Estimating treatment effects in networks using domain adversarial learning

Daan Caljon [1] **Jente Van Belle** [1] **Wouter Verbeke** [1]

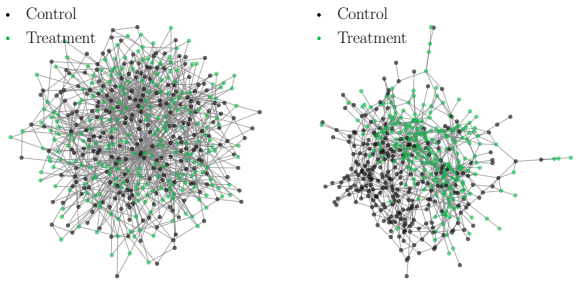
Abstract

Estimating heterogeneous treatment effects in networked settings is complicated by interference, meaning that an instance's outcome can be influenced by the treatment status of others. Existing causal machine learning approaches often assume a known exposure mapping that summarizes how the outcome of a given instance is influenced by others' treatments, a simplification that is often unrealistic. Furthermore, the interaction between homophily—the tendency of similar instances to connect—and the treatment assignment mechanism has not been explicitly studied before. This interaction can induce a network-level covariate shift, potentially biasing the estimated treatment effects. To address these challenges, we propose HINet—a novel method that integrates Graph Neural Networks (GNNs) with domain adversarial learning. This combination enables learning treatment-invariant node representations, thereby mitigating potential bias caused by covariate shift. Our empirical evaluations on synthetic and semi-synthetic network datasets demonstrate that our approach outperforms existing methods.

1. Introduction

Individualized treatment effect estimation enables data-driven optimization of decision-making. Traditionally, *no interference* is assumed, meaning that the treatment assigned to one instance does not affect the outcome of others. However, in many real-world settings this assumption is violated due to *spillover effects*. For example, a vaccine not only protects its recipient but also indirectly benefits their social contacts because of the recipient's enhanced protection.

Recent advances in causal machine learning have introduced methods for estimating treatment effects in network settings (Ma and Tresp, 2021; Jiang and Sun, 2022; Chen et al., 2024). These methods often rely on a predefined exposure mapping, which specifies how the treatments of other instances in a network influence the outcome of a given instance. A common approach is to aggregate these treatments using the sum or proportion of treated one-hop neighbors (Ma and Tresp, 2021; Forastiere et al., 2021; Jiang and Sun, 2022). While this simplifies the modeling of spillover effects, it is often unrealistic in real-world scenarios where the exact mechanisms behind these effects are unknown. Moreover, spillover effects may be heterogeneous, i.e., dependent on the features of the instances involved (Adhikari and Zheleva, 2023; Huang et al., 2023; Zhao et al., 2024).



(a) Network without homophily (b) Network with homophily

Figure 1: Homophily and the treatment assignment mechanism interact to create clusters of treated and untreated nodes within the network, i.e., network-level covariate shift.

In this work, we propose *Heterogeneous Interference Network (HINet)*, a novel method that combines expressive GNN layers—which enable the learning of *heterogeneous spillover effects*—with domain adversarial learning to obtain *balanced representations* for estimating treatment effects in the presence of interference. Importantly, HINet does not rely on a prespecified exposure mapping. Additionally, we analyze how homophily interacts with the treatment assignment mechanism. This interaction might induce clusters of treated and untreated nodes within the network (see Figure 1). Homophily connects nodes with similar features, while the treatment assignment mechanism increases the likelihood that such nodes receive similar treatment. Together, these mechanisms introduce a *covariate shift at the network level*. For instance, older individuals may be more likely to form connections with each other in a social network (due to homophily) and are also more likely to receive

[1]Faculty of Economics and Business, KU Leuven. Correspondence to: Daan Caljon <daan.caljon@kuleuven.be>.

Proceedings of the 42nd International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025. Copyright 2025 by the author(s).

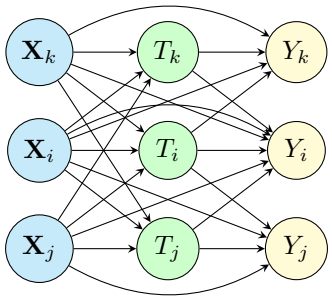


Figure 2: DAG representing the assumed causal structure.

a vaccine (due to the treatment assignment mechanism). This network-level covariate shift arises in addition to the standard covariate shift between treated and control units induced by the treatment assignment mechanism. We empirically demonstrate that HINet can effectively address both types of covariate shift.

Contributions. (1) We propose HINet, a new method for estimating treatment effects in the presence of interference. HINet combines expressive GNN layers—to learn an exposure mapping—and domain adversarial learning—to address (network-level) covariate shift; (2) we empirically show HINet's ability to estimate treatment effects in the presence of interference; (3) we propose two new metrics for the evaluation of treatment effect estimates in the presence of interference; and (4) we analyze how homophily interacts with the treatment assignment mechanism and demonstrate that domain adversarial training mitigates the impact of the resulting bias.

2. Problem Setup

Notation. We consider an undirected network $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ where $\mathcal{V}$ is the set of nodes/vertices and $\mathcal{E}$ the set of edges connecting the nodes. The set of edges of node i is denoted as $\mathcal{E}_i$. Each node i is an instance or unit in the network with covariates $\mathbf{X}_i \in \mathcal{X} \subseteq \mathbb{R}^d$, a treatment $T_i \in \mathcal{T} = \{0, 1\}$, and an outcome $Y_i \in \mathcal{Y} \subseteq \mathbb{R}$. In marketing, for example, $\mathbf{X}_i$ can represent customer features, T_i whether a customer was targeted with a marketing campaign, and Y_i customer expenditure. The set of directly connected instances, or neighbors, of instance i are denoted $\mathcal{N}_i$. $\mathcal{N}_i$ is used as a subscript to describe the set of covariates $\mathbf{X}_{\mathcal{N}_i}$ or treatments $\mathbf{T}_{\mathcal{N}_i}$ of i's neighbors. The potential outcome for unit i with treatment t_i and the set of treatments of its neighbors $\mathbf{t}_{\mathcal{N}_i}$ is denoted as $Y_i(t_i, \mathbf{t}_{\mathcal{N}_i})$.

Assumptions. We assume that only directly connected instances affect each other (Markov assumption). The assumed causal structure is visualized as a Directed Acyclic Graph (DAG) in Figure 2 (Greenland et al., 1999; Ogburn and VanderWeele, 2014). Three mutually connected instances i, j, and k are shown. The features of a unit i, $\mathbf{X}_i$, affect the treatment T and outcome Y of both itself and its neighbors. The treatment, in turn, affects the outcome of itself and its neighbors. The arrows from $\mathbf{X}_k$ and T_k to Y_j, and from $\mathbf{X}_j$ and T_j to Y_k are omitted for visual clarity.

We assume that we have access to observational data $\mathcal{D} = \big(\{\mathbf{x}_i, t_i, y_i, \}_{i=1}^{|\mathcal{V}|}; \mathcal{G}\big)$. Importantly, this data does not necessarily come from a randomized controlled trial (RCT), and a treatment assignment mechanism might be present. This is represented in the DAG by the arrows from a unit's features to its own treatment and the treatments of its neighbors.

Previous work assumed a predefined exposure mapping that summarizes how the treatments of neighbors influence the outcome of a given instance. In this work, we do not assume this function is known; instead, we aim to *learn* an exposure mapping from data.

The classic assumptions from causal inference are slightly modified to ensure identifiability in a networked setting (Forastiere et al., 2021; Jiang and Sun, 2022):

Consistency: If $T_i = t_i$ and $\mathbf{T}_{\mathcal{N}_i} = \mathbf{t}_{\mathcal{N}_i}$, then $Y_i = Y_i(t_i, \mathbf{t}_{\mathcal{N}_i})$, with $\mathbf{t}_{\mathcal{N}_i}$ the set of treatments of i's neighbors.

Overlap: $\exists\ \delta \in (0, 1)$ such that $\delta < p(T_i = 1 | \mathbf{X}_i = \mathbf{x}_i, \mathbf{X}_{\mathcal{N}_i} = \mathbf{x}_{\mathcal{N}_i}) < 1 - \delta$.

Strong ignorability: $Y_i(T_i = t_i, \mathbf{T}_{\mathcal{N}_i} = \mathbf{t}_{\mathcal{N}_i}) \perp\!\!\!\perp T_i, \mathbf{T}_{\mathcal{N}_i} \mid \mathbf{X}_i, \mathbf{X}_{\mathcal{N}_i},\ \forall t_i \in \mathcal{T}, \mathbf{t}_{\mathcal{N}_i} \in \mathcal{T}^{|\mathcal{N}_i|}, \mathbf{X}_i \in \mathcal{X}, \mathbf{X}_{\mathcal{N}_i} \in \mathcal{X}^{|\mathcal{N}_i|}$.

Objective. We aim to estimate the Individual Total Treatment Effect (ITTE) (Caljon et al., 2024), defined as:

$$\omega_i(t_i, \mathbf{t}_{\mathcal{N}_i}) = \mathbb{E}\big[Y_i(t_i, \mathbf{t}_{\mathcal{N}_i}) - Y_i(0, \mathbf{0}) \mid \mathbf{x}_i, \mathbf{x}_{\mathcal{N}_i}\big]. \quad (1)$$

To this end, we train a model $\mathcal{M}(\mathbf{x}_i, t_i, \mathbf{x}_{\mathcal{N}_i}, \mathbf{t}_{\mathcal{N}_i})$ to predict $Y_i(t_i, \mathbf{t}_{\mathcal{N}_i})$, which is used twice to obtain a predicted ITTE: $\hat{\omega}_i(t_i, \mathbf{t}_{\mathcal{N}_i}) = \mathcal{M}(\mathbf{x}_i, t_i, \mathbf{x}_{\mathcal{N}_i}, \mathbf{t}_{\mathcal{N}_i}) - \mathcal{M}(\mathbf{x}_i, 0, \mathbf{x}_{\mathcal{N}_i}, \mathbf{0})$.

Homophily and the treatment assignment mechanism. In observational data, the treatment assignment mechanism—such as a policy or self-selection—can induce a *covariate shift*, where the treatment and control groups have different covariate distributions. Consequently, treatment effect estimates can be biased (Shalit et al., 2017). In a setting with interference, this issue may also be present, and *possibly even amplified by homophily*. Homophily is a social phenomenon that refers to the tendency of people with similar features to be connected in a social network (McPherson et al., 2001). When the features that drive homophily also influence treatment assignment, an additional form of covariate shift arises—namely, *network-level covariate shift*—since similar instances are not only more likely to be connected but also more likely to receive the same treatment. Consequently, an instance that is likely to be treated (due to the treatment assignment mechanism) will also be more likely to have treated neighbors (due to the interaction between homophily and the treatment assignment mechanism).

This can lead to clusters of treated and untreated instances within a network, as depicted in Figure 1b, creating a covariate shift at the network level—where instances with many treated neighbors and those with few treated neighbors have different feature distributions. We hypothesize that this can lead to biased treatment effect estimates, and that learning balanced node representations will aid in reducing this bias.

3. Methodology

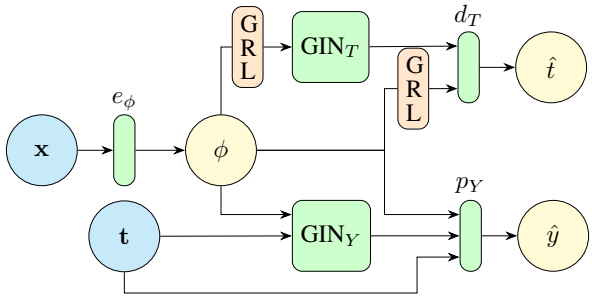


Figure 3: HINet architecture.

The *architecture of HINet*—our proposed neural model for estimating treatment effects in the presence of interference, which models heterogeneous spillover effects and uses domain adversarial training to learn balanced representations—is visualized in Figure 3. For each instance, the features $\mathbf{x}_i$ are transformed to a representation ϕ_i through a multi-layer perceptron (MLP) e_ϕ. This representation is subsequently used to predict both the treatment $\hat{t}_i$ and the outcome $\hat{y}_i$. Following Bica et al. (2020), the neural network first learns a shared representation and then splits into two branches.

The *lower branch* predicts y_i. To account for network information, a Graph Isomorphism Network (GIN) is used (Xu et al., 2019). Unlike some other GNN architectures, such as GCN (Kipf and Welling, 2016) and GraphSAGE (Hamilton et al., 2017), GIN offers maximal representational capacity. Consequently, it is particularly well-suited for learning different exposure mapping functions and thus heterogeneous spillover effects. The output of the GIN is combined with ϕ_i and t_i, and fed into the MLP p_Y to predict $\hat{y}_i$.

The *upper branch* predicts the treatment t_i for each instance. Its setup is very similar to that of the lower branch, with two key differences: $\mathbf{t}$ is not used as an input, and Gradient Reversal Layers (GRLs) (Ganin et al., 2016) are used. GRLs do not apply any transformation in the forward pass but reverse the gradient in the backward pass. This trains ϕ_i to become a treatment-invariant representation of the features, thereby reducing treatment assignment bias (Shalit et al., 2017; Bica et al., 2020; Berrevoets et al., 2020).

HINet is trained by combining two different losses: the outcome loss and the treatment prediction loss, defined respectively as $\mathcal{L}_y = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$ and $\mathcal{L}_t = \frac{1}{n}\sum_{i=1}^{n}\text{BCE}(t_i, \hat{t}_i)$, where BCE is the binary cross-entropy loss. Thanks to the GRL, we can optimize the combined loss

$$\mathcal{L}_{\text{comb}} = \mathcal{L}_y + \alpha \cdot \mathcal{L}_t, \tag{2}$$

where α determines the importance of adversarial balancing. Note that $\mathcal{L}_y$ does not affect the upper (treatment) branch, while $\mathcal{L}_t$ does not affect the lower (outcome) branch.

4. Experiments and Discussion

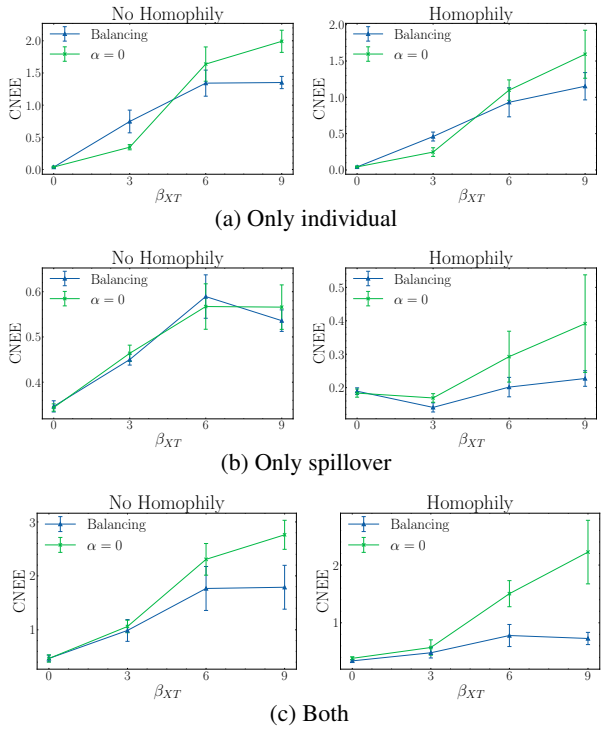
Data. Simulated data is commonly used in causal machine learning to evaluate treatment effect estimators, as ground truth effects are unobservable in real-world datasets (Berrevoets et al., 2020; Feuerriegel et al., 2024). As in related work (Ma and Tresp, 2021; Jiang and Sun, 2022; Chen et al., 2024), we use the *Flickr* and *BlogCatalog (BC)* datasets. To further evaluate generalization across different network structures, we also simulate two fully synthetic datasets: one using the *Barabási–Albert (BA)* random network model (Barabási and Albert, 1999), and another using a procedure that generates *homophilous* graphs based on cosine similarity. For each dataset, a training, validation, and test set is generated. Full details of the data-generating process (DGP) are provided in Appendix A. Appendix B provides details on the quantification of homophily.

Methods for comparison. We compare HINet to the following methods for estimating treatment effects: *TARNet* (Shalit et al., 2017), which ignores network information; *NetDeconf* (Guo et al., 2020), which incorporates network information but does not account for spillover effects; *NetEst* (Jiang and Sun, 2022), which relies on a predefined exposure mapping to estimate spillover effects; and *SPNet* (own implementation) (Zhao et al., 2024), which aims to estimate heterogeneous spillover effects using a masked attention mechanism. Finally, we include a *GIN model*, which uses node features and treatments as inputs to a GIN layer that is followed by an MLP to predict $\hat{y}_i$. Details on implementation and hyperparameter selection—including the selection of α—can be found in Appendix C.

Performance metrics. In a traditional no-interference setting with binary treatment, there is a uniquely defined treatment effect. In contrast, in a network setting, this is no longer the case, as there are *many possible treatment assignments*, each potentially resulting in different potential outcomes. It remains an open question how best to evaluate treatment effect estimation methods in the presence of interference using simulated data. Previous work has typically assessed performance based on only one *counterfactual network*—i.e., a network in which at least one unit receives a different treatment than in the observed network (Jiang and Sun, 2022; Chen et al., 2024). However, some models may be accurate in certain counterfactual networks (e.g.,

Dataset	Metric	TARNet	NetDeconf	NetEst	GIN model	SPNet	HINet (ours)
Flickr	PEHNE	3.39 ± 0.02	4.74 ± 0.14	1.67 ± 0.22	1.94 ± 0.37	5.87 ± 0.18	$\mathbf{0.88} \pm \mathbf{0.32}$
	CNEE	5.54 ± 0.02	6.14 ± 0.14	2.91 ± 0.28	1.07 ± 0.06	7.39 ± 0.08	$\mathbf{0.98} \pm \mathbf{0.31}$
BC	PEHNE	3.25 ± 0.02	6.97 ± 0.31	2.47 ± 0.25	1.86 ± 0.27	5.48 ± 0.78	$\mathbf{1.11} \pm \mathbf{0.24}$
	CNEE	3.95 ± 0.02	7.17 ± 0.34	1.34 ± 0.09	1.34 ± 0.09	5.66 ± 0.83	$\mathbf{1.17} \pm \mathbf{0.29}$
Simulated BA	PEHNE	3.01 ± 0.02	5.05 ± 0.17	0.93 ± 0.10	1.60 ± 0.08	3.94 ± 0.10	$\mathbf{0.64} \pm \mathbf{0.06}$
	CNEE	5.96 ± 0.02	7.13 ± 0.18	1.88 ± 0.10	1.27 ± 0.04	5.88 ± 0.10	$\mathbf{0.70} \pm \mathbf{0.05}$
Simulated homophilous	PEHNE	2.28 ± 0.03	1.50 ± 0.09	0.90 ± 0.04	0.59 ± 0.04	1.65 ± 0.04	$\mathbf{0.36} \pm \mathbf{0.02}$
	CNEE	4.50 ± 0.03	1.44 ± 0.08	1.04 ± 0.05	0.70 ± 0.03	1.61 ± 0.04	$\mathbf{0.35} \pm \mathbf{0.01}$

Table 1: Test set results (averaged over five different initializations) for the Flickr, BC, and the simulated BA and homophilous datasets. Lower is better for both metrics. The best-performing method is shown in bold; the second-best is underlined.



(a) Only individual

(b) Only spillover

(c) Both

Figure 4: Impact of balancing node representations in HINet on test CNEE (averaged over five different initializations). The x-axis shows increasing treatment assignment mechanism strength (β_{XT}). Each row represents a different DGP.

those with a low treatment rate) but perform poorly in others. Therefore, we argue that a good evaluation procedure should account for *performance across multiple counterfactual networks*. Yet, since there are $2^{|\mathcal{V}|} - 1$ possible counterfactual networks, it is computationally infeasible to evaluate all of them in larger networks. To address this, we propose two *novel evaluation metrics* that sample multiple counterfactual networks and report the average estimation error. First, the *Precision in Estimation of Heterogeneous Network Effects (PEHNE)* measures the estimation error for ITTE. Second, the *Counterfactual Network Estimation Error (CNEE)* measures the estimation error for counterfactual outcomes. The difference is that the latter puts less emphasis on the estimation of potential outcomes without any treatment, $Y_i(0, \mathbf{0})$. Further details are provided in Appendix D.

4.1. Performance on (semi-)synthetic data

Table 1 shows the test set results for the different datasets in terms of the PEHNE and CNEE metrics. HINet outperforms all comparison methods on both metrics. The GIN model consistently ranks second best on the CNEE metric, while NetEst outperforms the GIN model in terms of PEHNE on two of the four datasets. The latter is somewhat unexpected, as NetEst assumes an incorrect exposure mapping.

4.2. Impact of homophily

Figure 4 shows the performance of HINet on simulated homophilous and non-homophilous (BA) networks for three different DGP settings and for varying levels of treatment assignment mechanism strength (β_{XT}, with $\beta_{XT} = 0$ resembling an RCT). The DGP settings are: (a) individual (direct) treatment effects without interference, (b) spillover effects without direct effects, and (c) both individual and spillover effects. We visualize the test set results in terms of CNEE with and without balancing node representations. As expected, CNEE generally increases with higher values of β_{XT}, due to stronger covariate shift (Shalit et al., 2017). Notably, we also observe that balancing node representations has a greater positive impact on performance in the homophilous networks compared to the non-homophilous ones when spillover effects—both with and without individual effects—are present. This supports our hypothesis that network-level covariate shift can be (partly) mitigated by balancing representations. Nevertheless, additional results (Appendix E) indicate that while the impact of balancing is consistently larger when both homophily and both types of effects are present, it does not always lead to improved performance in settings with only spillover effects.

5. Conclusion

We introduced HINet, a novel method for estimating heterogeneous treatment effects in the presence of interference. HINet learns an exposure mapping directly from data to capture how neighbors' treatments influence an instance's outcome, while simultaneously balancing node representations. We showed that this enables effective mitigation of network-level covariate shift arising from the interaction between homophily and the treatment assignment mechanism.

References

Adhikari, S. and Zheleva, E. (2023). Inferring causal effects under heterogeneous peer influence. arXiv preprint arXiv:2305.17479.

Barabási, A.-L. and Albert, R. (1999). Emergence of scaling in random networks. Science, 286(5439):509–512.

Berrevoets, J., Jordon, J., Bica, I., Gimson, A., and van der Schaar, M. (2020). OrganITE: Optimal transplant donor organ offering using an individual treatment effect. Advances in Neural Information Processing Systems, 33:20037–20050.

Bica, I., Alaa, A. M., Jordon, J., and van der Schaar, M. (2020). Estimating counterfactual treatment outcomes over time through adversarially balanced representations. arXiv preprint arXiv:2002.04083.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. Journal of Machine Learning Research, 3(Jan):993–1022.

Caljon, D., Van Belle, J., Berrevoets, J., and Verbeke, W. (2024). Optimizing treatment allocation in the presence of interference. arXiv preprint arXiv:2410.00075.

Chen, W., Cai, R., Yang, Z., Qiao, J., Yan, Y., Li, Z., and Hao, Z. (2024). Doubly robust causal effect estimation under networked interference via targeted learning. In International Conference on Machine Learning.

Curth, A. and van der Schaar, M. (2023). In search of insights, not magic bullets: Towards demystification of the model selection dilemma in heterogeneous treatment effect estimation. In International Conference on Machine Learning, pages 6623–6642. PMLR.

Feuerriegel, S., Frauen, D., Melnychuk, V., Schweisthal, J., Hess, K., Curth, A., Bauer, S., Kilbertus, N., Kohane, I. S., and van der Schaar, M. (2024). Causal machine learning for predicting treatment outcomes. Nature Medicine, 30(4):958–968.

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

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., March, M., and Lempitsky, V. (2016). Domain-adversarial training of neural networks. Journal of Machine Learning Research, 17(59):1–35.

Greenland, S., Pearl, J., and Robins, J. M. (1999). Causal diagrams for epidemiologic research. Epidemiology, 10(1):37–48.

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

Hamilton, W., Ying, Z., and Leskovec, J. (2017). Inductive representation learning on large graphs. Advances in Neural Information Processing Systems, 30.

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

Holland, P. W. (1986). Statistics and causal inference. Journal of the American statistical Association, 81(396):945–960.

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

Jiang, S. and Sun, Y. (2022). Estimating causal effects on networked observational data via representation learning. In Proceedings of the 31st ACM International Conference on Information & Knowledge Management, pages 852–861.

Karypis, G. and Kumar, V. (1998). A fast and high quality multilevel scheme for partitioning irregular graphs. SIAM Journal on scientific Computing, 20(1):359–392.

Kingma, D. and Ba, J. (2015). Adam: A method for stochastic optimization. In International Conference on Learning Representations.

Kipf, T. N. and Welling, M. (2016). Semi-supervised classification with graph convolutional networks. In International Conference on Learning Representations.

Ma, Y. and Tresp, V. (2021). Causal inference under networked interference and intervention policy enhancement. In International Conference on Artificial Intelligence and Statistics, pages 3700–3708. PMLR.

McPherson, M., Smith-Lovin, L., and Cook, J. M. (2001). Birds of a feather: Homophily in social networks. Annual Review of Sociology, 27(1):415–444.

Newman, M. E. (2002). Assortative mixing in networks. Physical Review Letters, 89(20):208701.

Newman, M. E. (2003). Mixing patterns in networks. Physical Review E, 67(2):026126.

Ogburn, E. L. and VanderWeele, T. J. (2014). Causal diagrams for interference. Statistical Science, 29(4):559–578.

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

Shalit, U., Johansson, F. D., and Sontag, D. (2017). Estimating individual treatment effect: generalization bounds and algorithms. In International Conference on Machine Learning, pages 3076–3085. PMLR.

Xu, K., Hu, W., Leskovec, J., and Jegelka, S. (2019). How powerful are graph neural networks? In International Conference on Learning Representations.

Zhao, Z., Bai, Y., Xiong, R., Cao, Q., Ma, C., Jiang, N., Wu, F., and Kuang, K. (2024). Learning individual treatment effects under heterogeneous interference in networks. ACM Transactions on Knowledge Discovery from Data, 18(8):1–21.

A. Data-generating process

We adjust the DGP proposed by Jiang and Sun (2022). Instead of using a predefined exposure mapping $z_i = \frac{\sum_{j \in \mathcal{N}_i} t_j}{|\mathcal{N}_i|}$, we define a function that allows for heterogeneous spillover effects.

For the fully synthetic datasets, we first generate 10 features (following Jiang and Sun (2022)) from a standard normal distribution: $x_i^j \sim \mathcal{N}(0,1), j = 1, \ldots, 10$. For the semi-synthetic datasets (Flickr and BC), we follow Jiang and Sun (2022) to split the network into train, validation and test using METIS (Karypis and Kumar, 1998). Then, following Guo et al. (2020); Jiang and Sun (2022), we use Latent Dirichlet Allocation (Blei et al., 2003) to reduce the sparse features to a lower-dimensional representation. We also set the feature dimension for these datasets to 10.

For the fully synthetic datasets, we generate the network structure as follows. For the simulated BA dataset, each network of 5000 nodes (i.e., training, validation, and test) is simulated based on the Barabasi-Albert random network model (Barabási and Albert, 1999). The hyperparameter m is set to 2. For the simulated homophilous dataset, homophilous networks with 5000 nodes are generated based on the cosine similarity between the feature vectors of all node pairs in the network (some noise is added to the cosine similarity to allow unlikely edges to occur). Then, the node pairs are sorted according to cosine similarity. Edges are created between nodes with the highest cosine similarity until the average degree (number of edges per node) is equal to the average degree of the simulated BA network ($\bar{deg} = 4$).

To induce the causal structure (see Figure 2), we generate the following parameters:

$$
\begin{aligned}
&w_j^{XT} \sim \text{Unif}(-1,1) \quad \text{for } j \in \{1,2,\ldots,10\} && \mathbf{w}^{XT} = [w_1^{XT}, w_2^{XT}, \ldots, w_{10}^{XT}],\\
&w_j^{XY} \sim \text{Unif}(-1,1) \quad \text{for } j \in \{1,2,\ldots,10\} && \mathbf{w}^{XY} = [w_1^{XY}, w_2^{XY}, \ldots, w_{10}^{XY}],\\
&w_j^{TY} \sim \text{Unif}(-1,1) \quad \text{for } j \in \{1,2,\ldots,10\} && \mathbf{w}^{TY} = [w_1^{TY}, w_2^{TY}, \ldots, w_{10}^{TY}],\\
&w_j^{X_{\mathcal{N}}Y} \sim \text{Unif}(-1,1) \quad \text{for } j \in \{1,2,\ldots,10\} && \mathbf{w}^{X_{\mathcal{N}}Y} = [w_1^{X_{\mathcal{N}}Y}, w_2^{X_{\mathcal{N}}Y}, \ldots, w_{10}^{X_{\mathcal{N}}Y}],\\
&w_j^{T_{\mathcal{N}}Y} \sim \text{Unif}(-1,1) \quad \text{for } j \in \{1,2,\ldots,10\} && \mathbf{w}^{T_{\mathcal{N}}Y} = [w_1^{T_{\mathcal{N}}Y}, w_2^{T_{\mathcal{N}}Y}, \ldots, w_{10}^{T_{\mathcal{N}}Y}].
\end{aligned}
$$

These parameters quantify the effect of $\mathbf{X}_i$ on T_i, $\mathbf{X}_i$ on Y_i, the heterogeneous effect of T_i on Y_i, the effect of $\mathbf{X}_{\mathcal{N}_i}$ on Y_i, and the heterogeneous spillover effect of $\mathbf{T}_{\mathcal{N}_i}$ on Y_i, respectively. The treatment t_i is generated as follows. We first calculate ν_i as:

$$\nu_i = \beta_{XT} \cdot \mathbf{w}^{XT} \cdot \mathbf{x}_i,$$

with $\beta_{XT} \geq 0$ the treatment assignment mechanism strength and $\mathbf{x}_i = [x_1, x_2, \ldots, x_{10}]'$. Next, to set the percentage of nodes treated to approximately 25%, we calculate the 75-th percentile ν_{75} and transform $\nu' = \nu - \nu_{75}$. Finally, we apply the sigmoid function σ to ν', and obtain t_i by sampling: $t_i \sim \text{Bernoulli}(\sigma(\nu_i'))$.

To generate the outcomes, we first generate a transformed feature vector $\tilde{\mathbf{x}}_i$ by transforming half the features using the sigmoid function σ to add nonlinearities. The outcomes are obtained as follows:

$$y_i = \beta_{\text{individual}} \cdot h_i \cdot t_i + \beta_{\text{spillover}} \cdot z_i + \beta_{XY} \cdot u_i + \beta_{X_{\mathcal{N}}Y} \cdot u_{\mathcal{N}_i} + \beta_\epsilon \cdot \epsilon; \qquad \epsilon \sim \mathcal{N}(0,1),$$

with

$$
\begin{aligned}
h_i &= \mathbf{w}^{TY} \cdot \tilde{\mathbf{x}}_i, & u_i &= \mathbf{w}^{XY} \cdot \tilde{\mathbf{x}}_i,\\
z_i &= \frac{\sum_{j \in \mathcal{N}_i} t_j \cdot \mathbf{w}^{T_{\mathcal{N}}Y} \cdot \tilde{\mathbf{x}}_j}{|\mathcal{N}_i|}, & u_{\mathcal{N}_i} &= \frac{\sum_{j \in \mathcal{N}_i} \mathbf{w}^{X_{\mathcal{N}}Y} \cdot \tilde{\mathbf{x}}_j}{|\mathcal{N}_i|}.
\end{aligned}
$$

We use the following parameter values in the experiments presented in Section 4.1: $\beta_{XT} = 3, \beta_{\text{individual}} = 2, \beta_{\text{spillover}} = 2, \beta_{XY} = 1.5, \beta_{X_{\mathcal{N}}Y} = 1.5, \beta_\epsilon = 0.2$. The parameters for the experiments analyzing the impact of homophily (Section 4.2) are the same, except that $\beta_{X_{\mathcal{N}}Y} = 0$. Additionally, $\beta_{\text{individual}}$ and $\beta_{\text{spillover}}$ are set to 0 when isolating the effect of spillover and direct individual effects, respectively.

B. Measuring homophily

Homophily (McPherson et al., 2001) or assortative mixing (Newman, 2002; 2003) refers to the tendency of nodes in a network to associate with others that are similar to themselves. For example, individuals with similar interests are more

likely to be friends. The degree of assortative mixing for a given feature can be quantified using the assortativity coefficient. This coefficient is positive when an attribute is assortative, negative when it is disassortative, and zero when there is no assortativity. By calculating this coefficient for the treatment variable, we can objectively assess whether treated nodes are more likely to have treated neighbors. Similarly, outcome assortativity is positive if the outcomes of neighbors are positively correlated. These measures are reported in Table 2 for the four datasets used in our experiments. There is considerable assortativity in both outcomes and treatments in the simulated homophilous dataset. In other words, knowing the treatment and outcome of a neighbor of a randomly selected node provides information about the treatment and outcome of that node.

	Flickr	BC	Simulated BA	Simulated homophilous
Treatment assortativity	0.01	0.04	0.01	0.47
Outcome assortativity	-0.01	0.12	-0.13	0.67

Table 2: Treatment and outcome assortativity for the different datasets used in our experiments.

C. Hyperparameter selection and implementation details

Due to the fundamental problem of causal inference (Holland, 1986), individualized treatment effects are unobservable. As a result, selecting hyperparameters is challenging, since we cannot directly optimize them based on treatment effect estimation error. For standard machine learning hyperparameters, such as hidden layer size or learning rate, we can rely on the factual validation loss for hyperparameter selection. The factual validation loss is the average estimation error for outcomes actually observed in the validation set and can always be calculated. However, the factual loss may not reflect the treatment effect estimation performance. Nevertheless, this approach has been shown to work reasonably well (Curth and van der Schaar, 2023).

The weight for adversarial balancing, α, is a special type of hyperparameter. A positive α may cause the model to discard relevant information for predicting y_i in favor of constructing treatment-invariant representations, which will likely impair the factual validation loss. Consequently, if the factual loss is used to select this hyperparameter, α will often be chosen as zero—meaning that the upper branch of HINet would not be used.

However, both theoretical and empirical work suggests that balancing representations can improve treatment effect estimates (Shalit et al., 2017; Bica et al., 2020; Berrevoets et al., 2020). Based on this, we propose the following approach for hyperparameter selection. First, the standard machine learning hyperparameters are tuned using the factual validation loss. Once these hyperparameters are set, the factual loss is calculated for different values of α. As α increases, the factual loss typically increases as well. Our intuition is that a modest increase in factual loss is acceptable and merely indicates that treatment assignment bias is being mitigated. However, a substantial increase may suggest that valuable information is being discarded in favor of learning treatment-invariant representations. As a heuristic, we propose selecting the largest value of α for which the factual loss remains below $(1+p) \cdot \text{loss}_{\alpha=0}$. As a rule of thumb, we set $p = 0.10$, meaning that we allow for a maximum increase in validation error of 10%. An important advantage of this approach is that it allows $\alpha = 0$ to be selected when representation balancing would otherwise result in excessive information being discarded.

For HINet, NetEst, and SPNet, the range for α is $\{0, 0.025, 0.05, 0.1, 0.2, 0.3\}$. The other hyperparameters are selected from the ranges shown in Table 3. The GIN layers use a 2-layer MLP. The encoder block e_ϕ in HINet consists of two hidden layers.

Parameter	Value
Hidden size	$\{16, 32\}$
Num. epochs	$\{1500, 2000, 3000\}$
Learning rate	$\{0.001, 0.0005, 0.0001\}$

Table 3: Hyperparameter ranges.

The MLP blocks d_T and p_Y, as well as the MLP block in the GIN model, each consist of three hidden layers. All MLP blocks (for every method) use ReLU activations after each layer. Other hyperparameters are set to author-recommended values. Each model is trained using the Adam optimizer (Kingma and Ba, 2015) with weight decay set to 0.001. For all models except SPNet, we use the implementation provided by Jiang and Sun (2022). Since there is no publicly available

implementation of SPNet, we implemented it ourselves based on the description in Zhao et al. (2024). Reported results are averages over five different initializations, affecting both weight initialization and training data shuffling. Our code is available at https://github.com/daan-caljon/HINet.

D. Performance metrics

In a traditional no-interference setting with binary treatment, there is only one counterfactual: the outcome under the opposite treatment, $Y_i(1-t_i)$. In network settings, however, counterfactuals must be considered at the level of the entire network, since the potential outcome of any given unit may depend on the treatments of others. Therefore, we argue that a good evaluation procedure should account for *counterfactual networks* rather than only individual-level counterfactuals. A counterfactual network is a network in which at least one unit receives a different treatment than in the observed network. Note that the number of counterfactual networks is $2^{|\mathcal{V}|}-1$, each with $|\mathcal{V}|$ potential outcomes.

When simulated data is available, the Precision in Estimation of Heterogeneous Effects (PEHE) (Hill, 2011) is often used to evaluate methods in a traditional no-interference setting with binary treatment. PEHE is defined as the root mean squared error of the estimated Conditional Average Treatment Effects (CATEs), which is uniquely defined since there is a single counterfactual. In the presence of interference, however, this is no longer the case and estimated ITTEs could in principle be *evaluated for each counterfactual network* $j = 1, 2, \ldots, 2^{|\mathcal{V}|}-1$ in a similar manner. For large networks, however, this becomes computationally intractable. Therefore, we propose two new metrics that sample a diverse set of counterfactual networks. These metrics are inspired by the Mean Integrated Squared Error (MISE), which is used for evaluating treatment effects with a continuous treatment (Schwab et al., 2020). In the continuous setting, a similar challenge arises due to the existence of more than two counterfactuals per unit.

The first proposed metric is the Precision in Estimation of Heterogeneous Network Effects (PEHNE), which evaluates ITTE estimation error over a range of counterfactual networks. The calculation of PEHNE is described in Algorithm 1. In total, m counterfactual networks are sampled. By sampling treatments according to the percentage p_j, we ensure that models are evaluated across a variety of treatment rates.

Algorithm 1 PEHNE calculation

for $j = 1, 2, ..., m$ **do**
  Percentage of nodes to treat $p_j = \frac{100 \cdot j}{m}\%$
  Sample treatment for each node i: $t_i^j \sim \text{Bernoulli}(p_j)$
  Estimate ITTE for each node i: $\hat{\omega}_i^j(t_i^j, \mathbf{t}_{\mathcal{N}_i}^j) = \hat{y}_i^j(t_i^j, \mathbf{t}_{\mathcal{N}_i}^j) - \hat{y}_i^j(0, \mathbf{0})$
  Calculate $\text{MSE}_j = \frac{1}{|\mathcal{V}|}\sum_i(\omega_i^j(t_i^j, \mathbf{t}_{\mathcal{N}_i}^j) - \hat{\omega}_i^j(t_i^j, \mathbf{t}_{\mathcal{N}_i}^j))^2$
end for
Return $\text{PEHNE} = \frac{1}{m}\sum_j \text{MSE}_j$

The second proposed metric is the Counterfactual Network Estimation Error (CNEE), which evaluates counterfactual outcome estimation error over a range of counterfactual networks. The calculation of CNEE is described in Algorithm 2. PEHNE places strong emhasis on the estimation of the "zero" counterfactual network, i.e., the network in which no unit receives treatment, because of the term $Y_i(0, \mathbf{0})$ in Equation (1). If a model estimates these outcomes poorly, its performance in terms of PEHNE will be significantly penalized. In contrast, CNEE assigns equal importance to all sampled counterfactual networks.

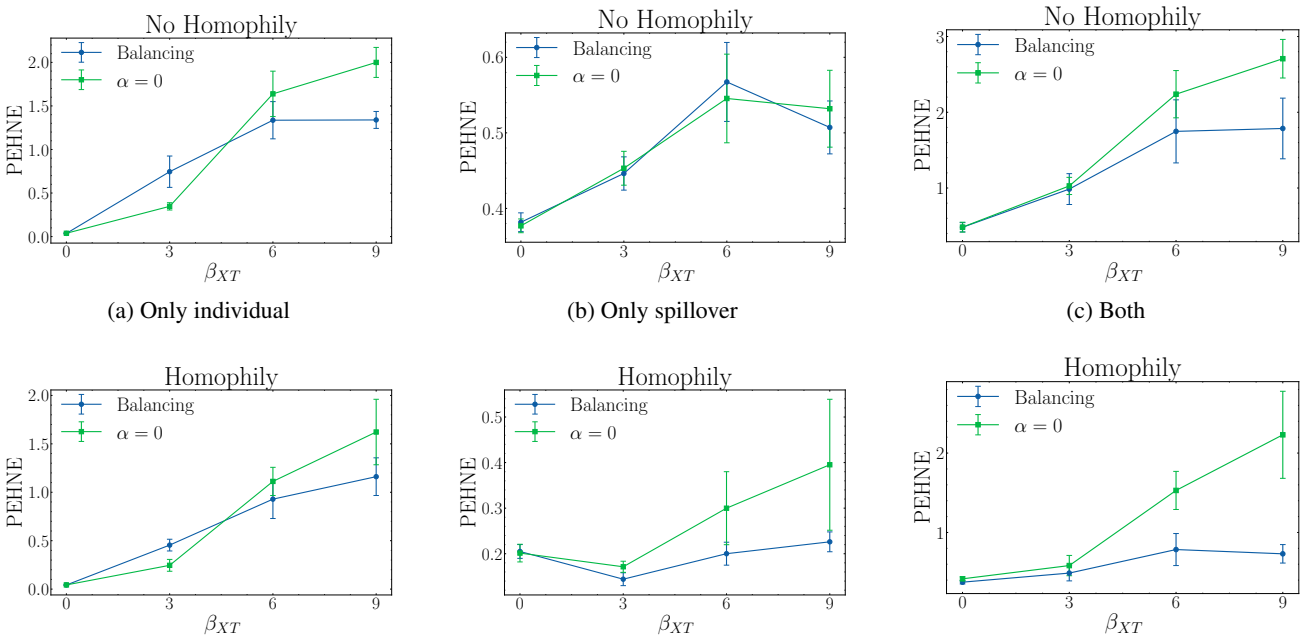
Algorithm 2 CNEE calculation

for $j = 1, 2, ..., m$ **do**
  Percentage of nodes to treat $p_j = \frac{100 \cdot j}{m}\%$
  Sample treatment for each node i: $t_i^j \sim \text{Bernoulli}(p_j)$
  Estimate the potential outcome $Y_i(t_i, \mathbf{t}_{\mathcal{N}_i})$ for each node i: $\hat{y}_i^j(t_i^j, \mathbf{t}_{\mathcal{N}_i}^j)$
  Calculate $\text{MSE}_j = \frac{1}{|\mathcal{V}|}\sum_i(y_i^j(t_i^j, \mathbf{t}_{\mathcal{N}_i}^j) - \hat{y}_i^j(t_i^j, \mathbf{t}_{\mathcal{N}_i}^j))^2$
end for
Return $\text{CNEE} = \frac{1}{m}\sum_j \text{MSE}_j$

In our experiments, we set $m = 50$ for both PEHNE and CNEE. Note that these metrics are used solely for performance evaluation, not for hyperparameter tuning (see Appendix C), as they cannot be calculated in practice from observational data—they require that all potential outcomes be known. Consequently, validation PEHNE/CNEE cannot be used for hyperparameter selection.

E. Additional results

In Figure 5, we visualize the impact of balancing node representations on the test set results in terms of PEHNE. The results are very similar to those presented for CNEE in Figure 4. When there are only individual effects, balancing improves performance in both the homophilous and non-homophilous networks at high values of β_{XT}. However, a difference between the non-homophilous and homophilous networks emerges when spillover effects are present. When there are only spillover effects, balancing considerably improves performance for the homophilous network, but does not have the same effect for the non-homophilous network. When both individual and spillover effects are present, the performance gain from balancing is relatively larger under homophily.

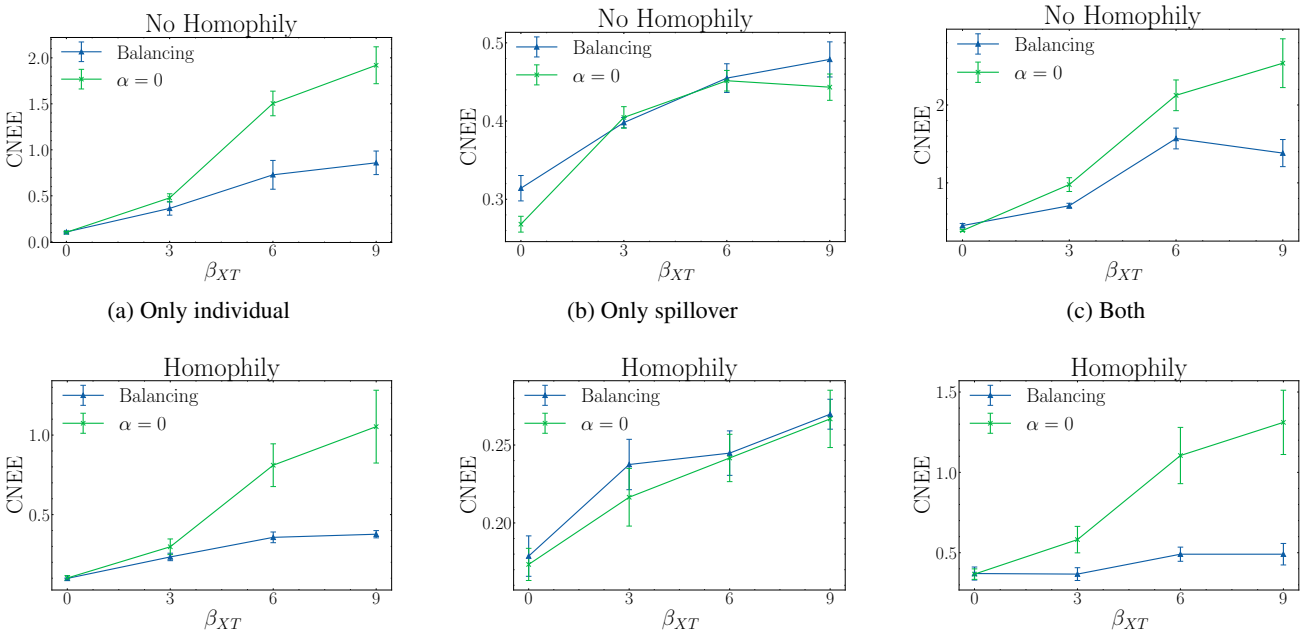


(a) Only individual (b) Only spillover (c) Both

(d) Only individual (e) Only spillover (f) Both

Figure 5: Impact of balancing node representations in HINet on test PEHNE (averaged over five different initializations). The x-axis shows increasing treatment assignment strength (β_{XT}). Each column corresponds to a different DGP. The top and bottom rows present the results for the non-homophilous (BA) and homophilous networks, respectively.

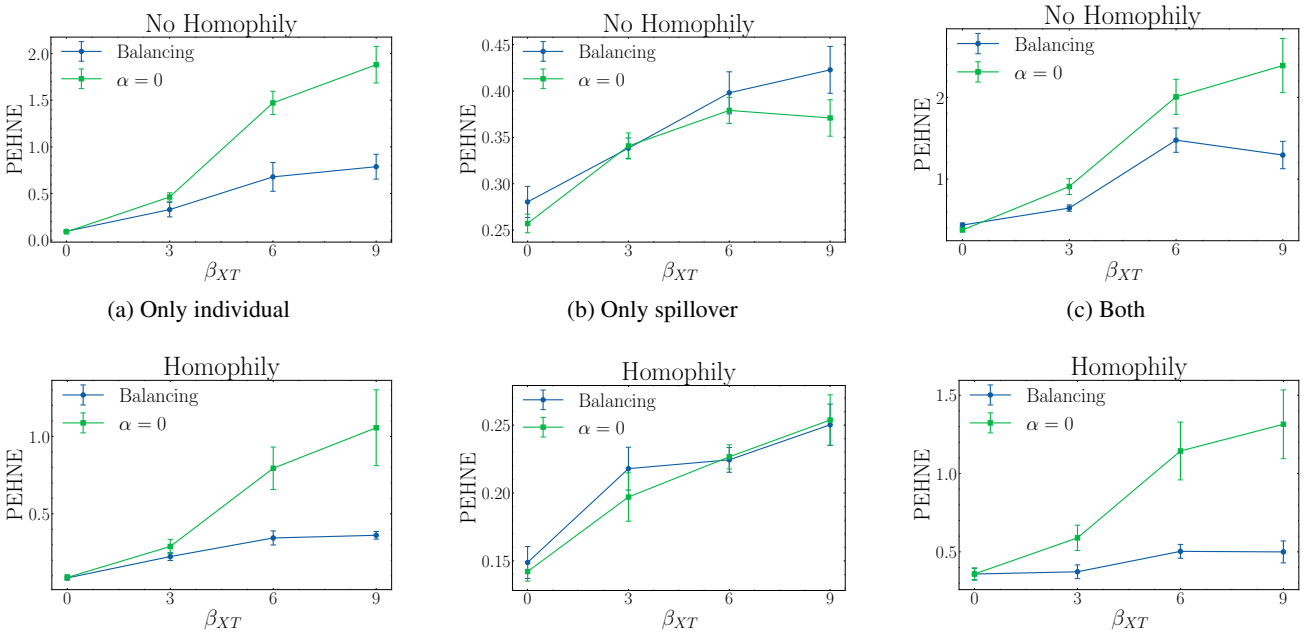
In Figures 6 and 7, we visualize the impact of balancing node representations on the test set result for $\beta_{X_{\mathcal{N}}Y} = 1.5$ (as used in the experiments in Section 4.1). In this setting, balancing is important for accurately estimating treatment effects when the DGP includes either only an individual treatment effect or both an individual and a spillover effect. In contrast to the findings in Section 4.2, balancing does not seem to affect estimation performance when only spillover effects are present. A possible explanation is that, in this setting, the effect of $\mathbf{X}_{\mathcal{N}_i}$ on Y_i is substantially stronger than the effect of $\mathbf{T}_{\mathcal{N}_i}$ on Y_i. As a result, learning treatment-invariant representations may discard too much relevant information. However, this hypothesis requires further investigation. Nevertheless, consistent with the findings in Section 4.2, the effect of balancing appears to be relatively larger in the homophilous setting when both effects are present.



(a) Only individual (b) Only spillover (c) Both

(d) Only individual (e) Only spillover (f) Both

Figure 6: Impact of balancing node representations in HINet on test CNEE (averaged over five different initializations) for $\beta_{X_{\mathcal{N}}Y} = 1.5$. The x-axis shows increasing treatment assignment strength (β_{XT}). Each column corresponds to a different DGP. The top and bottom rows present the results for the non-homophilous (BA) and homophilous networks, respectively.



(a) Only individual (b) Only spillover (c) Both

(d) Only individual (e) Only spillover (f) Both

Figure 7: Impact of balancing node representations in HINet on test PEHNE (averaged over five different initializations) for $\beta_{X_{\mathcal{N}}Y} = 1.5$. The x-axis shows increasing treatment assignment strength (β_{XT}). Each column corresponds to a different DGP. The top and bottom rows present the results for the non-homophilous (BA) and homophilous networks, respectively.