bar{Y}^1 - \bar{Y}^0). \tag{3.37}$$

With the hyperparameter optimization algorithms we are looking for the number of hidden units and layers for generator n^g_{fc}, n^g_{hid}, discriminator n^d_{fc}, n^d_{hid}, and inference network $n^i_{fc_0}$, $n^i_{hid_0}$, $n^i_{fc_1}$, and $n^i_{hid_1}$. Additionally we are looking for batch size and learning rates for inference network bs_i, lr_i, and generator network bs_g, lr_g.

CEVAE

Variational autoencoder (VAE) is a probabilistic graphical model with a Bayesian foundation, approximating the observed distribution $p(X|Z)$ (decoder) conditioned on latent variables Z, which are sampled from the latent posterior distribution $q(Z|X)$ (encoder). Both the decoder and encoder are trained simultaneously to maximize the Evidence Lower Bound (ELBO) [149]. In the context of causal inference, VAEs are adapted to DAGs and define the process by which observations are generated. CEVAE [43] samples the proxy covariate distribution $p(X|Z)$, the binary treatment distribution $p(T|Z)$, and the outcome distribution

$p(Y|T,Z)$ from hidden variables. The inference network learns the posterior approximation through the complete input set, $q(Z|X,Y,T)$. The overall training objective is determined by maximizing the variational lower bound of the model, with the addition of auxiliary distributions $q(T|X)$ and $q(Y|X,T)$. The loss function for CEVAE is presented below:

$$\begin{aligned}\mathcal{L}_{\text{CEVAE}} = &\mathbb{E}_{q(Z|X,Y,T)}[\log p(X,T|Z) + \log p(Y|T,Z) \\ &+ \log p(Z) - \log q(Z|X,Y,T)] \\ &+ \log q(T|X) + \log q(Y|X,T).\end{aligned} \tag{3.38}$$

In the case of CEVAE we are searching for number of fully connected layers and units for decoder and encoder networks transforming: outcomes y: n^y_{fc}, n^y_{hid}, covariates X: n^X_{fc}, n^X_{hid}, and treatment T: n^T_{fc}, n^T_{hid} to latent space and back.

TEDVAE

TEDVAE [107] was based on the ideas of CEVAE. However, unlike CEVAE, which learns a combined latent representation to infer X, Y, and T, TEDVAE disentangles the latent factors into three independent components: Z_T, Z_Y, and Z_C. The instrumental factor Z_T influences only the treatment assignment, Z_Y affects only the outcome, and Z_C acts as a confounding factor, influencing both the treatment and the outcome. Each disentangled factor is represented not by a single value but by a distribution, which is learned through separate encoders: $q_T(Z_T|X)$, $q_C(Z_C|X)$, and $q_Y(Z_Y|X)$. The parameters for each distribution are generated by fully connected neural networks. The TEDVAE inference model consists of a decoder $p_X(X|Z_T,Z_C,Z_Y)$ reconstructing X, two disjoint decoders $pY(Y|T=1,Z_C,Z_Y)$ and $pY(Y|T=0,Z_C,Z_Y)$ predicting counterfactual outcomes, and a decoder $p_T(T|Z_T,Z_C)$ recovering the assigned treatment. The loss of the TEDVAE is presented below:

$$\begin{aligned}\mathcal{L}_{\text{TEDVAE}} = &\mathcal{L}_{\text{ELBO}}(X,Y,T) \\ &+ \alpha_T \, \mathbb{E}_{q_T q_C}[\log p_T(T|Z_T,Z_C)] \\ &+ \alpha_Y \, \mathbb{E}_{q_Y q_C}[\log p_Y(Y|T,Z_Y,Z_C)],\end{aligned} \tag{3.39}$$

where $\alpha_T > 0,\ \alpha_Y > 0$ are hyperparameters, and $\mathcal{L}_{\text{ELBO}}(X,Y,T)$ is:

$$\begin{aligned}\mathcal{L}_{\text{ELBO}} = &\mathbb{E}_{q_T q_C q_Y}[\log p_X(X|Z_T,Z_C,Z_Y)] \\ &- D_{\text{KL}}(q_T(Z_T|X)||p_T(Z_T)) \\ &- D_{\text{KL}}(q_C(Z_C|X)||p_C(Z_C)) \\ &- D_{\text{KL}}(q_Y(Z_Y|X)||p_Y(Z_Y)).\end{aligned} \tag{3.40}$$

Here, $q_T(Z_T|X)$, $qC(Z_C|X)$, and $qY(Z_Y|X)$ are Gaussian or Bernoulli distributions, depending on whether the outcome variable is continuous or binary. The mean and variance

of these distributions are parameterized by neural networks. The priors $p_T(Z_T)$, $p_C(Z_C)$, and $p_Y(Z_Y)$ are represented by Gaussian normal distributions. The Kullback-Leibler divergence (KL divergence), denoted by D_{KL}, is used to measure the difference between these posterior distributions and their corresponding priors. For TEDVAE we a searching for the number of hidden layers and units for the encoder n_{fc}^{enc}, n_{hid}^{enc} and decoder n_{fc}^{dec}, n_{hid}^{dec} as well as the learning rate lr.

Chapter 4

Results

In this chapter we present the results of comparison between GNN-TARnet, GAT-TARnet and other models. We also present results on the artificial SUM dataset and the disclosed PerPain dataset. For clarity, we present the results in separate sections for clarity.

4.1 Implementation Details

Existing and Artificial Datasets

We evaluate the performance of the proposed graph-based ITE estimation models, including both GNN-TARnet and GAT-TARnet, using different graph construction strategies. For GNN-TARnet, we consider the identity (ident.) graph (GNN-TARnet (ident.)), as well as graphs derived from the previously mentioned causal discovery algorithms: LiNGAM [86] (GNN-TARnet (LiNGAM)), PC [69] (GNN-TARnet (PC)), and GES [87] (GNN-TARnet (GES)). The GAT-TARnet model is evaluated using the same graph construction methods to ensure a consistent comparison. These models are benchmarked against established treatment effect estimators, including meta-learners such as SLearner and TLearner [93], as well as representation learning methods like CFR-Wass [24], TEDVAE [107], and GANITE [103]. The evaluation focuses on identifying the strengths and limitations of the proposed graph-based architectures when applied to standard tabular benchmarks, specifically the JOBS, IHDP_A, and IHDP_B datasets.

For GNN-TARnet, we optimize the number of layers and hidden units in the treatment-specific branches using the random search hyperparameter optimization algorithm from the Keras Tuner library [124]. The search space was defined as follows: the number of hidden units in the treatment-specific branches, n_{hid}^0 and n_{hid}^1, ranged from 16 to 256 in increments of 16, while the number of fully connected layers, n_{fc}^0 and n_{fc}^1, varied from 2 to 10 in steps of 1. The optimal hyperparameters selected for the IHDP_A, IHDP_B, JOBS, and SUM datasets are summarized in Table 4.1. In this table, n_{hid}^{gnn} denotes the number of hidden units and

Table 4.1. Parameters of GNN-TARnet model.

	n_{fc}^0	n_{fc}^1	n_{hid}^0	n_{hid}^1	n_{fc}^{gnn}	n_{hid}^{gnn}	bs	lr
IHDP$_A$	4	6	96	240	4	128	64	1e-4
IHDP$_B$	4	6	208	240	4	128	32	1e-4
JOBS	5	6	240	176	2	16	64	1e-2
SUM	2	2	16	16	2	16	2	1e-3

Table 4.2. Parameters of GAT-TARnet model.

	n_{fc}^0	n_{fc}^1	n_{hid}^0	n_{hid}^1	n_{heads}^{att}	n_{layers}^{att}	bs	lr
IHDP$_A$	4	6	96	240	2	2	64	1e-4
IHDP$_B$	4	6	208	240	2	2	64	1e-4
JOBS	4	6	240	176	2	2	64	1e-2
SUM	2	2	16	16	2	2	2	1e-3

n_{fc}^{gnn} represents the number of fully connected layers in the GNN block, with bs indicating the batch size and lr the learning rate.

For GAT-TARnet, we set the number of layers and hidden units in the treatment-specific branches to match the optimal configurations found for GNN-TARnet (see Table 4.1). This ensures a consistent comparison between the two architectures. In addition to these shared parameters, two attention-specific hyperparameters were introduced: the number of attention heads, n_{heads}^{att}, and the number of attention layers, n_{layers}^{att}. These were fixed at $n_{heads}^{att} = 2$ and $n_{layers}^{att} = 2$ across all datasets. The complete configuration of GAT-TARnet for the IHDP$_A$, IHDP$_B$, JOBS, and SUM datasets is presented in Table 4.2, with bs denoting batch size and lr the learning rate.

For the IHDP$_A$, IHDP$_B$, and JOBS datasets, all models were tuned on the first sub-dataset using the Random Search Tuner from Keras [124], supplemented by TensorFlow callback functions such as *EarlyStopping* and *ReduceLROnPlateau* to enhance training efficiency and performance. We used the Stochastic Gradient Descent (SGD) optimizer [150] with a momentum of 0.9, as it performed better empirically on all considered models than the ADAM optimizer [151] on the tested datasets. Since the graph structures for IHDP$_A$, IHDP$_B$, and JOBS were unknown, the graph adjacency matrix A was identified using the LiNGAM algorithm from the *LiNGAM* package [152], as well as the PC and GES algo-

rithms from the *causalai* library [153]. Neural network parameters were initialized with values drawn from the normal distribution $\mathcal{N}(0, 0.05^2)$. Optimal parameters for other models were selected according to the procedure outlined in the previous chapter and reported in the Appendix. All computations were performed on a system with an NVIDIA GeForce RTX 2060 SUPER, an AMD Ryzen 7 3800X 8-core processor, and 32 GB of RAM.

Disclosed Dataset

All models were trained using the SGD optimizer [150]. Model parameters were tuned using 200-fold cross-validation, and final performance was evaluated on a held-out test set not used during training or validation. For GNN-TARnet, the model included 58 hidden units following the embedding layer. The treatment-specific branches were configured with 60, 100, and 64 hidden units, respectively. The number of fully connected layers was set to 4 in the graph neural network block, and to 5, 2, and 10 in the treatment-specific branches. Random normal kernel initialization and a linear activation function were used in the treatment-specific branches. Within the graph layers, both the combination and aggregation types were implemented as summation operations. For GAT-TARnet, we used 4, 8, and 2 layers for treatment-specific branches with 48, 60, and 68 hidden units respectively. Additionally, the attention mechanism was configured with $n^{att}_{heads} = 2$ and $n^{att}_{layers} = 2$.

For TARnet, the shared representation block consisted of 10 fully connected layers with 102 hidden units. The treatment-specific branches included 4, 5, and 3 layers with 120, 72, and 72 hidden units, respectively. The learning rate was set to 1×10^{-4}. For the T-Learner, the architecture consisted of 4, 5, and 3 fully connected layers with 52, 104, and 120 hidden units, respectively.

All model configurations were determined using the random search hyperparameter optimization algorithm from the Keras Tuner library [124], with a validation split of 0.2. The search space was reduced compared to that used for the existing datasets to minimize the risk of overfitting. We searched the number of layers in the treatment-specific branches between 2 and 10. The number of hidden units was selected from the interval between 4 and 128 with a step of 4. Experiments were conducted on a system equipped with an NVIDIA GeForce RTX 2060 SUPER GPU, an AMD Ryzen 7 3800X 8-core processor, and 32 GB of RAM.

4.2 Existing Datasets

The comparison results between GNN-TARnet and other models are presented in Table 4.3 and Table 4.4. The results indicate that the proposed method performs comparably with other approaches.

Table 4.3. Comparison of models $\mathcal{R}_{pol}$ and $\sqrt{\epsilon_{\text{PEHE}}}$ on different datasets. The best-performing models are highlighted in bold.

	JOBS($\mathcal{R}_{pol}$)		IHDP$_{\text{A}}$($\sqrt{\epsilon_{\text{PEHE}}}$)		IHDP$_{\text{B}}$($\sqrt{\epsilon_{\text{PEHE}}}$)	
	Train	Test	Train	Test	Train	Test
SLearner	0.22 ± 0.00	0.23 ± 0.01	0.41 ± 0.03	0.43 ± 0.04	2.14 ± 0.04	2.30 ± 0.06
TLearner	0.22 ± 0.00	0.23 ± 0.01	0.50 ± 0.01	0.53 ± 0.02	1.87 ± 0.05	2.01 ± 0.06
RLearner	0.22 ± 0.00	0.24 ± 0.01	0.58 ± 0.06	0.59 ± 0.06	2.29 ± 0.06	2.40 ± 0.07
XLearner	0.22 ± 0.00	0.23 ± 0.01	0.70 ± 0.06	0.72 ± 0.07	1.95 ± 0.05	2.07 ± 0.06
TARnet	0.22 ± 0.00	0.23 ± 0.01	0.37 ± 0.01	0.39 ± 0.01	1.84 ± 0.04	1.99 ± 0.05
CFR-Wass	0.28 ± 0.00	0.28 ± 0.01	0.37 ± 0.03	0.38 ± 0.04	1.97 ± 0.04	2.10 ± 0.05
CFR-MMDSQ	0.21 ± 0.00	0.25 ± 0.01	0.53 ± 0.02	0.54 ± 0.03	$\mathbf{1.83 \pm 0.04}$	$\mathbf{1.97 \pm 0.05}$
CFR-Weight	0.23 ± 0.00	0.25 ± 0.01	0.49 ± 0.03	0.51 ± 0.03	1.89 ± 0.04	2.02 ± 0.05
Dragonnet	0.24 ± 0.00	0.26 ± 0.01	0.37 ± 0.01	0.39 ± 0.02	1.85 ± 0.04	2.00 ± 0.05
DKLITE	0.21 ± 0.00	0.23 ± 0.01	$\mathbf{0.34 \pm 0.02}$	$\mathbf{0.36 \pm 0.03}$	2.24 ± 0.06	2.40 ± 0.06
TEDVAE	$\mathbf{0.19 \pm 0.00}$	$\mathbf{0.23 \pm 0.01}$	0.52 ± 0.03	0.56 ± 0.05	2.01 ± 0.04	2.16 ± 0.05
CEVAE	0.22 ± 0.00	0.23 ± 0.01	0.83 ± 0.08	0.83 ± 0.08	2.51 ± 0.06	2.62 ± 0.06
GANITE	0.24 ± 0.00	0.25 ± 0.01	0.51 ± 0.06	0.53 ± 0.07	2.53 ± 0.08	2.63 ± 0.09
GNN-TARnet (LiNGAM)	0.22 ± 0.00	0.23 ± 0.01	0.40 ± 0.02	0.42 ± 0.03	2.12 ± 0.09	2.29 ± 0.11
GNN-TARnet (GES)	0.22 ± 0.00	0.23 ± 0.01	0.46 ± 0.04	0.48 ± 0.04	2.31 ± 0.08	2.48 ± 0.09
GNN-TARnet (PC)	0.22 ± 0.00	0.23 ± 0.01	0.49 ± 0.06	0.51 ± 0.07	2.78 ± 0.15	2.99 ± 0.17
GNN-TARnet (ident.)	0.22 ± 0.00	0.23 ± 0.01	0.53 ± 0.01	0.54 ± 0.02	1.85 ± 0.04	1.98 ± 0.05
GAT-TARnet (LiNGAM)	0.25 ± 0.00	0.25 ± 0.01	0.43 ± 0.02	0.46 ± 0.02	2.10 ± 0.09	2.24 ± 0.09
GAT-TARnet (GES)	0.28 ± 0.00	0.29 ± 0.01	0.46 ± 0.04	0.48 ± 0.04	2.16 ± 0.09	2.31 ± 0.09
GAT-TARnet (PC)	0.22 ± 0.00	0.23 ± 0.01	0.53 ± 0.06	0.57 ± 0.07	2.52 ± 0.14	2.69 ± 0.15
GAT-TARnet (ident.)	0.30 ± 0.00	0.30 ± 0.01	0.67 ± 0.02	0.70 ± 0.03	1.90 ± 0.05	2.05 ± 0.05

Table 4.3 compares the performance of GNN-TARnet and GAT-TARnet to a range of baseline and state-of-the-art models across three datasets: JOBS, IHDP$_{\text{A}}$, and IHDP$_{\text{B}}$. On the JOBS dataset, GNN-TARnet variants (LiNGAM, GES, PC, and identity graphs) consistently achieve test scores around 0.23, performing on par with strong baselines such as TARnet and Dragonnet. While TEDVAE reports the lowest policy risk on this dataset (0.23 ± 0.01), GNN-TARnet remains highly competitive. GAT-TARnet models, on the other hand, show greater variance, with test scores ranging from 0.23 to 0.30. GAT-TARnet (PC) matches the top performers, but its other variants, especially those using identity or GES graphs, exhibit reduced accuracy.

On the IHDP$_{\text{A}}$ dataset, GNN-TARnet (LiNGAM) demonstrates solid performance with a test error of 0.42 ± 0.03, closely following TARnet and Dragonnet, both of which report

Table 4.4. Comparison of models ϵ_{ATT} and ϵ_{ATE} on different datasets. The best-performing models are highlighted in bold.

	JOBS(ϵ_{ATT})		IHDP$_{\text{A}}$(ϵ_{ATE})		IHDP$_{\text{B}}$(ϵ_{ATE})	
	Train	Test	Train	Test	Train	Test
SLearner	0.22 ± 0.00	0.23 ± 0.01	0.09 ± 0.01	0.10 ± 0.01	0.24 ± 0.03	0.26 ± 0.04
TLearner	0.15 ± 0.00	0.17 ± 0.01	0.09 ± 0.01	0.11 ± 0.01	$\mathbf{0.18 \pm 0.03}$	$\mathbf{0.22 \pm 0.04}$
RLearner	0.15 ± 0.00	0.17 ± 0.01	0.11 ± 0.01	0.13 ± 0.01	0.31 ± 0.05	0.33 ± 0.04
XLearner	0.21 ± 0.00	0.22 ± 0.00	0.10 ± 0.01	0.13 ± 0.02	0.24 ± 0.03	0.27 ± 0.04
TARnet	0.15 ± 0.00	0.16 ± 0.01	0.09 ± 0.01	0.10 ± 0.01	0.19 ± 0.03	0.23 ± 0.03
CFR-Wass	0.09 ± 0.02	0.12 ± 0.02	0.10 ± 0.01	0.10 ± 0.01	0.33 ± 0.05	0.36 ± 0.05
CFR-MMDSQ	0.15 ± 0.01	0.16 ± 0.02	0.09 ± 0.01	0.10 ± 0.01	0.22 ± 0.03	0.24 ± 0.04
CFR-Weight	0.18 ± 0.01	0.20 ± 0.02	0.09 ± 0.01	0.10 ± 0.01	0.21 ± 0.03	0.24 ± 0.04
Dragonnet	$\mathbf{0.21 \pm 0.02}$	$\mathbf{0.23 \pm 0.03}$	0.09 ± 0.01	0.10 ± 0.01	0.21 ± 0.03	0.26 ± 0.03
DKLITE	0.19 ± 0.01	0.20 ± 0.02	$\mathbf{0.09 \pm 0.01}$	$\mathbf{0.09 \pm 0.01}$	0.23 ± 0.03	0.27 ± 0.04
TEDVAE	0.16 ± 0.00	0.17 ± 0.01	0.09 ± 0.01	0.11 ± 0.01	0.22 ± 0.03	0.25 ± 0.03
CEVAE	0.16 ± 0.01	0.17 ± 0.02	0.11 ± 0.01	0.14 ± 0.02	0.25 ± 0.03	0.30 ± 0.04
GANITE	0.28 ± 0.02	0.30 ± 0.02	0.15 ± 0.02	0.16 ± 0.02	0.37 ± 0.05	0.41 ± 0.06
GNN-TARnet (LiNGAM)	0.12 ± 0.00	0.14 ± 0.01	0.09 ± 0.01	0.10 ± 0.01	0.24 ± 0.02	0.28 ± 0.04
GNN-TARnet (GES)	0.12 ± 0.00	0.14 ± 0.01	0.09 ± 0.01	0.11 ± 0.01	0.24 ± 0.03	0.26 ± 0.04
GNN-TARnet (PC)	0.12 ± 0.00	0.14 ± 0.01	0.09 ± 0.01	0.10 ± 0.01	0.31 ± 0.04	0.34 ± 0.05
GNN-TARnet (ident.)	0.12 ± 0.00	0.14 ± 0.01	0.09 ± 0.01	0.11 ± 0.01	0.19 ± 0.02	0.23 ± 0.03
GAT-TARnet (LiNGAM)	0.32 ± 0.02	0.34 ± 0.02	0.10 ± 0.01	0.11 ± 0.01	0.24 ± 0.02	0.28 ± 0.04
GAT-TARnet (GES)	0.12 ± 0.01	0.14 ± 0.02	0.10 ± 0.01	0.11 ± 0.01	0.24 ± 0.03	0.26 ± 0.04
GAT-TARnet (PC)	0.36 ± 0.02	0.37 ± 0.02	0.11 ± 0.01	0.13 ± 0.01	0.31 ± 0.04	0.34 ± 0.05
GAT-TARnet (ident.)	0.30 ± 0.00	0.19 ± 0.01	0.12 ± 0.01	0.14 ± 0.02	0.20 ± 0.03	0.24 ± 0.03

slightly lower errors. The best results are obtained by DKLITE (0.36 ± 0.03), indicating room for further optimization of GNN-TARnet in this setting. GAT-TARnet (LiNGAM) achieves comparable results but does not significantly outperform simpler baselines. The trend continues on the IHDP$_{\text{B}}$ dataset, where GNN-TARnet (ident.) performs especially well, matching or surpassing many baselines with a test error of 1.98 ± 0.05. Notably, this is on par with TARnet and better than methods such as DKLITE and CEVAE. However, GNN-TARnet (PC) underperforms significantly on IHDP$_{\text{B}}$, likely due to the limitations of structure learning in high-dimensional settings. Similarly, GAT-TARnet shows inconsistent results: while some variants like GAT-TARnet (LiNGAM) maintain competitive accuracy, others (e.g., PC and identity) perform poorly, with test errors exceeding 2.60.

Overall, GNN-TARnet demonstrates stable and competitive performance across datasets,

particularly when using reasonable graph structures such as LiNGAM or identity matrices. In contrast, GAT-TARnet tends to be more sensitive to graph structure quality and dataset complexity, performing well in some cases but lacking robustness in others.

4.3 Artificial Dataset

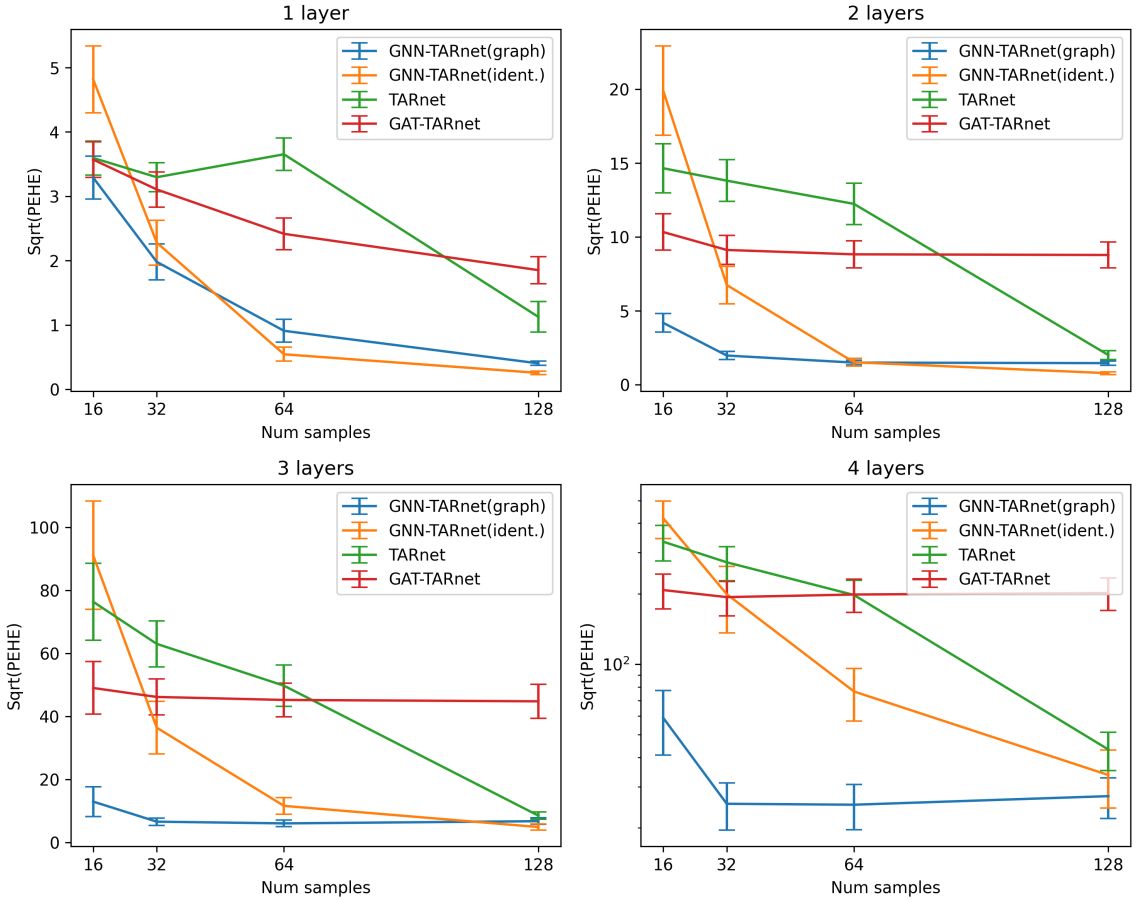


Figure 4.1. Dependency of number of layers and $\sqrt{\epsilon_{\text{PEHE}}}$ for different number of samples in the training set.

As illustrated in Figure 4.1, on the artificial dataset, the model using a real graph outperformed its counterpart using an identity graph when the number of training samples was small. As the training set size increased, the performance of GNN-TARnet with the identity graph steadily improved and eventually surpassed that of the model using real graphs. This pattern held consistently across datasets with varying depths (i.e., different numbers of layers). TARnet struggled to match the performance of GNN-TARnet under conditions of limited data, particularly when fewer than 128 samples were available, due to the presence of masked covariates, which posed a considerable challenge to accurate outcome estimation. Notably, the use of an identity graph in GNN-TARnet required more training samples

to achieve parity with the performance of GNN-TARnet using the actual graph, especially as the number of layers increased. Among all models, GAT-TARnet with the real graph demonstrated the second-best performance when the training sample size was 16. However, unlike other models, its performance plateaued and did not show significant improvement once the number of training samples exceeded 64.

4.4 Disclosed Dataset

From Table 4.5, we observe that both GNN-TARnet and GAT-TARnet significantly outperform the T-Learner and TARnet on the validation portion of the disclosed dataset. While the performance difference between GNN-TARnet and GAT-TARnet is not statistically significant, a two-sample Kolmogorov-Smirnov test [154] indicates that the predicted factual outcomes of GNN-TARnet are not significantly different from the actual outcomes, with a p-value of 0.99. Furthermore, an additional Kolmogorov-Smirnov test confirms that the outcomes resulting from reassigning treatments based on the highest average treatment effect are significantly different from the original assignment (p-value = 0.04). This level of alignment with actual outcomes was not observed for the predictions of the other methods. In terms of ATE, the original RCT-based assignment yields an ATE of 0.53 ± 1.11. In contrast, using the proposed strategy, which assigns treatment based on the highest predicted effect, the estimated ATE increases to 1.04 ± 0.75, indicating a substantial improvement. The results of applying GNN-TARnet, the best-performing model on the validation set, to the test set are shown in Figure 4.5. The model predicts PERT to yield superior treatment effects compared to alternative methods for the majority of participants. Subfigures in Figure 4.2 present the sorted treatment effects of EDDT predicted by GNN-TARnet on both the test and training sets as well as their standard deviations. Most real outcomes fall within one standard deviation of the predictions, demonstrating that GNN-TARnet effectively estimates the effects of this treatment. Similar trends are observed for EMDI and PERT, as illustrated in Figures 4.3 and 4.4, respectively.

	$\epsilon^{\text{train}}_{\text{error}}$	$\epsilon^{\text{val}}_{\text{error}}$
T-Learner	1.22 ± 0.01	2.33 ± 0.08
TARnet	1.37 ± 0.01	1.94 ± 0.06
GNN-TARnet	1.62 ± 0.02	$1.70 \pm 0.05^*$
GAT-TARnet	1.62 ± 0.02	$1.71 \pm 0.05^*$

Table 4.5. Performance of the models on the train and validation sets, where $*$ indicates significantly better results.

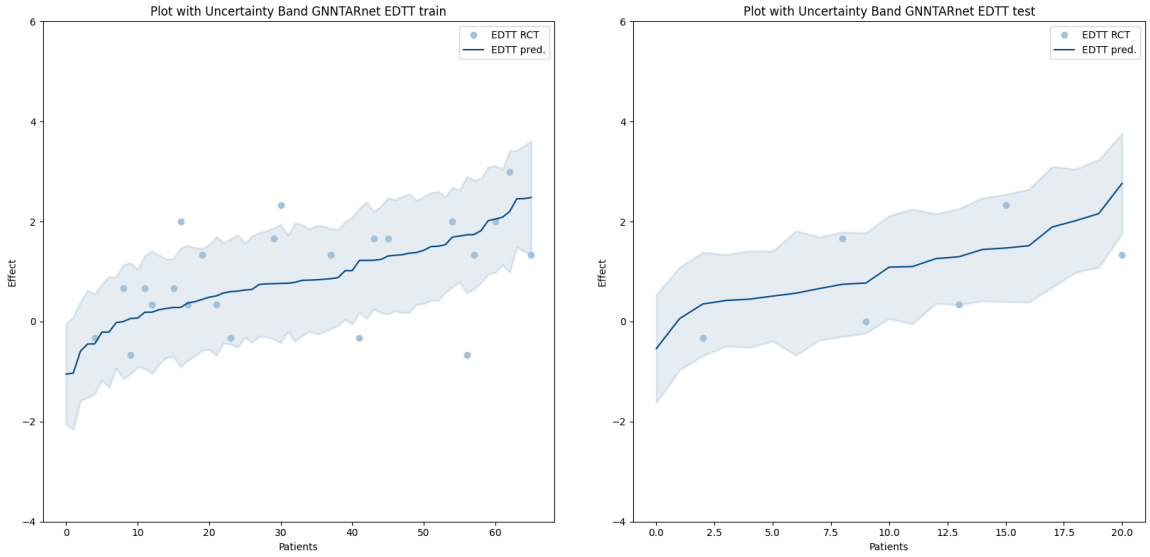


Figure 4.2. Combined Plots with Uncertainty Bands for GNN-TARnet on the train set (left) and test set (right) of the EDTT treatment. The shaded area denotes the uncertainty of the predicted treatment effects represented by the solid line. Observed treatment effects are marked by dots. It can be seen that most of them are located inside of the uncertainty area.

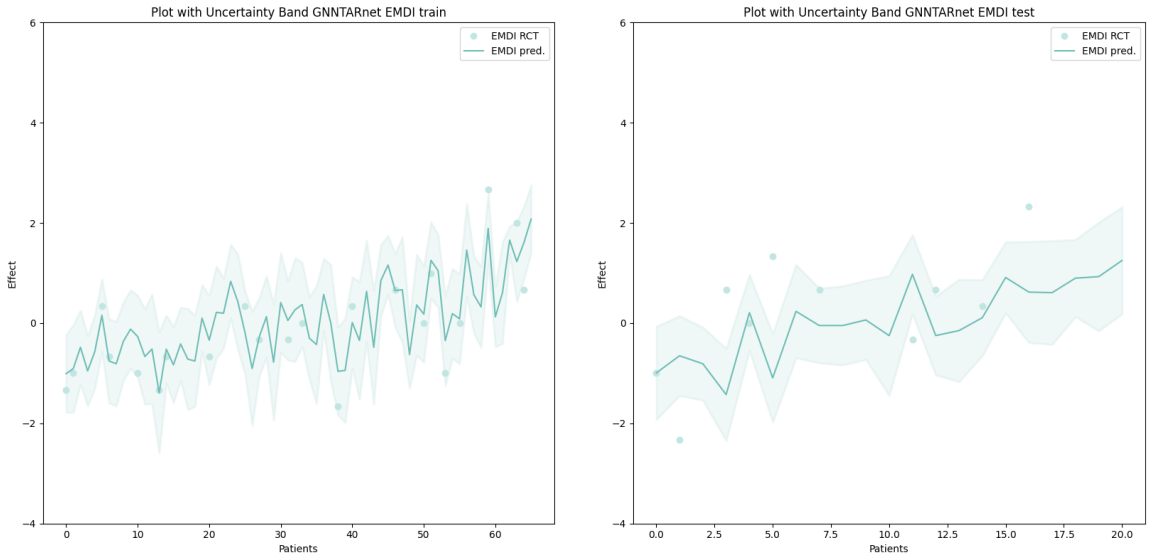


Figure 4.3. Combined Plots with Uncertainty Bands for GNN-TARnet on the train set (left) and test set (right) of the EMDI treatment. The shaded area denotes the uncertainty of the predicted treatment effects represented by the solid line. Observed treatment effects are marked by dots. It can be seen that most of them are located inside of the uncertainty area.

Figure 4.4. Combined Plots with Uncertainty Bands for GNN-TARnet on the train set (left) and test set (right) of the PERT treatment. The shaded area denotes the uncertainty of the predicted treatment effects represented by the solid line. Observed treatment effects are marked by dots. It can be seen that most of them are located inside of the uncertainty area.

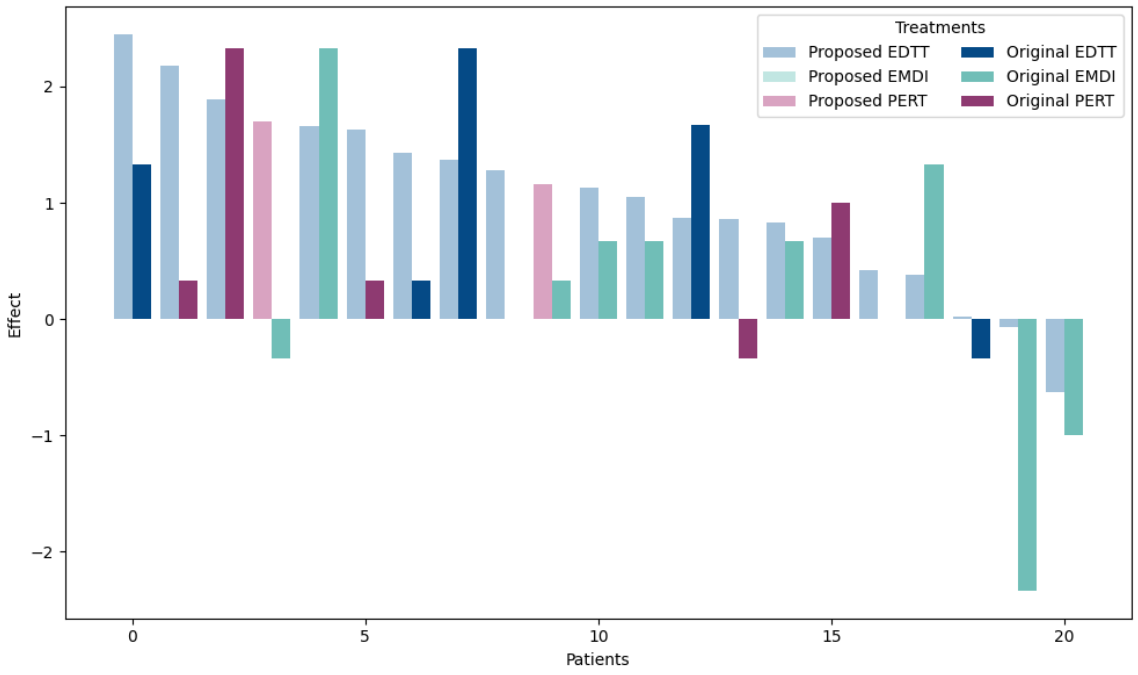


Figure 4.5. Effects of the proposed by GNN-TARnet treatments vs original treatments on the test set.

Chapter 5

Discussion

5.1 State-of-the-Art Comparison

Unlike TARnet or CFR, which rely on feature vectors alone, GNN-TARnet incorporates structural relationships (e.g., covariate dependencies) via graph convolutions, enhancing representation learning for ITE. This bridges graph-based learning with traditional causal inference on non-networked data. Unlike TEDVAE and CEVAE other covariate-confounding learning methods, our methods relies on causal graphs rather than grouping or disentangling variables by the influence type such as only influencing treatment, only influencing outcomes and influencing both treatment and the outcomes. If causal graphs are available these dependencies can be used for the ITE estimation without learning them.

While inspired by TARnet, our approach differs fundamentally from it by using graph-based layers to model structural covariate dependencies instead of fully-connected layers. Our method is designed to efficiently leverage connectivity information among covariates for ITE estimation. It can be used in scenarios when not enough training data is available. Other researchers have also used the power of GNNs to work with limited data. Panagopoulos et al. proposed UMGNET, a GNN framework for uplift modeling in e-commerce, reducing the training set size from $70\% - 80\%$ to $5\% - 20\%$ by leveraging bipartite user-product graphs and active learning [156]. However, the method was not directly used for ITE esti-

Connected Subjects	Connected Features
GIAL [84]	DCGs [85]
NN-CGC [155]	UMGNET [156]
	GNN-TARnet, GAT-TARnet

Table 5.1. Methods for estimating ITE using information about connectivity between subjects and within features of a single subject.

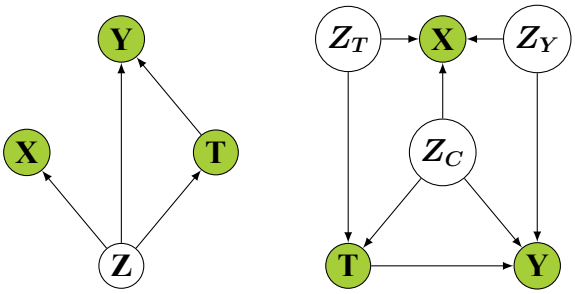


Figure 5.1. DAG of CEVAE (left) and TEDVAE (right) [3].

mation and thus different from our approach.

Parafita and Vitrià [85] proposed Deep Causal Graphs (DCGs). In contrast to traditional estimand-based methods, deriving query-specific formulas (e.g., back-door adjustment [85]), DCGs propose a general model using SCMs with Deep Causal Units (DCUs) to estimate any identifiable causal query, whether observational, interventional, or counterfactual, on arbitrary DAGs. Leveraging Normalizing Flows and a Graphical Conditioner, DCGs model complex distributions and scale efficiently. GNN-TARnet and GAT-TARnet models, built upon the TARnet architecture [24], share similarities with DCGs in utilizing graph structures but diverge in scope and methodology. While DCGs aim for query-agnostic generality, our approach focuses specifically on ITE estimation by predicting potential outcomes with GNNs. GNN-TARnet exploits structural relationships in a DAG (e.g., derived via the causal discovery algorithms, Section 3.2.1) to enhance ITE accuracy. Unlike DCGs, which model full joint distributions with expressive flows, our method prioritizes outcome prediction rather than explicitly modeling all variables or latent confounders.

Chu et al. [84] introduced Graph Infomax Adversarial Learning (GIAL) for ITE and ATE estimation on networked data with imbalanced structures combining GCN/GAT, structure mutual information, and adversarial learning using imbalance as a confounder proxy. Unlike our method it is designed to work with connected subjects and thus is different from our approach.

Pros and Vitrià [155] introduced Neural Networks with Causal Graph Constraints (NN-CGC) which is similar to our approach leverages causal graphs for ITE estimation. NN-CGC integrates causal constraints into existing architectures (e.g., TARNet, BCAUSS), enforcing a distribution

$$f_Y(X) \sim f_Y(f(Pa(Y)\backslash\{T\}), f(G_{x_1}), \ldots, f(G_{x_n})) \tag{5.1}$$

to reduce bias. In contrast, our GNN-TARnet and GAT-TARnet replace fully-connected layers of TARnet with GNN or GAT layers to process covariate relationships via an adjacency matrix. While NN-CGC offers robustness and broad applicability, its complexity limit flexibility. Our approach offers computational efficiency and strong performance on small, structured datasets. The list of methods sorted by their type of interaction is presented

in the Table 5.1

5.2 Existing Datasets

We begin by evaluating GNN-TARnet on the IHDP_A dataset. For this dataset, GNN-TARnet, using a graph obtained with the LiNGAM method, outperformed the model using an identity graph. However, the results were not as favorable as those achieved by CFR-Wass.

Unexpectedly, GNN-TARnet performed better with an identity graph than with a graph estimated by causal discovery methods on the IHDP_B dataset. This anomaly suggests that the complex nonlinear relationships between covariates and the outcome in IHDP_B present a challenge for the causal discovery algorithm, negatively impacting the results of GNN-TARnet when using the inferred graph. These findings indicate that, in cases where estimating the causal graph is particularly challenging, using an identity graph may be a viable alternative approach.

Table 5.2. The results of GNN-TARnet (LiNGAM) on various datasets, comparing cases where either all data or only nodes influencing outcomes Y are masked with zeros.

	Train	Test
IHDP_A (masked infl. Y)	1.75 ± 0.35	1.75 ± 0.35
IHDP_A (masked ALL)	1.94 ± 0.41	1.89 ± 0.39
IHDP_B (masked infl. Y)	4.57 ± 0.18	4.60 ± 0.20
IHDP_B (masked ALL)	4.67 ± 0.18	4.67 ± 0.21
JOBS (masked infl. Y)	0.30 ± 0.00	0.30 ± 0.00
JOBS (masked ALL)	0.30 ± 0.01	0.30 ± 0.01

In IHDP_A and IHDP_B, only a few randomly selected nodes are designed to influence the outcomes. This suggests that the causal discovery method did not identify all influential nodes; if it had, the results would have been the same when masking either the nodes influencing the outcomes or all nodes with zeros. However, as shown in Table 5.2, this was not the case. In the JOBS dataset, on the other hand, masking only the nodes influencing the outcomes produced almost identical results to masking all nodes with zeros, indicating that the causal discovery method successfully identified most outcome-influencing nodes. This led to results comparable to other those obtained with TARnet. This also suggests that other nodes appear to neither directly affect outcomes nor influence those that do. Overall,

for all three existing datasets, there appear to be no significant causal relationships among the features. This lack of connectivity likely contributes to the suboptimal performance of GNN-TARnet and GAT-TARnet on these datasets.

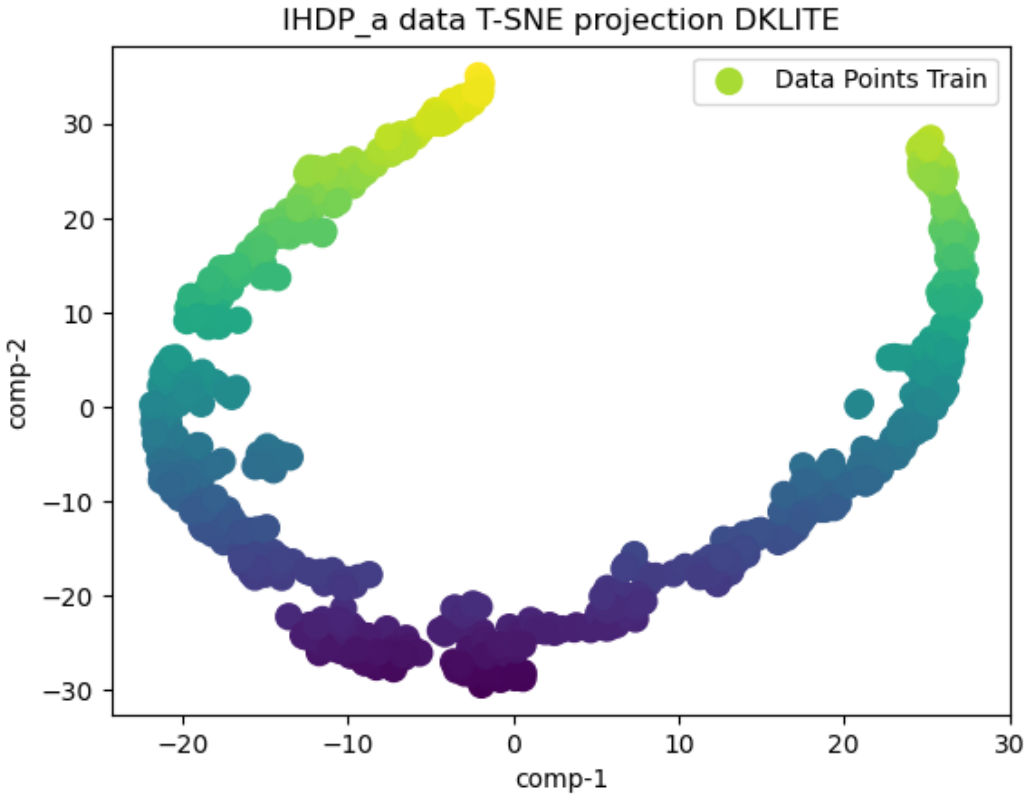


Figure 5.2. Two-dimensional T-SNE visualization of the latent space learned by the DKLITE model.

The results from existing datasets support our hypothesis that GNN-TARnet can compete with state-of-the-art methods in causal inference. This is unsurprising, as our approach builds upon the TARnet structure by preserving treatment-specific branches while enhancing it with the capability to incorporate causal graphs. However, the performance could have been even stronger if the actual adjacency matrices, with accurately defined nodes influencing the outcome, were available. The quality of the adjacency matrix is crucial, as it has a substantial impact on model effectiveness and can lead to either improved or diminished performance.

In order to find out why the DKLITE performed better than GNN-TARnet and GAT-TARnet on the IHDP_A dataset we made a plot of its hidden representation with reduced dimensionality to two using T-SNE algorithm (Figure 5.2). This indicates that for an optimal performance on this dataset, a model should project its hidden representation to a line-like structure. This can be confirmed by setting the hidden dimension of the TARnet before

branching to one. The results in Table 5.3 confirm our theory. This supports the conclusion that another reason why GNN-TARnet and GAT-TARnet did not show the best performance is the multi-dimensional structure of their hidden spaces.

Table 5.3. Comparison of the results on the IHDP_A dataset of DKLITE and the TARnet having one as an output of the representation layers.

	$\sqrt{\epsilon_{\text{PEHE}}}$		ϵ_{ATE}	
	Train	Test	Train	Test
DKLITE	0.34 ± 0.02	0.36 ± 0.03	0.09 ± 0.01	0.09 ± 0.01
TARnet	0.28 ± 0.02	0.31 ± 0.03	0.09 ± 0.01	0.09 ± 0.01

5.3 Artificial Dataset

For the SUM dataset, GNN-TARnet outperformed TARnet, even with the presence of zero-masked nodes that influence the outcome. GNN-TARnet adjusts the values of these nodes to be equal to the sum of their parent nodes. When using summation as the parent aggregation and node update function, an update step for a node $d_i^{out} = 0$ that influences the outcome in the SUM dataset with one layer is presented below.

$$h_i^{out} = \phi(d_i^{out} + \sum_{j \in \mathcal{N}_i} \psi(d_j)) = \phi(\sum_{j \in \mathcal{N}_i} \psi(d_j)) = \tag{10}$$

$$= w_i^{\phi}(\sum_{j \in \mathcal{N}_i} w_j^{\psi} d_j)) + b_i^{\phi} = \tag{11}$$

$$= \sum_{j \in \mathcal{N}_i} d_j. \tag{12}$$

In Equation (10), h_i^{out} represents the updated value of the masked node, d_j is the j-th parent of the node d_i^{out}, where j is an index from the set $\mathcal{N}_i$ of parent node indices. The functions ϕ and ψ correspond to the update and prepare functions from the GNN block (Figure 3.4). In Equation (11), w_i^{ϕ} and b_i^{ϕ} are the weight and bias of the update function ϕ, while w_j^{ψ} is the weight for the aggregate function ψ. Assuming that weights and biases were correctly identified during training as one and zero, respectively, Equation (12) shows that the updated masked node values are indeed equal to the sum of their parents.

Since the outcomes of the SUM dataset are calculated as the sum or average of nodes influencing the outcomes, knowing the values of these nodes renders ITE estimation trivial. This effect is particularly pronounced when the number of training samples is low. When a sufficient amount of data is available, however, the fully connected layers of TARnet can learn any relationship between features, matching the performance of GNN-TARnet.

5.4 Disclosed Dataset

From Figure 3.2 we can see that our algorithm recommends EDDT for most patients. When the other treatment outcomes are close to the predictions for EDDT, the differences are not significant. From the results, we can say that assigning EDDT to all patients seems to be a viable strategy, which warrants validation in future studies. We believe that the new treatment assignment strategy using GNN-TARnet can potentially be applied to all types of data coming from RCTs with a similar design as the PerPain RCT. This means that there should be randomization into personalized and non-personalized groups, followed by treatment assignment using clustering or some other baseline algorithm, as was done in the PerPain RCT. Applying this strategy to existing datasets can help determine if the original treatment assignment algorithm worked well or if there is a need to use an alternative approach. Our approach may not be applicable to pure RCTs without treatment assignment personalization.

5.5 Potential

Our proposed methods, GNN-TARnet and its extension GAT-TARnet, demonstrate strong potential for estimating ITE, delivering competitive performance across a variety of datasets, even when relying on estimated causal graphs. On benchmark datasets such as IHDP (747 samples) and JOBS (3,212 samples), GNN-TARnet effectively integrates GNNs with TARnet to leverage DAGs, resulting in precise ITE estimates. For instance, on the IHDP_A dataset with a LiNGAM-estimated graph, it achieves $\sqrt{\epsilon_{\text{PEHE}}} = 0.42$, comparable to state-of-the-art models like TARnet and the S-Learner. On the IHDP_B variant, it achieves one of the best performances even when using a simple identity graph, further showcasing its robustness.

On the synthetic SUM dataset (16-128 training samples), GNN-TARnet demonstrates strong performance in data-scarce settings. This sample efficiency stems from the ability of GNNs to propagate structural dependencies, mitigating overfitting risks commonly observed in traditional deep models such as TARnet.

When applied to real-world datasets like PerPain [10], GNN-TARnet and GAT-TARnet significantly outperform both TARnet and the T-Learner in terms of prediction error, while the difference between GNN-TARnet and GAT-TARnet was not significant. Importantly, we also show that our methods can also be extended to handle multiple treatments and uncertainty estimation, which are critical features for broader adoption in practical scenarios.

Collectively, these findings underscore the suitability of GNN-TARnet for real-world applications, particularly in fields such as healthcare (e.g., personalized medicine), economics, and policy-making, domains where small-sample causal inference is essential and experi-

mental data are limited. Its ability to deliver accurate ITE estimates with minimal data makes it a valuable tool for informed decision-making in resource-constrained environments.

5.6 Limitations

The primary limitation of the proposed method lies in its dependency on the quality of the causal graphs. This is because an incorrectly specified graph can significantly reduce the performance of the methods, as observed in the IHDP_A and IHDP_B datasets. Therefore, if no graph information is available and the training set is sufficiently large, methods that do not depend on graph information may be more beneficial and computationally efficient. It is also important to note that our approach requires more computational resources than other methods, which could limit its applicability in certain contexts. This is due to the increased number of parameters that scales with the number of features as a result of feature embedding. However, by computing the outcomes using only a subset of nodes that influence the outcomes, the method can handle relatively large input spaces, which is advantageous compared to other methods and makes it feasible for ITE estimation.

Another drawback of the proposed method is that the new treatment assignment strategy for the disclosed dataset needs to be verified in a new RCT because not all potential outcomes are available. However, an RCT can be designed in such a way that subjects in the personalized group are first treated with an original treatment assignment strategy, and after the first results, the treatment assignment is redefined using the proposed algorithm. Treatments for subjects in the randomized group should also be adjusted accordingly. An RCT that is updated in this way may be more beneficial for the treated subjects, as they may receive treatment that is more tailored to their needs. We also note that our algorithm is designed to assume a normal distribution of outcomes. Otherwise, one would have to compute uncertainties using alternative methods, such as bootstrapping [157], nonparametric statistics [158], or Bayesian inference [159].

5.7 Future Work

Future extensions, such as incorporating time-series modeling, hold promise for further expanding the applicability of our methods across a wider range of domains. Reducing the computational burden of our methods is another exciting avenue for future work. Finally, creating publicly available datasets where features are causally dependent on each other and the outcome is an important topic, as it can lead to better data validation and foster the development of improved GNN-based methods for causal effect estimation.

Chapter 6

Summary and Conclusion

This dissertation has explored the advancement of ITE estimation through the development and validation of GNN-based methods that leverage causal graph structures. The primary objective was to enhance the precision of ITE estimation, particularly in scenarios with limited training data, by incorporating structural dependencies among covariates. This chapter synthesizes the key findings, highlights the contributions, and reflects on the broader implications of this work for causal inference and personalized decision-making, particularly in healthcare applications.

6.1 Key Findings

The research began with the hypothesis that integrating causal graph structures into GNN-based models would improve ITE estimation accuracy compared to traditional methods that ignore such relationships. This hypothesis was tested across a range of datasets: synthetic datasets with controlled causal graphs, benchmark datasets such as IHDP and JOBS, with derived causal graphs, and real-world data from the PerPain consortium [10]. The proposed models, GNN-TARnet and GAT-TARnet, extend the TARnet framework [24] by embedding graph-based learning, enabling the capture of covariate dependencies encoded in DAGs.

Empirical results, detailed in Chapter 4, demonstrate that GNN-based methods outperform non-structural baselines in data-scarce scenarios. For instance, on synthetic datasets where the true causal graph is known, GNN-TARnet achieved lower $\sqrt{\epsilon_{\text{PEHE}}}$ values compared to standard neural networks, indicating superior accuracy in predicting individual treatment effects. On real datasets such as PerPain, the method showed competitive performance, particularly when combined with hyperparameter optimization strategies informed by our review [3]. These findings validate the hypothesis that structural information simplifies model training and enhances outcome estimation when data is limited, while matching the performance of state-of-the-art methods in data-rich settings.

A significant practical outcome was the application to the PerPain RCT, where the GNN-based approach improved treatment allocation compared to clustering-based method. By predicting potential outcomes and assigning treatments based on optimal predictions, the method offered a more robust framework for personalized chronic pain management.

6.2 Contributions

This thesis contributes to causal inference in several ways. First, it introduces novel GNN-based architectures, GNN-TARnet and GAT-TARnet, which integrate SCMs [29] with deep learning. The GNN-TARnet model was evaluated and validated in a peer-reviewed publication [1], demonstrating its efficacy on synthetic and benchmark datasets. Second, the co-authored preprint [3] provides a comprehensive review of deep learning methods for ITE estimation, proposing automatic hyperparameter optimization, a strategy that enhanced the performance of our models. Third, the practical contribution to the PerPain consortia, including data processing and the development of a clustering web application, demonstrates the practical utility of this research in real-world settings. The forthcoming publication on this work will further substantiate its impact. Finally, the open-source release of our implementation ensures reproducibility and fosters further development, in the spirit of scientific transparency and collaboration.

Theoretically, this work bridges SCMs with GNNs, advancing the understanding of how relational data can inform counterfactual reasoning [28]. Practically, it offers a tool for personalized treatment strategies, with implications beyond healthcare to fields like education and policy analysis.

6.3 Implications and Future Directions

The integration of GNNs with causal inference has significant implications for personalized decision-making. In healthcare, accurate ITE estimation can optimize treatment plans, reducing costs and improving patient outcomes. The success of GNN-TARnet in the PerPain context suggests potential scalability to other complex RCTs or observational studies where covariate relationships are critical [47]. Beyond medicine, the methodology could inform targeted interventions in social sciences or economics, where heterogeneous effects are prevalent [6].

However, limitations remain. The reliance on accurate causal graphs poses a challenge, as real-world graph estimation can be noisy or incomplete [73]. Scalability to large datasets and computational efficiency also warrant further investigation. Future research could expand the method to time-series data and create more datasets with causally dependent fea-

tures.

In conclusion, this dissertation establishes a robust foundation for GNN-based ITE estimation, demonstrating the advantage of leveraging structural knowledge. By combining theoretical innovation with practical tools, it advances the precision and applicability of causal inference, paving the way for future graph-based machine learning research in personalized decision-making.

Chapter 7

Appendix

The tables provide results for all models and different hyperparameters optimization techniques. Below one can also find the used hyperparameters of the mentioned methods.

Table 7.1. Results and their 95% confidence intervals of the Representation-Based methods on the IHDB_B dataset

	$\text{IHDP}_\text{B}(\sqrt{\epsilon_{\text{PEHE}}})$		$\text{IHDP}_\text{B}(\epsilon_{ATE})$	
	Train	Test	Train	Test
TARnet	1.87 ± 0.04	2.01 ± 0.05	0.20 ± 0.03	0.24 ± 0.04
TARnet-Bayesian	1.86 ± 0.04	2.01 ± 0.05	0.19 ± 0.03	0.24 ± 0.04
TARnet-Random	1.89 ± 0.04	2.04 ± 0.05	0.20 ± 0.02	0.25 ± 0.03
TARnet-Hyperband	1.84 ± 0.04	1.99 ± 0.05	0.19 ± 0.02	0.23 ± 0.03
CFR-MMDSQ	1.88 ± 0.04	2.01 ± 0.05	0.22 ± 0.03	0.24 ± 0.04
CFR-MMDSQ-Bayesian	1.83 ± 0.04	1.97 ± 0.05	0.20 ± 0.02	0.23 ± 0.04
CFR-MMDSQ-Random	1.84 ± 0.04	1.98 ± 0.05	0.21 ± 0.03	0.23 ± 0.04
CFR-MMDSQ-Hyperband	1.97 ± 0.04	2.10 ± 0.06	0.22 ± 0.02	0.26 ± 0.04
CFR-Wass	1.97 ± 0.04	2.10 ± 0.05	0.33 ± 0.05	0.36 ± 0.05
CFR-Wass-Bayesian	2.07 ± 0.05	2.19 ± 0.07	0.37 ± 0.06	0.39 ± 0.06
CFR-Wass-Random	2.10 ± 0.05	2.23 ± 0.06	0.43 ± 0.08	0.46 ± 0.08
CFR-Wass-Hyperband	1.97 ± 0.05	2.10 ± 0.06	0.36 ± 0.05	0.40 ± 0.05
CFR-Weight	1.89 ± 0.04	2.02 ± 0.05	0.21 ± 0.03	0.24 ± 0.04
CFR-Weight-Bayesian	2.14 ± 0.05	2.26 ± 0.06	0.21 ± 0.02	0.25 ± 0.04
CFR-Weight-Random	1.94 ± 0.04	2.10 ± 0.05	0.21 ± 0.02	0.25 ± 0.04
CFR-Weight-Hyperband	2.06 ± 0.05	2.18 ± 0.06	0.22 ± 0.03	0.26 ± 0.04
DragonNet	1.89 ± 0.04	2.03 ± 0.05	0.23 ± 0.03	0.26 ± 0.04
DragonNet-Bayesian	1.85 ± 0.04	2.00 ± 0.05	0.21 ± 0.03	0.26 ± 0.04
DragonNet-Random	1.88 ± 0.04	2.03 ± 0.06	0.22 ± 0.03	0.25 ± 0.04
DragonNet-Hyperband	1.88 ± 0.04	2.04 ± 0.05	0.24 ± 0.03	0.28 ± 0.04
DKLITE	2.54 ± 0.05	2.63 ± 0.07	0.27 ± 0.03	0.33 ± 0.04
DKLITE-Bayesian	3.71 ± 0.15	3.75 ± 0.16	0.45 ± 0.06	0.49 ± 0.07
DKLITE-Random	2.28 ± 0.05	2.40 ± 0.07	0.23 ± 0.03	0.27 ± 0.04
DKLITE-Hyperband	2.24 ± 0.06	2.40 ± 0.06	0.28 ± 0.03	0.30 ± 0.04

Table 7.2. Results and their 95% confidence intervals of the Representation-Based methods on the JOBS dataset

	JOBS($\mathcal{R}_{pol}$)		JOBS(ϵ_{ATT})	
	Train	Test	Train	Test
TARnet	0.26 ± 0.00	0.26 ± 0.01	0.11 ± 0.00	0.12 ± 0.01
TARnet-Bayesian	0.29 ± 0.00	0.29 ± 0.01	0.10 ± 0.00	0.12 ± 0.01
TARnet-Random	0.22 ± 0.00	0.23 ± 0.01	0.15 ± 0.00	0.16 ± 0.00
TARnet-Hyperband	0.30 ± 0.00	0.30 ± 0.01	0.08 ± 0.00	0.10 ± 0.01
CFR-MMDSQ	0.21 ± 0.00	0.26 ± 0.01	0.15 ± 0.01	0.16 ± 0.02
CFR-MMDSQ-Bayesian	0.21 ± 0.00	0.25 ± 0.01	0.19 ± 0.02	0.21 ± 0.03
CFR-MMDSQ-Random	0.24 ± 0.00	0.24 ± 0.01	0.31 ± 0.03	0.32 ± 0.03
CFR-MMDSQ-Hyperband	0.22 ± 0.00	0.25 ± 0.01	0.17 ± 0.00	0.18 ± 0.01
CFR-Wass	0.28 ± 0.00	0.28 ± 0.01	0.09 ± 0.02	0.12 ± 0.02
CFR-Wass-Bayesian	0.30 ± 0.00	0.30 ± 0.01	0.03 ± 0.01	0.08 ± 0.01
CFR-Wass-Random	0.28 ± 0.00	0.29 ± 0.01	0.11 ± 0.03	0.15 ± 0.03
CFR-Wass-Hyperband	0.30 ± 0.00	0.30 ± 0.01	0.02 ± 0.00	0.08 ± 0.01
CFR-Weight	0.23 ± 0.00	0.25 ± 0.01	0.18 ± 0.01	0.20 ± 0.02
CFR-Weight-Bayesian	0.26 ± 0.00	0.27 ± 0.01	0.19 ± 0.03	0.20 ± 0.04
CFR-Weight-Random	0.24 ± 0.00	0.26 ± 0.01	0.80 ± 0.07	0.81 ± 0.08
CFR-Weight-Hyperband	0.28 ± 0.00	0.28 ± 0.01	0.23 ± 0.05	0.24 ± 0.05
DragonNet	0.27 ± 0.00	0.27 ± 0.01	0.08 ± 0.00	0.10 ± 0.01
DragonNet-Bayesian	0.24 ± 0.00	0.26 ± 0.01	0.21 ± 0.02	0.23 ± 0.03
DragonNet-Random	0.26 ± 0.00	0.27 ± 0.01	0.52 ± 0.08	0.53 ± 0.08
DragonNet-Hyperband	0.30 ± 0.00	0.30 ± 0.01	0.03 ± 0.01	0.08 ± 0.01
DKLITE	0.21 ± 0.00	0.23 ± 0.01	0.19 ± 0.00	0.20 ± 0.01
DKLITE-Bayesian	0.22 ± 0.00	0.24 ± 0.01	0.20 ± 0.00	0.21 ± 0.01
DKLITE-Random	0.22 ± 0.00	0.24 ± 0.01	0.22 ± 0.00	0.23 ± 0.01
DKLITE-Hyperband	0.22 ± 0.00	0.24 ± 0.01	0.22 ± 0.00	0.23 ± 0.01

Table 7.3. Results and their 95% confidence intervals of the Meta-learners on IHDB_A dataset

	$\text{IHDP}_\text{A}(\sqrt{\epsilon_\text{PEHE}})$		$\text{IHDP}_\text{A}(\epsilon_{ATE})$	
	Train	Test	Train	Test
SLearner	0.61 ± 0.06	0.64 ± 0.07	0.09 ± 0.01	0.11 ± 0.01
SLerner-Bayesian	0.48 ± 0.03	0.50 ± 0.05	0.09 ± 0.01	0.11 ± 0.01
SLearner-Random	0.41 ± 0.03	0.43 ± 0.04	0.09 ± 0.01	0.10 ± 0.01
SLearner-Hyperband	1.19 ± 0.04	1.19 ± 0.05	0.14 ± 0.01	0.18 ± 0.02
TLearner	0.65 ± 0.03	0.68 ± 0.05	0.10 ± 0.01	0.13 ± 0.01
TLearner-Bayesian	0.72 ± 0.04	0.75 ± 0.06	0.10 ± 0.01	0.13 ± 0.01
TLearner-Random	0.50 ± 0.01	0.53 ± 0.02	0.09 ± 0.01	0.11 ± 0.01
TLearner-Hyperband	0.74 ± 0.02	0.77 ± 0.03	0.10 ± 0.01	0.14 ± 0.01
RLearner	0.81 ± 0.08	0.81 ± 0.08	0.13 ± 0.02	0.14 ± 0.02
RLearner-Bayesian	1.95 ± 0.41	1.90 ± 0.40	0.24 ± 0.05	0.32 ± 0.08
RLearner-Random	0.58 ± 0.06	0.59 ± 0.06	0.11 ± 0.01	0.13 ± 0.01
RLearner-Hyperband	0.61 ± 0.05	0.62 ± 0.06	0.11 ± 0.01	0.12 ± 0.01
XLearner	0.95 ± 0.09	0.97 ± 0.10	0.16 ± 0.02	0.17 ± 0.02
Learner-Bayesian	0.80 ± 0.06	0.82 ± 0.07	0.11 ± 0.01	0.14 ± 0.02
XLearner-Random	0.76 ± 0.05	0.76 ± 0.06	0.11 ± 0.01	0.13 ± 0.02
XLearner-Hyperband	0.70 ± 0.06	0.72 ± 0.07	0.10 ± 0.01	0.13 ± 0.02

Table 7.4. Results and their 95% confidence intervals of the Meta-learners on IHDB_B dataset

	$\text{IHDP}_\text{B}(\sqrt{\epsilon_\text{PEHE}})$		$\text{IHDP}_\text{B}(\epsilon_{ATE})$	
	Train	Test	Train	Test
SLearner	2.69 ± 0.07	2.77 ± 0.08	0.28 ± 0.03	0.33 ± 0.04
SLearner-Bayesian	2.23 ± 0.05	2.41 ± 0.06	0.27 ± 0.04	0.33 ± 0.04
SLearner-Random	2.15 ± 0.04	2.32 ± 0.06	0.24 ± 0.03	0.28 ± 0.04
SLearner-Hyperband	2.14 ± 0.04	2.30 ± 0.06	0.24 ± 0.03	0.26 ± 0.04
TLearner	1.96 ± 0.04	2.09 ± 0.06	0.19 ± 0.02	0.24 ± 0.04
TLearner-Bayesian	2.29 ± 0.06	2.45 ± 0.07	0.21 ± 0.03	0.28 ± 0.04
TLearner-Random	1.87 ± 0.05	2.01 ± 0.06	0.18 ± 0.03	0.22 ± 0.04
TLearner-Hyperband	2.00 ± 0.05	2.14 ± 0.06	0.20 ± 0.02	0.24 ± 0.04
RLearner	2.29 ± 0.06	2.40 ± 0.07	0.31 ± 0.05	0.33 ± 0.04
RLearner-Bayesian	2.55 ± 0.06	2.64 ± 0.08	0.36 ± 0.05	0.39 ± 0.06
RLearner-Random	2.45 ± 0.07	2.55 ± 0.08	0.27 ± 0.04	0.33 ± 0.04
RLearner-Hyperband	2.48 ± 0.07	2.56 ± 0.08	0.56 ± 0.07	0.59 ± 0.08
XLearner	1.95 ± 0.05	2.07 ± 0.06	0.24 ± 0.03	0.27 ± 0.04
XLearner-Bayesian	2.69 ± 0.07	2.77 ± 0.08	0.28 ± 0.03	0.33 ± 0.04
XLearner-Random	2.22 ± 0.05	2.36 ± 0.06	0.28 ± 0.04	0.32 ± 0.05
XLearner-Hyperband	2.13 ± 0.05	2.29 ± 0.06	0.24 ± 0.03	0.29 ± 0.04

Table 7.5. Results and their 95% confidence intervals of the Meta-learners on JOBS dataset

	JOBS($\mathcal{R}_{pol}$)		JOBS(ϵ_{ATT})	
	Train	Test	Train	Test
SLearner	0.23 ± 0.00	0.26 ± 0.01	0.17 ± 0.00	0.19 ± 0.01
SLerner-Bayesian	0.23 ± 0.00	0.26 ± 0.01	0.17 ± 0.00	0.19 ± 0.01
SLearner-Random	0.22 ± 0.00	0.23 ± 0.01	0.22 ± 0.00	0.23 ± 0.01
SLearner-Hyperband	0.22 ± 0.00	0.25 ± 0.01	0.18 ± 0.00	0.19 ± 0.01
TLearner	0.22 ± 0.00	0.23 ± 0.01	0.15 ± 0.00	0.17 ± 0.01
TLearner-Bayesian	0.27 ± 0.00	0.28 ± 0.01	0.07 ± 0.00	0.10 ± 0.01
TLearner-Random	0.22 ± 0.00	0.23 ± 0.01	0.15 ± 0.00	0.16 ± 0.01
TLearner-Hyperband	0.30 ± 0.00	0.30 ± 0.01	0.06 ± 0.00	0.08 ± 0.01
RLearner	0.23 ± 0.00	0.25 ± 0.01	0.16 ± 0.00	0.17 ± 0.01
RLearner-Bayesian	0.22 ± 0.00	0.24 ± 0.01	0.15 ± 0.00	0.17 ± 0.01
RLearner-Random	0.23 ± 0.00	0.25 ± 0.01	0.16 ± 0.00	0.17 ± 0.01
RLearner-Hyperband	0.23 ± 0.00	0.25 ± 0.01	0.17 ± 0.00	0.18 ± 0.01
XLearner	0.22 ± 0.00	0.23 ± 0.01	0.22 ± 0.00	0.23 ± 0.01
XLearner-Bayesian	0.22 ± 0.00	0.23 ± 0.01	0.21 ± 0.00	0.22 ± 0.01
XLearner-Random	0.22 ± 0.00	0.23 ± 0.01	0.21 ± 0.00	0.22 ± 0.01
XLearner-Hyperband	0.22 ± 0.00	0.23 ± 0.01	0.21 ± 0.00	0.22 ± 0.01

Table 7.6. Results and their 95% confidence intervals of Covariate-Confounding methods on IHDB_A dataset

	$\text{IHDP}_\text{A}(\sqrt{\epsilon_{\text{PEHE}}})$		$\text{IHDP}_\text{A}(\epsilon_{ATE})$	
	Train	Test	Train	Test
GANITE	0.60 ± 0.08	0.62 ± 0.09	0.16 ± 0.02	0.18 ± 0.02
GANITE-Bayesian	0.65 ± 0.06	0.68 ± 0.07	0.18 ± 0.03	0.18 ± 0.03
GANITE-Random	0.71 ± 0.08	0.71 ± 0.08	0.20 ± 0.02	0.22 ± 0.03
GANITE-Hyperband	0.51 ± 0.06	0.53 ± 0.07	0.15 ± 0.02	0.16 ± 0.03
CEVAE	0.83 ± 0.08	0.83 ± 0.08	0.11 ± 0.01	0.14 ± 0.02
CEVAE-Bayesian	0.91 ± 0.12	0.89 ± 0.11	0.12 ± 0.01	0.15 ± 0.02
CEVAE-Random	0.92 ± 0.12	0.90 ± 0.12	0.12 ± 0.01	0.14 ± 0.02
CEVAE-Hyperband	0.96 ± 0.13	0.94 ± 0.12	0.12 ± 0.02	0.15 ± 0.02
TEDVAE	0.57 ± 0.05	0.61 ± 0.07	0.09 ± 0.01	0.11 ± 0.01
TEDVAE-Bayesian	0.90 ± 0.06	1.00 ± 0.08	0.10 ± 0.01	0.16 ± 0.02
TEDVAE-Random	0.52 ± 0.03	0.56 ± 0.05	0.09 ± 0.01	0.11 ± 0.01
TEDVAE-Hyperband	0.66 ± 0.04	0.70 ± 0.05	0.10 ± 0.01	0.12 ± 0.02

Table 7.7. Results and their 95% confidence intervals of Covariate-Confounding methods on IHDB_B dataset

	$\text{IHDP}_\text{B}(\sqrt{\epsilon_\text{PEHE}})$		$\text{IHDP}_\text{B}(\epsilon_{ATE})$	
	Train	Test	Train	Test
GANITE	3.49 ± 0.10	3.54 ± 0.11	0.46 ± 0.06	0.55 ± 0.08
GANITE-Bayesian	2.68 ± 0.05	2.77 ± 0.07	0.30 ± 0.04	0.36 ± 0.05
GANITE-Random	2.84 ± 0.08	2.91 ± 0.09	0.51 ± 0.07	0.55 ± 0.07
GANITE-Hyperband	2.53 ± 0.08	2.63 ± 0.09	0.37 ± 0.05	0.41 ± 0.06
CEVAE	2.51 ± 0.06	2.62 ± 0.06	0.27 ± 0.03	0.32 ± 0.04
CEVAE-Bayesian	2.68 ± 0.07	2.75 ± 0.08	0.25 ± 0.03	0.30 ± 0.04
CEVAE-Random	2.54 ± 0.06	2.64 ± 0.07	0.26 ± 0.04	0.31 ± 0.04
CEVAE-Hyperband	2.68 ± 0.07	2.77 ± 0.09	0.28 ± 0.04	0.30 ± 0.05
TEDVAE	2.34 ± 0.06	2.46 ± 0.07	0.23 ± 0.03	0.27 ± 0.04
TEDVAE-Bayesian	2.01 ± 0.04	2.16 ± 0.05	0.22 ± 0.03	0.25 ± 0.04
TEDVAE-Random	2.11 ± 0.05	2.25 ± 0.07	0.23 ± 0.03	0.26 ± 0.04
TEDVAE-Hyperband	2.17 ± 0.05	2.30 ± 0.07	0.21 ± 0.03	0.23 ± 0.04

Table 7.8. Results and their 95% confidence intervals of Covariate-Confounding methods on JOBS dataset

	$\text{JOBS}(\mathcal{R}_{pol})$		$\text{JOBS}(\epsilon_{ATT})$	
	Train	Test	Train	Test
GANITE	0.30 ± 0.00	0.30 ± 0.01	0.10 ± 0.00	0.12 ± 0.01
GANITE-Bayesian	0.23 ± 0.00	0.26 ± 0.01	0.27 ± 0.00	0.28 ± 0.02
GANITE-Random	0.24 ± 0.00	0.25 ± 0.01	0.28 ± 0.02	0.30 ± 0.02
GANITE-Hyperband	0.25 ± 0.00	0.27 ± 0.01	0.25 ± 0.01	0.26 ± 0.02
CEVAE	0.22 ± 0.00	0.23 ± 0.01	0.21 ± 0.02	0.22 ± 0.03
CEVAE-Bayesian	0.24 ± 0.00	0.26 ± 0.01	2.85 ± 0.62	2.84 ± 0.58
CEVAE-Random	0.22 ± 0.00	0.24 ± 0.01	0.16 ± 0.00	0.17 ± 0.01
CEVAE-Hyperband	0.22 ± 0.00	0.26 ± 0.01	0.28 ± 0.08	0.29 ± 0.07
TEDVAE	0.20 ± 0.00	0.24 ± 0.01	0.16 ± 0.01	0.17 ± 0.01
TEDVAE-Bayesian	0.19 ± 0.00	0.23 ± 0.01	0.16 ± 0.00	0.17 ± 0.02
TEDVAE-Random	0.20 ± 0.00	0.23 ± 0.01	0.16 ± 0.01	0.17 ± 0.02
TEDVAE-Hyperband	0.20 ± 0.00	0.24 ± 0.01	0.17 ± 0.01	0.19 ± 0.02

Table 7.9. Results and their 95% confidence intervals of Representation-Based methods on IHDB_A dataset

	$\text{IHDP}_\text{A}(\sqrt{\epsilon_{\text{PEHE}}})$		$\text{IHDP}_\text{A}(\epsilon_{ATE})$	
	Train	Test	Train	Test
TARnet	0.47 ± 0.11	0.49 ± 0.03	0.09 ± 0.01	0.11 ± 0.01
TARnet-Bayesian	1.23 ± 0.06	1.28 ± 0.06	0.11 ± 0.01	0.18 ± 0.02
TARnet-Random	0.37 ± 0.01	0.39 ± 0.02	0.09 ± 0.01	0.10 ± 0.01
TARnet-Hyperband	1.20 ± 0.05	1.24 ± 0.06	0.11 ± 0.01	0.18 ± 0.02
CFR-MMDSQ	0.53 ± 0.03	0.55 ± 0.04	0.09 ± 0.01	0.11 ± 0.01
CFR-MMDSQ-Bayesian	0.58 ± 0.02	0.59 ± 0.03	0.11 ± 0.01	0.12 ± 0.02
CFR-MMDSQ-Random	0.53 ± 0.02	0.54 ± 0.03	0.09 ± 0.01	0.10 ± 0.01
CFR-MMDSQ-Hyperband	0.59 ± 0.02	0.62 ± 0.03	0.11 ± 0.01	0.12 ± 0.01
CFR-Wass	0.47 ± 0.05	0.49 ± 0.06	0.09 ± 0.01	0.10 ± 0.01
CFR-Wass-Bayesian	0.63 ± 0.02	0.64 ± 0.03	0.18 ± 0.03	0.19 ± 0.03
CFR-Wass-Random	0.37 ± 0.03	0.38 ± 0.04	0.10 ± 0.01	0.10 ± 0.01
CFR-Wass-Hyperband	0.39 ± 0.02	0.41 ± 0.03	0.12 ± 0.01	0.12 ± 0.02
CFR-Weight	0.56 ± 0.05	0.58 ± 0.06	0.09 ± 0.01	0.11 ± 0.01
CFR-Weight-Bayesian	2.15 ± 0.38	2.10 ± 0.36	0.47 ± 0.05	0.55 ± 0.06
CFR-Weight-Random	0.49 ± 0.02	0.51 ± 0.03	0.09 ± 0.01	0.10 ± 0.01
CFR-Weight-Hyperband	0.59 ± 0.03	0.60 ± 0.04	0.10 ± 0.01	0.12 ± 0.01
DragonNet	0.55 ± 0.02	0.58 ± 0.03	0.10 ± 0.01	0.12 ± 0.01
DragonNet-Bayesian	0.76 ± 0.04	0.78 ± 0.05	0.17 ± 0.04	0.21 ± 0.04
DragonNet-Random	0.37 ± 0.01	0.39 ± 0.02	0.09 ± 0.01	0.10 ± 0.01
DragonNet-Hyperband	0.48 ± 0.01	0.50 ± 0.03	0.10 ± 0.01	0.11 ± 0.01
DKLITE	0.84 ± 0.15	0.85 ± 0.15	0.09 ± 0.01	0.15 ± 0.02
DKLITE-Bayesian	1.66 ± 0.38	1.55 ± 0.36	0.13 ± 0.02	0.19 ± 0.04
DKLITE-Random	0.34 ± 0.02	0.36 ± 0.03	0.09 ± 0.01	0.09 ± 0.01
DKLITE-Hyperband	0.67 ± 0.22	0.65 ± 0.19	0.09 ± 0.01	0.12 ± 0.02

Table 7.10. Default and optimal parameters for S-Learner

	n_{fc}	n_{hid}	lr	bs
Default				
IHDP_A	3	300	1e-3	64
IHDP_B	3	300	1e-3	64
JOBS	3	300	1e-3	64
Optimal				
IHDP_A (Random)	7	80	1e-3	64
IHDP_B (Hyperband)	9	512	1e-3	64
JOBS (Bayesian)	2	16	1e-3	128

Table 7.11. Default and optimal parameters for T-Learner

	n_{fc}^0	n_{hid}^0	n_{fc}^1	n_{hid}^1	lr^0	bs^0	lr^1	bs^1
Default								
IHDP_A	3	300	3	300	1e-2	64	1e-2	64
IHDP_B	3	300	3	300	1e-2	64	1e-2	64
JOBS	3	64	3	64	1e-2	256	1e-2	256
Optimal								
IHDP_A (Random)	4	96	4	240	0.01	128	0.01	64
IHDP_B (Random)	4	96	10	128	1e-2	128	1e-2	128
JOBS (Bayesian)	2	512	2	512	1e-3	64	1e-3	64

Table 7.12. Default and optimal parameters for R-Learner

	n_{fc}^{mu}	n_{hid}^{mu}	n_{fc}^{g}	n_{hid}^{g}	n_{fc}^{tau}	n_{hid}^{tau}	lr^{mu}	lr^{g}	lr^{r}	bs^{mu}	bs^{g}	bs^{r}
Default												
$IHDP_A$	3	200	3	100	3	200	1e-3	1e-3	1e-3	32	32	32
$IHDP_B$	3	200	3	100	3	200	1e-3	1e-3	1e-3	32	32	32
JOBS	3	200	3	100	3	200	1e-3	1e-3	1e-3	32	32	32
Optimal												
$IHDP_A$ (Random)	2	128	5	18	6	32	1e-3	1e-2	1e-3	128	512	32
$IHDP_B$ (Bayesian)	3	16	10	8	3	16	1e-2	1e-4	1e-2	128	128	32
JOBS (Baysian)	2	160	2	24	2	512	1e-2	1e-2	1e-3	32	512	32

Table 7.13. Default and optimal parameters for X-Learner

	$n_{fc}^{mu_0}$	$n_{hid}^{mu_0}$	$n_{fc}^{mu_1}$	$n_{hid}^{mu_1}$	$n_{fc}^{d_0}$	$n_{hid}^{d_0}$	$n_{fc}^{d_1}$	$n_{hid}^{d_1}$	n_{fc}^{g}	n_{hid}^{g}	bs_{mu_0}	bs_{mu_1}	bs_{d_0}	bs_{d_1}	bs_g
Default															
$IHDP_A$	3	200	3	200	3	200	3	200	3	200	256	256	512	512	256
$IHDP_B$	3	200	3	200	3	200	3	200	3	200	256	256	512	512	256
JOBS	3	300	3	300	3	300	3	300	3	300	256	256	512	512	256
Optimal															
$IHDP_A$ (Random)	3	128	4	80	7	176	4	80	3	40	32	32	256	32	64
$IHDP_B$ (Default)	3	200	3	200	3	200	3	200	3	200	256	256	256	256	256
JOBS (Hyperband)	2	80	3	80	2	80	2	240	2	56	32	80	32	32	32

Table 7.14. Default and optimal parameters for TAR-Net

	n_{fc}	n_{hid}	n^0_{fc}	n^0_{hid}	n^1_{fc}	n^1_{hid}	lr	bs
Default								
$IHDP_A$	3	200	3	100	3	100	1e-4	32
$IHDP_B$	3	200	3	200	3	200	1e-4	32
JOBS	3	200	3	200	3	200	1e-2	256
Optimal								
$IHDP_A$ (Random)	7	192	6	464	4	432	1e-4	32
$IHDP_B$ (Random)	4	240	5	240	2	400	1e-4	32
JOBS (Random)	2	16	4	464	4	256	1e-2	1024

Table 7.15. Default and optimal parameters for CFR-Wass

	n_{fc}	n_{hid}	n^0_{fc}	n^0_{hid}	n^1_{fc}	n^1_{hid}	lr	bs
Default								
$IHDP_A$	3	200	3	100	3	100	1e-4	1024
$IHDP_B$	3	200	3	100	3	100	1e-3	1024
JOBS	3	200	3	200	3	200	1e-2	1024
Optimal								
$IHDP_A$ (Random)	8	208	10	416	4	480	1e-4	1024
$IHDP_B$ (Default)	3	200	3	100	3	100	1e-3	1024
JOBS (Random)	4	240	5	240	2	400	1e-2	1024

Table 7.16. Default and optimal parameters for CFR-MMDSQ

	n_{fc}	n_{hid}	n^0_{fc}	n^0_{hid}	n^1_{fc}	n^1_{hid}	lr	bs
Default								
$IHDP_A$	3	200	3	100	3	100	1e-4	1024
$IHDP_B$	3	200	3	100	3	100	1e-3	1024
JOBS	3	200	3	200	3	200	1e-2	1024
Optimal								
$IHDP_A$ (Random)	8	208	10	416	4	480	1e-4	1024
$IHDP_B$ (Random)	4	240	5	240	2	400	1e-3	1024
JOBS (Bayesian)	3	416	5	128	3	192	1e-2	1024

Table 7.17. Default and optimal parameters for CFR-Weight

	n_{fc}	n_{hid}	n^0_{fc}	n^0_{hid}	n^1_{fc}	n^1_{hid}	n^t_{fc}	n^t_{hid}	bs	lr
Default										
$IHDP_A$	3	200	3	100	3	100	3	100	1024	1e-4
$IHDP_B$	3	200	3	100	3	100	3	100	1024	1e-3
JOBS	3	200	2	100	2	100	1	1	1024	1e-2
Optimal										
$IHDP_A$ (Random)	7	192	6	464	4	432	6	406	1024	1e-4
$IHDP_B$ (Default)	3	200	2	100	2	100	1	1	1024	1e-3
JOBS (Default)	3	200	2	100	2	100	1	1	1024	1e-2

Table 7.18. Default and optimal parameters for Dragon-Net

	n_{fc}	n_{hid}	n^0_{fc}	n^0_{hid}	n^1_{fc}	n^1_{hid}	n^t_{fc}	n^t_{hid}	bs	lr
Default										
$IHDP_A$	3	200	3	100	3	100	1	1	64	1e-4
$IHDP_B$	3	200	3	100	3	100	1	1	32	1e-4
JOBS	3	200	3	100	3	100	1	1	512	1e-3
Optimal										
$IHDP_A$ (Random)	7	192	6	464	4	432	6	496	64	1e-4
$IHDP_B$ (Bayesian)	3	416	5	128	3	192	3	112	32	1e-4
JOBS (Bayesian)	2	192	4	96	7	64	8	336	512	1e-3

Table 7.19. Default and optimal parameters for DKLITE

	n^{enc}_{fc}	n^{enc}_{hid}	n^{dec}_{fc}	n^{dec}_{hid}	lr	bs
Default						
$IHDP_A$	2	50	2	50	1e-3	1024
$IHDP_B$	2	50	2	50	1e-3	1024
JOBS	2	50	2	50	1e-4	512
Optimal						
$IHDP_A$ (Random)	7	80	6	416	1e-3	1024
$IHDP_B$(Hyperband)	9	48	9	512	1e-3	1024
JOBS (Default)	2	50	2	50	1e-3	512

Table 7.20. Default and optimal parameters for CEVAE

	n_{fc}^{y}	n_{hid}^{y}	n_{fc}^{X}	n_{hid}^{X}	n_{fc}^{t}	n_{hid}^{t}	lr	bs
Default								
$IHDP_A$	3	200	3	200	3	200	1e-3	64
$IHDP_B$	3	200	3	200	3	200	1e-3	64
JOBS	3	200	3	200	3	200	1e-3	1024
Optimal								
$IHDP_A$ (Default)	3	200	3	200	3	200	1e-3	64
$IHDP_B$ (Default)	3	200	3	200	3	200	1e-3	64
JOBS (Default)	3	200	3	200	3	200	1e-3	1024

Table 7.21. Default and optimal parameters for TEDVAE

	n_{fc}^{enc}	n_{hid}^{enc}	n_{fc}^{dec}	n_{hid}^{dec}	lr	bs
Default						
$IHDP_A$	4	500	4	500	1e-4	1024
$IHDP_B$	4	500	4	500	1e-3	1024
JOBS	4	500	4	500	1e-3	256
Optimal						
$IHDP_A$ (Random)	8	496	6	448	1e-4	1024
$IHDP_B$ (Bayesian)	5	272	3	112	1e-3	1024
JOBS (Bayesian)	3	416	5	128	1e-3	256

Table 7.22. Default and optimal parameters for GANITE

	n^g_{fc}	n^g_{hid}	n^d_{fc}	n^g_{hid}	$n^i_{fc_0}$	$n^i_{hid_0}$	$n^i_{fc_1}$	$n^i_{hid_1}$	bs_g	bs_i	lr_g	lr_i
Default												
$IHDP_A$	5	8	5	5	3	200	3	200	64	64	1e-3	1e-3
$IHDP_B$	5	8	5	5	3	200	3	200	64	64	1e-3	1e-3
JOBS	3	4	3	4	2	100	2	100	128	128	1e-3	1e-3
Optimal												
$IHDP_A$ (Hyperband)	4	28	3	48	3	200	3	200	64	512	1e-3	1e-4
$IHDP_B$ (Hyperband)	9	448	9	160	3	112	2	464	256	256	1e-3	1e-3
JOBS (Bayesian)	9	448	9	160	4	300	4	300	64	64	1e-3	1e-3

Bibliography

[1] Andrei Sirazitdinov, Marcus Buchwald, Vincent Heuveline, and Jürgen Hesser. Graph Neural Networks for Individual Treatment Effect Estimation. *IEEE Access*, 12:106884–106894, 2024.

[2] Marco Scutari. Learning Bayesian Networks with the bnlearn R Package. *Journal of Statistical Software*, 35(3):1–22, 2010.

[3] Andrei Sirazitdinov, Marcus Buchwald, Jürgen Hesser, and Vincent Heuveline. Review of Deep Learning Methods for Individual Treatment Effect Estimation with Automatic Hyperparameter Optimization, December 2022.

[4] Uri Shalit. Can we learn individual-level treatment policies from clinical data? *Biostatistics*, page kxz043, November 2019.

[5] M. A. Hernán. A definition of causal effect for epidemiological research. *Journal of Epidemiology & Community Health*, 58(4):265–271, April 2004.

[6] Guido W. Imbens and Donald B. Rubin. *Causal Inference for Statistics, Social, and Biomedical Sciences*. Cambridge University Press, 2015.

[7] Donald B. Rubin. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of Educational Psychology*, 66(5):688–701, 1974.

[8] Paul R. Rosenbaum. Observational Studies. In *Observational Studies*, pages 1–17. Springer New York, New York, NY, 2002.

[9] Ronald A. Fisher. *The Design of Experiments*. Oliver and Boyd, 1935.

[10] E. Beiner, D. Baumeister, D. Buhai, M. Löffler, A. Löffler, and others. The PerPAIN trial: a pilot randomized controlled trial of personalized treatment allocation for chronic musculoskeletal pain – a protocol. *Pilot and Feasibility Studies*, 8(1):251, December 2022.

[11] James F Burke and Jeremy B Sussman. Statistical Analysis of Subgroup Effects in Randomized Trials. *Clinical Trials*, 12(5):477–485, 2015.

[12] Sander Greenland and James M Robins. Confounding and Exposure Trends in Case-Control Studies. *Epidemiology*, 10(1):37–48, 1999.

[13] Paul R. Rosenbaum and Donald B. Rubin. The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1):41–55, April 1983.

[14] James M Robins, Miguel A Hernan, and Babette Brumback. Marginal Structural Models and Causal Inference in Epidemiology. *Epidemiology*, 11(5):550–560, 2000.

[15] Zongyu Li, Xiaobo Guo, and Siwei Qiang. A survey of deep causal models and their industrial applications. *Artificial Intelligence Review*, 57(11):298, September 2024.

[16] Benjamin Freedman. Equipoise and the Ethics of Clinical Research. *New England Journal of Medicine*, 317(3):141–145, 1987.

[17] A. D. Nichol, M. Bailey, and D. J. Cooper. Challenging issues in randomised controlled trials. *Injury*, 41:20–23, 2010.

[18] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer, 2nd edition, 2009.

[19] Jennifer L. Hill. Bayesian Nonparametric Modeling for Causal Inference. *Journal of Computational and Graphical Statistics*, 20(1):217–240, January 2011.

[20] Roderick J A Little and Donald B Rubin. *Statistical Analysis with Missing Data*. Wiley, 3rd edition, 2019.

[21] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The Graph Neural Network Model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2009.

[22] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *AI Open*, 1:57–81, January 2020.

[23] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. A Comprehensive Survey on Graph Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1):4–24, January 2021.

[24] Uri Shalit, Fredrik D. Johansson, and David Sontag. Estimating individual treatment effect: generalization bounds and algorithms. In *Proceedings of the 34th International Conference on Machine Learning*, pages 3076–3085, 2017.

[25] Stefan Wager and Susan Athey. Estimation and Inference of Heterogeneous Treatment Effects using Random Forests. *Journal of the American Statistical Association*, 113(523):1228–1242, 2018.

[26] Judea Pearl. Causal inference in statistics: An overview. *Statist. Surv.*, 3(none), January 2009.

[27] Y Li and others. Graph Neural Networks for Causal Inference. *arXiv preprint arXiv:2010.09876*, 2020.

[28] Matej Zečević, Devendra Singh Dhami, Petar Veličković, and Kristian Kersting. Relating Graph Neural Networks to Structural Causal Models, October 2021. arXiv:2109.04173 [cs, stat].

[29] Judea Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, 2nd edition, 2009.

[30] Jacob Cohen. *Statistical Power Analysis for the Behavioral Sciences*. Lawrence Erlbaum Associates, 2nd edition, 1988.

[31] Paul W. Holland. Statistics and Causal Inference. *Journal of the American Statistical Association*, 81(396):945–960, December 1986.

[32] Martin Bland. *An Introduction to Medical Statistics*. Oxford University Press, 4th edition, 2015.

[33] Austin Bradford Hill. The Environment and Disease: Association or Causation? *Proceedings of the Royal Society of Medicine*, 58(5):295–300, 1965.

[34] Joshua D. Angrist and Alan B. Krueger. Does Compulsory School Attendance Affect Schooling and Earnings? *The Quarterly Journal of Economics*, 106(4):979–1014, 1991.

[35] Miguel A Hernan and James M Robins. Causal Inference: What If (the book), October 2012.

[36] Robert J. Sampson. Great American City: Chicago and the Enduring Neighborhood Effect. *University of Chicago Press*, 2012.

[37] Douglas G Altman and J Martin Bland. Statistics Notes: The Use of Correlation and Regression. *British Medical Journal*, 310(6988):1120, 1995.

[38] Tyler J Vander Weele and Miguel A Hernan. Causal Inference Under Multiple Versions of Treatment. *Journal of Causal Inference*, 1(1):1–20, 2013.

[39] Paul R Rosenbaum. *Design of Observational Studies*. Springer, 2010.

[40] Joshua D Angrist, Guido W Imbens, and Donald B Rubin. Identification of Causal Effects Using Instrumental Variables. *Journal of the American Statistical Association*, 91(434):444–455, 1996.

[41] Donald L. Thistlethwaite and Donald T. Campbell. Regression-Discontinuity Analysis: An Alternative to the Ex Post Facto Experiment. *Journal of Educational Psychology*, 51(6):309–317, 1960.

[42] David Card and Alan B. Krueger. Minimum Wages and Employment: A Case Study of the Fast-Food Industry in New Jersey and Pennsylvania. *The American Economic Review*, 84(4):772–793, 1994.

[43] Christos Louizos, Uri Shalit, Joris Mooij, David Sontag, Richard Zemel, and Max Welling. Causal Effect Inference with Deep Latent-Variable Models. *arXiv:1705.08821 [cs, stat]*, November 2017. arXiv: 1705.08821.

[44] Hidetoshi Shimodaira. Improving Predictive Inference Under Covariate Shift by Weighting the Log-Likelihood. *Journal of Statistical Planning and Inference*, 90(2):227–244, 2000.

[45] Yaroslav Ganin, Evgeniya Ustinova, and others. Domain-Adversarial Training of Neural Networks. *Journal of Machine Learning Research*, 17(59):1–35, 2016.

[46] Karl Weiss, Taghi M. Khoshgoftaar, and DingDing Wang. A Survey of Transfer Learning. *Journal of Big Data*, 3(1):9, 2016.

[47] Susan Athey and Guido W Imbens. State of the Art in Causal Inference. *Econometric Reviews*, 36(1-3):1–5, 2017.

[48] Fredrik Johansson, Uri Shalit, and David Sontag. Learning representations for counterfactual inference. In *International conference on machine learning*, pages 3020–3029. PMLR, 2016.

[49] Yao Zhang, Alexis Bellot, and Mihaela van der Schaar. Learning Overlapping Representations for the Estimation of Individualized Treatment Effects. *arXiv:2001.04754 [cs, stat]*, February 2020. arXiv: 2001.04754.

[50] Andrew Gelman and Jennifer Hill. *Data Analysis Using Regression and Multilevel/Hierarchical Models*. Cambridge University Press, 2006.

[51] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

[52] Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual Explanations Without Opening the Black Box. *Harvard Journal of Law & Technology*, 31(2), 2017.

[53] Scott M Lundberg and Su-In Lee. A Unified Approach to Interpreting Model Predictions. *Advances in Neural Information Processing Systems*, 30:4765–4774, 2017.

[54] Matt J Kusner, Joshua Loftus, Chris Russell, and Ricardo Silva. Counterfactual Fairness. *Advances in Neural Information Processing Systems*, 30:4066–4076, 2017.

[55] Sahil Verma and Julia Rubin. Fairness Definitions Explained. *2018 IEEE/ACM International Workshop on Software Fairness*, pages 1–7, 2018.

[56] Donald B Rubin. Causal Inference Using Potential Outcomes: Design, Modeling, Decisions. *Journal of the American Statistical Association*, 100(469):322–331, March 2005.

[57] Judea Pearl. On the Consistency Rule in Causal Inference: Axiom, Definition, Assumption, or Theorem? *Epidemiology*, 21(6):872–875, November 2010.

[58] Daniel Westreich and Stephen R. Cole. Invited Commentary: Positivity in Practice. *American Journal of Epidemiology*, 171(6):674–677, February 2010.

[59] Jeroen Hoogland and others. A tutorial on individualized treatment effect prediction from randomized trials with a binary endpoint. *Statistics in Medicine*, 40(26):5961–5981, 2021.

[60] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

[61] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Networks. *arXiv:1406.2661 [cs, stat]*, June 2014. arXiv: 1406.2661.

[62] Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. *arXiv:1312.6114 [cs, stat]*, May 2014. arXiv: 1312.6114.

[63] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, and others. Attention is All You Need. *Advances in Neural Information Processing Systems*, 30:5998–6008, 2017.

[64] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, and others. Graph Attention Networks, February 2018.

[65] Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A Kernel Two-Sample Test. *Journal of Machine Learning Research*, 13(25):723–773, 2012.

[66] Cédric Villani. *Optimal transport: old and new*. Number 338 in Grundlehren der mathematischen Wissenschaften. Springer, Berlin, 2009.

[67] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.

[68] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.

[69] Peter Spirtes, Clark Glymour, and Richard Scheines. *Causation, Prediction, and Search*. MIT Press, 2nd edition, 2000.

[70] Gideon Schwarz. Estimating the Dimension of a Model. *The Annals of Statistics*, 6(2):461–464, 1978.

[71] Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. *Elements of Causal Inference: Foundations and Learning Algorithms*. MIT Press, 2017.

[72] Thomas N. Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks, February 2017. arXiv:1609.02907 [cs, stat].

[73] Clark Glymour, Kun Zhang, and Peter Spirtes. Review of Causal Discovery Methods Based on Graphical Models. *Frontiers in Genetics*, 10, 2019.

[74] Zhipu Xie, Weifeng Lv, Shangfo Huang, Zhilong Lu, and others. Sequential Graph Neural Network for Urban Road Traffic Speed Prediction. *IEEE Access*, 8:63349–63358, 2020.

[75] Austin Derrow-Pinion, Jennifer She, David Wong, Oliver Lange, Todd Hester, Luis Perez, Marc Nunkesser, Seongjae Lee, Xueying Guo, Brett Wiltshire, Peter W. Battaglia, Vishal Gupta, Ang Li, Zhongwen Xu, Alvaro Sanchez-Gonzalez, Yujia Li, and Petar Velickovic. ETA Prediction with Graph Neural Networks in Google Maps. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, CIKM '21, pages 3767–3776, New York, NY, USA, October 2021. Association for Computing Machinery.

[76] Danfei Xu, Yuke Zhu, Christopher Choy, and Li Fei-Fei. Scene Graph Generation by Iterative Message Passing. In *Computer Vision and Pattern Recognition (CVPR)*, 2017.

[77] Junheng Hao, Tong Zhao, Jin Li, Xin Luna Dong, Christos Faloutsos, Yizhou Sun, and Wei Wang. P-Companion: A Principled Framework for Diversified Complementary Product Recommendation. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 2517–2524, Virtual Event Ireland, October 2020. ACM.

[78] Dejun Jiang, Zhenxing Wu, Chang-Yu Hsieh, Guangyong Chen, Ben Liao, Zhe Wang, Chao Shen, Dongsheng Cao, Jian Wu, and Tingjun Hou. Could graph neural networks learn better molecular representation for drug discovery? A comparison study of descriptor-based and graph-based models. *J Cheminform*, 13(1):12, December 2021.

[79] William L Hamilton, Rex Ying, and Jure Leskovec. Inductive Representation Learning on Large Graphs. *Advances in Neural Information Processing Systems*, 30:1024–1034, 2017.

[80] Nurul A Asif and others. Graph neural network: A comprehensive review on non-euclidean space. *IEEE Access*, 9:60588–60606, 2021.

[81] Cheng-Te Li, Yu-Che Tsai, Chih-Yao Chen, and Jay Chiehen Liao. Graph Neural Networks for Tabular Data Learning: A Survey with Taxonomy and Directions, January 2024.

[82] Panyu Zhai, Yanwu Yang, and Chunjie Zhang. Causality-based CTR prediction using graph neural networks. *Information Processing & Management*, 60(1):103137, January 2023.

[83] S. Wein, W. M. Malloni, A. M. Tomé, S. M. Frank, G.-I. Henze, S. Wüst, M. W. Greenlee, and E. W. Lang. A graph neural network framework for causal inference in brain networks. *Sci Rep*, 11(1):8061, April 2021.

[84] Zhixuan Chu, Stephen L. Rathbun, and Sheng Li. Graph Infomax Adversarial Learning for Treatment Effect Estimation with Networked Observational Data. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 176–184, Virtual Event Singapore, August 2021. ACM.

[85] Alvaro Parafita and Jordi Vitria. Estimand-Agnostic Causal Query Estimation With Deep Causal Graphs. *IEEE Access*, 10:71370–71386, 2022.

[86] Shohei Shimizu, Patrik O. Hoyer, Aapo Hyvärinen, and Antti Kerminen. A Linear Non-Gaussian Acyclic Model for Causal Discovery. *Journal of Machine Learning Research*, 7(72):2003–2030, 2006.

[87] David Maxwell Chickering. Optimal structure identification with greedy search. *J. Mach. Learn. Res.*, 3(null):507–554, March 2003.

[88] James Robins. A New Approach to Causal Inference in Mortality Studies with a Sustained Exposure Period—Application to Control of the Healthy Worker Survivor Effect. *Mathematical Modelling*, 7(9-12):1393–1512, 1986.

[89] Elizabeth A Stuart. Matching Methods for Causal Inference: A Review and a Look Forward. *Statistical Science*, 25(1):1–21, 2010.

[90] Stefano M Iacus, Gary King, and Giuseppe Porro. Causal Inference Without Balance Checking: Coarsened Exact Matching. *Political Analysis*, 20(1):1–24, 2012.

[91] Daniel Westreich, Justin Lessler, and Michele Jonsson Funk. Propensity Score Estimation: Machine Learning and Classification Methods as Alternatives to Logistic Regression. *Journal of Clinical Epidemiology*, 63(8):826–833, 2010.

[92] Leo Breiman. Random forests. *Machine learning*, 45:5–32, 2001.

[93] Sören R. Künzel, Jasjeet S. Sekhon, Peter J. Bickel, and Bin Yu. Metalearners for estimating heterogeneous treatment effects using machine learning. *Proc. Natl. Acad. Sci. U.S.A.*, 116(10):4156–4165, March 2019.

[94] Susan Athey and Stefan Wager. Estimating Treatment Effects with Causal Forests: An Application. *arXiv:1902.07409 [stat]*, February 2019. arXiv: 1902.07409.

[95] Hugh A. Chipman, Edward I. George, and Robert E. McCulloch. BART: Bayesian additive regression trees. *Ann. Appl. Stat.*, 4(1):266–298, March 2010. arXiv: 0806.3286.

[96] Ahmed M. Alaa, Michael Weisz, and Mihaela van der Schaar. Deep Counterfactual Networks with Propensity-Dropout. *arXiv:1706.05966 [cs, stat]*, June 2017. arXiv: 1706.05966.

[97] Liuyi Yao, Sheng Li, Yaliang Li, Mengdi Huai, Jing Gao, and Aidong Zhang. Representation learning for treatment effect estimation from observational data. *Advances in neural information processing systems*, 31, 2018.

[98] Liuyi Yao, Sheng Li, Yaliang Li, Mengdi Huai, Jing Gao, and Aidong Zhang. ACE: Adaptively Similarity-Preserved Representation Learning for Individual Treatment Effect Estimation. In *2019 IEEE International Conference on Data Mining (ICDM)*, pages 1432–1437, 2019.

[99] Claudia Shi, David M. Blei, and Victor Veitch. Adapting Neural Networks for the Estimation of Treatment Effects. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. Curran Associates Inc., Red Hook, NY, USA, 2019.

[100] Serge Assaad, Shuxi Zeng, Chenyang Tao, Shounak Datta, Nikhil Mehta, Ricardo Henao, Fan Li, and Lawrence Carin Duke. Counterfactual Representation Learning with Balancing Weights. In Arindam Banerjee and Kenji Fukumizu, editors, *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pages 1972–1980. PMLR, April 2021.

[101] Fredrik D. Johansson, Uri Shalit, Nathan Kallus, and David Sontag. Generalization Bounds and Representation Learning for Estimation of Potential Outcomes and Causal Effects. *arXiv:2001.07426 [cs, stat]*, February 2022. arXiv: 2001.07426.

[102] Xinshu Li and Lina Yao. Contrastive individual treatment effects estimation. In *2022 IEEE International Conference on Data Mining (ICDM)*, pages 1053–1058. IEEE, 2022.

[103] Jinsung Yoon, James Jordon, and Mihaela van der Schaar. GANITE: Estimation of Individualized Treatment Effects using Generative Adversarial Nets. In *International Conference on Learning Representations*, 2018.

[104] Xin Du, Lei Sun, Wouter Duivesteijn, Alexander Nikolaev, and Mykola Pechenizkiy. Adversarial balancing-based representation learning for causal effect inference with observational data. *Data Min Knowl Disc*, 35(4):1713–1738, July 2021.

[105] Zhenyu Guo, Shuai Zheng, Zhizhe Liu, Kun Yan, and Zhenfeng Zhu. Cetransformer: Casual effect estimation via transformer based representation learning. In *Pattern Recognition and Computer Vision: 4th Chinese Conference, PRCV 2021, Beijing, China, October 29–November 1, 2021, Proceedings, Part IV 4*, pages 524–535. Springer, 2021.

[106] Guanglin Zhou, Lina Yao, Xiwei Xu, Chen Wang, and Liming Zhu. Cycle-balanced representation learning for counterfactual inference. In *Proceedings of the 2022 SIAM International Conference on Data Mining (SDM)*, pages 442–450. SIAM, 2022.

[107] Weijia Zhang, Lin Liu, and Jiuyong Li. Treatment effect estimation with disentangled latent factors. In *Proceedings of the Thirty-fifth AAAI Conference on Artificial Intelligence (AAAI'21)*, 2021.

[108] Anpeng Wu, Junkun Yuan, Kun Kuang, Bo Li, Runze Wu, Qiang Zhu, Yue Ting Zhuang, and Fei Wu. Learning Decomposed Representations for Treatment Effect Estimation. *IEEE Transactions on Knowledge and Data Engineering*, page 1, February 2022.

[109] Tobias Hatt and Stefan Feuerriegel. Estimating average treatment effects via orthogonal regularization. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 680–689, 2021.

[110] Xinkun Nie and Stefan Wager. Quasi-Oracle Estimation of Heterogeneous Treatment Effects. *arXiv:1712.04912 [econ, math, stat]*, August 2020. arXiv: 1712.04912.

[111] Gino Tesei, Stefanos Giampanis, Jingpu Shi, and Beau Norgeot. Learning end-to-end patient representations through self-supervised covariate balancing for causal treatment effect estimation. *Journal of Biomedical Informatics*, 140:104339, 2023.

[112] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Networks. *Commun. ACM*, 63(11):139–144, October 2020. Place: New York, NY, USA Publisher: Association for Computing Machinery.

[113] Negar Hassanpour and Russell Greiner. Learning Disentangled Representations for CounterFactual Regression. In *International Conference on Learning Representations*, 2020.

[114] Mathias Niepert, Mohamed Ahmed, and Konstantin Kutzkov. Learning convolutional neural networks for graphs. In *International conference on machine learning*, pages 2014–2023. PMLR, 2016.

[115] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and others. Neural message passing for quantum chemistry. In *International conference on machine learning*, pages 1263–1272. PMLR, 2017.

[116] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How Powerful are Graph Neural Networks? *International Conference on Learning Representations*, 2018.

[117] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[118] Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. Gated graph sequence neural networks. *arXiv preprint arXiv:1511.05493*, 2015.

[119] Peter W. Battaglia, Jessica B. Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, Caglar Gulcehre, Francis Song, Andrew Ballard, Justin Gilmer, George Dahl, Ashish Vaswani, Kelsey Allen, Charles Nash, Victoria Langston, Chris Dyer, Nicolas Heess, Daan Wierstra, Pushmeet Kohli, Matt Botvinick, Oriol Vinyals, Yujia Li, and Razvan Pascanu. Relational inductive biases, deep learning, and graph networks, October 2018. arXiv:1806.01261 [cs, stat].

[120] Michael M Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv preprint arXiv:2104.13478*, 2021.

[121] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. Rectifier Nonlinearities Improve Neural Network Acoustic Models. *ICML Workshop on Deep Learning for Audio, Speech, and Language Processing*, 2013.

[122] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs). *arXiv preprint arXiv:1511.07289*, 2015.

[123] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2015.

[124] Tom O'Malley, Elie Bursztein, James Long, François Chollet, Haifeng Jin, Luca Invernizzi, and others. KerasTuner, 2019.

[125] J. Brooks-Gunn, F. R. Liaw, and P. K. Klebanov. Effects of early intervention on cognitive function of low birth weight preterm infants. *J Pediatr*, 120(3):350–359, March 1992.

[126] Rajeev H. Dehejia and Sadek Wahba. Causal Effects in Nonexperimental Studies: Reevaluating the Evaluation of Training Programs. *Journal of the American Statistical Association*, 94(448):1053–1062, December 1999.

[127] John Binder, Daphne Koller, Stuart Russell, and Keiji Kanazawa. Adaptive Probabilistic Networks with Hidden Variables. *Machine Learning*, 29(2):213–244, November 1997.

[128] J Tesarz, GH Seidler, and W Eich. Treatment of Pain with EMDR. *Stuttgart: Klett-Cotta*, 2015.

[129] Jonas Tesarz, Sabine Leisner, Andreas Gerhardt, Susanne Janke, Günter H Seidler, Wolfgang Eich, and Mechthild Hartmann. Effects of eye movement desensitization and reprocessing (EMDR) treatment in chronic pain patients: a systematic review. *Pain Medicine*, 15(2):247–263, 2014.

[130] Andreas Gerhardt, Sabine Leisner, Mechthild Hartmann, Susanne Janke, Günter H Seidler, Wolfgang Eich, and Jonas Tesarz. Eye movement desensitization and reprocessing vs. treatment-as-usual for non-specific chronic back pain patients with psychological trauma: A randomized controlled pilot study. *Frontiers in Psychiatry*, 7:201, 2016.

[131] Annette Löffler, Martin Löffler, Carolin Schütz, Josepha Zimmer, and Herta Flor. Therapiemanual für das Schmerz-Extinktions-Retraining [Therapy Manual for a pain extinction training]. *OSF*, 2023.

[132] Leonie Ader, Anita Schick, Martin Löffler, Annette Löffler, Eva Beiner, Wolfgang Eich, Stephanie Vock, Andrei Sirazitdinov, Christopher Malone, Jürgen Hesser, and others. Refocusing of attention on positive events using monitoring-based feedback and microinterventions for patients with chronic musculoskeletal pain in the PerPAIN randomized controlled trial: Protocol for a microrandomized trial. *JMIR Research Protocols*, 12(1):e43376, 2023.

[133] Robert Kerns, Dennis Turk, and Thomas Rudy. The West Haven-Yale multidimensional pain inventory (WHYMPI). *Pain*, 23:345–356, January 1986.

[134] Thomas Rudy, Dennis Turk, Hussein Zaki, and Hugh Curtin. An empirical taxometric alternative to traditional classification of temporomandibular disorder. *Pain*, 36:311–320, March 1989.

[135] Jacob Benesty, Jingdong Chen, Yiteng Huang, and Israel Cohen. Pearson Correlation Coefficient. In *Noise Reduction in Speech Processing*, pages 1–4. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[136] Michael Von Korff, Johan Ormel, Francis J. Keefe, and Samuel F. Dworkin. Chronic Pain Grade Questionnaire, April 2012.

[137] David P. Bernstein, Laura Fink, Leonard Handelsman, and Jeffrey Foote. Childhood Trauma Questionnaire, September 2011.

[138] Eva M. Klein, Elmar Brähler, Michael Dreier, Leonard Reinecke, Kai W. Müller, Gabriele Schmutzer, Klaus Wölfling, and Manfred E. Beutel. The German version of the Perceived Stress Scale – psychometric characteristics in a representative German community sample. *BMC Psychiatry*, 16(1):159, December 2016.

[139] Klaus Linde, Barbara Riedl, Stefanie Kehrer, Antonius Schneider, Bernd Löwe, and Andreas Toussaint. Der Fragebogen SSD-12 zur Erfassung der psychischen Belastung bei somatischer Belastungsstörung – erste Erfahrungen und Validierung bei Hausarztpatienten. In *51. Kongress der Deutschen Gesellschaft für Allgemeinmedizin und Familienmedizin (DEGAM), Düsseldorf, 21.-23.09.2017*, page Doc17degam143. German Medical Science GMS Publishing House, 2017.

[140] Per Fink, Henrik Ewald, Jörgen Jensen, Lisbeth Sörensen, Marianne Engberg, Martin Holm, and Povl Munk-Jörgensen. Whiteley-7 Scale, April 2013.

[141] J. L. Hodges. The significance probability of the smirnov two-sample test. *Arkiv för Matematik*, 3(5):469 – 486, 1958.

[142] Joshua Durso-Finley, Jean-Pierre Falet, Raghav Mehta, Douglas L. Arnold, Nick Pawlowski, and Tal Arbel. Improving Image-Based Precision Medicine with Uncertainty-Aware Causal Models. In Hayit Greenspan, Anant Madabhushi, Parvin Mousavi, Septimiu Salcudean, James Duncan, Tanveer Syeda-Mahmood, and Russell Taylor, editors, *Medical Image Computing and Computer Assisted Intervention – MICCAI 2023*, pages 472–481, Cham, 2023. Springer Nature Switzerland.

[143] Matthias Feurer and Frank Hutter. Hyperparameter Optimization. In Frank Hutter, Lars Kotthoff, and Joaquin Vanschoren, editors, *Automated Machine Learning: Methods, Systems, Challenges*, The Springer Series on Challenges in Machine Learning, pages 3–33. Springer International Publishing, Cham, 2019.

[144] Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. Hyperband: a novel bandit-based approach to hyperparameter optimization. *J. Mach. Learn. Res.*, 18(1):6765–6816, January 2017.

[145] James Bergstra and Yoshua Bengio. Random Search for Hyper-Parameter Optimization. *Journal of Machine Learning Research*, 13:281–305, 2012.

[146] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical Bayesian Optimization of Machine Learning Algorithms. *Advances in Neural Information Processing Systems*, 25:2951–2959, 2012.

[147] Peter Robinson. Root- N-Consistent Semiparametric Regression. *Econometrica*, 56(4):931–54, 1988.

[148] Jean Kaddour, Yuchen Zhu, Qi Liu, Matt J. Kusner, and Ricardo Silva. Causal Effect Inference for Structured Treatments, October 2021.

[149] Daniel Jiwoong Im, Kyunghyun Cho, and Narges Razavian. Causal Effect Variational Autoencoder with Uniform Treatment. *arXiv:2111.08656 [cs]*, November 2021. arXiv: 2111.08656.

[150] Sebastian Ruder. An overview of gradient descent optimization algorithms, June 2017. arXiv:1609.04747 [cs].

[151] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization, January 2017. arXiv:1412.6980 [cs].

[152] Takashi Ikeuchi, Mayumi Ide, Yan Zeng, Takashi Nicholas Maeda, and Shohei Shimizu. Python package for causal discovery based on LiNGAM. *Journal of Machine Learning Research*, 24(14):1–8, 2023.

[153] Devansh Arpit and others. Salesforce CausalAI Library: A Fast and Scalable Framework for Causal Analysis of Time Series and Tabular Data, 2023.

[154] Frank J Massey Jr. The kolmogorov-smirnov test for goodness of fit. *Journal of the American statistical Association*, 46(253):68–78, 1951.

[155] Roger Pros and Jordi Vitrià. Neural Networks with Causal Graph Constraints: A New Approach for Treatment Effects Estimation, April 2024.

[156] George Panagopoulos, Daniele Malitesta, Fragkiskos D Malliaros, and Jun Pang. Graph Neural Networks for Treatment Effect Prediction. *arXiv e-prints*, pages arXiv–2403, 2024.

[157] Bradley Efron and Robert J Tibshirani. *An introduction to the bootstrap*. Chapman and Hall/CRC, 1994.

[158] Larry Wasserman. *All of nonparametric statistics*. Springer Science & Business Media, 2006.

[159] A. Gelman, J.B. Carlin, H.S. Stern, D.B. Dunson, A. Vehtari, and D.B. Rubin. *Bayesian Data Analysis, Third Edition*. Chapman & Hall/CRC Texts in Statistical Science. Taylor & Francis, 2013.

Acknowledgements

I am deeply grateful to the many individuals whose support, guidance, and inspiration have made this dissertation possible. Their contributions have been invaluable in shaping this work and my journey as a researcher. First and foremost, I extend my heartfelt thanks to my parents, whose unwavering love, support, and belief in me have been the bedrock of my academic and personal growth. Your sacrifices and encouragement have driven me to face challenges and pursue my goals with determination. To my wife, your boundless love, patience, and understanding have been my cornerstone. Your steadfast support and ability to bring balance to my life have made this journey not only achievable but deeply meaningful. Thank you for being my partner in every way. To my sister, I owe a special debt of gratitude for inspiring the very topic of this thesis. Your insights and enthusiasm sparked my interest in this field, and your positivity and companionship have been a constant source of motivation and joy throughout this process.

I am immensely grateful to my supervisor, Prof. Dr. Jürgen Hesser, for his mentorship, intellectual guidance, and trust in my abilities. Your expertise and thoughtful feedback have profoundly influenced this research and my development as a scientist. To my colleagues at the Ruprecht-Karls-Universität Heidelberg and the PerPain consortium, thank you for your collaboration, thought-provoking discussions, and camaraderie. Your diverse perspectives and willingness to share knowledge have enriched this work and made the research process truly rewarding. I also wish to thank my friends, who have provided laughter, encouragement, and grounding during this journey. Your presence has made the highs more joyful and the lows more manageable. Finally, I acknowledge the broader academic community, including peers and researchers whose work in causal inference and graph neural networks has inspired and informed this dissertation. Your contributions have laid a strong foundation for my own exploration. This thesis is a testament to the collective support of everyone mentioned and many others not named. Thank you all for being part of this significant chapter of my life.