CAVIN: A Community-Aware Variance-Weighted ITE Network for Uplift Modeling

Ruochen Fan [1] **Zheng Dong** [2]

Abstract

In the domain of online marketing, accurately estimating Individual Treatment Effects (ITE) is crucial for precise advertisement targeting and personalized content recommendation. Recent works combine Graph Neural Networks (GNNs) and meta-learners to estimate ITE while accounting for mutual influences. However, existing methods overlook the heteroscedasticity inherent in users' causal effects and consider only first-order neighbors, thereby failing to account for the community-level information within the network. Consequently, these methods face challenges in accurately estimating ITE in scenarios characterized by heteroscedasticity in user behavior, network interference, and community preferences. To address these challenges, we propose a novel community-aware variance-weighted network that integrates graph positional encoding into a GNN to capture community effects and employs variance-weighted learning to mitigate heteroscedastic noise. Evaluations on semi-synthetic datasets from CoraFull, DBLP, and PubMed show that our model significantly outperforms baseline methods that combine GNN with S-, T-, X-, and DR-learner, in accurately estimating ITEs.

1. Introduction

Accurate estimation of ITE and Conditional Average Treatment Effect (CATE) is fundamental for data-driven decision-making in domains ranging from personalized marketing (Parshakov et al., 2020) to public health interventions (Barkley et al., 2020). Traditional methods, including meta-learners such as S-learner (Künzel et al., 2019) and Doubly Robust (DR) learner (Kennedy, 2023), typically assume independent and identically distributed (i.i.d.) data. However, this assumption fails to account for the complex interdependencies inherent in real-world networked systems, such as social platforms and online marketing ecosystems, where user behaviors are influenced by both individual traits and networked inference (Forastiere et al., 2021; Zivich & Breskin, 2021). Recent advances in GNNs have enabled the integration of network structures into causal inference, addressing hidden confounders through neighbor feature aggregation. Despite these advancements, three major challenges remain: (1) *Overlooking positional encoding*, which inadequately disentangles the influence of high level community biases from individual interactions, resulting in biased outcomes; (2) *Uneven data variability*, where certain users exhibit highly erratic engagement patterns, significantly degrading data quality; and (3) *The flawed dataset assumption*, where most existing works rely on semi-synthetic datasets, whose generation process often fails to align with real-world complexities (Chen et al., 2024).

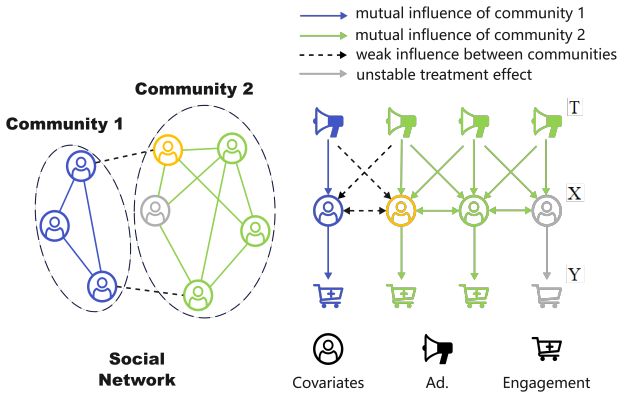

Figure 1. A diagram illustrating the mutual influence among users in a social network. Users in community 1 are shown in blue, and users in community 2 are shown in green. The orange user represents the target to be analyzed. Grey color indicates the user has a larger random error within the treatment outcome.

First, network structures play a pivotal role in shaping user behavior. In social networks, users within the same community often amplify treatment effects through peer influence,

[1]Department of Electrical and Electricity,Imperial College, London, United Kingdom [2]ByteDance, Beijing, China. Correspondence to: Ruochen Fan <ruochen.fan24@imperial.ac.uk>.

Proceedings of the 42^{nd} International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025.

a phenomenon observed in vaccination campaigns and marketing (Aral & Walker, 2011; Luo et al., 2021; de Valck et al., 2009). Existing GNN-based ITE estimators treat all neighbors uniformly, overlooking the hierarchical nature of community-driven interactions. Moreover, existing works only focus on the influence of first-order neighbors, making it even more difficult to model community influence (Sirazitdinov et al., 2024). Second, real-world outcomes frequently exhibit heteroscedasticity (Abrevaya & Xu, 2023). The behavior patterns of some users may become highly unstable due to measurement errors or anomalous factors, resulting in treatment outcomes that exhibit significant randomness. If such data is used for training, it can lead to overfitting and degrade the overall performance of the model. While variance-weighted methods (Shi et al., 2024; Mo & Liu, 2022) have been explored in traditional ITE estimation, their integration with graph-based approaches remains nascent. Third, because only one outcome, either under treatment or control, can be observed for each individual, most existing methods rely on semi-synthetic datasets (Huang et al., 2023; Guo et al., 2020; Kazemi & Ester, 2024). However, their assumptions are often unrealistic, as most works only consider the influence of first-order neighbors, and their data synthesis methods fail to incorporate the treatment effects of neighboring nodes, making it incapable to model real-world community influences.

To address these gaps, we propose CAVIN (Community-Aware Variance-Weighted ITE Network), a novel framework that synergizes community-aware graph learning with variance-weighted meta-learning. Our contributions are threefold: (1) *Community-Aware Positional Encoding (GPE)*: We develop a self-attention mechanism that integrates node features with their positional embeddings, effectively capturing both the nuanced local interactions and the global community influences stemming from node positions. We further construct a multi-layer Graph Attention Network (GAT) proposed by Velickovic et al. (2017) that adeptly aggregates and synthesizes information from multi-hop neighbors, thereby enhancing the model's ability to discern and leverage complex patterns within the network; (2) *Variance-Weighted X-Learning*: By estimating ITE variances and reweighting samples adaptively, CAVIN mitigates heteroscedastic noise. Theoretical analysis demonstrates that our weighting scheme achieves lower mean squared error (MSE) than unweighted X-learners under heteroskedasticity. (3) Enhanced Graph Meta-Learning Dataset: We extend the work of Guo et al. (2020) by improving multi-hop aggregation of neighbor features and treatment states. Specifically, we first detect potential communities by using Louvain algorithm (Blondel et al., 2008). Then we amplify network inference between users in same community. We also introduce community bias and individual heteroscedasticity into the dataset to better simulate real-world social structures and varying treatment responses.

Experiments on semi-synthetic datasets demonstrate CAVIN's superior performance over state-of-the-art baselines. Notably, on each dataset, CAVIN reduces rooted Precision in Estimation of Heterogeneous Effect (PEHE) by more than 29% compared to the best baseline. Moreover, while some baselines fail to converge under challenging conditions, CAVIN consistently achieves stable results. Our ablation studies further validate the necessity of GPE across all datasets, models with GPE outperform their counterparts without GPE, reinforcing the effectiveness of structured positional embeddings in ITE estimation.

2. Related Work

ITE estimation and uplift modeling, which mainly focus on ITE and CATE estimation respectively, have shown significant practical impact in both industrial and medical fields. Traditional uplift modeling approaches encompass tree and forest-based methods (Radcliffe & Surry, 2011; McCulloch et al., 2018; Athey & Imbens, 2016), Knapsack problem-based approaches (Goldenberg et al., 2020; Albert & Goldenberg, 2022), and meta-learners. Tree and forest-based methods recursively partition data to assign individuals into subgroups with similar treatment effects. The Knapsack problem-based approach transforms treatment effect estimation into a combinatorial optimization problem, seeking optimal matching under budget constraints. Meta-learners estimate outcomes through vectors.

Deep learning approaches for ITE estimation have gained increasing attention from researchers with the advancement of deep learning. Compared to traditional uplift modeling methods, deep uplift modeling demonstrates superior capability in fitting non-linear causal relationships, significantly enhancing expressive power. Representative works in this domain include BNN (Johansson et al., 2016a), TARNet, CFRNet (Shalit et al., 2017), SNet (Curth & Van der Schaar, 2021), TNet (Chen et al., 2024), and EFIN (Liu et al., 2023). These methods primarily adopt the meta-learner framework, utilizing deep learning models to estimate components defined in meta-learners, i.e., BNN builds upon the S-learner framework, while TARNet and CFRNet extend the T-learner approach, and TNet resembles the DR-learner structure. Due to inherent limitations, tree and forest-based methods and Knapsack problem-based approaches are less suitable for deep learning integration. Consequently, the current research mainstream focuses on constructing network architectures based on meta-learner principles for ITE estimation. However, in domains such as online marketing, samples often violate the i.i.d. assumption, making the investigation of network interference a critical research frontier.

Causal inference under network interference has become

a popular topic in recent years because traditional methods, which assume no interference between units (SUTVA), often fall short in real-world scenarios where units influence each other through networks. Early work by Hudgens & Halloran (2008) set the stage by introducing the idea of partial interference, where interference happens only within predefined clusters of individuals. This idea has since been expanded to handle more complex settings, including cases where interference depends on spatial or temporal relationships (Papadogeorgou et al., 2022). One of the biggest challenges in this area is figuring out how to define how the treatment of one unit affects others in the network. Ugander et al. (2013) tackled this by spliting networks into clusters to ensure balanced treatment assignment and mitigate interference. Recent advances have focused on leveraging machine learning to model complex interference patterns. Forastiere et al. (2021) developed a Bayesian framework for estimating causal effects in the presence of network interference, incorporating graph convolutional networks (GCNs) to capture network structure. Guo et al. (2020) proposed a deep learning-based approach to estimate casual effects by embedding network information into treatment effect models. However, these methods fail to account for real-world complexities, such as community preferences and heterogeneous effect variances. While there exist approaches for addressing heteroscedasticity, no methods currently integrate these with network interference, not to mention influence of node positions and community. Moreover, the dataset semi-synthesis methods overlook the influence of multi-hop neighbors and their treatments (Chen et al., 2024), which have shown significant influence on treatment effects.

To address these problems, we propose CAVIN, which uses the node2vec algorithm (Grover & Leskovec, 2016) to encode node positions in the graph and employs a self-attention mechanism to fuse node features with position encodings. The fused features are then embedded through a GAT. Based on the X-learner framework, we use a variance-weighted loss to backpropagate and update the parameters of the ITE estimation network. Both of our modifications show significant improvements over other methods on our dataset. Notably, to our knowledge, we are also the first to extend the X-learner to network inference, implement it as a baseline in this domain.

3. Preliminary

In this section, we introduce all the problem definitions, assumptions, and prerequisites used in our model and introduce our notations respectively in each subsection. In section 3.1, we introduce uplift modeling. In section 3.2, we extends the uplift modeling problem to graph data, defining uplift modeling in graph-structured settings. In section 3.3, we details the X-learner used in CAVIN.

3.1. Uplift Modeling

Mathematically, for each user i who characterized by a set of covariates $X_i \in \mathcal{X} \subset \mathbb{R}^d$, we define the observed data as $\mathcal{D} = \{(X_i, T_i, Y_i)\}_{i=1}^n$, where $T_i \in \{0, 1\}$ indicates whether the treatment was received. Specifically, $T_i = 1$ if the individual received the treatment, and $T_i = 0$ otherwise. Y_i represents the observed outcome of the i-th individual under the assigned treatment T_i. We denote $Y_i(1)$ as the potential outcome when the individual receives the treatment, and $Y_i(0)$ as the potential outcome when the individual does not receive the treatment. Then Y_i can be written as $Y_i = T_i \cdot Y_i(1) + (1 - T_i)Y_i(0)$.

Definition 3.1. (ITE). We define the ITE of user i as:

$$D_i = Y_i(1) - Y_i(0). \tag{1}$$

Notably, in most senarios, the individual outcome is either from treatment group or control group and can never observed both simultaneously, making it impossible to observe true ITE value by using equation 1 directly. Therefore, in order to select the optimal treatment, we need to estimate the true value of the ITE, which is noted as CATE.

Definition 3.2. (CATE). Mathematically:

$$\tau(X_i) = \mathbb{E}[Y_i(1) - Y_i(0) \mid X = X_i], \tag{2}$$

where $\tau(X_i)$ represents the estimated ITE value for user i.

The objective of uplift modeling is to find the best ITE estimator $\hat{\tau}$ within a hypothesis space $\mathcal{H}$ such that the MSE between $\hat{\tau}(X_i)$ and the true ITE D_i is minimized:

$$\arg\min_{\hat{\tau} \in \mathcal{H}} \mathbb{E}\left[(D_i - \hat{\tau}(X_i))^2 \mid X = X_i\right]. \tag{3}$$

Theorem 3.3. *Assuming that* $\mathbb{E}\left[D_i - \tau(X_i) \mid X_i = x\right] = 0$*, then the objective simplifies to minimizing the MSE between* $\hat{\tau}(X_i)$ *and the CATE* $\tau(X_i)$ *(proof in appendix section 3.3).*

$$\arg\min_{\hat{\tau} \in \mathcal{H}} \mathbb{E}\left[(\tau(X_i) - \hat{\tau}(X_i))^2 \mid X = X_i\right]. \tag{4}$$

Hence finding best ITE estimator and finding best CATE estimator in terms of minimizing MSE are equivalent.

3.2. Graph-Based ITE Estimation

First, we introduce the notations in our work. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ represent a graph with node set $\mathcal{V}$ and edge set $\mathcal{E}$. Each node i is characterized by a feature vector $X_i \in \mathbb{R}^d$, a treatment indicator $T_i \in \{0, 1\}$, and an observed outcome Y_i. Our dataset is defined as:

$$\mathcal{D} = \{(X_i, T_i, Y_i), \mathcal{A}\}_{i=1}^n, \tag{5}$$

where $\mathcal{A}$ denotes the adjacency matrix. The dataset adjacency matrix is undirected, with all edge weights set to 1. In

the assumptions of most related works (Ma et al., 2022; Cai et al., 2023; Chen et al., 2024), it is typically posited that individuals are only influenced by their first-order neighbors. Our work relaxes this assumption and introduces the concept of multi-hop neighbor influence, thereby modeling community bias relationships. We note the set of k-hop neighbors of i as $\mathcal{N}_i^k$. The influence of the features and treatment status of k-hop neighbors on an individual's outcome is denoted as $X_{\mathcal{N}_i^k}$ and $T_{\mathcal{N}_i^k}$, respectively. We make the following assumptions:

Assumption 3.4. (Network Consistency). The individual outcome depends not only on their own treatments but also on the influence of their neighbors. Mathematically: $Y_i = Y_i(T_i, \mathcal{T}_i)$, where $\mathcal{T}_i = \mathcal{T}_i(T_{\mathcal{N}_i^{1,\cdots r}})$ represents the treatments influence of multi-hop neighbor. Here, we denote $T_{\mathcal{N}_i^{1,\cdots r}}$ as $T^i_{\mathcal{N}}$, $X_{\mathcal{N}_i^{1,\cdots r}}$ as $X^i_{\mathcal{N}}$.

Assumption 3.5. (Independent Treatment Assignment). The treatment assignment for each user is an independent event. Therefore, we also have $T_i \perp \mathcal{T}_i$.

Assumption 3.6. (Network Unconfoundedness). After fixing X_i and its multi-hop neighbors' features $X^i_{\mathcal{N}}$, the treatment assignment of both the individual and their neighbors should be independent of the potential outcomes. Mathematically: $Y_i(T_i, \mathcal{T}_i) \perp (T_i, T^i_{\mathcal{N}}) | X_i, X^i_{\mathcal{N}}$.

Assumption 3.7. (Network Overlap). For any individual and their neighbors, every possible combination of treatments has a non-zero probability of occurring in the data. Mathematically: $0 < p(T_i, T^i_{\mathcal{N}} \mid X_i, X^i_{\mathcal{N}}) < 0$.

Assumption 3.8. (Neighborhood Interference). The influence of neighbors on an individual's outcome is mediated through an aggregation function agg, which defines a equivalence class. Mathematically: $\mathcal{T}_i \sim \mathcal{T}_i'$ (means $Y(T_i, \mathcal{T}_i) = Y(T_i, \mathcal{T}_i')$) if and only if $agg(\mathcal{T}_i) = agg(\mathcal{T}_i')$.

We extend the standard ITE estimation framework to incorporate the graph structure. In our work, the objective is to estimate the network ITE in the sense of minimizing the MSE, which equates to estimating the network CATE:

$$\arg\min_{\hat{\tau} \in \mathcal{H}} \mathbb{E}[(\tau - \hat{\tau})^2 \mid X = X_i, X_{\mathcal{N}} = X^i_{\mathcal{N}}], \tag{6}$$

$$\tau = \mathbb{E}[Y(1, \mathcal{T}_i) - Y(0, \mathcal{T}_i) \mid X = X_i, X_{\mathcal{N}} = X^i_{\mathcal{N}}]. \tag{7}$$

3.3. X-learner for Network ITE

The X-learner first estimates the outcomes of individuals under different treatments separately:

$$\mu_t = \mathbb{E}[Y(t, \mathcal{T}_i) \mid X = X_i, X_{\mathcal{N}} = X^i_{\mathcal{N}}] \tag{8}$$

$$= \mathbb{E}[Y \mid T = t, \mathcal{T} = \mathcal{T}_i, X = X_i, X_{\mathcal{N}} = X^i_{\mathcal{N}}], \tag{9}$$

where $t \in \{0, 1\}$. During optimization, the objective of μ_0 is to minimize the MSE on the control group samples, and vice versa. Next, it calculate conditional residual:

$$\mathrm{D}_1 = Y(1) - \mu_0(X, X_{\mathcal{N}}), \mathrm{D}_0 = \mu_1(X, X_{\mathcal{N}}) - Y(0) \tag{10}$$

We estimate D_1 by using $Y(1)$ and estimated outcome μ_0 when $Y(0)$ is missing, vice versa. In practice, two separate regression models, $g_1(X, X_{\mathcal{N}})$ and $g_0(X, X_{\mathcal{N}})$, are trained to predict D_1 and D_0. Here, we distinguish between D (the true ITE), $\mathcal{D}$ (the dataset), and D (the residual).

Assumption 3.9. (Conditional Independence of Residuals). There is no significant difference between conditional residual and ITE, ensuring that the ITE derived from the two distinct regressors is reasonable. Mathematically: $\mathbb{E}[Y(1, \mathcal{T}_i) - Y(0, \mathcal{T}_i) \mid X, X_{\mathcal{N}}] \approx \mathbb{E}[\mathrm{D} \mid X, X_{\mathcal{N}}]$. Here, D represents both D_1 and D_0.

In the final ITE estimation, the propensity score e is used as a weight. Formally:

$$\hat{\tau}(X, X_{\mathcal{N}}) = w g_1 + (1 - w) g_0, \tag{11}$$

where $e = p(T = 1 \mid \mathcal{T}, X, X_{\mathcal{N}})$, and $w = e/(1 - e)$.

Definition 3.10. (Criterions). Two widely used evaluation metrics, rooted PEHE and absolute error on ATE, are employed in our work. Their definitions are as follows:

$$\sqrt{\epsilon_{PEHE}} \triangleq \sqrt{\sum\nolimits_{i=1}^{n} (\tau_i - \hat{\tau}_i)^2 / n} \tag{12}$$

$$\epsilon_{ATE} \triangleq \left| \sum\nolimits_{i=1}^{n} (\tau_i - \hat{\tau}_i) / n \right|. \tag{13}$$

4. CAVIN

In this section, we present the key components of our framework for uplift modeling using GATs and variance-weighted meta-learners. In section 4.1, we introduce the integration of GATs which leveraging multi-head attention to fuse node features and positional encodings for enhanced representation learning. Section 4.2 details our proposed variance-weighted extension of the X-learner. We provide a theoretical analysis demonstrating the variance reduction achieved by our method. Finally, we outline the overall framework of CAVIN. The model structure is visualized in Figure 2. Our model is designed to train all components concurrently, eliminating the necessity for a staged or phased learning approach.

4.1. GATs in Uplift Modeling

4.1.1. Attention-Based Feature Fusion

In our framework, we enhance GAT by integrating node features $h \in \mathbb{R}^{n \times d_h}$ with positional encodings $p \in \mathbb{R}^{n \times d_p}$ using multi-head attention. Our Fusion module projects h to queries Q, and p to keys K and values V via linear layer:

$$Q = hW_Q,\ K = pW_K,\ V = pW_V. \tag{14}$$

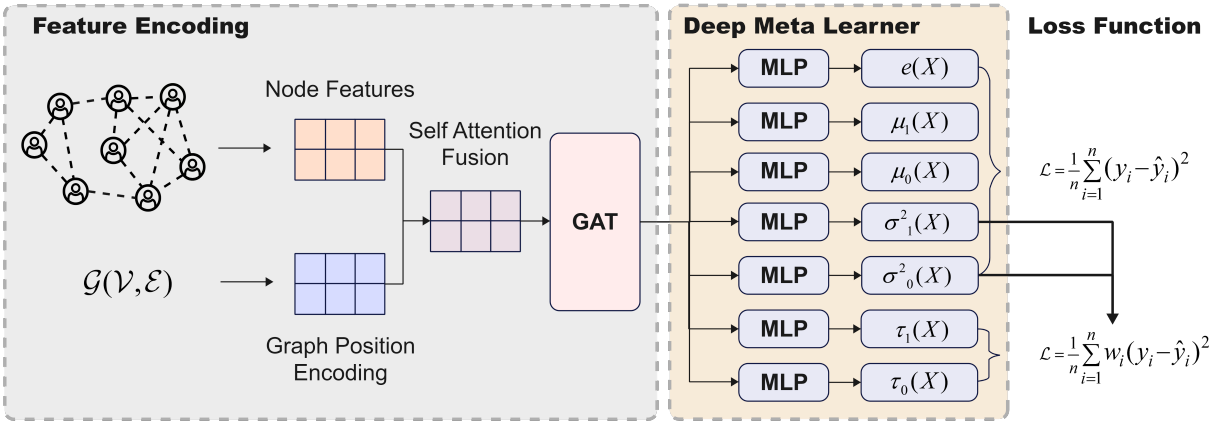

Figure 2. The graph structure represents the raw data $\mathcal{D} = \{(X_i, T_i, Y_i), \mathcal{A}\}_{i=1}^n$. $\mathcal{G}(\mathcal{V}, \mathcal{E})$ denotes the graph structure. We first use node2vec (Grover & Leskovec, 2016) to embed node positions, then apply multi-head self-attention to fuse node features and positional features as input to the GAT. Through multiple GAT layers, features are embedded into a low-dimensional space, capturing neighborhood influences. multilayer perceptrons (MLPs) are then used to output core estimations, with a weighted MSE loss for τ and standard MSE loss for other terms.

The i-th attention head outputs can be calculated by:

$$\text{head}_i(Q_i, K_i, V_i) = \text{softmax}\left(Q_i K_i^T / \sqrt{d_h}\right) V_i. \quad (15)$$

Finally, the multi-head attention outputs are:

$$\text{MHA}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_H) W_o, \quad (16)$$

where W_o is the learned linear projection matrix.

4.1.2. Graph Attention Networks

We use a GAT to extract node representations from the graph. Given an input graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, each node $v \in \mathcal{V}$ has an initial feature vector $h_v = X_v \in \mathbb{R}^d$. The GAT operates as follows:

Linear Projection: Each node's feature is linearly projected into a new space: $h'_v = W h_v, W \in \mathbb{R}^{d' \times d}$, where F' is the new feature dimension.

Attention Coefficients: For a node v and its first order neighbor $u \in \mathcal{N}(v)$, compute the unnormalized attention coefficient:

$$e_{vu} = \text{LeakyReLU}(a^T \cdot [h'_v || h'_u]), \quad (17)$$

where a is a learnable parameter vector and $[\cdot||\cdot]$ denotes concatenation.

Normalization: Normalize the attention coefficients using softmax:

$$\alpha_{vu} = \frac{\exp(e_{vu})}{\sum_{k \in \mathcal{N}(v)} \exp(e_{vk})}, \quad (18)$$

Feature Aggregation: Compute the new node representation by aggregating neighbors' features weighted by α_{vu}:

$$h_v^{\text{new}} = \sigma\left(\sum\nolimits_{u \in \mathcal{N}(v)} \alpha_{vu} h'_u\right), \quad (19)$$

where σ is Leaky ReLU function. In our implementation, we use multi-head attention to enhance the model's ability to capture diverse relationships between nodes. To be specific, we stack 5 layers of GAT to aggregate the influence of first 5-hop neighbors.

4.2. Variance-Weighted Meta Learners

We present a variance weighting mechanism based extension of the X-learner (Künzel et al., 2019), specifically designed to mitigate estimation error in heteroskedastic settings. We also provide theoretical analysis demonstrating how our approach reduces the MSE. We also include pseudocode to illustrate the implementation of our framework.

4.2.1. Variance-Weighted X-learner

In real-world scenarios, the true value of ITE can be expressed as $D_i = \tau(X_i, X_{\mathcal{N}}^i) + \varepsilon_i$, where ε_i is a zero-mean random noise of treatment effect. The heteroskedastic variance implies that for different individual, $\text{Var}[\varepsilon_i]$ also exhibits significant differences.

Assumption 4.1. We assume that σ and X_i are not independent, where the conditional variance of ε_i given X_i is defined as $\sigma^2(X_i) \triangleq \text{Var}[\varepsilon_i \mid X_i]$.

We propose a variance-weighted extension for meta-learners to eliminate the noise influence and lower the estimation MSE. For simplicity, we still denote the features represented by the GAT which already incorporated network influences

as $X \in \mathbb{R}^{n\times d}$. The conditional variance is:

$$\begin{aligned}\sigma^2(X_i) &= \mathrm{Var}[\varepsilon_i \mid X = X_i] = \mathrm{Var}[D \mid X = X_i] \\ &= \mathbb{E}[D^2 \mid X = X_i] - \mathbb{E}^2[D \mid X = X_i] \\ &= \mathbb{E}[D^2 \mid X = X_i] - \tau^2(X_i). \end{aligned} \tag{20}$$

Refer to section 3.2, we already got the estimation $\hat{\tau}$. We can proceed to optimize $\hat{\sigma}^2(X_i)$ by performing a regression of $\mathrm{D}^2 - \hat{\tau}^2$ on X_i, where D is the residual in Formula 10. Similar to the X-learner, if the sample belongs to control group, we then estimate it using D_0 and τ_0, and vice versa. The sample weights are defined as:

$$w_i = \frac{1}{\max\{\hat{\sigma}^2(X_i), \delta\}},\ \delta > 0, \tag{21}$$

where δ is a small constant to avoid extremely large weights. The variance-Weighted extension learner performs weighted regression:

$$\hat{\tau} = \arg\min_{\hat{\tau}} \sum_{i=1}^{n} w_i(\mathrm{D}_i - \hat{\tau}(X_i)). \tag{22}$$

Typically, we also employ cross-fitting: $\hat{\mu}$, $\hat{\tau}$, and $\hat{\sigma}^2(X_i)$ are optimized only on training dataset.

4.2.2. Theoretical Analysis and Proof Sketch

Here we show how our proposed variance-weighted learners achieves a lower MSE than the baseline learners under heteroskedasticity. Our method is based on the X-learner. Therefore, we compare the MSE of our model with that of the X-learner in this context. Define the MSE of any estimator $\hat{\tau}(X)$ as $\mathrm{MSE}(\hat{\tau}) = \mathbb{E}\left[(\tau(X) - \hat{\tau}(X))^2\right]$, where X is the neighbor influence embedded feature represented by GAT. Decomposing MSE into bias and variance, we then have:

$$\mathrm{MSE}(\hat{\tau}) = \underbrace{[\tau(X) - \mathbb{E}[\hat{\tau}(X)]]^2}_{\mathrm{Bias}^2} + \underbrace{\mathrm{Var}[\hat{\tau}(X)]}_{\mathrm{Variance}}. \tag{23}$$

Since our approach only introduces weighting in the loss function without modifying the structure of the X-learner, the bias remains unchanged. Therefore, the MSE improvement comes from a variance reduction. Assume the ITEs are given by $D_i = \tau(X_i) + \varepsilon_i$, where ε_i is a random noise. The weighted least squares used by our method minimizes $\sum_{i=1}^{n} w_i(\mathrm{D}_i - \hat{\tau}(X_i))^2$. Without loss of generality, assume τ as a linear model (same as linearize τ locally), our model can then be assumed to be $\mathrm{D} = X\beta + \varepsilon$. According to Gauss–Markov theory (Aitken, 1936),

$$\hat{\beta}_{\mathrm{OLS}} = (X^T X)^{-1} X^T \mathrm{D}, \tag{24}$$

$$\hat{\beta}_{\mathrm{WLS}} = (X^T W X)^{-1} X^T W \mathrm{D}, \tag{25}$$

denote the best odinary and weighted least square solution respectively, where W is diagonal matrix which elements are $\{1/\sigma^2(X_i)\}_{i=1}^n$. The covariance matrix of $\hat{\beta}_{\mathrm{OLS}}$ is:

$$\begin{aligned}\mathrm{Var}(\hat{\beta}_{\mathrm{OLS}}) &= \mathrm{Var}\left[(X^T X)^{-1} X^T (X\beta + \varepsilon)\right] \\ &= \mathrm{Var}\left[\beta + (X^T X)^{-1} X^T \varepsilon\right] \\ &= \mathrm{Var}\left[(X^T X)^{-1} X^T \varepsilon\right] \\ &= (X^T X)^{-1} X^T \Sigma X (X^T X) \end{aligned} \tag{26}$$

where Σ is the covariance matrix of ε, $\Sigma = W^{-1}$. Similarly:

$$\begin{aligned}\mathrm{Var}(\hat{\beta}_{\mathrm{WLS}}) &= (X^T W X)^{-1} X^T W \Sigma W X (X^T W X) \\ &= (X^T \Sigma^{-1} X)^{-1} \end{aligned} \tag{27}$$

When performing WLS regression, the heteroskedasticity issue is transformed into a scenario with homogeneous noise variances. Therefore, by the Gauss–Markov theorem, the WLS estimator becomes the minimum variance Best Linear Unbiased Estimator (BLUE). We give the detailed proof in appendix section B. Thus we have:

$$\mathrm{Var}(\hat{\beta}_{\mathrm{WLS}}) \preceq \mathrm{Var}(\hat{\beta}_{\mathrm{OLS}}), \tag{28}$$

$$\begin{aligned}&\mathrm{Var}(\hat{\beta}_{\mathrm{WLS}}(x)) = x^T (X^T \Sigma^{-1} X)^{-1} x \\ &\leq x^T (X^T X)^{-1} X^T \Sigma X (X^T X) x = \mathrm{Var}(\hat{\beta}_{\mathrm{OLS}}(x)),\end{aligned} \tag{29}$$

where "$\preceq$" denotes the Löwner order, which means $\mathrm{Var}(\hat{\beta}_{\mathrm{OLS}}) - \mathrm{Var}(\hat{\beta}_{\mathrm{WLS}})$ is positive semi-definite.

4.2.3. The Overall Framework

Here, we present the overall framework and training pipeline of CAVIN. The model architecture is illustrated in Figure 2. First, based on the graph structure, we employ the node2vec algorithm to generate feature embeddings for node positions, which are then fused using multi-head attention. The fused data is passed through multiple layers of GAT to obtain the repression incorporating neighborhood influences. Subsequently, we use MLPs to model the key estimands, and weighted regression is applied in the g_t model. The pseudocode of ITE estimation part is provided in Algorithm 1.

5. Experiments

In this section, we describe the experimental setup for evaluating CAVIN, comparing it with baselines under various settings of the outcome noise range ρ. We first review our semi-synthetic data generation procedure and implementation details, then discuss the quantitative results, including an ablation study on whether to use GPE.

5.1. Dataset

In this paper, we build upon existing semi-synthetic dataset construction methods to develop a novel graph-based data

Algorithm 1 CAVIN (ITE Estimation Part)

Input: Data: $\mathcal{D} = \{(X_i, T_i, Y_i)\}_{i=1}^n$
Parameter: δ, number of epochs K
Output: ITE Estimator ITE $\hat{\tau}(x)$
for each epoch $k = 1, \ldots, K$ **do**
  Compute $\hat{e}(X)$, $\hat{\mu}_t(X)$, $\hat{\tau}_t(X)$, $\hat{\sigma}_t^2(X)$ using MLPs.
  Compute sample weight $w_{t,i} = 1/(\hat{\sigma}_t^2(X_i) + \delta)$.
  Compute residual $\mathrm{D}_t = (-1)^t[\hat{\mu}_{1-t}(X) - Y]$.
  Compute $\sigma^2(X) = \mathrm{D}_t^2 - \hat{\tau}_t^2(X)$.
  Regression: update parameters $\hat{e}(X)$, $\hat{\mu}_t(X)$, $\hat{\sigma}_t^2(X)$ using MSE loss.
  Weighted Regression: optimize $\hat{\tau}$ by minimizing $\sum_{i=1}^n w_i(\mathrm{D}_i - \hat{\tau}(X_i))^2$, the weighted MSE loss.
end for
return $\hat{\tau}(x)$.

synthesis approach for ITE estimation (Johansson et al., 2016b; Louizos et al., 2017; Shalit et al., 2017; Schwab et al., 2020; Guo et al., 2020). Unlike previous methods, which overlook the influence of multi-hop neighbors, peer treatment effects, and community preferences, our approach better captures real-world complexities. We utilize the CoraFull, DBLP, and PubMed (Bojchevski & Günnemann, 2018) datasets, homogeneous graphs with undirected edges, for data synthesis. Before using the dataset, we first apply the Louvain algorithm to identify communities, setting intra-community edges to weight 3 and inter-community edges to weight 1. We then normalize this new adjacency matrix $\mathcal{A}$ for a more stable graph representation. Formally, $\mathcal{A} \leftarrow D^{-\frac{1}{2}}(\mathcal{A} + I)D^{-\frac{1}{2}}$, where D is the diagonal degree matrix of $\mathcal{A}$, D_{ii} represents the neighbor number of i.

We employ the original node features X and adjacency matrix $\mathcal{A}$ from the raw datasets for semi-synthetic data generation. To reduce computational complexity and align with the task requirements, we first compress the features using Principal Component Analysis (PCA). Specifically, we perform Singular Value Decomposition (SVD) on the feature matrix X, expressed as $X = U\Sigma V^T$. We retain only the top $k = 128$ largest singular values, setting the rest to zero, and reconstruct the matrix as $Z = XV_k^T$, where V_k^T contains the top k singular vectors. The resulting matrix Z represents the low-dimensional node embeddings. Next, we select the vector with the largest L_2-norm as the treatment representative vector c_1, while the mean vector of all reduced-dimension vectors serves as c_0. The treatment probability $P(T_i = 1 \mid T_{\mathcal{N}}^i, Z_i, Z_{\mathcal{N}}^i)$ for each node i is then computed based on these representations:

$$P = \exp(p_1^i)/[\exp(p_1^i) + \exp(p_0^i)], \tag{30}$$

$$p_1 = \kappa_1 Z^T c_1 + \kappa_2 \sum\nolimits_{r=1}^{h} \gamma^r \mathcal{A}^r Z c_1, \tag{31}$$

$$p_0 = \kappa_1 Z^T c_0 + \kappa_2 \sum\nolimits_{r=1}^{h} \gamma^r \mathcal{A}^r Z c_0, \tag{32}$$

where κ_1, κ_2 are the coefficients representing the influence of self and r hop neighbors, γ represents the influence decay by each hop, and $\mathcal{A}^r Z$ represents the set of r hop neighbors of node i. We set T based on a binomial distribution and the treatment probabilities. Based on the node features, network structure, and the treatment T_i, we compute the potential outcomes Y_1, Y_0 and the observed outcome Y as follows:

$$f_1 = p_1 + \beta_1 + \kappa_3 \sum\nolimits_{r=1}^{h} \gamma^r \mathcal{A}^r T + \kappa_4 c_b \tag{33}$$

$$f_0 = p_0 + \beta_0 + \kappa_3 \sum\nolimits_{r=1}^{h} \gamma^r \mathcal{A}^r T + \kappa_4 c_b \tag{34}$$

$$Y = C(f_0 + T \odot f_1) + \epsilon \tag{35}$$

where $\beta_{1,0}$ are the bias, κ_3 is the impact factor of neighbor treatment, c_b is the community bias, κ_4 is the impact factor of community bias. $\epsilon \sim \mathcal{N}(0, \sigma(Z))$, where $\sigma(Z)$ can be computed by taking the dot product of Z and a random vector ω and linearly mapped to the range $[-\rho, \rho]$. The detailed parameter settings are given in appendix section C.1.

5.2. Implementation Details

We follow the data synthesis method introduced in Section 5.1 to generate outcomes on the three graph datasets, CoraFull, DBLP, and PubMed. Each node's features, adjacency structure, and potential outcomes are derived to simulate realistic heterophily and community biases.We set the noise variance parameter $\rho \in \{5, 10, 30\}$ to examine how scaling the magnitude of heteroskedastic noise impacts estimation error. For all experiments, we adopt the following configurations: (1) Model Architecture: For GNN-based methods (including CAVIN), we use five GAT layers with multi-head attention to obtain node embeddings, with a hidden size of [256,128,128,128,128]. For the meta-learner modules, we only use four-layer MLPs (ReLU activation, hidden size 128) as fundation block to estimate the ITE and any other important values. (2) Variance Weighting: In CAVIN, we further train a variance prediction branch to estimate $\sigma^2(x)$, setting $\delta = 0.01$ to avoid excessively large sample weights. (3) Training Procedure: Models are trained by minimizing the MSE w.r.t. the selected learner's objective. All parameters are optimized via Adam with a learning rate of 0.05, using random 80/20 splits for training and testing, and repeating each experiment 5 times to report the average error. The detailed hyperparameter settings and search ranges used for our experiments are provided in the appendix section C.1.

We evaluate all methods on two metrics: the Precision in rooted PEHE and the absolute error in ATE. We compare four baseline structures, BNN (Johansson et al., 2016a), TARNet (Shalit et al., 2017), TNet (Chen et al., 2024), established on S-, T-, DR-learner. We are also the first implemented deep X-learner with network inference, which we noted as GAT+X. Table 1 reports the results under different

Table 1. For each dataset, the top three rows show PEHE values for $\rho \in \{5, 10, 30\}$, and the bottom three rows show ATE values for the same ρ values. Each value is averaged over 5 independent experiments. We record best value of rooted PEHE and ATE in the same epoch. For the values underlined, the rooted PEHE or ATE has been continuously increasing from the beginning of the experiment, indicating that the model fails to converge. In such cases, we take the reading from the last epoch (200 epochs) as the final value.

Dataset	Criterion	ρ	Learner Performance					
			BNN	TARNet	TNet	GAT+X	CAVIN	
			NoGPE	NoGPE	NoGPE	NoGPE	NoGPE	GPE
Corafull	PEHE	5	$5.947^{\pm 0.128}$	$5.780^{\pm 0.101}$	$5.570^{\pm 0.095}$	$5.481^{\pm 0.087}$	$4.703^{\pm 0.045}$	$\mathbf{2.962}^{\pm 0.028}$
		10	$5.966^{\pm 0.233}$	$5.919^{\pm 0.215}$	$5.904^{\pm 0.217}$	$5.782^{\pm 0.198}$	$5.034^{\pm 0.041}$	$\mathbf{3.315}^{\pm 0.023}$
		30	$5.975^{\pm 0.092}$	$\underline{17.715}^{\pm 1.582}$	$\underline{13.683}^{\pm 0.964}$	$\underline{11.139}^{\pm 1.249}$	$5.416^{\pm 0.075}$	$\mathbf{4.287}^{\pm 0.035}$
	ATE	5	$1.560^{\pm 0.017}$	$1.557^{\pm 0.020}$	$1.710^{\pm 0.028}$	$1.643^{\pm 0.019}$	$0.360^{\pm 0.009}$	$\mathbf{0.016}^{\pm 0.005}$
		10	$1.584^{\pm 0.022}$	$1.570^{\pm 0.021}$	$1.575^{\pm 0.021}$	$1.544^{\pm 0.027}$	$0.187^{\pm 0.014}$	$\mathbf{0.022}^{\pm 0.004}$
		30	$1.646^{\pm 0.035}$	$1.726^{\pm 0.038}$	$1.843^{\pm 0.142}$	$0.969^{\pm 0.515}$	$0.124^{\pm 0.007}$	$\mathbf{0.012}^{\pm 0.004}$
DBLP	PEHE	5	$5.681^{\pm 0.115}$	$5.272^{\pm 0.096}$	$4.899^{\pm 0.066}$	$4.866^{\pm 0.058}$	$3.056^{\pm 0.030}$	$\mathbf{1.687}^{\pm 0.025}$
		10	$5.743^{\pm 0.160}$	$5.340^{\pm 0.095}$	$5.385^{\pm 0.098}$	$5.152^{\pm 0.088}$	$3.387^{\pm 0.032}$	$\mathbf{2.229}^{\pm 0.028}$
		30	$\underline{5.965}^{\pm 0.239}$	$\underline{19.839}^{\pm 1.697}$	$\underline{15.828}^{\pm 1.181}$	$\underline{12.596}^{\pm 0.878}$	$4.246^{\pm 0.038}$	$\mathbf{3.453}^{\pm 0.032}$
	ATE	5	$1.412^{\pm 0.018}$	$1.409^{\pm 0.018}$	$1.608^{\pm 0.025}$	$1.643^{\pm 0.028}$	$0.161^{\pm 0.015}$	$\mathbf{0.015}^{\pm 0.003}$
		10	$1.403^{\pm 0.019}$	$1.373^{\pm 0.017}$	$1.193^{\pm 0.016}$	$1.819^{\pm 0.030}$	$\mathbf{0.125}^{\pm 0.016}$	$0.022^{\pm 0.003}$
		30	$1.337^{\pm 0.018}$	$2.440^{\pm 0.045}$	$3.092^{\pm 0.272}$	$3.070^{\pm 0.368}$	$0.144^{\pm 0.009}$	$\mathbf{0.015}^{\pm 0.004}$
PubMed	PEHE	5	$6.085^{\pm 0.141}$	$4.215^{\pm 0.055}$	$3.933^{\pm 0.046}$	$4.089^{\pm 0.062}$	$2.966^{\pm 0.028}$	$\mathbf{1.822}^{\pm 0.021}$
		10	$\underline{6.221}^{\pm 0.155}$	$4.935^{\pm 0.064}$	$4.387^{\pm 0.058}$	$4.962^{\pm 0.059}$	$3.277^{\pm 0.033}$	$\mathbf{2.093}^{\pm 0.023}$
		30	$\underline{6.887}^{\pm 0.177}$	$\underline{19.399}^{\pm 1.592}$	$5.893^{\pm 0.080}$	$5.586^{\pm 0.072}$	$4.658^{\pm 0.048}$	$\mathbf{3.937}^{\pm 0.043}$
	ATE	5	$1.085^{\pm 0.013}$	$1.015^{\pm 0.015}$	$1.128^{\pm 0.024}$	$1.394^{\pm 0.025}$	$0.110^{\pm 0.007}$	$\mathbf{0.008}^{\pm 0.001}$
		10	$1.119^{\pm 0.020}$	$1.029^{\pm 0.016}$	$0.507^{\pm 0.018}$	$1.482^{\pm 0.028}$	$\mathbf{0.017}^{\pm 0.003}$	$0.021^{\pm 0.004}$
		30	$1.069^{\pm 0.018}$	$2.955^{\pm 0.048}$	$1.892^{\pm 0.034}$	$2.182^{\pm 0.042}$	$0.125^{\pm 0.013}$	$\mathbf{0.016}^{\pm 0.003}$

values of ρ. As ρ increases, all methods experience larger estimation errors, indicating that scaling up the heteroscedasticity of noise makes the ITE estimation more challenging. Nonetheless, for each fixed ρ, CAVIN consistently achieves lower PEHE and smaller ATE errors compared with other baselines, demonstrating its robustness against larger outcome magnitudes. For almost every setting of CAVIN, GPE variants outperform NoGPE, underscoring the importance of explicitly modeling node positions in a graph to account for subtle community-level biases that purely local feature aggregation might miss.

Overall, the proposed CAVIN framework yields the highest estimation accuracy across all experimental configurations, validating the efficacy of combining variance-weighted meta-learning with positional encoding in GNNs. Position-aware embeddings are beneficial in capturing community-driven effects, which is key in online marketing or social recommendation scenarios, where user communities can significantly influence treatment outcomes. As the outcome range grows, all methods incur higher PEHE and ATE errors, but CAVIN remains comparatively more stable. These results confirm the advantage of incorporating both community-level context and heteroskedastic noise modeling, shedding light on how future real-world systems can benefit from a graph-based uplift modeling approach.

6. Conclusion

In this paper, we introduce CAVIN, a graph-based ITE estimation framework that addresses both community-level influences and heteroskedastic noise. By integrating a self-attention-based positional encoding into GATs, it captures subtle community biases often overlooked by existing methods, while a variance-weighted meta-learner mitigates noise by adaptively reweighting predictions. Empirical results on semi-synthetic data derived from real-world graphs show consistent gains over traditional baselines, including GNN-augmented S-, T-, X-, and DR-learners. Although our experiments are encouraging, further work is needed to evaluate performance on larger, more diverse datasets and to tackle the challenges posed by user graphs with hundreds of millions of nodes. Through these efforts, we aim to develop CAVIN into a robust, practical solution for reliable ITE estimation in complex networked systems.

Impact Statement

Our work advances the field of network-based individual treatment effect estimation by introducing the CAVIN framework, which simultaneously accounts for community-level influences and heteroskedastic noise. This enables a more nuanced understanding of how interventions affect individuals within interconnected environments such as social networks. This capability has broad applications in areas like personalized marketing and public health interventions, enhancing decision-making accuracy and optimizing resource allocation. However, our approach relies on user data embedded within large-scale graph structures, which, if misused, could raise ethical concerns related to privacy and user consent. To address these issues, we validate our method using semi-synthetic datasets. We are committed to resolving data security challenges and strive to develop safe and reliable recommendation models. In the future, our research and responsible deployment will focus on robust data governance, fairness assessments, and transparent reporting to ensure that these methodological advancements translate into equitable and ethically sound outcomes in real-world applications.

References

Abrevaya, J. and Xu, H. Estimation of treatment effects under endogenous heteroskedasticity. *Journal of Econometrics*, 234(2):451–478, 2023.

Aitken, A. C. Iv.—on least squares and linear combination of observations. *Proceedings of the Royal Society of Edinburgh*, 55:42–48, 1936. doi: 10.1017/S0370164600014346.

Albert, J. and Goldenberg, D. E-commerce promotions personalization via online multiple-choice knapsack with uplift modeling. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, CIKM '22, pp. 2863–2872, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450392365. doi: 10.1145/3511808.3557100. URL https://doi.org/10.1145/3511808.3557100.

Aral, S. and Walker, D. Creating social contagion through viral product design: A randomized trial of peer influence in networks. *Management science*, 57(9):1623–1639, 2011.

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

Barkley, B. G., Hudgens, M. G., Clemens, J. D., Ali, M., and Emch, M. E. Causal inference from observational studies with clustered interference, with application to a cholera vaccine study. *The Annals of Applied Statistics*, 14(3):1432 – 1448, 2020. doi: 10.1214/19-AOAS1314. URL https://doi.org/10.1214/19-AOAS1314.

Blondel, V. D., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008, 2008.

Bojchevski, A. and Günnemann, S. Deep gaussian embedding of graphs: Unsupervised inductive learning via ranking. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=r1ZdKJ-0W.

Cai, R., Yang, Z., Chen, W., Yan, Y., and Hao, Z. Generalization bound for estimating causal effects from observational network data. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pp. 163–172, 2023.

Chen, W., Cai, R., Yang, Z., Qiao, J., Yan, Y., Li, Z., and Hao, Z. Doubly robust causal effect estimation under networked interference via targeted learning. *arXiv preprint arXiv:2405.03342*, 2024.

Curth, A. and Van der Schaar, M. Nonparametric estimation of heterogeneous treatment effects: From theory to learning algorithms. In *International Conference on Artificial Intelligence and Statistics*, pp. 1810–1818. PMLR, 2021.

de Valck, K., van Bruggen, G. H., and Wierenga, B. Virtual communities: A marketing perspective. *Decision Support Systems*, 47(3):185–203, 2009. ISSN 0167-9236. doi: https://doi.org/10.1016/j.dss.2009.02.008. URL https://www.sciencedirect.com/science/article/pii/S016792360900058X. Online Communities and Social Network.

Forastiere, L., Airoldi, E. M., and Mealli, F. Identification and estimation of treatment and interference effects in observational studies on networks. *Journal of the American Statistical Association*, 116(534):901–918, 2021.

Goldenberg, D., Albert, J., Bernardi, L., and Estevez, P. Free lunch! retrospective uplift modeling for dynamic promotions recommendation within roi constraints. In *Proceedings of the 14th ACM Conference on Recommender Systems*, pp. 486–491, 2020.

Grover, A. and Leskovec, J. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 855–864, 2016.

Guo, R., Li, J., and Liu, H. Learning individual causal effects from networked observational data. In *Proceedings of the 13th international conference on web search and data mining*, pp. 232–240, 2020.

Huang, Q., Ma, J., Li, J., Guo, R., Sun, H., and Chang, Y. Modeling interference for individual treatment effect estimation from networked observational data. *ACM Transactions on Knowledge Discovery from Data*, 18(3): 1–21, 2023.

Hudgens, M. G. and Halloran, M. E. Toward causal inference with interference. *Journal of the American Statistical Association*, 103(482):832–842, 2008.

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

Johansson, F., Shalit, U., and Sontag, D. Learning representations for counterfactual inference. In Balcan, M. F. and Weinberger, K. Q. (eds.), *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 3020–3029, New York, New York, USA, 20–22 Jun 2016b. PMLR. URL https://proceedings.mlr.press/v48/johansson16.html.

Kazemi, A. and Ester, M. Adversarially balanced representation for continuous treatment effect estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 13085–13093, 2024.

Kennedy, E. H. Towards optimal doubly robust estimation of heterogeneous causal effects. *Electronic Journal of Statistics*, 17(2):3008–3049, 2023.

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

Liu, D., Tang, X., Gao, H., Lyu, F., and He, X. Explicit feature interaction-aware uplift network for online marketing. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 4507–4515, 2023.

Louizos, C., Shalit, U., Mooij, J. M., Sontag, D., Zemel, R., and Welling, M. Causal effect inference with deep latent-variable models. *Advances in neural information processing systems*, 30, 2017.

Luo, S., Xin, M., Wang, S., Zhao, J., Zhang, G., Li, L., Li, L., and Lau, J. T.-f. Behavioural intention of receiving covid-19 vaccination, social media exposures and peer discussions in china. *Epidemiology & Infection*, 149: e158, 2021.

Ma, J., Wan, M., Yang, L., Li, J., Hecht, B., and Teevan, J. Learning causal effects on hypergraphs. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1202–1212, 2022.

McCulloch, R., Sparapani, R., Gramacy, R., Spanbauer, C., and Pratola, M. Bart: Bayesian additive regression trees. *R package version*, 1, 2018.

Mo, W. and Liu, Y. Efficient learning of optimal individualized treatment rules for heteroscedastic or misspecified treatment-free effect models. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 84(2): 440–472, 2022.

Papadogeorgou, G., Imai, K., Lyall, J., and Li, F. Causal inference with spatio-temporal data: estimating the effects of airstrikes on insurgent violence in iraq. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 84(5):1969–1999, 2022.

Parshakov, P., Naidenova, I., and Barajas, A. Spillover effect in promotion: Evidence from video game publishers and esports tournaments. *Journal of Business Research*, 118: 262–270, 2020.

Radcliffe, N. J. and Surry, P. D. Real-world uplift modelling with significance-based uplift trees. *White Paper TR-2011-1, Stochastic Solutions*, pp. 1–33, 2011.

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

Shalit, U., Johansson, F. D., and Sontag, D. Estimating individual treatment effect: generalization bounds and algorithms. In *International conference on machine learning*, pp. 3076–3085. PMLR, 2017.

Shi, P., Zhang, X., and Zhong, W. Estimating conditional average treatment effects with heteroscedasticity by model averaging and matching. *Economics Letters*, 238:111679, 2024. ISSN 0165-1765. doi: https://doi.org/10.1016/j.econlet.2024.111679. URL https://www.sciencedirect.com/science/article/pii/S0165176524001629.

Sirazitdinov, A., Buchwald, M., Heuveline, V., and Hesser, J. Graph neural networks for individual treatment effect estimation. *IEEE Access*, 12:106884–106894, 2024. doi: 10.1109/ACCESS.2024.3437665.

Ugander, J., Karrer, B., Backstrom, L., and Kleinberg, J. Graph cluster randomization: Network exposure to multiple universes. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 329–337, 2013.

Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y., et al. Graph attention networks. *stat*, 1050 (20):10–48550, 2017.

Zivich, P. N. and Breskin, A. Machine learning for causal inference: on the use of cross-fit estimators. *Epidemiology*, 32(3):393–401, 2021.

A. Proof of Theorem 3.3

Proof. This objective can be decomposed as:

$$\begin{aligned}&\mathbb{E}\left[(D_i-\hat{\tau}(X_i))^2 \mid X=X_i\right]=\mathbb{E}\left[(D_i-\tau(X_i)+\tau(X_i)-\hat{\tau}(X_i))^2 \mid X=X_i\right]\\&=\mathbb{E}\left[(D_i-\tau(X_i))^2 \mid X=X_i\right]+\mathbb{E}\left[(\tau(X_i)-\hat{\tau}(X_i))^2 \mid X=X_i\right]\\&+2\mathbb{E}\left[(D_i-\tau(X_i))(\tau(X_i)-\hat{\tau}(X_i)) \mid X=X_i\right].\end{aligned} \tag{36}$$

Because $\mathbb{E}\left[D_i-\tau(X_i) \mid X_i=x\right]=0$,

$$\mathbb{E}\left[(D_i-\hat{\tau}(X_i))^2 \mid X=X_i\right]=\mathbb{E}\left[(D_i-\tau(X_i))^2 \mid X=X_i\right]+\mathbb{E}\left[(\tau(X_i)-\hat{\tau}(X_i))^2 \mid X=X_i\right], \tag{37}$$

where $\mathbb{E}\left[(D_i-\tau(X_i))^2 \mid X=X_i\right]$ is a constant value. Therefore, the optimization object equals to:

$$\arg\min_{\hat{\tau}\in\mathcal{H}}\mathbb{E}\left[(\tau(X_i)-\hat{\tau}(X_i))^2 \mid X=X_i\right]. \tag{38}$$

□

B. Proof of BLUE

According to section 4.2.2, the OLS and WLS estimators are given by

$$\hat{\beta}_{\text{OLS}}=(X^TX)^{-1}X^TD,\quad \hat{\beta}_{\text{WLS}}=(X^TWX)^{-1}X^TWD \tag{39}$$

Their covariance matrices are:

$$\text{Var}(\hat{\beta}_{\text{OLS}})=(X^TX)^{-1}X^T\Sigma X(X^TX)^{-1},\quad \text{Var}(\hat{\beta}_{\text{WLS}})=(X^TWX)^{-1}=(X^T\Sigma^{-1}X)^{-1}. \tag{40}$$

Under heteroskedasticity, the Gauss–Markov theorem states that WLS is the BLUE when $W=\Sigma^{-1}$. This implies:

$$\text{Var}(\hat{\beta}_{\text{WLS}})\preceq\text{Var}(\hat{\beta}_{\text{OLS}}), \tag{41}$$

Proof. Let $A=X^T\Sigma^{-1/2}$ and $B=X^T\Sigma^{1/2}$, where $\Sigma^{1/2}$ is the symmetric square root of Σ. Note that:

$$A^TA=X^T\Sigma^{-1}X,\quad B^TB=X^T\Sigma X. \tag{42}$$

For WLS:

$$\text{Var}(\hat{\beta}_{\text{WLS}})=(A^TA)^{-1}. \tag{43}$$

For OLS:

$$\text{Var}(\hat{\beta}_{\text{OLS}})=(X^TX)^{-1}B^TB(X^TX)^{-1}. \tag{44}$$

The Cauchy–Schwarz inequality for matrices states that for any positive definite matrices P and Q:

$$P^TP\succeq(P^TQ^{-1}P)^{-1}. \tag{45}$$

Applying this to $P=X^T\Sigma^{1/2}$ and $Q=\Sigma^{-1/2}$, we obtain:

$$B^TB=X^T\Sigma X\succeq(X^T\Sigma^{-1}X)^{-1}. \tag{46}$$

Multiplying both sides by $(X^TX)^{-1}$ from the left and right:

$$(X^TX)^{-1}B^TB(X^TX)^{-1}\succeq(X^T\Sigma^{-1}X)^{-1}. \tag{47}$$

This simplifies to:

$$\text{Var}(\hat{\beta}_{\text{OLS}})\succeq\text{Var}(\hat{\beta}_{\text{WLS}}). \tag{48}$$

□

C. Hyperparameters And Configurations

C.1. Dataset Semi-synthetic

Different datasets have different feature dimensions and norms. To ensure that the ranges of the noise-free output values Y_0 and Y_1 are approximately consistent across the three datasets, we set different parameters for each dataset as shown in Table 2.

In the training process, we set the all head number of all the muiti-head attention layer as 4, including GAT and feature fusion stages. We set total epochs as 200, learning rate as 0.05. We use weight decay of 1×10^{-4} and did not use dropout method.

C.2. Graph Position Embedding

We utilize the node2vec method to generate graph positional embeddings, where each node is assigned an embedding vector that encodes its structural position within the graph. The embedding process is primarily based on random walks, with the following parameter settings: an embedding dimension of 128, a walk length (number of steps per random walk) of 30, 200 random walks per node, and a window size of 10. Under this configuration, the graph embedding process typically requires approximately 30 minutes to complete.

In our framework, we employ GAT as the graph neural network architecture for representation learning. Across all experiments, we consistently utilize the same GAT module to aggregate network information, ensuring a unified and reliable approach to capturing structural dependencies within the graph. We construct a 5-layer GAT architecture with layer dimensions $[256, 128, 128, 128, 128]$ for representation learning. For the S-learner-based BNN and T-learner-based TARNet models, we employ a 4-layer MLP with each layer sized at 128 as the estimation model for vector prediction. In the case of TNet, we deviate from its original design by omitting the explicit modeling of neighbor treatment effects, focusing exclusively on its DR-learner implementation, which we include as a baseline in our experiments. Additionally, to the best of our knowledge, we are the first to integrate the X-learner into deep network-based causal inference, and thus, the X-learner baseline is our novel contribution.

Table 2. Parameter settings for different datasets

DATASET	C	β_0	β_1	EMB_DIM	γ	κ_1	κ_2	κ_3	κ_4	HOPS
CORAFULL	5	-0.3	0.3	256	0.6	0.2	0.1	0.03	0.05	5
DBLP	5	-0.3	0.3	256	0.6	1.5	0.75	0.2	0.3	5
PUBMED	5	-0.3	0.3	256	0.6	0.15	0.07	0.025	0.035	5

D. Additional Experiment Results

As supplementary experiments, we tested the effects of introducing GPE into other GAT-based meta-learners. The results are given in Table 3. The experiments demonstrate that GPE reduces rooted PEHE and ATE in the vast majority of cases. This indicates that the effectiveness of our graph embedding approach is general.

To ensure reproducibility, the code and dataset details will be made publicly available upon acceptance of the paper.

Table 3. Comparison of learner performance for different datasets. For each dataset, the top three rows show PEHE values for $\rho \in \{5, 10, 30\}$, and the bottom three rows show ATE values for the same values. Each value is averaged over 5 independent experiments. For the values underlined, the rooted PEHE or ATE has been continuously increasing from the beginning of the experiment, indicating that the model fails to converge. In such cases, we take the reading from the last epoch (200 epochs) as the final value. For each model, we test both estimation error with and without GPE.

Dataset	Criterion	ρ	Learner Performance									
			BNN		TARNet		TNet		GAT+X		CAVIN	
			GPE	NoGPE	GPE	NoGPE	GPE	NoGPE	GPE	NoGPE	GPE	NoGPE
Corafull	PEHE	5	5.923	5.947	4.537	5.780	4.201	5.570	4.300	5.481	**2.962**	4.703
		10	5.939	5.966	5.725	5.919	4.661	5.904	4.879	5.782	**3.315**	5.034
		30	6.046	5.975	15.143	17.715	5.289	13.683	5.742	11.139	**4.287**	5.416
	ATE	5	1.615	1.560	1.276	1.557	0.747	1.710	1.013	1.643	**0.016**	0.360
		10	1.636	1.584	0.451	1.570	0.936	1.575	0.689	1.544	**0.022**	0.187
		30	1.618	1.646	1.586	1.726	1.699	1.843	0.153	0.969	**0.012**	0.124
DBLP	PEHE	5	5.629	5.681	4.173	5.272	3.459	4.899	3.890	4.866	**1.687**	3.056
		10	5.671	5.743	4.794	5.340	4.408	5.385	4.620	5.152	**2.229**	3.387
		30	5.710	5.965	21.146	19.839	5.411	15.828	9.695	12.596	**3.453**	4.246
	ATE	5	1.484	1.412	0.154	1.409	**0.013**	1.608	0.948	1.643	0.015	0.161
		10	1.499	1.403	0.200	1.373	**0.009**	1.193	1.162	1.819	0.022	0.125
		30	1.496	1.337	2.577	2.440	**0.010**	3.092	1.610	3.070	0.015	0.144
PubMed	PEHE	5	5.888	6.085	3.168	4.215	2.843	3.933	2.945	4.089	**1.822**	2.966
		10	5.914	6.221	4.321	4.935	3.551	4.387	3.593	4.962	**2.093**	3.277
		30	6.121	6.887	5.814	19.399	4.281	5.893	5.437	5.586	**3.937**	4.658
	ATE	5	1.377	1.085	0.355	1.015	0.017	1.128	0.424	1.394	**0.008**	0.110
		10	1.453	1.119	0.092	1.029	0.048	0.507	0.516	1.482	**0.017**	0.017
		30	1.549	1.069	0.035	2.955	1.306	1.892	1.033	2.182	**0.016**	0.125