Graph Neural Networks for Treatment Effect Prediction

George Panagopoulos[1], Daniele Malitesta[2], Fragkiskos D. Malliaros[2], and Jun Pang[1]

[1] University of Luxembourg, 6 avenue de la Fonte, Esch-sur-Alzette, Luxembourg
{georgios.panagopoulos, jun.pang}@uni.lu
[2] Université Paris-Saclay, CentraleSupélec, Inria, Gif-sur-Yvette, France
{daniele.malitesta, fragkiskos.malliaros}@centralesupelec.fr

Abstract. Estimating causal effects in e-commerce tends to involve costly treatment assignments which can be impractical in large-scale settings. Leveraging machine learning to predict such treatment effects without actual intervention is a standard practice to diminish the risk. However, existing methods for treatment effect prediction tend to rely on training sets of substantial size, which are built from real experiments and are thus inherently risky to create. In this work we propose a graph neural network to diminish the required training set size, relying on graphs that are common in e-commerce data. Specifically, we view the problem as node regression with a restricted number of labeled instances, develop a two-model neural architecture akin to previous causal effect estimators, and test varying message-passing layers for encoding. Furthermore, as an extra step, we combine the model with an acquisition function to guide the creation of the training set in settings with extremely low experimental budget. The framework is flexible since each step can be used separately with other models or policies. The experiments on real large-scale networks indicate a clear advantage of our methodology over the state of the art, which in many cases performs close to random underlining the need for models that can generalize with limited labeled samples to reduce experimental risks.

Keywords: Graph Neural Networks · Causal Inference · Active Learning.

1 Introduction

Statistical tests for causal inference are ubiquitous, from drug discovery [50] and psychology [56] to social studies and online platforms—being at the core of decision-making. A prevalent example is A/B testing [18], the de facto way to evaluate the potential of a system change before applying it in scenarios such as e-commerce. Typically, the population is split into two random groups, and the treatment is assigned to one of them ($T = 1$). The difference between their response variable Y (e.g., time spent on the system) quantifies the average treatment effect (ATE) of the randomized control trial (RCT) [40], i.e., $Y_T - Y_C = \mathbb{E}[Y|T = 1] - \mathbb{E}[Y|T = 0]$, where T and C refer to the treatment and control

groups, respectively. In this setting, however, we risk causing churn on a massive scale if the proposed change is not effective [2]. It is thus more wise to test on a smaller scale and try to extrapolate our findings. In these cases, we can build a model that predicts the outcome without actually intervening, based on the samples' confounders, e.g., for a user, this could be the purchase history and demographics. The core hypothesis is that given the common assumption of unconfoundedness [9], samples with similar confounders will exhibit similar outcomes under the same treatment. Developing a model that can predict the effect of a treatment on unseen samples, is referred to as *uplift modeling* (UM) [38], mainly stemming from business context applications. The uplift refers to the prediction of the outcome's change due to the treatment i.e. the ATE, for a set of samples [11]. The problem is akin to individual treatment effect (ITE) estimation, i.e., predicting the outcome of the sample with and without treatment (counterfactual) and computing their difference.

In this work, our attention is directed towards a real-world scenario of a marketing campaign where promoting is costly/risky/time-consuming—a typical uplift modeling setting [11]. In this context, we aim to rank the whole user base based on how they will respond to the campaign, i.e., how much more they will consume if they receive a coupon. The coupon resembles a treatment and the difference in consumption is the effect. We assume there is a fixed budget for the number of users that can participate in the experiment and a predefined random balanced treatment allocation, to diminish the risks and costs. The problem can then be broken down into two subproblems:

- Which users should participate in the experiment?
- How to use this experiment to predict the outcome of the rest of the dataset?

We approach the first problem with an active learning formulation [45], where we sequentially choose subsets of the samples to label in order to maximize the model's effectiveness until we reach our budget. The latter problem pertains to the model's capacity to generalize with limited supervision, which can be cast as a semi-supervised learning problem where graph neural networks (GNNs) are rather effective [28]. GNNs utilize the relational structure of samples to facilitate generalization when the training labels are scarce. Moreover, the majority of online systems can be represented as heterogeneous graphs, e.g., user-product purchases (e-commerce [54]) or follow relationships (social media [12]).Thus, we frame the problem as an active semi-supervised learning task on a bipartite graph and address it in alternating rounds.

Contributions. Our contributions can be summarized as follows:

1. We develop a novel modular framework, based on two steps, a GNN, and an active learning method, to address the need for limited supervision in UM, moving from the standard "70%-80%" train set rule [10,20] down to 5%-20%.
2. We formulate UM for networks and test it in an open, large-scale network with real experimental annotations. To the best of our knowledge, this is the first such attempt in the literature. Moreover, we focus on continuous

outcomes, which are relatively understudied compared to binary UM [53] but equally prevalent in the real world.
3. We conduct experiments with models from the UM and ITE literature, including neural, tree-based, and graph-aware methods, and showcase that the proposed methodology surpasses the state-of-the-art substantialy.

2 Related Work

One of the most prevalent applications of machine learning (ML) in causality is estimating the probability of a sample being assigned to a treatment i.e. the propensity scores using the confounders, which can uncover design bias [30]. Models that predict the propensity score and the outcome can be used in tandem in the double ML framework [5] which is provably doubly robust (DR), in the sense that it is consistent if either the propensity or the outcome predictive model is consistent. DR has been utilized for heterogeneous causal effect estimation [27], which is similar to UM. Simpler models use treatment as an extra input (S-Learner). Two models that learn the output of treatment and control groups (T-Learner) are prevalent [37] and are the basis for the successful causal meta-learners (R/X-Learner) [29]. Class transformation with regularized Logistic Regression [42] or XGBoost [48] is also very effective for binary outcomes and under balanced treatment assignments. Finally, UpliftTrees [44] have custom splitting criteria based on the outcome distribution in the treatment groups of the tree nodes.

The effect of the network has not been examined in the context of UM, but it has for ITE. One of the first works that proposed an adjustment to the causal effect estimation based on the network was Arbour et al. [1]. Veitch et al. [52] developed a neural method to identify hidden confounders in interconnected samples. They extend the doubly robustness for non-iid data, and build a model that relies on random-walk-based node representations. The same team proposed Dragonnet [47], a combined neural architecture that is split into predicting the outcomes under each treatment and the propensity score.

Guo et al. [19] developed an ITE model with GNN encoding in the initial layers and split output layers for each treatment. Furthermore, it minimizes the distance between the representations' distribution under different treatments. This stems from a seminal work on bounding the generalization error of ITE models by the sum of the given model's standard error and the distance between the treatment and control confounders' distributions [46]. Similar GNNs have been developed based on multi-task and adversarial learning [6,24], the geometric curvature of the network [13], or network with multiple relations [31]. An effort to predict ITE considering both the network confounding and the interference was made using hypergraph neural networks [34]. We refer the reader to Appendix A where we make a distinction between confounding and interference and justify why we follow the literature [52,6,13]on the assumption for the latter.

One important difference between UM and ITE is that the latter defines distinctly the counterfactual prediction and requires the respective ground truth

for evaluation, which renders the use of simulated data imperative [35]. Hence, the aforementioned works on network-based ITE are evaluated with simulated experiments on observational data and hardcoded network confounding. Our work uses, for the first time, an open network with ground-truth experimental variables. Furthermore, we address a network with heterogeneous nodes, a setting prevalent in e-commerce. HINITE [31], which uses one type of nodes but multiple relations, is in a similar heterogeneous direction but not comparable.

3 Proposed Methodology

We consider the e-commerce scenario where data is commonly structured through bipartite, undirected graphs e.g. user-product. Given the effectiveness of GNNs in semi-supervised learning [21], we create a general framework that can be tested with several GNN layers. The model can be used as is, but we additionally define an active learning method to build the training set iteratively based on the model's uncertainty and the samples' properties. In the following, the scalars are represented by a lowercase letter, the vectors or sets with an upper case, and the matrices with an upper case bold letter.

3.1 Uplift Modeling with Graph Neural Networks (UMGNet)

Let $u \in U$ and $p \in P$ be a user and a product respectively in the recommendation system, having $|U| = n$ users and $|P| = m$ products in total. Then, let $\mathbf{R} \in \mathbb{R}^{n \times m}$ be the user-product interaction matrix, where $R_{up} = 1$ if there is a recorded interaction between user u and product p, 0 otherwise. Moreover, let each user u be represented by a tuple $Z_u = (\mathbf{X}_u, T_u, Y_u)$ where $\mathbf{X}_U \in \mathbb{R}^{n \times d}$ are the covariates representing features of dimensionality d from all n users, while $Y_U \in \mathbb{R}^n$ are the continuous trial outcomes, and $T_U \in \{0, 1\}^n$ is the treatment assignment.

Based on the Neyman-Rubin framework of potential outcomes [41] and using the *do* operator introduced by Pearl [36], we define the average treatment effect for the population as:

$$ATE = \mathbb{E}[Y_U | do(T_U = 1)] - \mathbb{E}[Y_U | do(T_U = 0)]. \tag{1}$$

If we consider the law of total expectation and the unconfoundedness [9], i.e., that the confounders $\mathbf{X}$ block all possible ways that the outcome depends on treatment assignment, we can contend the observed change is an unbiased causal effect from the treatment. Taking into consideration once again the user-product interaction matrix $\mathbf{R}$ as an extra confounder, we have:

$$ATE = \mathbb{E}[Y_U | \mathbf{X}_U, \mathbf{R}, T_U = 1] - \mathbb{E}[Y_U | \mathbf{X}_U, \mathbf{R}, T_U = 0]. \tag{2}$$

Training one model to predict $\mathbb{E}[Y_U | \mathbf{X}_U, \mathbf{R}, T_U]$, e.g., the S-Learner [29], has a specific disadvantage: we expect that Y and $\mathbf{X}$ differ between $T = 0$ and $T = 1$. To this end, the two-model methods have exhibited improved performance [29], especially in scenarios with imbalanced assignments. However,

two model approaches [29] do not allow for the sharing of obvious knowledge overlaps. As mentioned in Section 2, this can be alleviated by developing a common neural architecture for the first layers [25], including treatment prediction [47] and random walk-based confounders [52]. That motivates us to adapt DRAGONNET [47] for bipartite graphs with GNN encoding.

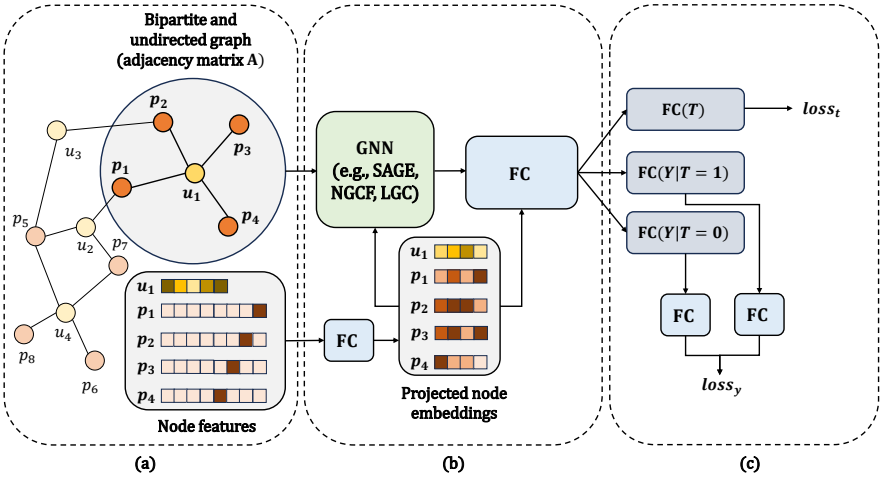

Fig. 1. Schematic representation of UMGNET. First **(a)**, the bipartite and undirected user-product graph, along with the node features, are injected into the framework. Second **(b)**, node features for users and products are projected to the same latent space through an FC layer and used as input to the GNN model; another FC layer takes the GNN's output and input to predict the regression outcome. Third **(c)**, outputs for $T = 1$ and $T = 0$ are considered separately and injected into two different FC layers to calculate $loss_y$; the general output of the regression FC is used to calculate $loss_t$.

We define the user-product bipartite and undirected graph as $G = (U, P, \mathbf{A})$, where we already introduced U and P, while $\mathbf{A} \in \mathbb{R}^{(n+m)\times(n+m)}$ is the adjacency matrix of G:

$$\mathbf{A} = \begin{pmatrix} 0 & \mathbf{R} \\ \mathbf{R}^\top & 0 \end{pmatrix}. \tag{3}$$

The users' features are represented as $\mathbf{X}_U \in \mathbb{R}^{n\times d}$, and the product features as $\mathbf{X}_P \in \mathbb{R}^{m\times m}$. First, we project users and products to the same latent space through a fully-connected (FC) layer. The obtained projections are horizontally concatenated to get a common set of node embeddings $\mathbf{X} \in \mathbb{R}^{(n+m)\times w}$, as follows:

$$\mathbf{X} = \text{concat}\big([\text{ReLU}(\mathbf{X}_U\mathbf{W}_U), \text{ReLU}(\mathbf{X}_P\mathbf{W}_P)]\big), \tag{4}$$

where $\mathbf{W}_U \in \mathbb{R}^{d\times w}$ and $\mathbf{W}_P \in \mathbb{R}^{m\times w}$.

Subsequently, we learn graph features leveraging GNN layers. For this part, we have examined different architectures, including GraphSAGE [21], NGCF [54],

and LGC [22]:

$$\mathbf{H}_1 = \mathrm{GNN}(\mathbf{A}, \mathbf{X}). \tag{5}$$

Through another FC layer, the representations are then broken into two separate paths for $T = 1$ and $T = 0$ denoted as t and c, respectively. We also use a residual connection from $\mathbf{X}$, thus for $T = 1$ we have:

$$\mathbf{H}^t_{i+1} = \mathrm{ReLU}\big(\big[\mathbf{X}[0:n], \mathbf{H}_1[0:n]\big]\mathbf{W}^t_i\big), \tag{6}$$

where we are slicing the matrices to keep the first n rows corresponding to the user representations assuming the horizontal concatenation in Eq. (4). Two final FC layers (with F as depth) predict the outcome under treatment t and control c:

$$\hat{Y}^c = \mathbf{H}^c_F \mathbf{W}^c_F, \qquad \hat{Y}^t = \mathbf{H}^t_F \mathbf{W}^t_F.$$

The loss takes into consideration only the factual outcomes, i.e., where the treatment vector $\mathbf{T}$ has 1 for the t output and 0 for the c:

$$loss_y = \frac{1}{n}\sum_{i=0}^{n}\big(T_i(\hat{Y}^t_i - Y_i)^2 + (1 - T_i)(\hat{Y}^c_i - Y_i)^2\big). \tag{7}$$

We refer to this generic GNN-based uplift modeling framework as UMGNet. A schematic representation of the model is shown in Fig. 1. As mentioned above, we have examined various instances of the model with different GNN architectures. Lastly, inspired by [47], we consider a variant denoted by UMGNet-Dr, in which we add an extra output layer that predicts the treatment. For this model, we add the following loss term to the one in Eq. (7):

$$loss_t = \frac{1}{n}\sum_{i=0}^{n}\mathrm{CrossEntropy}\big(T, \mathrm{Sigmoid}(\mathbf{H}^\top_F \mathbf{W}^\top_F)\big). \tag{8}$$

3.2 Active Learning for Uplift GNNs (UMGNet-AL)

Active learning relies on the structure of the data and the uncertainty of the model to build iteratively the train set in scenarios where data labeling is costly [45]. We can break it down into two parts: the first is the uncertainty estimation and the acquisition function over non-labeled samples, while the second is the active learning policy we use to gather new samples to label i.e. to test in our case.

Uncertainty estimation. Uncertainty estimation in graph learning is an active research topic [49,23]. The uncertainty can be distinguished based on its source: the model (epistemic) or the data (Aleatoric). We will employ an unprincipled yet effective practice to quantify epistemic uncertainty by measuring the variance on the responses of an ensemble of models, or simply performing dropout multiple times [17]. The dropout mask is random during inference, so

the model will produce different outputs for each test sample. This represents a Bayesian approximation [14].

Aleatoric uncertainty can be measured using the structure of the graph and the feature distance. Acquisition functions based on diverse criteria, including both types of uncertainties have proven more effective in node-level tasks [57,16]. We adopt a similar approach and define $D_u, \forall u \in U$ to be the degree, Q_u is the model's uncertainty, and b is the budget for new training samples in each iteration. According to the literature [57,16], we define diversity based on feature clustering. Specifically, we cluster the samples with a k-means algorithm in a predefined number of clusters and store the assignments in $C \in \mathbb{Z}^n$. Then we calculate the distance $M_u \in \mathbb{R}^n$ between each sample and its cluster centroid and compute the budget for each cluster based on its relative size and b, in $C_b \in \mathbb{Z}^{|C|}$. We can cast the problem of choosing a batch to add to the training set as ranking based on U, D and M with the first constraint being on the batch size. The second constraint, which represents diversity, pertains to how many samples of a given cluster should be included in the train set in each round. Finally, we want to keep the train set balanced in terms of treatment and control, hence the third constraint:

$$
\begin{aligned}
\text{maximize} \quad & O = \sum_{u=1}^{N} x_u (Q_u + D_u + M_u) \\
\text{subject to}: & \sum_{u=1}^{n} x_u \leq b \\
& \sum_{u=1}^{n} x_u C_u \leq C_b[C_u] \\
& \sum_{u=1}^{n} x_u T_u \leq \frac{b}{2} \\
& x_u \in \{0,1\}, \quad \text{for } u = 1, 2, \ldots, n.
\end{aligned}
$$

The problem can be solved greedily in each iteration of batch selection, while D, C, and C_b are precomputed. In practice, we weigh each of the three criteria in the objective function with a coefficient in $[0, 1]$ based on the results of validation.

Acquisition policy. Thompson sampling [43], and upper-confidence bandits [15] are some of the most popular policies to perform active learning [45]. The lack of knowledge at the initial iterations justifies the use of stochastic policies. However, greedy selection has been effective in batch active learning, and it is actually provably near-optimal in some cases [55]. Since our acquisition function does not rely solely on the model's output, which is less trustworthy due to the initial lack of samples, we are going to use a greedy policy and solve the ranking problem in every iteration. The whole procedure can be seen in Algorithm 1 for 5 iterations, assuming UMGNet includes by default the parameters $\mathbf{X}, T, Y, \mathbf{A}$ for clarity.

Algorithm 1 UMGNet AL

Input: $O, D, M, b,$ UMGNet
  $S \leftarrow \{\text{argmax}_{s \subset U} O(D, M, T, b)\}$
  while $|S| \leq 5 * b$ **do**
    Train UMGNet(S)
    $Q, \hat{Y} \leftarrow$ UMGNet$(U \setminus S)$
    $S \leftarrow S \cup \{\text{argmax}_{s \subset U} O(Q, D, M, T, b)\}$
  end while
Output: $\hat{Y}$

4 Experimental Evaluation

We report on the experimental results to empirically justify the soundness of our framework. First, we present the main datasets and how we built them. Second, we outline the benchmarking models adopted for this work. Finally, we discuss the obtained results and answer different questions with an ablation study.

4.1 Datasets

RetailHero. The *RetailHero* [39] dataset is comprised of two equal groups of users undergoing a marketing campaign, along with their product purchase history and some relevant features. The treatment group has received promotional SMS texts and the binary outcome corresponds to whether the user has made a purchase or not after the promotional SMS. We follow suit from the literature and the original competition and utilize features such as age, sex, coupon issue time, coupon redeem time, and delay between issue and redeem time. The products are represented by one-hot encoding. We created a user-product graph based on the purchases performed before the time that the promotional coupon was redeemed from treated users. Note that issue and redeem timestamps are added by the organizers for untreated users as well, and they follow similar properties with the redeem time for treated users. Similar to a real-life experiment, purchases done after the intervention are not accessible in the initial form of the graph, except for training samples, i.e., users that have indeed taken part in the experiment in real life. We set two types of continuous outcomes Y:

- *RHC*: The difference between the average money spent before and after coupon redeem time. $ATE = 2.60, \bar{Y} = 266, \sigma(Y) = 521$.
- *RHP*: The average money spent after coupon redeem time. $ATE = 1.95, \bar{Y} = 423, \sigma(Y) = 387$.

The second outcome measures how much more prone treated users are to spend money compared to the control. The first is similar but takes into account the spending of each user before the treatment as a means of normalization.

Table 1. Statistics of the bipartite datasets (directed).

Dataset	Nodes	E (before T)	E (after)	$T = 0$	$T = 1$
RetailHero	(180,653+40,542)	2,522,096	12,021,243	90,097	90,556
MovieLens	(59,047+162,541)	9,369,966	15,630,129	29,518	29,529

MovieLens. We utilize the *Movielens25* and filter the users based on the minimum number of ratings.We extract the features of the movie nodes using a Universal Sentence Encoder-Lite [3] on the concatenation of the title, year, and genres and bring them down to 16 dimensions using principal component analysis. The adjacency is defined as in Eq. 3 from the movie-viewer bipartite network. We define the treatment as $T = 1$ if the movie's average rating is over the median of 3.15 or $T = 0$ otherwise, akin to [34,31]. Similar to the same literature, the regression outcome is simulated 5 times: $y_s = \text{ReLU}(w_s x + w_t t + e_s)$, where $w_s \in \mathbb{U}(10, 20)$ and $e_s \in\sim \mathcal{N}(10, 5^2)$ are random variables of the simulation.

4.2 Benchmark Models

To facilitate comparison with the state-of-the-art, we utilize a variety of methodologies from meta learners S, T, X, R [29] and doubly robust DR [27] to uplift trees UpliftTree [38], and neural models CEVAE [32], and Dragonnet [47], which outperform CFR and TARNET [46] in similar settings. We rely on the implementations from the CausalML package [4] for all, using XGBoost as the output and the build-in *elastic net* as the propensity model whenever required. We include NetDeconf [19] as the graph-aware benchmark from ITE literature but remove the Wasserstein distance cost because it fails to run on a GPU with 32GB. All methods are run with their suggested parameters.

4.3 Experiments

In contrast to the aforementioned studies in ITE [19,35], we can not measure the counterfactual error in the sample level because it does not exist. Moreover, since our task is not binary, we can not utilize the uplift curve [11]. Even if we could, our application focuses on the potential of the top ranked users in order to target them with the campaign, hence it is not sensible to focus on the estimation of the whole dataset. Instead, we rely on the realistic evaluation of our scenario that was also the success criterion for our main dataset[3]:
up@40/20: We take the top 40% and 20% of the test samples sorted based on their predicted uplifts, and measure the real ATE in this set [39].

Settings. To evaluate the models' capacity in semi-supervised generalization, we utilize 5 and 20-fold cross-validation where the test part is set for training and

[3] `https://ods.ai/competitions/x5-retailhero-uplift-modeling`.

vice versa i.e. we will have 20% and 5% training set size and 5 and 20 iterations respectively. We run each method 5 times with random seeds from 0 to 4 and log the average and standard deviation of the respective metric in k-fold validation. For all datasets, we set the learning rate to 0.01 with a weight decay of 0.00001. We used ReLU as activation function. The dropout rate is set to 0.4, the number of epochs to 2000 and the hidden layers to (64, 64, 32). Finally, the number of clusters is set to 50 and the coefficients for the acquisition function are 0.2, 0.1, and 0.7, respectively.

In the active learning setting, denoted as UMGNet-AL which utilizes the UMGNet-SAGE model with the proposed acquisition function, instead of 20 and 5-fold, we train an initial model in 1% and 4% of the dataset and increase the train set up to 5%/20% respectively with 5 batch queries using a greedy policy. The experiments are performed with an Nvidia V100 16 GB and 32 GB RAM. The results for *RetailHero* can be seen in Tables 2, 3. The best result is indicated in **boldface** and the second-best is underlined.

Results and discussion. It can be seen that the proposed methods outperform the benchmarks for both outcome variables. Some benchmarks even exhibit negative uplifts, which means the predicted sets have higher Y_c then Y_t. The regular ATE is the $\bar{Y}_t - \bar{Y}_c$ of the whole dataset, and it is what we expect to get by a random balanced subset of users without any ranking. If we consider it as a baseline, we see that in most cases the benchmarks tend to be under the ATE. This is justified by the reduced supervision, which severely diminishes the benchmarks' generalization, moving their results closer to a random sample. In contrast, the proposed methods achieve consistently and sometimes considerably higher uplift than the ATE, signifying their ability to generalize at this setting. We actually see that the difference between the proposed methods and the benchmarks is close to or sometimes larger than the actual ATE.

Table 2. Uplift metrics of the predicted sets in the *RHC*. Regular $ATE = 2.60$.

Model	20% training size		5% training size	
	up@40	up@20	up@40	up@20
S-XGB	-1.63 ± 1.85	-0.67 ± 3.6	1.13 ± 1.09	2.43 ± 1.79
T-XGB	0.58 ± 1.48	3.25 ± 2.72	-0.05 ± 0.32	1.33 ± 1.51
X-XGB	-2.55 ± 1.97	-2.62 ± 1.47	0.21 ± 1.02	1.19 ± 1.64
R-XGB	1.77 ± 0.97	3.70 ± 1.62	2.01 ± 0.53	5.2 ± 1.81
DR-XGB	-0.26 ± 1.04	1.18 ± 1.60	0.61 ± 1.00	2.65 ± 1.05
UpliftTree	$\underline{5.95 \pm 2.43}$	2.38 ± 7.00	$\underline{3.74 \pm 0.77}$	2.73 ± 1.83
CEVAE	3.10 ± 0.59	3.23 ± 1.63	2.88 ± 0.75	3.34 ± 0.70
Dragonnet	2.24 ± 0.14	3.08 ± 0.34	2.26 ± 0.04	3.01 ± 0.10
NetDeconf	1.89 ± 1.86	2.73 ± 1.39	1.08 ± 0.46	2.06 ± 0.62
UMGNet-SAGE	3.20 ± 0.25	$\mathbf{6.48 \pm 0.70}$	2.66 ± 0.75	$\underline{5.69 \pm 1.01}$
UMGNet-AL	$\mathbf{6.27 \pm 3.00}$	$\underline{4.64 \pm 3.60}$	$\mathbf{5.83 \pm 2.75}$	$\mathbf{6.83 \pm 3.77}$

Table 3. Uplift metrics of the predicted sets in the *RHP*. Regular $ATE = 1.95$.

Model	20% training size		5% training size	
	up@40	up@20	up@40	up@20
S-XGB	3.15 ± 2.07	4.54 ± 1.10	2.48 ± 0.93	3.56 ± 0.96
T-XGB	2.67 ± 0.26	$\underline{5.65 \pm 1.16}$	2.53 ± 0.39	4.65 ± 0.65
X-XGB	2.96 ± 1.01	5.33 ± 1.57	2.29 ± 0.35	4.04 ± 1.06
R-XGB	2.66 ± 1.10	4.58 ± 0.88	2.90 ± 0.32	4.11 ± 0.67
DR-XGB	3.23 ± 1.22	3.86 ± 1.38	2.77 ± 0.19	3.61 ± 1.00
UpliftTree	2.84 ± 0.75	5.01 ± 0.45	1.77 ± 0.70	3.26 ± 0.86
CEVAE	2.37 ± 1.67	3.26 ± 1.89	1.89 ± 0.54	2.10 ± 0.54
Dragonnet	0.21 ± 0.16	-0.01 ± 0.26	0.21 ± 0.01	-0.104 ± 0.01
NetDeconf	0.84 ± 1.28	2.00 ± 1.00	0.45 ± 0.63	0.770 ± 1.20
UMGNet-SAGE	$\mathbf{5.01 \pm 2.16}$	$\mathbf{7.28 \pm 4.34}$	$\underline{3.99 \pm 2.00}$	$\underline{5.07 \pm 3.61}$
UMGNet-AL	$\underline{4.11 \pm 2.59}$	4.69 ± 2.88	$\mathbf{5.89 \pm 2.48}$	$\mathbf{6.04 \pm 2.46}$

To be more specific, UMGNet-SAGE is overall more effective in settings with 20% training size, and UMGNet-AL produces the strongest average uplift when the training size is limited to 5%, which is sensible due to active learning choosing the most informative training set. UpliftTree has the second best performance in *RHC* albeit with the highest standard deviation (7). Furthermore, an effective model should exhibit higher **up@20** than **up@40**, meaning the top 20% predictions should have higher real uplift than the top 40% if the predictive ranking is consistent. Our methods clearly follow this pattern in 15 out of 16 comparisons, in contrast to UpliftTree in *RHC*. The propensity-based methods, i.e., DR-XGB, DragonNet, R-XGB, are not as accurate because the treatment assignment in the dataset is balanced. Thus that the potential bias is minimized and, accordingly the help of the propensity score that can alleviate this bias is diminished.

Throughout the experiments, the standard deviation grows with the average value, with exceptions such as UMGNet-SAGE in *RHC* and T-XGB in *RHP*. This makes sense given the range of values and their std. It should be noted that although both tasks are challenging, *RHC* has significantly greater std (521 to 387) with a lower average (266 to 423) and it is arguably more informative. *RHC* normalizes the effect of the treatment with the user's normal behavior, while *RHP* ranks based on absolute purchase average, which can be biased on the user's preferences.

Finally, the five realizations of the semi-simulated *MovieLens* dataset are used to compare UMGNet-SAGE with the best benchmarks from the first experiment. The average results are shown in Fig. 2, where it is visible that it consistently outperforms them. In this case, however the benchmarks perform better compared to *RetailHero* if we consider the ATE, possibly due to outcome values being smaller and the input features of the movies being considerably more informative.

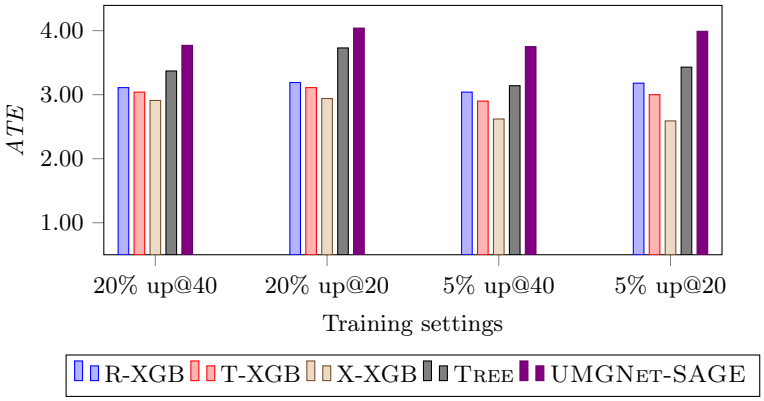

Fig. 2. Uplift of the predicted sets on the *MovieLens* dataset. Regular $ATE = 0.457$.

Table 4. The effect of GNN compared to vanilla DRAGONNET.

Dataset	*Model*	20% training size		5% training size	
		up@40	up@20	up@40	up@20
RHC	DRAGONNET	2.24 ± 0.14	3.08 ± 0.34	2.26 ± 0.04	3.01 ± 0.10
	UMGNET-DR	2.41 ± 1.27	5.20 ± 1.64	3.00 ± 0.48	5.70 ± 0.50
Improvement (%)		+7.6%	+68.8%	+32.7%	+89.4%
RHP	DRAGONNET	0.21 ± 0.16	-0.01 ± 0.26	0.21 ± 0.01	-0.10 ± 0.01
	UMGNET-DR	4.19 ± 1.33	6.15 ± 2.42	3.47 ± 0.82	4.33 ± 0.90
Improvement (%)		+1895.2%	+2700%	+1552.4%	+4430%

Ablation study. We perform a number of extra experiments to answer certain questions of interest.

• *Does the GNN help?* DRAGONNET [47] is one of the most popular neural architectures for *ITE*, though it has not been extensively utilized for uplift modeling or in semi-supervised settings. Although they are trained with different parameters e.g., our architecture has fewer layers, UMGNET-DR resembles a version of DRAGONNET with bipartite SAGE encoding. In Table 4, we see that adding information from the network produces significantly better results. The lack of supervision is detrimental for a deep neural network like DRAGONNET, as is prevalent in RHP.

• *Does the GNN layer impact the prediction?* We compare the results with GNNs like SAGE [21], NGCF [54], and LGC [22]. These GNN models are used to derive different instances of the UMGNET framework. The interested reader may refer to Appendix B for a detailed presentation of each GNN layer. Note that the layers also differ i.e., SAGE has only 1 layer, but NGCF has 3. The results can be

Table 5. The effect of different GNN layers in UMGNet.

Dataset	UMGNet	20% training size		5% training size	
		up@40	up@20	up@40	up@20
RHC	NGCF	$\mathbf{5.34 \pm 0.86}$	5.49 ± 1.15	$\mathbf{3.71 \pm 0.61}$	3.74 ± 0.97
	LGC	4.17 ± 1.55	3.96 ± 2.70	3.70 ± 0.73	3.72 ± 0.87
	SAGE	3.20 ± 0.25	$\mathbf{6.48 \pm 0.70}$	2.66 ± 0.75	$\mathbf{5.69 \pm 1.01}$
RHP	NGCF	3.53 ± 2.30	3.06 ± 2.51	1.97 ± 0.24	2.22 ± 0.23
	LGC	3.50 ± 2.00	2.89 ± 2.25	3.31 ± 0.52	3.30 ± 0.77
	SAGE	$\mathbf{5.01 \pm 2.16}$	$\mathbf{7.28 \pm 4.34}$	$\mathbf{3.99 \pm 2.00}$	$\mathbf{5.07 \pm 3.61}$

Table 6. The effect of different active learning policies in UMGNet.

Dataset	UMGNet	20% training size		5% training size	
		up@40	up@20	up@40	up@20
RHC	EG	6.01 ± 4.33	3.94 ± 3.18	4.12 ± 5.00	4.18 ± 3.00
	AL	6.27 ± 3.00	4.64 ± 3.60	5.83 ± 2.75	6.83 ± 3.77
Improvement (%)		+4.3%	+17.8%	+41.5%	+63.4%
RHP	EG	1.91 ± 2.40	3.59 ± 4.51	3.91 ± 3.20	5.66 ± 3.45
	AL	4.11 ± 2.59	4.69 ± 2.88	5.89 ± 2.48	6.04 ± 2.46
Improvement (%)		+115.2%	+30.6%	+50.6%	+6.7%

seen in Table 5, where it is clear that SAGE overall performs better, but NGCF is equally effective in *RHC*.

• *Is active learning helpful?* In Table 6 we are utilizing an e-greedy policy, that uses random batches with probability $\epsilon = 0.5$, noted as UMGNet-EG, as a baseline to clarify the need for the acquisition function. We see a clear improvement in each metric if we optimize the acquisition function in each step.

5 Conclusion

The creation of a large enough training set to use for uplift modeling can be a costly, time-consuming, or risky task. It is thus important to develop methodologies that select the right samples to intervene on and extrapolate efficiently on the rest of the dataset. We propose a two-step modular methodology, that addresses both of these needs. The main problem is formulated as semi-supervised uplift modeling and we solve it with bipartite graph neural networks. Additionally, a batch active learning method is defined based on the model's uncertainty, structural importance, and feature diversity to build the training set. We utilize, for the first time, a large-scale graph with ground-truth experimental information to test our hypothesis. The proposed methodology is compared to a breadth of

benchmarks from both uplift modeling and individual treatment effect literature. Our results indicate a clear advantage of the proposed methodology compared to the benchmarks, which sometimes perform near random in semi-supervised setting. Moreover, active learning enhances the results as the supervision diminishes. It is important to note that each step can be utilized separately e.g. the acquisition function can be used with other models. This framework aspires to be an initial step towards addressing the realistic yet overlooked problem of uplift modeling under budget for experimental interventions.

6 Ethics

This study only involved public datasets that are freely available for academic purposes. There are no obvious ethical considerations regarding negative impacts from the broader application of the method.

References

1. Arbour, D., Garant, D., Jensen, D.: Inferring network effects from observational data. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 715–724 (2016)
2. Bakshy, E., Eckles, D., Yan, R., Rosenn, I.: Social influence in social advertising: evidence from field experiments. In: Proceedings of the 13th ACM Conference on Electronic Commerce. pp. 146–161 (2012)
3. Cer, D., Yang, Y., Kong, S.y., Hua, N., Limtiaco, N., John, R.S., Constant, N., Guajardo-Cespedes, M., Yuan, S., Tar, C., et al.: Universal sentence encoder. arXiv preprint arXiv:1803.11175 (2018)
4. Chen, H., Harinen, T., Lee, J.Y., Yung, M., Zhao, Z.: Causalml: Python package for causal machine learning. arXiv preprint arXiv:2002.11631 (2020)
5. Chernozhukov, V., Chetverikov, D., Demirer, M., Duflo, E., Hansen, C., Newey, W.: Double/debiased/neyman machine learning of treatment effects. American Economic Review **107**(5), 261–265 (2017)
6. Chu, Z., Rathbun, S.L., Li, S.: Graph infomax adversarial learning for treatment effect estimation with networked observational data. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining. pp. 176–184 (2021)
7. Cortez, M., Eichhorn, M., Yu, C.: Staggered rollout designs enable causal inference under interference without network knowledge. Advances in Neural Information Processing Systems **35**, 7437–7449 (2022)
8. Cristali, I., Veitch, V.: Using embeddings for causal estimation of peer influence in social networks. Advances in Neural Information Processing Systems **35**, 15616–15628 (2022)
9. Dawid, A.P.: Conditional independence in statistical theory. Journal of the Royal Statistical Society Series B: Statistical Methodology **41**(1), 1–15 (1979)
10. Devriendt, F., Moldovan, D., Verbeke, W.: A literature survey and experimental evaluation of the state-of-the-art in uplift modeling: A stepping stone toward the development of prescriptive analytics. Big Data **6**(1), 13–41 (2018)

11. Diemert, E., Betlei, A., Renaudin, C., Amini, M.R.: A large scale benchmark for uplift modeling. In: Proceedings of the KDD Workshop on Artificial Intelligence for Computational Advertising (2018)
12. Fan, W., Ma, Y., Li, Q., He, Y., Zhao, Y.E., Tang, J., Yin, D.: Graph neural networks for social recommendation. In: Proceeding of the 28th ACM Web Conference. pp. 417–426. ACM (2019)
13. Farzam, A., Tannenbaum, A., Sapiro, G.: Curvature and causal inference in network data. In: Causal Representation Learning Workshop at NeurIPS 2023 (2023)
14. Gal, Y., Ghahramani, Z.: Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In: International Conference on Machine Learning. pp. 1050–1059. PMLR (2016)
15. Garivier, A., Moulines, E.: On upper-confidence bound policies for switching bandit problems. In: International Conference on Algorithmic Learning Theory. pp. 174–188. Springer (2011)
16. Gilhuber, S., Busch, J., Rotthues, D., Frey, C.M., Seidl, T.: Diffusal: Coupling active learning with graph diffusion for label-efficient node classification. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. pp. 75–91. Springer (2023)
17. Graff, D.E., Shakhnovich, E.I., Coley, C.W.: Accelerating high-throughput virtual screening through molecular pool-based active learning. Chemical Science **12**(22), 7866–7881 (2021)
18. Gui, H., Xu, Y., Bhasin, A., Han, J.: Network a/b testing: From sampling to estimation. In: Proceedings of the 24th International Conference on World Wide Web. pp. 399–409 (2015)
19. Guo, R., Li, J., 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)
20. Gutierrez, P., Gérardy, J.Y.: Causal inference and uplift modelling: A review of the literature. In: Proceedings of the 4th International Conference on Predictive Applications and APIs. pp. 1–13. PMLR (2017)
21. Hamilton, W., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. Advances in Neural Information Processing Systems **30** (2017)
22. He, X., Deng, K., Wang, X., Li, Y., Zhang, Y., Wang, M.: LightGCN: Simplifying and powering graph convolution network for recommendation. In: Proceedings of the 43rd ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 639–648. ACM (2020)
23. Huang, K., Jin, Y., Candes, E., Leskovec, J.: Uncertainty quantification over graph with conformalized graph neural networks. Advances in Neural Information Processing Systems **36** (2023)
24. Jiang, S., Sun, Y.: Estimating causal effects on networked observational data via representation learning. In: Proceedings of the 31st ACM International Conference on Information & Knowledge Management. pp. 852–861 (2022)
25. Johansson, F., Shalit, U., Sontag, D.: Learning representations for counterfactual inference. In: Proceedings of the 33rdh International Conference on Machine Learning. pp. 3020–3029. PMLR (2016)
26. Karrer, B., Shi, L., Bhole, M., Goldman, M., Palmer, T., Gelman, C., Konutgan, M., Sun, F.: Network experimentation at scale. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining. pp. 3106–3116 (2021)
27. Kennedy, E.H.: Towards optimal doubly robust estimation of heterogeneous causal effects. Electronic Journal of Statistics **17**(2), 3008–3049 (2023)

28. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
29. Künzel, S.R., Sekhon, J.S., Bickel, P.J., Yu, B.: Metalearners for estimating heterogeneous treatment effects using machine learning. Proceedings of the National Academy of Sciences **116**(10), 4156–4165 (2019)
30. Lee, B.K., Lessler, J., Stuart, E.A.: Improving propensity score weighting using machine learning. Statistics in Medicine **29**(3), 337–346 (2010)
31. Lin, X., Zhang, G., Lu, X., Bao, H., Takeuchi, K., Kashima, H.: Estimating treatment effects under heterogeneous interference. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. pp. 576–592. Springer (2023)
32. Louizos, C., Shalit, U., Mooij, J.M., Sontag, D., Zemel, R., Welling, M.: Causal effect inference with deep latent-variable models. Advances in Neural Information Processing Systems **30** (2017)
33. Ma, J., Guo, R., Chen, C., Zhang, A., Li, J.: Deconfounding with networked observational data in a dynamic environment. In: Proceedings of the 14th ACM International Conference on Web Search and Data Mining. pp. 166–174 (2021)
34. Ma, J., Wan, M., Yang, L., Li, J., Hecht, B., 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)
35. Ma, Y., Tresp, V.: Causal inference under networked interference and intervention policy enhancement. In: International Conference on Artificial Intelligence and Statistics. pp. 3700–3708. PMLR (2021)
36. Pearl, J.: Causality. Cambridge university press (2009)
37. Radcliffe, N.: Using control groups to target on predicted lift: Building and assessing uplift model. Direct Marketing Analytics Journal pp. 14–21 (2007)
38. Radcliffe, N.J., Surry, P.D.: Real-world uplift modelling with significance-based uplift trees. White Paper TR-2011-1, Stochastic Solutions pp. 1–33 (2011)
39. Rafla, M., Voisine, N., Crémilleux, B.: Evaluation of uplift models with non-random assignment bias. In: International Symposium on Intelligent Data Analysis. pp. 251–263. Springer (2022)
40. Rubin, D.B.: Estimating causal effects of treatments in randomized and nonrandomized studies. Journal of Educational Psychology **66**(5), 688 (1974)
41. Rubin, D.B.: Causal inference using potential outcomes: Design, modeling, decisions. Journal of the American Statistical Association **100**(469), 322–331 (2005)
42. Rudaś, K., Jaroszewicz, S.: Regularization for uplift regression. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. pp. 593–608. Springer (2023)
43. Russo, D.J., Van Roy, B., Kazerouni, A., Osband, I., Wen, Z., et al.: A tutorial on thompson sampling. Foundations and Trends® in Machine Learning **11**(1), 1–96 (2018)
44. Rzepakowski, P., Jaroszewicz, S.: Decision trees for uplift modeling with single and multiple treatments. Knowledge and Information Systems **32**, 303–327 (2012)
45. Settles, B.: Active learning literature survey (2009)
46. Shalit, U., Johansson, F.D., Sontag, D.: Estimating individual treatment effect: generalization bounds and algorithms. In: Proceedings of the 34th International Conference on Machine Learning. pp. 3076–3085. PMLR (2017)
47. Shi, C., Blei, D., Veitch, V.: Adapting neural networks for the estimation of treatment effects. Advances in Neural Information Processing Systems **32** (2019)
48. Sołtys, M., Jaroszewicz, S.: Boosting algorithms for uplift modeling. arXiv preprint arXiv:1807.07909 (2018)

49. Stadler, M., Charpentier, B., Geisler, S., Zügner, D., Günnemann, S.: Graph posterior network: Bayesian predictive uncertainty for node classification. Advances in Neural Information Processing Systems **34**, 18033–18048 (2021)
50. Tye, H.: Application of statistical 'design of experiments' methods in drug discovery. Drug discovery today **9**(11), 485–491 (2004)
51. Ugander, J., Karrer, B., Backstrom, L., 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)
52. Veitch, V., Wang, Y., Blei, D.: Using embeddings to correct for unobserved confounding in networks. Advances in Neural Information Processing Systems **32** (2019)
53. Verhelst, T., Petit, R., Verbeke, W., Bontempi, G.: Uplift vs. predictive modeling: a theoretical analysis. arXiv preprint arXiv:2309.12036 (2023)
54. Wang, X., He, X., Wang, M., Feng, F., Chua, T.: Neural graph collaborative filtering. In: SIGIR. pp. 165–174. ACM (2019)
55. Wei, K., Iyer, R., Bilmes, J.: Submodularity in data subset selection and active learning. In: Proceedings of the 32nd International Conference on Machine Learning. pp. 1954–1963. PMLR (2015)
56. Wright, D.B.: Comparing groups in a before–after design: When t test and ancova produce different results. British Journal of Educational Psychology **76**(3), 663–675 (2006)
57. Wu, Y., Xu, Y., Singh, A., Yang, Y., Dubrawski, A.: Active learning for graph neural networks via node feature propagation. arXiv preprint arXiv:1910.07567 (2019)

Appendix

A Network as a confounder and as a source for interference

Generally, we can make a distinction between the methods that predict the ATE i) using the network as a confounder [52,19,33,6,24,13], ii) assuming interference bias caused by the network [35,7,31], iii) separating between confounding and interference bias [34,8]. That said, all cases are evaluated based on the predicted ITE's accuracy, meaning that the interference models do not address how to use the predictions to alleviate the bias from the observed outcome. Our work falls in the first category and follows the same assumptions as the literature regarding interference [52,33,6,13]. Interference in e-commerce is not studied as extensively as in social networks [51,26]. Its existence remains unproved by experimental studies, mainly because it relies on the type of recommendation algorithm and the nature of the network.

B GNN layers adopted for the ablation study

In the ablation study, we analyzed the possible impact of each GNN layer adopted in our framework, namely, SAGE [21], NGCF [54], and LGC [22]. We

deep dive into the technical details of each of them. Hamilton et al. [21] introduce GraphSAGE (abbreviated as SAGE in this work), a GNN able to work on inductive scenarios by predicting labels of nodes unseen in the training set; for each graph node, SAGE's layer aggregates and samples nodes from the neighborhood and leverages node features such as textual features. Neural graph collaborative filtering [54] (NGCF) is one of the first GNNs-based recommendation systems; during the message-passing, the model aggregates the information from the neighbor nodes and calculates the inter-dependencies among the ego and the neighborhood nodes. Through light graph convolutional network [22] (LGC), the authors propose to lighten the formulation provided by NGCF as they theoretically and empirically observe that this can lead to superior recommendation accuracy; the model is a lightweight version of NGCF as it removes node features transformation and non-linearities.