Learning Causality with Networked Observational Data

by

Ruocheng Guo

A Dissertation Presented in Partial Fulfillment
of the Requirement for the Degree
Doctor of Philosophy

Approved March 2021 by the
Graduate Supervisory Committee:

Huan Liu, Chair
K. Selçuk Candan
Guoliang Xue
Emre Kiciman

ARIZONA STATE UNIVERSITY

May 2021

ABSTRACT

This dissertation considers the question of how convenient access to copious networked observational data impacts our ability to learn causal knowledge. It investigates in what ways learning causality from such data is different from – or the same as – the traditional causal inference which often deals with small scale i.i.d. data collected from randomized controlled trials? For example, how can we exploit network information for a series of tasks in the area of learning causality? To answer this question, the dissertation is written toward developing a suite of novel causal learning algorithms that offer actionable insights for a series of causal inference tasks with networked observational data. The work aims to benefit real-world decision-making across a variety of highly influential applications. In the first part of this dissertation, it investigates the task of inferring individual-level causal effects from networked observational data. First, it presents a representation balancing-based framework for handling the influence of hidden confounders to achieve accurate estimates of causal effects. Second, it extends the framework with an adversarial learning approach to properly combine two types of existing heuristics: representation balancing and treatment prediction. The second part of the dissertation describes a framework for counterfactual evaluation of treatment assignment policies with networked observational data. A novel framework that captures patterns of hidden confounders is developed to provide more informative input for downstream counterfactual evaluation methods. The third part presents a framework for debiasing two-dimensional grid-based e-commerce search with observational search log data where there is an implicit network connecting neighboring products in a search result page. A novel inverse propensity scoring framework that models user behavior patterns for two-dimensional display in e-commerce websites is developed, which aims to optimize online performance of ranking algorithms with offline log data.

To my family for their support and love.

ACKNOWLEDGEMENTS

This dissertation is impossible without the help from my Ph.D. advisor Dr. Huan Liu. I would like give my special thanks to him for the great freedom I have been enjoying through my years in the DMML lab to explore interesting research problems. Without his outstanding guidance, I would have not experienced such a colorful, exciting and productive Ph.D. journey. It is very fortunate to have Dr. Huan Liu as my advisor. There are many things I learned from him that can benefit all my life: academic writing and presentation, research problem discovery and solving, and career development. Dr. Liu is way beyond a Ph.D. advisor. He provides important advise to various aspects of my life, which improves my communication with different people, attitude towards different situations, and decision making for the future career.

I would like to thank my committee members, Dr. K. Selçuk Candan, Dr. Emre Kiciman, and Dr. Guoliang Xue, for helpful suggestions and insightful comments on my draft. I am very grateful to the committee members for their insightful questions during my comprehensive exam and proposal defense, which inspires me towards broad and deep thinking for my ongoing and future research agenda. I was fortunate to collaborate with Dr. K. Selçuk Candan on both research papers and grant proposals. Without his great effort and help on the research proposal writing, I would not have the chance to be funded by the NSF grant which allows me to have more time focusing on my research on causal machine learning. I am very grateful to have to chance to work with Dr. Emre Kiciman as a research intern at Microsoft, which allows me to work on a cutting-edge causal machine learning research problem. Dr. Kiciman offered me great suggestions for my presentation and encouraged me to network with other researchers during my internship. Without his help, I could neither complete a paper on a unfamiliar topic nor receive great suggestions from a group of researchers at Microsoft. I was very fortunate to take the optimization course from

Dr. Guoliang Xue. The course prepared me with solid technical background for my Ph.D. research on machine learning and causal inference.

I really appreciate the time working at DMML with a diverse group of lab members. I would like to thank Faisal Alatawi, Ghazaleh Beigi, Tyler Black, Amrita Bhattacharjee, Lu Cheng, Matthew Davis, Kaize Ding, Philippe Christophe Faucon, Min Gao, Bohan Jiang, Ujun Jeong, Isaac Jones, Nur Shazwani Kamrudin, Mansooreh Karami, Nayoung Kim, David (Ahmadreza) Mosallanezhad, Deepak Mahudeswaran, Vineeth Rakesh Mohan, Raha Moraffah, Fred Morstatter, Jundong Li, Yichuan Li, Tahora Hossein Nazer, Alex Nou, Suhas Ranganath, Justin Sampson, Kai Shu, Paras Sheth, Zhen Tan, Anique Tahir, Tharindu Kumarage, Brian Vincent, Suhang Wang, Liang Wu, Qianru Wang, Qun Zhao for the invaluable interactions. Jundong is a role model to my research career and a true friend during my downtime. Liang and Ghazaleh also helped me out during my early stage.

I was lucky to work as interns in Microsoft Research, X and Etsy with amazing colleagues and mentors: Emre Kiciman, Pengchuan Zhang and Hao Liu in Microsoft Research; Charlotte Leroy and Hongxu Ma at X; Liangjie Hong, Xiaoting Zhao, Adam Henderson and Xuan Yin from Etsy. You made my life much easier to adapt to new enviroments; Because of you, I enjoyed the three productive internships. In addition, I am really grateful to have a group of great people as collaborators: Ashkan Aleali, Hamidreza Alvari, Nitin Agarwal, Abhinav Bhatnagar, Chen Chen, Renqin Cai, Mengnan Du, P. Richard Hahn, Xia Hu, Ninghao Liu, Jing Ma, Ericsson Marin, Soumajyoti Sarkar, Paulo Shakarian, Elham Shaabani and Aidong Zhang. I am very fortunate to be (partially) supported by the following grants: NSF #1909555, NSF #1614576, ONR N00014-16-1-2257 and ARL W911NF2020124.

TABLE OF CONTENTS

CHAPTER Page

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

Causality is a universal relationship between a certain cause and its corresponding effects. As human beings, we often can learn the existence of a causal relationship between two events either from existing knowledge or by interacting with the environment. For example, we know an ice cream shop is popular because of its delicious ice creams and good service. When we aim to use machine learning algorithms to capture causality from data, the difference between causality and statistical associations becomes a major challenge. This is because machine learning models that are unaware of causality can end up with capturing *spurious correlation*. For example, in summer, the ice cream shop owner can find that sales and electric bills are high at the same time. However, the correlation between these two quantities, electric bill and sales, is spurious because consuming more electricity is not likely to be helpful in terms of increasing the sales. To avoid spurious correlations, we are motivated to include learning causality as a fundamental component of developing human-level AI (Peters *et al.*, 2017; Marcus and Davis, 2019) or system 2 (Schölkopf *et al.*, 2021).

In traditional causal analysis, data are often limited in terms of both the scale and the number of measurable features (covariates). Therefore, strong prior causal knowledge is required to reach causal conclusions (Cartwright *et al.*, 1994). In many cases, data have to be collected through carefully designed experiments where prior causal knowledge can be guaranteed. For example, randomized controlled trials (RCTs) are often referred to as the golden standard of causal inference (Cook *et al.*, 2002; Pearl, 2009; Rubin, 2005). RCTs are widely used to identify and estimate the average causal effects on the population level. Specifically, in data collected from RCTs, the group

receiving treatments and the group under control are thought to be equivalent on average expect the treatment assignments, which naturally excludes the effects of all other factors. However, experiments like RCTs which can be rather expensive, time consuming, and even unethical in some cases (Kallus and Zhou, 2018).

The era of big data has been granting us the convenient access to massive observational data in a myriad of highly influential areas including but not limited to social networks (Zafarani *et al.*, 2014; Shakarian *et al.*, 2015b), online advertising (Bottou *et al.*, 2013), recommender systems (Schnabel *et al.*, 2016), economics (LaLonde, 1986; Dehejia and Wahba, 1999) and healthcare (Hill, 2011). Compared to the data collected through carefully designed experiments, an observational dataset is often effortless to obtain and comes with a large number of instances and an affluent set features. Meanwhile, we can also find useful side information in such data. For example, there exists an inherent network structures that connect individuals when an observational dataset is collected from a social network service or an e-commerce website.

1.1 Challenges of Learning Causality with Observational Data

However, learning causality from observational data poses the challenge of various types of bias including confounding bias and selection bias.

The existence of confounding bias is confirmed when the causal effect of the treatment on the outcome is confounded by confounders, the variables causally influence both the treatment and the outcome (Pearl, 2009). For example, when we study the causal effect of an expensive medicine on the health outcomes of individuals, the poor socioeconomic status of an individual can limit her access to the expensive medicine and have negative impact on her health condition at the same time. Thus, failure in controlling the influence of the socioeconomic status may result in overestimated

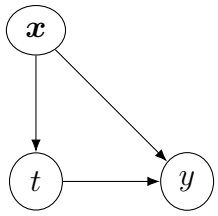


Figure 1.1: The causal graph explains the definition of a confounder: a variable that has simultaneous causal influence on both treatment and outcome.

treatment effect of the expensive medicine. Adjustment for the confounding bias is often recognized as the main challenge in various causal inference tasks with observational data such as causal effect estimation, counterfactual evaluation and optimization (Pearl, 2009; Rubin, 2005). To deal with confounding bias, in the literature, a series of work (Makar *et al.*, 2019; Wager and Athey, 2018; Schwab *et al.*, 2018; Hill, 2011; Johansson *et al.*, 2016; Shalit *et al.*, 2017; Yao *et al.*, 2018; Dudík *et al.*, 2011; Athey and Wager, 2017; Qian and Murphy, 2011; Kallus, 2018; Zou *et al.*, 2019) takes advantage of the strong ignoralbility (a.k.a. unconfoundedness) assumption (Rubin, 1978). This assumption can be interpreted as: all the confounders are measurable and have already been included in the set of observed features. As such, these methods overwhelmingly rely on the observed features to mitigate confounding bias. In the aforementioned example, most of existing efforts try to eliminate the influence of socioeconomic status on the chance to take the medicine and the health condition through controlling the impact of the related proxy variables such as annual income, age, and education. However, with massive observational data, it becomes extremely difficult to collect the causal relationships among variables as prior causal knowledge. As a result, the strong ignoralbility assumption can become untenable and is likely to be unrealistic due to the existence of hidden confounders (Pearl, 2009; Guo *et al.*, 2020a). Recently, a series of theoretical and empirical methods have been proposed to

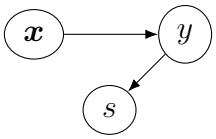


Figure 1.2: An example causal graph shows the selection bias in reporting annual income. s is a binary variable representing whether an individual reports her/his income.

leverage the advanced machine learning techniques to relax the assumption. However, they still need the assumption that, from observational data, it is possible to learn a set of latent variables to represent the confounders (Louizos *et al.*, 2017; Wang and Blei, 2018; Bennett and Kallus, 2019).

Selection bias also widely exists in real-world datasets (Bareinboim *et al.*, 2014; Bareinboim and Tian, 2015). Selection bias exists when only a biased subset of the population is observed in a dataset due to preferential selection. For example, in a study of the causal effect of job training on individuals' annual income, we may observe that people with higher income are more likely to report their outcomes than those earn less. A causal graph describes such selection bias is shown in Fig. 1.2. Bareinboim and Tian (2015) theoretically derive a set of conditions for recovering certain probabilistic quantities and causal effects from data under selection bias, given data generated according to a certain set of causal graphs. In the application of search ranking, there is a series of work on unbiased learning to rank which aims to recover the label distribution given biased search log data (Joachims *et al.*, 2017; Hu *et al.*, 2019; Ai *et al.*, 2018).

1.2 Role of Network Information in Causal Machine Learning Problems

In the existing efforts in causal identification and estimation, the importance of side information (e.g., network structures) that comes along with observational data

has been underestimated in the tasks of learning causality. When we confront the situations where some confounders are not feasible to measure, we can attempt in an alternative way to recognize their patterns and adjust for their influence by incorporating side information. For example, as measuring the socioeconomic status of an individual can be difficult, an alternative solution is to capture it by the individual's social network structural patterns such as centralities and community affiliation (Shakarian *et al.*, 2015b; Zafarani *et al.*, 2014). Recently, Veitch *et al.* (2019) use node embeddings learned from network information to compensate for hidden confounders. However, their method cannot properly utilize the observed features. We also attempt to utilize network information for mitigating selection bias. To the best of my knowledge, this has not been discussed in the literature.

In addition to using network information to compensate for unobserved confounders, there exists a series of work on network interference (a.k.a spillover effect). We say there exists interference iff an individual's treatment can causally influence its neighbors' outcomes. In fact, interference raises a big challenge for causal inference in terms of identification of causal effects. In a bipartite experiment setting where there are two types of nodes – items and buyers, Doudchenko *et al.* (2020) show that the causal identification of average treatment effect of different exposure levels can hold when (1) the weight of each edge is a known constant and (2) the treatment assignment is independent of potential outcomes given the edge weights. This implies that, in bipartite experiments, even with the ability to assign treatments (e.g., randomized experiment), the identification of average treatment effect is not trivial. This is because interference violates the stable unit treatment value assumption (SUTVA) which often used along with the Potential Outcome Framework (Rubin, 2005; Pearl and Mackenzie, 2018). For identifying the causal effect of ads block allocation on clicks under interference among ads, Nabi *et al.* (2020) use an assumption called

conditional network ignorability, which means potential outcomes are independent of treatment assignment given observed features. In market experiments, tech companies often rely on methods that redefine units such that interference can be ignored. For example, social network companies such as LinkedIn can use a community or a network cluster as a unit and do randomization over such units (Saint-Jacques *et al.*, 2019). This is based on the reasonable assumption that there is little interference between two redefined units. Note that in this dissertation we only assume that the causal influence conveyed by network is only relevant to the confounders or the the true propensity model.

In short, utilizing the rich side information (e.g., network information) that naturally exists in observational data to mitigate bias for downstream causal machine learning tasks still remains under-explored.

1.3 Thesis Statement and Research Questions

To bridge the gap, this dissertation focuses on the development of algorithms that leverage a type of universally existing side information – network structures to mitigate confounding bias and selection bias for highly-influential causal inference and machine learning tasks.

Thesis Statement. Network structures that embed causal dependencies among instances in observational data can be leveraged to mitigate confounding bias and selection bias in causal machine learning problems.

This dissertation provides evidence to support the thesis statement through investigating the two types of research questions below:

- Through investigations to answer the two following fundamental research questions, we show the general importance of network information in causal machine learning problems:

  - In causal effect estimation, does leveraging network structures improve causal inference algorithms in terms of mitigating confounding bias?
  - In counterfactual evaluation of novel treatment assignment functions, does properly utilizing the information embedded in the networks lead to better adjustment for confounding bias?

  In the application of debiasing e-commerce search, we showcase the power of network driven causal machine learning by answering the following research question:

- In e-commerce product search, does incorporating the network information of the two-dimensional display of products in search result pages help mitigate selection bias?

Toward answering these research questions, I present a summary of the main contributions of the dissertation as follows. I characterize the task of learning causality with observational data that distance it from traditional machine learning and classic causal analysis problems. Then, I propose novel frameworks for learning causality from observational data that can exploit network structures for mitigating confounding and selection bias. The rest of the dissertation is organized as follows. In Chapter 2, I present a comprehensive literature review of the related work. In Chapter 3, I describe two proposed frameworks that are capable of utilizing network information for deconfounding causal effects with networked observational data. In Chapter 4, I discuss a framework for counterfactual evaluation and optimization of novel treatment assignment functions in networked observational data. In Chapter 5, I present a framework for debiasing learning to rank algorithms in two-dimensional grid-based display for e-commerce product search.

Chapter 2

LITERATURE REVIEW

In this chapter, a comprehensive literature review of the related work is presented from the five following perspectives: (1) causal inference with network data; (2) causal inference with proxy variables; (3) learning individual treatment effects from i.i.d. data; (4) counterfactual evaluation and optimization of treatment assignment functions; (5) graph neural networks; (6) unbiased learning to rank; (7) grid-based search; (8) e-commerce search.

2.1 Causal Inference with Network Data

Researchers aim to utilize networks to approximate hidden confounders using observational studies. Shalizi et al. (Shalizi and McFowland III, 2016) propose a two-stage approach to estimate causal effects in networks based on predefined generative models. To avoid misspecified generative models, Veitch et al. (Veitch *et al.*, 2019) propose causal network embedding (CNE) which learns node embeddings from pure network data to represent confounders. However, CNE suffers from the following limitations: It relies on treatment prediction alone to handle confounding bias. CNE requires observable edge weights. It only infers ATE and cannot properly utilize the observed features. However, none of the existing methods can satisfy the two desiderata of handling confounding bias together. Networks can propagate the treatment received by an instance to interfere the outcomes or treatments of its neighbors. This phenomena can be referred to as contagion (Shalizi and Thomas, 2011), treatment entanglement (Toulis *et al.*, 2018), or spillover effect (Arbour *et al.*, 2016; Rakesh *et al.*, 2018). Different from them, we follow previous work (Veitch *et al.*, 2019) to

assume that conditioning on latent confounders decouples each individual's treatment and outcome from those of others.

2.2 Causal Inference with Proxy Variables

When hidden confounders exist, observed proxy variables can be utilized to approximate them. (Pearl, 2012; Kuroki and Pearl, 2014; Miao *et al.*, 2018; Louizos *et al.*, 2017; Veitch *et al.*, 2019). Most of the existing work assumes that the observed data is i.i.d. and generated by latent confounders. Theoretically, in (Pearl, 2012; Kuroki and Pearl, 2014), authors showed that causal effects can be identified by proxy variables. Miao et al. (Miao *et al.*, 2018) showed that it is feasible to restore the causal effects without knowing anything but the size of the latent confounders $\boldsymbol{z}$. Louizos et al. (Louizos *et al.*, 2017) showed that ITE (CATE) can be identified given the joint distribution $P(\boldsymbol{x}, t, y, \boldsymbol{z})$ and proposed a deep latent-varibale model to estimate ITEs. Recently, results in (Veitch *et al.*, 2019) show that network information, as proxy variables, can also help mitigate confounding bias.

2.3 Learning Individual Treatment Effects from i.i.d. Data

Learning ITEs from i.i.d. observational data has attracted great attention. Causal Forest (CF) (Wager and Athey, 2018) is a method that recursively partitions the original feature space through treatment prediction. Its hypothesis is that within each subspace, the instances are very similar in terms of their estimated propensity score. Therefore, we can think the treatment assignment in each subspace is random and the instances in the same subspace share the same ITE. So, CF infers ITEs via applying the naive estimator in each subspace. CFR (Johansson *et al.*, 2016; Shalit *et al.*, 2017) is a pioneer method for learning ITEs by representation learning. Both theoretical analysis and empirical results indicate that balancing the distributions of

the treated and controlled instances in the representation space can improve the performance in learning ITE. However, the methods mentioned above rely on the strong ignorability assumption, which is often untenbale in observational data. Louizos et al. (Louizos *et al.*, 2017) proposed to consider observed features as proxy variables of hidden confounders and use a deep latent-variable model to learn representation of confounders via variational inference. However, this line of work does not consider to utilize network information for learning causal effects.

2.4 Counterfactual Evaluation of Treatment Assignment Functions

Counterfactual evaluation methods with i.i.d. data can be classified into three major categories: direct methods (Qian and Murphy, 2011), weighted estimators (Kallus, 2018; Bennett and Kallus, 2019; Swaminathan and Joachims, 2015a,b), and doubly robust estimators (Dudík *et al.*, 2011; Athey and Wager, 2017). Directed methods achieve counterfactual evaluation by inferring counterfactual outcomes. However, existing direct methods are known to suffer from biased estimates (Beygelzimer *et al.*, 2008). This is mainly because that they rely on the strong ignorability assumption. Moreover, the supervision of the observed treatments remains to be utilized. Weighted estimators avoid the problem of inferring counterfactual outcomes. Instead, they estimate the utility of treatment assignment functions through a weighted average of observed outcomes. In particular, a sample weight is learned for each instance. The goal is to let the reweighted factual outcomes approximate their counterparts that would have been observed if the treatments had been assigned by the function to be evaluated. Inverse propensity scoring (IPS) (Kitagawa and Tetenov, 2018; Hirano *et al.*, 2003) is the most widely adopted strategy for reweighting. IPS estimators can suffer from the issue of high variance when the estimated propensity scores take extreme values. Therefore, a series of clipping and normalization based

methods (Bottou *et al.*, 2013; Swaminathan and Joachims, 2015a,b) have been proposed to mitigate this issue. However, IPS estimators' performance is still limited by the accuracy of estimated propensity scores. To combine the advantages of the two types of methods, doubly robust estimators are proposed (Chernozhukov *et al.*, 2018; Bang and Robins, 2005). Each doubly robust estimator consists of a direct method and a weighted estimator. Previous work (Dudík *et al.*, 2011; Chernozhukov *et al.*, 2018) has shown doubly robust estimators can maintain good performance even if either its direct method or its IPS estimator suffers from large bias. Different from the aforementioned work, this work investigates the effectiveness of incorporating network information in counterfactual evaluation.

2.5 Graph Neural Networks

Previous work on Graph Convolutional Networks (GCN) mainly focused on the development of spatially localized [1] and computationally efficient convolutional filters for various types of network data including citation networks and social networks. Bruna et al. (Bruna *et al.*, 2013) proposed to use the first-order graph Laplacian matrix as the basic of filters in the spectrum domain. However, this filter has a large number of trainable parameters and its the spatial locality is not guaranteed. In (Defferrard *et al.*, 2016), Defferrard et al. proposed a more efficient and properly localized filter for the graph convolution operator. This filter is parameterized as l-th order polynomials of the graph Laplacian matrix to ensure the locality, where l is a positive integer and is often greater than 1. Then the polynomials are approximated by their Chebyshev expansion to reduce the computational cost. Then, Kipf and Welling (Kipf and Welling, 2016) proposed the renormalization trick to further improve the com-

[1] Here, spatial locality refers to the constraint that information of a node only propagates to its neighbors in a certain number of hops.

putational efficiency of GCN. Recently, variants of GCN has also been proposed to a myriad of applications using network data such as recommendation (Wang *et al.*, 2019), content recommendation in social networks (Ying *et al.*, 2018), anomaly detection in attributed networks (Ding *et al.*, 2019), entity classification and link prediction in knowledge graphs (Schlichtkrull *et al.*, 2018) and link sign prediction in signed networks (Derr *et al.*, 2018). Different from the existing work, work presented in this dissertation prospectus is the first work exploiting graph neural networks for learning causal with observational data.

2.6 Unbiased Learning to Rank

Unbiased Learning to Rank is an area where causal inference (Guo *et al.*, 2020a) helps learning to rank. Given the same attractiveness (relevance), the probability of products (documents) being clicked may change significantly with many factors in SERPs of product (web) search. Position is one of the most significant factor. It has been studied in list-wise web search (Wang *et al.*, 2016; Joachims *et al.*, 2017; Wang *et al.*, 2018; Ai *et al.*, 2018; Hu *et al.*, 2019). As the literature of unbiased learning to focuses on solving the problem of position bias in traditional information retrieval systems, here, we use the terms, document and relevance, instead of product and attractiveness. Joachims et al. (Joachims *et al.*, 2017) analyzed the inherent position bias in search log data with implicit feedback and proposed the Propensity SVM-Rank (Joachims, 2002) algorithm which applies inverse propensity scoring to each clicked document to mitigate the position bias. In particular, the propensity scores of each position is estimated through an randomized experiment which randomly picks and swaps items at the i-th and j-th positions (Joachims *et al.*, 2017). In (Agarwal *et al.*, 2018), the authors extended the Propensity SVM-Rank model to directly optimize additive information retrieval metrics such as DCG and proposed to replace

the SVM-Rank model with neural networks. However, such randomized experiments may degrade users' experience and would likely be time and labor consuming. Ai et al. (Ai *et al.*, 2018) treated estimating propensity scores as a dual problem of unbiased learning to rank (Joachims *et al.*, 2017). As the propensity scores can only be used to reweigh documents with clicks in their model and only relevant documents are clicked, so they reweigh each document with its probability to be relevant. Both the propensity model and the ranker are parameterized by neural networks. Then, listwise objectives (Cao *et al.*, 2007; Xia *et al.*, 2008) are employed to train the two models alternatively. In (Hu *et al.*, 2019), an unbiased learning to rank algorithm is proposed based on the pairwise ranking algorithm LambdaMART (Wu *et al.*, 2010). Similar to (Ai *et al.*, 2018), in unbiased LambdaMART, the propensity score model is learned along with the ranker by an alternating optimization algorithm. However, none of the existing unbiased learning to rank algorithms takes the unique context of e-commerce into consideration. Different from them, the work presented in this dissertation, the proposed framework is developed to handle multiple types of implicit feedback and incorporate the unique user behavior patterns in grid-based product search into inverse propensity scoring. In particular, compared to unbiased LambdaMART which also utilizes a pairwise debiasing strategy and adopts LambdaMART, the proposed framework incorporates prior knowledge of users' behavior patterns to guide the learning process of propensity score models.

2.7 Grid-based Search

Nowadays, various types of websites including e-commerce, video and music streaming services show SERPs in a grids. Recently, in eye-tracking experiments, Xie *et al.* (2019) observed three unique properties of users' behaviors in grid-based image search: middle bias, slower decay and row skipping. Based on the observations, for the sake of

developing better evaluation metrics for grid-based search, they propose three novel click models to quantify how users' attention decays in such scenarios. We did not adopt these new evaluation metrics because without eye-tracking experiments we cannot obtain ground truth for the parameters of these evaluation metrics which quantify the decay of attention. Different from their focus, we propose to incorporate the row skipping and slower decay click models for propensity score modeling toward unbiased learning to rank. At the same time, grid-based search is still an open question for many other research problems like grid-based sponsored search.

2.8 E-commerce Search

Compared to traditional information retrieval, e-commerce search is confronted with some unique challenges such as its multi-objective nature and the need to explore new items for fairness among sellers as well as long-term user engagement (Wu *et al.*, 2018; Goswami *et al.*, 2018). E-commerce search logs come with multiple types of implicit feedback (e.g., purchase and click). The target of e-commerce search is to maximize purchases or revenue of the website, however, due to the fact that purchases are much less frequently observed than other types of feedback such as clicks, it has been proposed to combine different types of feedback in the training objective (Karmaker Santu *et al.*, 2017; Wu *et al.*, 2018; Sorokina and Cantu-Paz, 2016). In (Sorokina and Cantu-Paz, 2016), authors found such hybrid objectives help improve the search performance of fashion products on Amazon. In (Wu *et al.*, 2018), a two-stage algorithm is proposed to integrate clicks and purchases through two separate machine learning models. In e-commerce search, we aim to help buyers explore unseen items, in (Goswami *et al.*, 2018), authors proposed a multi-armed bandit (MAB) method which allows exploration of items that are shown less than a certain times in a time interval. In terms of feature engineering, besides manu-

ally engineered features, recently, representation learning has been incorporated in e-commerce search (Van Gysel *et al.*, 2016; Ai *et al.*, 2017). Regarding other aspects, Goswami *et al.* (2019) also found that e-commerce search log data helps quantify the gap between customer demands and supplies. Different from them, our work is the first to develop a framework for unbiased learning to rank for e-commerce search.

Chapter 3

LEARNING INDIVIDUAL CAUSAL EFFECTS WITH NETWORKED OBSERVATIONAL DATA

Estimating the causal effect of a treatment on an outcome is one of the most fundamental tasks in learning causality. Studying this problem helps us derive actionable patterns from networked observational data for rational decision making in a wide range of applications. Given the fact that the underlying network structures can be useful in capturing patterns of hidden confounders, a vast majority of existing methods have not been developed in a proper way to utilize such information. In this chapter, we show that when hidden confounders are correlated with network patterns, the ability to exploit network information becomes of vital importance toward unbiased causal effect estimation.

3.1 Problem Statement

This section starts with introducing the notations and preliminaries. Then, the problem statement is formally presented.

Notations. First, we describe the notations used in this work. We denote a scalar, a vector, and a matrix with a lowercase letter (e.g., t), a boldface lowercase letter (e.g., $\boldsymbol{x}$), and a boldface uppercase letter (e.g., $\boldsymbol{A}$), respectively. Subscripts signify element indexes (e.g., $\boldsymbol{x}_i$ and $\boldsymbol{A}_{i,j}$). Superscripts of the a potential outcome variable denotes its corresponding treatment (e.g., y_i^t).

Networked Observational Data. Then we introduce networked observational data. In this work, we aim to learn individual treatment effects from networked observational data. Such data can be represented as $(\{\boldsymbol{x}_i, t_i, y_i\}_{i=1}^n, \boldsymbol{A})$ where $\boldsymbol{x}_i$,

t_i and y_i denote the features, the observed treatment, and the observed (factual) outcome of the i-th instance, respectively. The symbol $\boldsymbol{A}$ signifies the adjacency matrix of the auxiliary network information among different data instances. Here, we assume that the network is undirected and all the edges share the same weight [1] . Therefore, with the adjacency matrix $\boldsymbol{A} \in \{0,1\}^{n\times n}$, $\boldsymbol{A}_{i,j} = \boldsymbol{A}_{j,i} = 1$ ($\boldsymbol{A}_{i,j} = \boldsymbol{A}_{j,i} = 0$) denotes that there is an (no) edge between the i-th instance and the j-th instance. We focus on the cases where the treatment variable takes binary values $t \in \{0,1\}$. Without loss of generality, $t_i = 1$ ($t_i = 0$) means that the i-th instance is under treatment (control). We also let the outcome variable be a scalar and take values on real numbers as $y \in \mathbb{R}$.

Preliminaries. Then we introduce the background knowledge of learning individual causal effects. To define individual treatment effect (ITE), we start with the definition of potential outcomes, which is widely adopted in the causal inference literature (Rubin, 1978) [2] :

Definition 1. ***Potential Outcomes.*** *Given an instance i and the treatment t, the potential outcome of i under treatment t, denoted by y_i^t, is defined as the value of y would have taken if the treatment of instance i had been set to t.*

Then we are able to provide the formal definition of ITE for the i-th instance in the setting of networked observational data as:

$$\tau_i = \tau(\boldsymbol{x}_i, \boldsymbol{A}) = \mathbb{E}[y_i^1|\boldsymbol{x}_i, \boldsymbol{A}] - \mathbb{E}[y_i^0|\boldsymbol{x}_i, \boldsymbol{A}] \tag{3.1}$$

Intuitively, ITE is defined as the expected potential outcome of an instance under treatment subtracted by that under control, which reflects how much improvement in

[1] This work can be directly applied to weighted undirected networks. It can also be extended to directed networks using the Graph Convolutional Neural Networks for directed networks (Monti *et al.*, 2018).

[2] Note that we only use the concept of potential outcomes, but do not rely on the strong ignorability assumption that is often adopted along with this concept.

the outcome would be caused by the treatment. Note that with the network information, we are able to go beyond the limited information provided by the features and distinguish two instances with the similar features but different network patterns in the task of learning individual treatment effects. With ITE defined, we can formulate the average treatment effect (ATE) by taking the average of ITE over the instances as: $ATE = \frac{1}{n}\sum_{i=1}^{n}\tau_i$. Finally, we formally present the definition of the problem of learning individual treatment effects from networked observational data as follows:

Definition 2. ***Learning Individual Treatment Effects from Networked Observational Data.*** *Given the networked observational data* $(\{\boldsymbol{x}_i, t_i, y_i\}_{i=1}^{n}, \boldsymbol{A})$*, we aim to develop a causal inference framework which estimates the ITE of each individual such that a predefined error metric on the ITEs is minimized.*

3.2 Proposed Framework I – Network Deconfounder

In this section, we discuss the proposed framework, the *network deconfounder*. We start with an introduction of the background about the strong ignorability assumption and confounding bias. Then the proposed framework is described.

3.2.1 Background

It is not difficult to find that, in networked observational data, as only one of the two potential outcomes can be observed, the main challenge of learning individual treatment effects is the inference of *counterfactual outcomes* $y_i^{CF} = y_i^{1-t_i}$. In previous work (Hill, 2011; Wager and Athey, 2018; Johansson *et al.*, 2016; Shalit *et al.*, 2017), with the strong ignorability assumption, controlling observed features is often considered to be enough to eliminate confounding bias. Formally, strong ignorability can be defined as:

Definition 3. ***Strong Ignorability.*** *With strong ignorability, it is assumed that: (1) the potential outcomes of an instance are independent of whether it receives treatment or control given its features. (2) In addition, for each instance the probability to get treated is larger than 0 and less than 1. Formally, given the feature space $\mathcal{X}$, the strong ignorability assumption can be presented as:*

$$y^1, y^0 \perp\!\!\!\perp t|\boldsymbol{x} \quad and \quad 1 > Pr(t = 1|\boldsymbol{x}) > 0, \forall \boldsymbol{x} \in \mathcal{X}, t \in \{0, 1\}. \tag{3.2}$$

It implies that $\mathbb{E}[y^t|\boldsymbol{x}] = \mathbb{E}[y|\boldsymbol{x}, t]$. This is due to the conditional independence between the treatment and the potential outcomes, where y denotes the outcome resulting from the features $\boldsymbol{x}$ and the treatment t. Intuitively, strong ignorability means we can observe every single feature that describes the difference between the treatment and the control group. With the strong ignorability assumption, many existing methods (Johansson *et al.*, 2016; Shalit *et al.*, 2017; Hill, 2011; Wager and Athey, 2018) boil down the task to learning a machine learning model that approximates the function $f : \mathcal{X} \times \{0, 1\} \to \mathbb{R}$ that estimates the expected potential outcomes $\mathbb{E}[y|\boldsymbol{x}, t]$ given features and the treatment.

However, in this work, we consider a more realistic setting where we allow the existence of hidden confounders. As a result, inferring counterfactual outcomes based on the features and the treatment alone would result in a biased estimator. This can be written as $\mathbb{E}[y|\boldsymbol{x}, t] \neq \mathbb{E}[y^t|\boldsymbol{x}]$. This is because the dependencies between the treatment variable and the two potential outcomes are introduced by the hidden confounders.

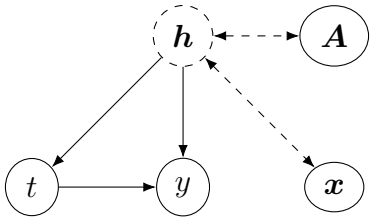


Figure 3.1: The causal diagram representing the assumptions of the proposed framework – network deconfounder

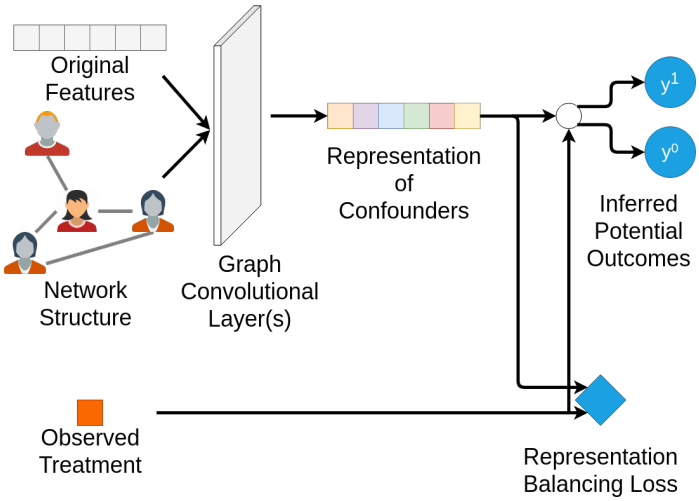


Figure 3.2: The work flow of the proposed framework network deconfounder

3.2.2 Network Deconfounder

In this subsection, we propose the network deconfounder, a novel framework that addresses the challenges of learning individual treatment effects from networked observational data. Fig. 3.2 illustrates the workflow of the proposed network deconfounder framework. Given the adjacency matrix $\boldsymbol{A}$, features $\boldsymbol{x}$, the treatment t, and the outcome y, Fig. 3.1 shows the causal diagram which represents the assumption used in the network deconfounder. Specifically, the network structure represented by adjacency matrix $\boldsymbol{A}$ along with the observed features $\boldsymbol{x}$ are proxy variables of the hidden confounders $\boldsymbol{h}$, which can be utilized to learn representations of hidden confounders.

The directed (bidirected) edges signify causal relations (correlations), solid circles represent observed variables, and the dashed circle stands for hidden confounders. Instead of relying on the strong ignorability assumption, the network deconfounder is based on a weaker assumption that the features and the network structure are two sets of proxy variables of the hidden confounders. This is a more practical assumption than the strong ignorability assumption in the sense that we do not require the observed features to capture all the information that describes the difference between the treated instances and the controlled ones. For example, although we cannot directly measure the socioeconomic status of an individual, we can collect features such as age, job type, zip code, and the social network to approximate her socioeconomic status. Based on this assumption, the proposed network deconfounder attempts to learn representations that approximate hidden confounders and estimate ITE from networked observational data simultaneously.

Unlike eliminating confounding bias based on the observed features alone, leveraging the underlying network structure for controlling confounding bias raises special challenges: (1) instances are inherently interconnected with each other through the network structure and hence their features are not independent identically distributed (i.i.d.) samples from a certain feature distribution, (2) the adjacency matrix of a network is often high-dimensional and can be very sparse ($\boldsymbol{A} \in \{0,1\}^{n \times n}$).

To tackle these special challenges of controlling confounding bias when network structure information exists, we propose the network deconfounder framework. The task can be divided into two steps. First, we aim to learn representations of hidden confounders by mapping the features and the network structure simultaneously into a shared representation space of confounders. Then an output function is learned to infer a potential outcome of an instance based on the treatment and the representation of hidden confounders. Then we present how the two tasks are accomplished by the

network deconfounder.

Learning Representation of Confounders. In previous work (Johansson *et al.*, 2016; Shalit *et al.*, 2017; Louizos *et al.*, 2017), representation learning techniques have been leveraged for estimating individual level causal effects. Different from them, the network deconfounder is the first one that is able to utilize auxiliary network information to improve the representation learned toward ITE estimation. The first component of the network deconfounder is a representation learning function g. The function g maps the features and the underlying network into the d-dimensional shared latent space of confounders, which can be formulated as $g : \mathcal{X} \times \mathcal{A} \to \mathbb{R}^d$. We parameterize the g function using Graph Convolutional Networks (GCN) (Defferrard *et al.*, 2016; Kipf and Welling, 2016), whose effectiveness have been verified in various machine learning tasks across different types of networked data (Ding *et al.*, 2019). To the best of our knowledge, this is the first work introducing GCN to the task of learning causal effects. In particular, the representation of confounders of the i-th instance is learned through GCN layers. Here, for the simplicity of notation, we describe the function g with a single GCN layer. The representation learning function g is parameterized as:

$$\boldsymbol{h}_i = g(\boldsymbol{x}_i, \boldsymbol{A}) = \sigma((\hat{\boldsymbol{A}}\boldsymbol{X})_i \boldsymbol{U}), \tag{3.3}$$

where $\hat{\boldsymbol{A}}$ denotes the normalized adjacency matrix, $(\hat{\boldsymbol{A}}\boldsymbol{X})_i$ signifies the i-th row of the matrix product $\hat{\boldsymbol{A}}\boldsymbol{X}$, $\boldsymbol{U} \in \mathbb{R}^{m \times d}$ represents the weight matrix to be learned, and σ stands for the ReLU activation function (Glorot *et al.*, 2011). Specifically, with the following notations, $\tilde{\boldsymbol{A}} = \boldsymbol{A} + \boldsymbol{I}_n$ and $\tilde{\boldsymbol{D}}_{j,j} = \sum_j \tilde{\boldsymbol{A}}_{j,j}$, the normalized adjacency matrix $\hat{\boldsymbol{A}}$ can be calculated using the renormalization trick (Kipf and Welling, 2016):

$$\hat{\boldsymbol{A}} = \tilde{\boldsymbol{D}}^{-\frac{1}{2}} \tilde{\boldsymbol{A}} \tilde{\boldsymbol{D}}^{-\frac{1}{2}} \tag{3.4}$$

We can compute $\hat{\boldsymbol{A}}$ in a pre-processing step to avoid repeating the computation. Then

the weight matrix $\boldsymbol{U} \in \mathbb{R}^{m \times d}$ along with the ReLU activation function maps the input signal into the low-dimensional representation space. Note that more than one GCN layers can be stacked to catch the non-linear relations between hidden confounders and the input data.

Inferring Potential Outcomes. Then we introduce the second component of network deconfounder, namely the output function $f : \mathbb{R}^d \times \{0, 1\} \to \mathbb{R}$. The function f maps the representation of hidden confounders as well as a treatment to the corresponding potential outcome. With $\boldsymbol{h}_i \in \mathbb{R}^d$ denoting the representation of the confounders of the i-th instance and $t \in \{0, 1\}$ signifying the treatment, to infer the corresponding potential outcome, the output function f is defined as:

$$f(\boldsymbol{h}_i, t) = \begin{cases} f_1(\boldsymbol{h}_i) \text{ if } t = 1 \\ \\ f_0(\boldsymbol{h}_i) \text{ if } t = 0 \end{cases}, \tag{3.5}$$

where f_1 and f_0 are the output functions for treatment $t = 1$ and $t = 0$. Specifically, we parameterize the output functions f_1 and f_0 using L fully connected layers followed by a regression layer as:

$$\begin{aligned} f_1 &= \boldsymbol{w}^1 \sigma(\boldsymbol{W}_L^1 ... \sigma(\boldsymbol{W}_1^1 \boldsymbol{h}_i)), \\ f_0 &= \boldsymbol{w}^0 \sigma(\boldsymbol{W}_L^0 ... \sigma(\boldsymbol{W}_1^0 \boldsymbol{h}_i)), \end{aligned} \tag{3.6}$$

where $\boldsymbol{h}_i$ is the representation of the i-th instance's confounders (output of the g function), $\{\boldsymbol{W}_l^t\}, l = 1, ..., L$ denote the weight matrices of the fully connected layers, and $\boldsymbol{w}^t$ is the weight for the regression layers. The bias terms of the fully connected layers and the output regression layer are dropped for simplicity of notation. We can either set $t = t_i$ to infer the observed factual outcome y_i^{CF} or $t = 1 - t_i$ to estimate the counterfactual outcome.

With the two components of the network deconfounder formulated, given the features of the i-th instance $\boldsymbol{x}_i$, the treatment t, and the adjacency matrix $\boldsymbol{A}$, we can

infer the potential outcome as:

$$\hat{y}_i^t = f(g(\boldsymbol{x}_i, \boldsymbol{A}), t), \tag{3.7}$$

where $\hat{y}_i^t$ denotes the inferred potential outcome of instance i corresponding to treatment t by the network deconfounder framework.

Objective Function. Then, we introduce the three essential components of the loss function for the proposed network confounder.

Factual Outcome Inference. First, we aim to minimize the error in the inferred factual outcomes. This leads to the first component of the loss function, the mean squared error in the inferred factual outcomes:

$$\min \frac{1}{n} \sum_{i=1}^{N} (\hat{y}_i^{t_i} - y_i)^2. \tag{3.8}$$

Representation Balancing. Minimizing the error in the factual outcomes (y_i) does not necessarily mean that the error in the counterfactual outcomes (y_i^{CF}) is also minimized. In other words, in the problem of learning ITE from networked observational data, we essentially confront the challenge of distribution shift (Johansson *et al.*, 2016; Shalit *et al.*, 2017). In particular, the network deconfounder would be trained on the conditional distribution of factual outcomes $Pr(y_i|\boldsymbol{x}_i, \boldsymbol{A}, t_i)$ but the task is to infer the conditional distribution of counterfactual outcomes $Pr(y_i^{CF}|\boldsymbol{x}_i, \boldsymbol{A}, 1 - t_i)$. In (Shalit *et al.*, 2017, Lemma 1.), the authors have shown that the error in the inferred counterfactual outcomes is upperbounded by a weighted sum of (1) the error in the inferred factual outcomes; and (2) an integral probability metric (IPM) measuring the difference between the distributions the treated instances and the controlled instances in terms of their confounder representations. Therefore, besides the error in inferred factual outcomes, we also aim to minimize the IPM measuring the how different the treatment group and the control group are regarding their distributions of confounders' representations. With $P(\boldsymbol{h}) = Pr(\boldsymbol{h}|t_i = 1)$ and $Q(\boldsymbol{h}) = Pr(\boldsymbol{h}|t_i = 0)$ being

the empirical distributions of representation of hidden confounders, we let $\rho_{\mathcal{Z}}(P, Q)$ denote the IPM defined in the functional space $\mathcal{Z}$ which measures the divergence between the two distributions of confounders' representations. Assuming that $\mathcal{Z}$ denotes the functional space of 1-Lipschitz functions, the IPM reduces to the Wasserstein-1 distance which is defined as:

$$\min \rho_{\mathcal{Z}}(P, Q) = \inf_{k \in \mathcal{K}} \int_{\boldsymbol{h} \in \{\boldsymbol{h}_i\}_{i:t_i=1}} ||k(\boldsymbol{h}) - \boldsymbol{h}|| P(\boldsymbol{h}) d\boldsymbol{h} \tag{3.9}$$

where $\mathcal{K} = \{k | k : \mathbb{R}^d \to \mathbb{R}^d \ s.t. \ Q(k(\boldsymbol{h})) = P(\boldsymbol{h})\}$ denotes the set of push-forward functions that can transform the representation distribution of the treated ($P(\boldsymbol{h})$) to that of the controlled ($Q(\boldsymbol{h})$). By minimizing $\alpha \rho_{\mathcal{Z}}(P, Q)$, we approximately minimize the divergence between the distributions of confounders' representations, where $\alpha \geq 0$ signifies the hyperparameter controlling the trade-off between penalizing the imbalance of confounders' representations and the other penalty terms in the loss function of the network deconfounder. We adopt the efficient approximation algorithm proposed by (Cuturi and Doucet, 2014) to compute the Wasserstein-1 distance in Eq. (3.9) and its gradients against the model parameters for training the network deconfounder.

ℓ_2 Regularization. Third, we let $\boldsymbol{\theta}$ signify the vector of the model parameters of the network deconfounder. Then a squared ℓ_2 norm regularization term on the model parameters - $\lambda ||\boldsymbol{\theta}||_2^2$, is added to mitigate the overfitting problem, where $\lambda \geq 0$ denotes the hyperparamter controlling the trade-off between the ℓ_2 regularization term and the other two terms.

Formally, we present the objective function of the network deconfounder as:

$$\mathcal{L}(\{\boldsymbol{x}_i, t_i, y_i\}_{i=1}^n, \boldsymbol{A}) = \frac{1}{n} \sum_{i=1}^{n} (\hat{y}_i^{t_i} - y_i)^2 + \alpha \rho_{\mathcal{Z}}(P, Q) + \lambda ||\boldsymbol{\theta}||_2^2, \tag{3.10}$$

3.3 Experimental Evaluation I

In this section, experiments are described. They are performed to investigate the effectiveness of the proposed framework in the task of learning ITEs with networked observational data. It starts with dataset description. Then experimental settings and results are presented.

3.3.1 Dataset Description

It is notoriously hard to obtain ground truth of ITEs because in most if not all cases, we can only observe one of the potential outcomes. For example, a patient can only choose to take the medicine or not to take it, but not both. So we can only observe the outcome resulting from her choice. However, we need benchmark datasets that provide ground truth of ITEs such that we can compare different methods that estimate ITEs with networked observational data. To resolve this problem, we follow the existing literature (Johansson *et al.*, 2016; Shalit *et al.*, 2017; Louizos *et al.*, 2017; Schwab *et al.*, 2019) to create semi-synthetic datasets. In particular, we introduce two benchmark datasets for the task of learning ITEs from networked observational. These datasets are semi-synthetic in the sense that they are based on features and network structures collected from real-world sources. Then we synthesize treatments, and outcomes for the task of learning ITEs from networked observational data in the presence of hidden confounders.

BlogCatalog. BlogCatalog [3] is an online community where users post blogs. In the dataset, each instance is a blogger. Each edge signifies the social relationship (friendship) between two bloggers. The features are bag-of-words representations of keywords in bloggers' descriptions. We extend the BlogCatalog dataset used in (Li

[3] https://www.blogcatalog.com/

et al., 2015, 2019a) by synthesizing (a) the outcomes – the opinions of readers on each blogger; and (b) the treatments – whether contents created by a blogger receive more views on mobile devices or desktops. Similar to the News dataset used in the previous work (Johansson *et al.*, 2016; Schwab *et al.*, 2018, 2019), we make the following assumptions: (1) Readers either read on mobile devices or desktops. We say a blogger get treated (controlled) if her blogs are read more on mobile devices (desktops). (2) Readers prefer to read some topics from mobile devices, others from desktops. (3) A blogger and her neighbors' topics causally influence her treatment assignment. (4) A blogger and her neighbors' topics also causally affect readers' opinions on them. Here, we aim to study the individual treatment effect of receiving more views on mobile devices (than desktops) on readers' opinions. To synthesize treatments and outcomes in accordance to the assumptions mentioned above, we first train a LDA topic model (Blei *et al.*, 2003) on a large set of documents. Then, two centroids in the topic space are defined as follows: (i) we randomly sample a blogger and let the topic distribution of her description be the centroid of the treated instances, denoted by r_1^c. (ii) The centroid of the controlled, r_0^c, is set to be the mean of the topic distributions of all the bloggers' descriptions. Then we introduce how the treatments and outcomes are synthesized based on the similarity between the topic distribution of a blogger's description and the two centroids. With $r(\boldsymbol{x}_i)$ denoting the topic distribution of the i-th blogger's description, we model the device preference of

the readers of the i-th blogger's content as:

$$
\begin{aligned}
Pr(t=1|\boldsymbol{x}_i, \boldsymbol{A}) &= \frac{\exp(p_1^i)}{\exp(p_1^i)+\exp(p_0^i)}; \\
p_1^i &= \kappa_1 r(\boldsymbol{x}_i)^T r_1^c + \kappa_2 \sum_{j\in\mathcal{N}(i)} r(\boldsymbol{x}_j)^T r_1^c \\
&= \kappa_1 r(\boldsymbol{x}_i)^T r_1^c + \kappa_2 (\boldsymbol{A} r(\boldsymbol{x}_j))^T r_1^c; \\
p_0^i &= \kappa_1 r(\boldsymbol{x}_i)^T r_0^c + \kappa_2 \sum_{j\in\mathcal{N}(i)} r(\boldsymbol{x}_j)^T r_0^c \\
&= \kappa_1 r(\boldsymbol{x}_i)^T r_0^c + \kappa_2 (\boldsymbol{A} r(\boldsymbol{x}_j))^T r_0^c,
\end{aligned}
\tag{3.11}
$$

where $\kappa_1, \kappa_2 \geq 0$ signifies the magnitude of the confounding bias resulting from a blogger's topics and her neighbors' topics, respectively. When $\kappa_1 = 0, \kappa_2 = 0$ the treatment assignment is random and the greater the values κ_1 and κ_2 are, the more significant the influence of a blogger's topics and her neighbors' topics on the device preference is. Then the factual outcome and the counterfactual outcome of the i-th blogger are simulated as:

$$y^F(\boldsymbol{x}_i) = y_i = C(p_0^i + t_i p_1^i) + \epsilon; \tag{3.12}$$

$$y^{CF}(\boldsymbol{x}_i) = C[p_0^i + (1-t_i)p_1^i] + \epsilon, \tag{3.13}$$

where C is a scaling factor and the noise is sampled as $\epsilon \sim \mathcal{N}(0,1)$. In this work, we set $C = 5, \kappa_1 = 10, \kappa_2 \in \{0.5, 1, 2\}$. Note that the outcomes of an individual are not influenced by the treatment assignment or outcomes of their neighbors, therefore, there is no interference or spillover effect in this scenario.

In the experiments, 50 LDA topics are learned from the training corpus. Then we reduce the vocabulary by taking the union of the most frequent 100 words from each topic. By doing this, we end up with 2,173 bag-of-words features. We perform the aforementioned simulation 10 times for each setting of κ_2. Figure 3.3 shows the distribution of topics in one of the simulations, which is projected to two-dimensional

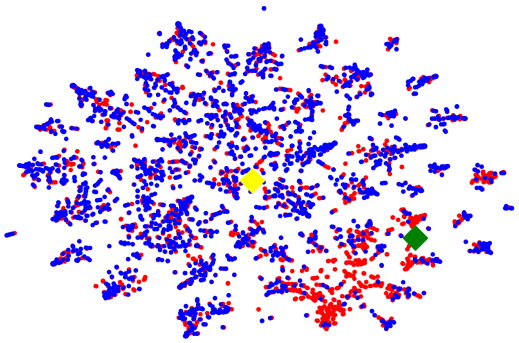


Figure 3.3: Distribution of the treated (red) and control (blue) instances in the LDA topic space.

space through the visualization technique TSNE (van der Maaten and Hinton, 2008). The green and yellow diamonds signify the two centroids r_1^c and r_0^c for the two treatment groups. We observe that there are more treated instances (red dots) near the centroid r_1^c (green diamond) and more control instances (blue dots) close to the centroid r_0^c (yellow diamond). In addition, a significant shift from the centroids can be perceived which shows the impact of the network structure.

Flickr. Flickr [4] is an online social network where users share images and videos. In this dataset, each instance is a user and each edge represents the social relationship (friendship) between two users. The features of each user represent a list of tags of interest. We adopt the same settings and assumptions as we do for the BlogCatalog dataset. Thus, we also study the individual-level causal effects of being viewed on mobile devices on readers' opinions on the user. In particular, we also learn 50 topics from the training corpus using LDA and concatenate the top 25 words of each topic. Thus, we reduce the data dimension to 1,210. We maintain the same settings of parameters as the BlogCatalog dataset ($C = 5, \kappa_1 = 10$ and $\kappa_2 \in \{0.5, 1, 2\}$).

In Table 4.1, we present a summary of the statistics of the semi-synthetic datasets

[4] https://www.flickr.com

Table 3.1: Dataset Description (BC stands for BlogCatalog)

	Instances	Edges	Features	κ_2	ATE Mean	STD
BC	5,196	173,468	2,173/8,189	0.5	4.366	0.553
				1	7.446	0.759
				2	13.534	2.309
Flickr	7,575	239,738	1,210/12,047	0.5	6.672	3.068
				1	8.487	3.372
				2	20.546	5.718

described in this subsection. The average and standard deviation of the ATEs are calculated over the 10 runs under each setting of parameters.

3.3.2 Experimental Settings

Following the original implementation of GCN (Kipf and Welling, 2016) [5] , we train the model with all the training instances along with the complete adjacency matrix of the auxiliary network information. ADAM (Kingma and Ba, 2014) is the optimizer we use to minimize the objective function of the network deconfounder (Eq. (3.20)). We randomly sample 60% and 20% of the instances as the training set and validation set and let the remaining be the test set. We perform 10 times of random sampling for each simulation of the datasets and report the average results. Grid search is applied to find the optimal combination of hyperparameters for the network deconfounder. In particular, we search learning rate in $\{10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}\}$, the number of output layers in $\{1, 2, 3\}$, dimensionality of the outputs of the GCN layers and the number of hidden units of the fully connected layers in $\{50, 100, 200\}$, α and λ in $\{10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}\}$. For the baselines, we adopt their default settings of

[5] https://github.com/tkipf/gcn

hyperparameters.

Note that, in this work, we consider the scenarios where each individual's potential outcomes are not influenced by the observed treatments or outcomes of others in the network, i.e., there is no interference or spillover effect. At the same time, the auxiliary network is utilized as a source of information to help us learn better representations of confounders. Also note that the proposed network deconfounder framework is the first framework which incorporates auxiliary network information to learn better representations for controlling confounding bias and estimating individual treatment effects. Therefore, there does not exist baseline methods that can naturally incorporate the auxiliary network information. But we can concatenate the corresponding row of adjacency matrix to the original features to enable the baselines to utilize the network information. However, due to the issues of high dimensionality and sparsity, we find such an approach cannot improve baselines' performance. Then we describe the baseline methods which represent the state-of-the-art methods for the task of learning ITEs from observational data.

Counterfactual Regression (CFR). CFR (Shalit *et al.*, 2017) is based on the strong ignorability assumption. It learns representations of confounders by mapping the original features into a latent space. CFR is trained by minimizing the error in inferred factual outcomes and tries to minimize the imbalance of confounders' representations between the treated and the controlled. Following (Shalit *et al.*, 2017), two types of representation balancing penalties are considered: the Wasserstein-1 distance (CFR-Wass) and the maximum mean discrepancy (CFR-MMD).

Treatment-agnostic Representation Networks (TARNet). TARnet (Shalit *et al.*, 2017)is a variant of CFR. It does not have the representation balancing term.

Causal Effect Variational Autoencoder (CEVAE). CEVAE (Louizos *et al.*, 2017) is a deep latent-variable model which estimates ITEs via modeling the joint

distribution $P(\boldsymbol{x}, t, y, \boldsymbol{h})$. It learns representations of confounders as Gaussian distributions. Then through variational inference, it is trained by maximizing the variational lower bound of the graphical model representing the causal relations between the four variables: the features, the treatment, the outcome and the confounders.

Causal Forest. Causal Forest (Wager and Athey, 2018) is an extension of Breiman's random forest (Breiman, 2001) for estimating heterogenous treatment effects in subgroups. Here, we treat the heterogenous treatment effect estimated by causal forest of a subgroup as the ITE of each instance in the subgroup. It works with the strong ignorability assumption.

Bayesian Additive Regression Trees (BART). BART (Hill, 2011) is a Bayesian regression tree based ensemble model which is widely adopted in the literature of causal inference. It is also based on the strong ignorability assumption.

Two widely used evaluation metrics, the Rooted Precision in Estimation of Heterogeneous Effect ($\sqrt{\epsilon_{PEHE}}$) and Mean Absolute Error on ATE (ϵ_{ATE}), are adopted by this work. Formally, they are defined as:

$$\begin{aligned} \sqrt{\epsilon_{PEHE}} &= \sqrt{\frac{1}{n}\sum_{i=1}(\hat{\tau}_i - \tau_i)^2}, \\ \epsilon_{ATE} &= |\frac{1}{n}\sum_{i=1}(\hat{\tau}_i) - \frac{1}{n}\sum_{i=1}(\tau_i)|, \end{aligned} \tag{3.14}$$

where $\hat{\tau}_i = \hat{y}_i^1 - \hat{y}_i^0$ and $\tau_i = y_i^1 - y_i^0$ denote the inferred ITE and the ground truth ITE for the i-th instance.

3.3.3 Results

Effectiveness. First, we compare the effectiveness of the proposed framework, the network deconfounder, with the aforementioned state-of-the-art methods. Table 3.2 summarizes the empirical results on the BlogCatalog and Flickr datasets with $C = 5, \kappa_1 = 10$ and $\kappa_2 \in \{0.5, 1, 2\}$. We summarize the observations from these

experimental results as follows:

- The proposed network deconfounder framework consistently outperforms the state-of-the-art baseline methods on the semi-synthetic datasets with treatments and outcomes generated under various settings. We also perform one-tailed T-test to verify the statistical significance. The results indicate that the network deconfounder achieves significantly better estimations on individual treatment effects with a significant level of 0.05.

- With the capability to recognize the patterns of hidden confounders from the network structure, the network deconfounder suffers the least when the influence of hidden confounders grows (from $\kappa_2 = 0.5$ to $\kappa_2 = 2$) in terms of the increase in the errors $\sqrt{\epsilon_{PEHE}}$ and ϵ_{ATE}.

3.4 Proposed Framework II – IGNITE

This section presents the two desiderata of handling confounding bias and the description of the proposed framework.

3.4.1 Two Desiderata of Handling Confounding Bias

Hidden confounders pose the main challenge of learning ITEs from networked observational data. To handle confounding bias, existing methods present two desiderata.

First, on the group level, it is desirable to balance the distributions of confounders (or their representations) between the treated and the controlled. A variety of representation balancing methods for learning ITEs from observational data have been developed based on this principle Shalit *et al.* (2017); Yao *et al.* (2018). Let $\hat{\mathbf{h}}_i$ denote the approximated latent confounders' representation of instance i, the representation

balancing methods that follow the first desideratum minimize a divergence metric (e.g., Wasserstein distance) between $P(\hat{\mathbf{h}}_i|\mathbf{x}_i, t_i = 1)$ and $P(\hat{\mathbf{h}}_i|\mathbf{x}_i, t_i = 0)$. The second desideratum, on the individual level, aims to capture the patterns of hidden confounders that are useful in predicting treatments. Following this idea, methods proposed in Louizos *et al.* (2017); Veitch *et al.* (2019) learn a function that predicts the observed treatment of each individual based on the confounders' representations. Intuitively, this treatment prediction function mimics the treatment assignment mechanism that generates the data. Therefore, through learning the treatment prediction function, we can capture the information of hidden confounders that explains how the observed treatments are assigned. However, none of the existing methods can satisfy the two desiderata together because they seem to contradict each other. Intuitively, when the divergence between $P(\hat{\mathbf{h}}_i|\mathbf{x}_i, t_i = 1)$ and $P(\hat{\mathbf{h}}_i|\mathbf{x}_i, t_i = 0)$ becomes smaller, it becomes more difficult to distinguish between a treated instance and a controlled one by their confounders' representations. We introduce how to resolve this issue with a minimax game in the next section.

3.4.2 The Proposed Framework: IGNITE

We observe that confounders' representations and treatment predictions are often computed by two separate modules. This implies we can develop a minimax game where they are iteratively optimized toward satisfying the two desiderata. We propose IGNITE to learn ITEs from networked observational data. Here, we first introduce the components of IGNITE, then we formulate its loss function including the minimax game for handling confounding bias.

Components of IGNITE. IGNITE has three components: the confounder representation function, the treatment group's critic function, and the outcome inference function.

Confounder Representation Function. Here, we define the confounder representation function $g : \mathcal{X} \times \mathcal{A} \to \mathbb{R}^d$. This function maps the features and the adjacency matrix of the network structure into a d-dimensional representation space to approximate the confounders. To quantify the importance of each edge in its influence on the confounders, we extend the Graph Attention Network layers (GAT) Veličković *et al.* (2018). The i-th instance's confounder representation is a function of its features and network structure. For the simplicity of notation, we formulate the confounder representation function g with a single GAT layer:

$$\hat{\mathbf{h}}_i = g(\mathbf{x}_i, \mathbf{A}) = \|_{k=1}^{K} \delta(\sum_{j \in \mathcal{N}_i} \alpha_{ij}^k \mathbf{W}^k \mathbf{x}_j) \tag{3.15}$$

where $\|$ denotes concatenation. $\mathcal{N}_i$ is the set of neighbors of the i-th instance in the network $\mathbf{A}$. K is the number of attention heads. Each head of the attention mechanism is a weighted aggregation of information from the neighbors. $\mathbf{W}^k \in \mathbb{R}^{d \times m}$ is the weight matrix of the k-th attention head. δ is the ELU unit. We compute the normalized attention coefficients α_{ij}^k as:

$$\alpha_{ij}^k = \frac{\exp(\delta'(\mathbf{a}^T[\mathbf{W}^k\mathbf{x}_i \parallel \mathbf{W}^k\mathbf{x}_j]))}{\sum_{l \in \mathcal{N}_i} \exp(\delta'(\mathbf{a}^T[\mathbf{W}^k\mathbf{x}_i \parallel \mathbf{W}^k\mathbf{x}_l]))}, \tag{3.16}$$

where δ' denotes the LeakyReLU unit and $\mathbf{a} \in \mathbb{R}^{2d}$ denotes a weight vector. Stacking multiple GAT layers can help us capture Multi-hop relations.

Treatment Group Critic Function. The critic function $D : \mathbb{R}^d \to \mathbb{R}$ maps the confounders' representation of an instance to a real value. Larger value of $D(\hat{\mathbf{h}}_i)$ indicates that instance i is more likely to receive treatment. Following Gulrajani *et al.* (2017), we parameterize it with a neural network that consists of fully connected layers and LeakyReLU units.

Outcome Inference Function. We infer outcomes of an instance based on its confounders' representation. We define the output function $f : \mathbb{R}^d \times \{0, 1\} \to \mathbb{R}$. We

parameterize the output function of each treatment with fully connected layers with ELU units (except the last layer). We can set $t = t_i$ or $1 - t_i$ to let the corresponding layers infer the factual or counterfactual outcome.

With these three components, given the features of the i-th instance $\mathbf{x}_i$, the treatment t, and the adjacency matrix $\mathbf{A}$, outcomes are inferred as $\hat{y}_i^t = f(g(\mathbf{x}_i, \mathbf{A}), t)$, where $\hat{y}_i^t$ is the inferred outcome of instance i under treatment t. After training, it can infer the ITE of instance i as $\hat{\tau}_i = \hat{y}_i^1 - \hat{y}_i^0$ and estimate the ATE as $\frac{1}{n}\sum_i \hat{\tau}_i$.

A Minimax Game for Handling Confounding Bias. Note that function g is used to compute confounders' representations $\hat{\mathbf{h}}_i$. Here, we formulate the two desiderata of handling confounding bias as a minimax game:

$$\min_g \max_D \mathcal{L}_{CB} = \frac{1}{n^1} \sum_{i:t_i=1} D(\hat{\mathbf{h}}_i) - \frac{1}{n^0} \sum_{i:t_i=0} D(\hat{\mathbf{h}}_i), \tag{3.17}$$

where n^1 and n^0 are the number of instances under treatment and control. In the maximization stage, the critic function D is trained to maximize the difference between the value it assigns for the treated instances and those for the controlled ones. In the minimization stage, the confounder representation function g is used to fool the treatment group critic D. This step balances the distributions of confounders' representations because it makes it more difficult to distinguish the confounders' representation of a treated instance from that of a controlled one. To avoid difficulty in training (e.g., vanishing gradients), we follow Gulrajani *et al.* (2017) to limit the functional space of the treatment group critic D to a subset of 1-Lipschitz functions. To achieve this, we add a gradient penalty term to the maximization stage. It is computed on n' randomly sampled pairs of treated and controlled instances:

$$\mathcal{L}_{GP} = -\frac{1}{n'} \sum_{i=1}^{n'} \lambda(|| \bigtriangledown_{\tilde{\mathbf{h}}_i} D(\tilde{\mathbf{h}}_i)||_2 - 1)^2, \tag{3.18}$$

where $\tilde{\mathbf{h}}_i = \epsilon \hat{\mathbf{h}}_j + (1 - \epsilon)\hat{\mathbf{h}}_k$, (j, k) is one of the n' randomly sampled pairs. Each pair contains a treated instance and a controlled one. $|| \cdot ||_2$ denotes L_2 norm and

$\epsilon \sim U[0,1]$. We set the parameter $\lambda = 10$ as in Gulrajani *et al.* (2017). In addition, we aim to achieve accurate inference of factual outcomes. We minimize the mean squared error on the inferred factual outcomes:

$$\mathcal{L}_{FO} = \frac{1}{n} \sum_i (\hat{y}_i^{t_i} - y_i)^2, \tag{3.19}$$

Finally, we present the objective functions of the proposed minimax game in two stages:

$$\begin{aligned} \max_D \mathcal{L}_D &= \beta(\mathcal{L}_{CB} + \mathcal{L}_{GP}), \\ \min_g \mathcal{L}_g &= \mathcal{L}_{FO} + \beta(\mathcal{L}_{CB}), \end{aligned} \tag{3.20}$$

where $\beta \geq 0$ is a hyperparameters controlling the trade-off between the objectives. IGNITE is trained with backpropagation by iteratively optimizing $\mathcal{L}_D$ and $\mathcal{L}_g$.

3.5 Experimental Evaluation II

In this section, we investigate the two following research questions: **RQ1.** In learning ITEs from networked observational data, is the proposed minimax game more effective in handling confounding bias than representation balancing, treatment prediction or a combination of them? **RQ2.** How does the hyperparameter β affect the performance of the proposed framework, IGNITE?

3.5.1 Dataset Description

The dataset is generated in a similar way as in (Guo *et al.*, 2020c). To mimic real-world situations, we consider unobserved edge weights.

BlogCatalog (BC) is a social network with blog service. Each instance is a blogger. Each edge signifies the friendship between two bloggers. The features are the keywords of each blogger's articles. We extend the BlogCatalog dataset (Li *et al.*,

2019b) by synthesizing (a) the outcomes – the number of readers who adopt fintech products after reading each blogger's work; and (b) the treatment assignments – whether work of a blogger is browsed more on desktops or on mobile devices. The following assumptions are made: (1) Readers either read on mobile devices or desktops. A blogger is treated (controlled) if her blogs are more popular on mobile devices (desktops). (2) A blogger's articles are either more popular on mobile devices or desktops. (3) A blogger's treatment and outcomes can be influenced by her topics and her neighbors' topics. To synthesize treatments and outcomes, we train an LDA topic model on a large corpus. Then the centroids of the two treatment groups are defined as: (i) the topic distribution of a randomly selected blogger is the centroid of the treatment group, denoted by $\bar{r}^1$; (ii) the centroid of the controlled, $\bar{r}^0$, is the average topic distribution of all the bloggers. Then the treatments and outcomes are generated based on the similarity between the topic distributions of bloggers and the two centroids. Let $r(\mathbf{x}_i)$ denote the topic distribution of the i-th blogger, we model the readers' preference of browsing devices on the blogger's content:

$$Pr(t = 1|\mathbf{x}_i, \tilde{\mathbf{A}}) = \frac{\exp(p_i^1)}{\exp(p_i^1) + \exp(p_i^0)}, \tag{3.21}$$

where p_i^t is calculated as:

$$p_i^t = \kappa_1 r(\mathbf{x}_i)^T \bar{r}^t + \kappa_2 (\tilde{\mathbf{A}} r(\mathbf{x}_j))^T \bar{r}^t, \tag{3.22}$$

where $t \in \{0, 1\}$. $\kappa_1 \geq 0$ ($\kappa_2 \geq 0$) signifies the strength of the confounding bias resulting from a blogger's (her neighbors') topics. When $\kappa_1 = \kappa_2 = 0$ the treatment assignment is random and the greater the value κ_1 and κ_2 are, the more significant the bias of device preference is. $\tilde{\mathbf{A}}$ denotes the weighted adjacency matrix, where each entry $\tilde{\mathbf{A}}_{ij}$ denotes the importance of an edge with related to the influence on confounding bias. To emphasize the fact that in many real-world networks the edge

weights are unknown, we only let the unweighted adjacency matrix $\mathbf{A}$ be observed in the data. However, the unobserved weighted adjacency matrix $\tilde{\mathbf{A}}$ is the one that influences the values of treatments and outcomes. Thus, an ideal causal inference approach needs to catch the weights of each edge. If $\mathbf{A}_{ij} = 1$, we sample $\tilde{\mathbf{A}}_{ij} = \tilde{\mathbf{A}}_{ji} \sim U(0.8, 1.2)$; otherwise, we set $\tilde{\mathbf{A}}_{ij} = \tilde{\mathbf{A}}_{ji} = 0$. Outcomes of a blogger are simulated as:

$$y^t(\mathbf{x}_i) = C(p_i^0 + tp_i^1) + \epsilon, \tag{3.23}$$

where C is a scaling factor and $\epsilon \sim \mathcal{N}(0, 1)$. We set $C = 5, \kappa_1 = 10, \kappa_2 \in \{0.5, 1, 2\}$. 50 LDA topics are learned from the training corpus. Then we reduce the vocabulary by taking the union of the most probable 100 words from each topic, which results in 2,173 bag-of-word features.

Flickr is an image and video sharing service. Each instance refers to a user and each edge represents the social relationship between two users. The features of each user represent a list of tags of interest. We adopt the same settings and assumptions as we do for the BC datasets. Thus, we study the ITE of being viewed on mobile devices on the number of readers' adoptions of fintech products recommended by the user's images and videos. We learn 50 topics from the training corpus using LDA and concatenate the top 25 words of each topic which reduces the feature dimension to 1,210. We set the parameters the same as the BC datasets.

In Table 4.1, we present the statistics of the semi-synthetic datasets. The average and standard deviation of ATE are calculated over the 10 runs under each setting of parameters. The ATE varies because the true edge weights are randomly sampled from the uniform distribution $U(0.8, 1.2)$.

3.5.2 Experimental Settings

We randomly split the data into training (60%), validation (20%), and test sets (20%), which is repeated ten times for each simulated dataset. We train IGNITE with Adam (Kingma and Ba, 2014) optimizer with weight decay set to 10^{-4}. We iteratively optimize the two objectives in Eq. (3.20). Grid search finds the optimal set of hyperparameters for IGNITE and the baselines. For IGNITE, we search learning rate in $\{5 \times 10^{-3}, 10^{-3}, 5 \times 10^{-4}, 10^{-4}\}$, the number of GAT layers and fully connected layers of the functions g, D and f in $\{1, 2, 3\}$, the number of hidden units of the GAT layers and the fully connected layers in $\{16, 32, 64, 128\}$, the number of attention heads in $\{2, 4, 8\}$, β in $\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$. Then, we list the baselines:

- **Network Deconfounder (ND)** (Guo *et al.*, 2020c) learns confounders' representations using GCN layer(s) (Kipf and Welling, 2016). It minimizes the Wasserstein distance between the two confounder representation distributions.

- **GATD** is a variant of ND with GAT layer(s) (Veličković *et al.*, 2018) for fair comparison.

- **GATD+ and GATDT.** To show the advantage of the proposed minimax game over a simple combination of representation balancing and treatment prediction, we further create two variants of GATD. GATD+ balances confounder representations and predicts treatments based on these representations. GATDT predicts treatments to handle confounding bias.

- **CNE** (Veitch *et al.*, 2019) learns confounders' representations by predicting observed outcomes, treatments and edges. It does not utilize observed features. CNE uses AIPW (Robins *et al.*, 1994), therefore, only infers ATE.

- **CNE-.** We create a variant of CNE w/o AIPW, which can infer both ATE and ITEs.

- **Counterfactual Regression (CFR) (Shalit *et al.*, 2017)** is a ITEs estimator for i.i.d. data. It minimizes errors on inferred factual outcomes and balances representation distributions. We report the optimal results of the three CFR models: representation balancing with Wasserstein distance, that with Maximum Mean Discrepancy and no representation balancing.

- **CEVAE (Louizos *et al.*, 2017)** is a deep latent-variable model for learning ITEs. It learns the joint distribution of features, latent confounders, treatments, and outcomes to infer ITEs.

- **Causal Forest (Wager and Athey, 2018)** is an ensemble model trained by predicting observed treatments.

For the evaluation metrics, the Rooted Precision in Estimation of Heterogeneous Effect ($\sqrt{\epsilon_{PEHE}}$) and Mean Absolute Error on ATE (ϵ_{ATE}), are used. They are defined as:

$$\sqrt{\epsilon_{PEHE}} = \sqrt{\frac{1}{n}\sum_{i=1}(\hat{\tau}_i - \tau_i)^2}, \epsilon_{ATE} = |\frac{1}{n}\sum_{i=1}(\hat{\tau}_i) - \frac{1}{n}\sum_{i=1}(\tau_i)|, \tag{3.24}$$

where $\hat{\tau}_i$ and $\tau_i = y_i^1 - y_i^0$ denote the inferred ITE and the ground truth ITE for the i-th instance.

3.5.3 Experimental Results

Effectiveness. Here, we compare the effectiveness of IGNITE with the baselines in the task of learning ITEs from networked observational data. Table 4.2 shows the results evaluated on the BC and Flickr datasets with $C = 1, \kappa_1 = 10$ and $\kappa_2 \in$

$\{0.5, 1, 2\}$. We summarize the observations made from these experimental results as follows:

- IGNITE outperforms the baselines consistently in almost all cases. One-tailed T-tests show that the boldfaced results are significantly better than others with a significant level of 0.05.

- IGNITE shows consistent superior performance than GATD+. This verifies that the proposed minimax game does a better job in satisfying the two desiderata than a simple combination of representation balancing and treatment prediction.

- The fact that IGNITE outperforms GATD and GATDT implies that the proposed minimax game handles confounding bias better than doing representation balancing or treatment prediction alone.

- We observe that GATD+ fails to outperform GATD and GATDT in a majority of cases. This implies that a naïve combination of representation balancing and treatment prediction may not achieve the two desiderata together. Instead, it may perform worse than representation balancing or treatment prediction alone.

- GATD outperforms ND under various settings. This is because GAT layers can capture the unobserved edge importance. Note that the unobserved edge importance plays may have a significant influence on the values of treatments and outcomes.

- The improvement of IGNITE over CNE and CNE- results from two aspects. First, the proposed minimax game shows better efficacy in dealing with confounding bias than treatment prediction alone. Second, the GAT layer(s) capture unobserved edge weights and incorporate observed features.

- Compared to the methods for i.i.d. data – CFR, CEVAE, and CF, IGNITE achieves better performance because it is trained by the proposed minimax game for handling confounding bias and it utilizes the network information to recognize patterns of latent confounders.

Parameter Study. Then we investigate how the variation in values of the important hyperparameter β affects the performance of IGNITE. β controls the trade-off between more accurate outcome inference and better confounding bias handling. We set β to $\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$. The following settings are applied: learning rate is 5×10^{-3}, the number of epochs is 300, the number of GAT layer is 2 and the numbers of fully connected layers for D and f are 2 and 1, the number of attention head is 8, the number of hidden units of each attention head and each fully connected layer of D and f are 128, 64 and 32. We show the results of this parameter study on the BC datasets in Table 3.5. We observe that IGNITE maintains reasonably consistent performance in terms of both evaluation metrics when $\beta \in [10^{-4}, 10^{-1}]$. In addition, IGNITE often achieves the optimal performance when $\beta \in [10^{-3}, 10^{-2}]$.

3.6 Summary

New challenges are presented by the prevalence of networked observational data for learning individual treatment effects. In this chapter, we formulate a novel problem, learning individual treatment effects from networked observational data. As the underlying network structure could capture useful information of hidden confounders, we propose two novel frameworks, which leverage the network structural patterns along with original features for learning better representations of confounders. Empirically, we perform extensive experiments across multiple real-world datasets. Empirical results verify that the proposed minimax game training paradigm learns better representation of confounders than the state-of-the-art methods.

Here, we also introduce two most interesting directions of future work. First, we are interested in leveraging other types of structure between instances for learning ITEs from observational data. For example, temporal dependencies can also be utilized to capture patterns of hidden confounders. Second, real-world networks can evolve over time (Marin *et al.*, 2017; Sarkar *et al.*, 2019; Shakarian *et al.*, 2015a). Hence, investigating how to exploit dynamics in evolving networks for learning ITEs creates new opportunities and poses new challenges.

Table 3.2: Experimental Results comparing effectiveness of the proposed network deconfounder with the baseline methods.

BlogCatalog						
κ_2	0.5		1		2	
	$\sqrt{\epsilon_{PEHE}}$	ϵ_{ATE}	$\sqrt{\epsilon_{PEHE}}$	ϵ_{ATE}	$\sqrt{\epsilon_{PEHE}}$	ϵ_{ATE}
NetDeconf (ours)	**4.532**	**0.979**	**4.597**	**0.984**	**9.532**	**2.130**
CFR-Wass	10.904	4.257	11.644	5.107	34.848	13.053
CFR-MMD	11.536	4.127	12.332	5.345	34.654	13.785
TARNet	11.570	4.228	13.561	8.170	34.420	13.122
CEVAE	7.481	1.279	10.387	1.998	24.215	5.566
Causal Forest	7.456	1.261	7.805	1.763	19.271	4.050
BART	4.808	2.680	5.770	2.278	11.608	6.418
Flickr						
	$\sqrt{\epsilon_{PEHE}}$	ϵ_{ATE}	$\sqrt{\epsilon_{PEHE}}$	ϵ_{ATE}	$\sqrt{\epsilon_{PEHE}}$	ϵ_{ATE}
NetDeconf (ours)	**4.286**	**0.805**	**5.789**	**1.359**	**9.817**	**2.700**
CFR-Wass	13.846	3.507	27.514	5.192	53.454	13.269
CFR-MMD	13.539	3.350	27.679	5.416	53.863	12.115
TARNet	14.329	3.389	28.466	5.978	55.066	13.105
CEVAE	12.099	1.732	22.496	4.415	42.985	5.393
Causal Forest	8.104	1.359	14.636	3.545	26.702	4.324
BART	4.907	2.323	9.517	6.548	13.155	9.643

Dataset	Instances	Edges	Features	κ_2	Average ATE $\pm$ STD
BC	5,196	173,468	8,189	0.5	6.079 $\pm$ 2.962
				1	9.012 $\pm$ 3.602
				2	20.003 $\pm$ 8.132
Flickr	7,575	239,738	12,047	0.5	5.130 $\pm$ 0.892
				1	7.576 $\pm$ 0.715
				2	13.445 $\pm$ 2.093

Table 3.3: Statistics of the Datasets

	BC					
	$\kappa_2 = 0.5$		$\kappa_2 = 1$		$\kappa_2 = 2$	
	$\sqrt{\epsilon_{PEHE}}$	ϵ_{ATE}	$\sqrt{\epsilon_{PEHE}}$	ϵ_{ATE}	$\sqrt{\epsilon_{PEHE}}$	ϵ_{ATE}
IGNITE	**4.415**	**0.506**	**6.163**	**0.971**	**10.998**	**2.514**
GATD+	5.132	0.666	8.442	2.159	17.167	10.74
GATD	5.170	1.070	7.989	1.779	16.574	5.942
GATDT	5.165	1.055	8.017	1.863	16.578	5.940
ND	5.386	2.070	10.403	4.811	20.286	10.350
CNE	–	7.314	–	13.212	–	24.298
CNE-	10.323	8.194	18.839	14.991	33.607	26.531
CFR	10.073	5.000	15.229	9.631	36.680	16.481
CEVAE	6.812	3.129	12.055	2.700	24.128	14.576
CF	5.941	3.349	10.413	3.336	19.145	16.812
	Flickr					
	$\kappa_2 = 0.5$		$\kappa_2 = 1$		$\kappa_2 = 2$	
	$\sqrt{\epsilon_{PEHE}}$	ϵ_{ATE}	$\sqrt{\epsilon_{PEHE}}$	ϵ_{ATE}	$\sqrt{\epsilon_{PEHE}}$	ϵ_{ATE}
IGNITE	**6.938**	**1.242**	**10.725**	2.006	**18.864**	**2.643**
GATD+	7.731	1.394	13.201	2.903	27.105	7.088
GATD	7.605	1.688	13.092	2.436	26.846	7.196
GATDT	7.602	1.681	13.075	2.452	26.781	7.099
ND	7.337	2.000	14.006	3.046	28.379	5.817
CNE	–	8.103	–	16.058	–	33.94
CNE-	14.109	9.001	26.536	17.275	54.906	35.262
CFR	9.826	3.619	16.859	7.240	45.150	12.787
CEVAE	11.836	2.678	22.171	3.493	48.840	7.360
CF	8.406	1.938	14.485	**1.821**	31.111	6.520

Table 3.4: Results on the two datasets with $\kappa_2 \in \{0, 1, 2\}$ measured by the two evaluation metrics $\sqrt{\epsilon_{PEHE}}$ and ϵ_{ATE}, the smaller the better.

		β	10^{-4}	10^{-3}	10^{-2}	10^{-1}
BC	$\kappa_2 = 0.5$	$\sqrt{\epsilon_{PEHE}}$	4.422	4.439	**4.415**	4.566
		ϵ_{ATE}	0.526	0.56	**0.506**	0.642
	$\kappa_2 = 1$	$\sqrt{\epsilon_{PEHE}}$	6.196	**6.163**	6.166	6.177
		ϵ_{ATE}	1.139	**0.971**	0.993	1.124
	$\kappa_2 = 2$	$\sqrt{\epsilon_{PEHE}}$	11.934	**10.998**	12.046	12.385
		ϵ_{ATE}	2.183	**2.514**	2.675	3.134

Table 3.5: Parameter study results on the BC datasets with $\kappa_2 \in \{0.5, 1, 2\}$ in terms of $\sqrt{\epsilon_{PEHE}}$ and ϵ_{ATE}, the smaller the better.

Chapter 4

COUNTERFACTUAL EVALUATION OF TREATMENT ASSIGNMENT FUNCTIONS WITH NETWORKED OBSERVATIONAL DATA

With massive population, it is a trend to deploy personalized treatment assignment using data-driven models. Examples that have been adopted in real-world applications include recommendation systems (Schnabel *et al.*, 2016), search ranking systems (Wu *et al.*, 2018; Joachims *et al.*, 2017) and computational advertising (Bottou *et al.*, 2013). Evaluating a novel treatment assignment function with observational data is a desirable application of learning causality. With observational data that can be collected effortlessly (e.g., log data from a recommendation system), counterfactual evaluation allows us to form a basic understanding of how a treatment assignment strategy performs without performing online randomized controlled trials. Although there exists a series of work on counterfactual evaluation (a.k.a offline policy evaluation) (Swaminathan and Joachims, 2015a; Zou *et al.*, 2019; Bennett and Kallus, 2019), the importance of auxiliary information that may contain patterns of hidden confounders has not been realized. In this section, we investigate the problem of how to effectively exploit network structure information for mitigating confounding bias in counterfactual evaluation.

4.1 Problem Statement

In this section, we present technical preliminaries and the problem statement.

First, we start with the notations. Lower alphabets (e.g., y_i) denote scalars, uppercase alphabets (e.g., N) signify constants, lower and upper boldface alphabets (e.g., $\boldsymbol{x}$ and $\boldsymbol{A}$) denote vectors and matrices.

In networked observational data, each instance i comes with a feature vector $\boldsymbol{x}_i \in \mathbb{R}^M$, an observed treatment $t_i \in \{0, 1\}$ and an observed (factual) outcome $y_i \in \mathbb{R}$. Besides, we observe a network connecting the instances, represented by its adjacency matrix $\boldsymbol{A} \in \{0, 1\}^{N \times N}$. To define the utility of a treatment function, we assume that there exists a potential outcome corresponding to each treatment-instance pair (t, i), i.e., $y_i(1)$ and $y_i(0)$ (Rubin, 2005). For each instance i, the observed outcome y_i (short for $y_i(t_i)$) takes the value on one of the potential outcomes, depending on the observed treatment (t_i). Formally, this can be written as $y_i(t_i) = t_i y_i(1) + (1 - t_i) y_i(0)$. Following the literature (Pearl, 2009), we call the unobserved outcomes $y_i(t),\ t \neq t_i$ the *counterfactual outcomes*.

As in almost all the cases, it is not possible to test whether the set of observed features contain all the confounders. Therefore, we adopt a realistic setting where hidden confounders exist. Thus, the strong ignorability assumption does not hold given observed features:

$$y_i(1), y_i(0) \not\perp\!\!\!\perp t_i | \boldsymbol{x}_i. \tag{4.1}$$

Instead, we only assume that there exist latent confounders $\boldsymbol{z}$ which satisfy

$$y_i(1), y_i(0) \perp\!\!\!\perp t_i | \boldsymbol{z}_i. \tag{4.2}$$

Note that the latent confounders are not observable in the data. But we can approximate them from the observed features and the network information.

Similar to its counterpart for i.i.d. data (Bennett and Kallus, 2019; Athey and Wager, 2017), in networked observational data, a treatment assignment function $\pi : \mathbb{R}^M \times \mathcal{A} \to (0, 1)$ maps an instance's feature vector to its probability to receive the treatment, where $\mathcal{A}$ is the set of possible adjacency matrices. Then $\pi^t(\boldsymbol{x})$ denotes the probability that treatment t assigned to an instance with features $\boldsymbol{x}$ by π. In accordance with (Kallus, 2018; Bennett and Kallus, 2019; Zou *et al.*, 2019), we define

the true utility of a treatment assignment function π as:

$$\tau(\pi) = \frac{1}{N} \sum_{i=1}^{N} \sum_{t \in \{0,1\}} \pi^t(\boldsymbol{x}_i, \boldsymbol{A}) y_i(t). \tag{4.3}$$

From this definition, we make the observation: when $t \neq t_i$, $y_i(t)$ is a counterfactual outcome which is not available in observational data. Therefore, counterfactual evaluation is a challenging problem.

Here, we present the problem statement. Based on the aforementioned notations and definitions, a formal statement of the problem is given as follows:

Problem 1. *Counterfactual Evaluation of the Treatment Assignment Functions with Networked Observational Data.*

Given: *networked observational data* $(\{(\boldsymbol{x}_i, t_i, y_i)\}_{i=1}^{N}, \boldsymbol{A})$ *and a novel treatment assignment function* π*.*

Estimate: *its true utility* $\tau(\pi)$ *on the given data.*

4.2 Background

Here, we review three types of long-established approaches for counterfactual evaluation with independent and identically distributed (i.i.d.) observational data: direct methods (Beygelzimer *et al.*, 2008), weighted estimators (Bottou *et al.*, 2013; Swaminathan and Joachims, 2015a), and doubly robust estimators (Dudík *et al.*, 2011). In this section, the symbol π denotes a treatment assignment function for i.i.d. observational data as $\pi : \mathbb{R}^M \to \mathbb{R}$. Given i.i.d. observational data $\{\boldsymbol{x}_i, t_i, y_i\}_{i=1}^{N}$ and policy π, the directed method (Qian and Murphy, 2011) estimates the policy value as:

$$\hat{\tau}(\pi) = \frac{1}{N} \sum_{i=1}^{N} \sum_{t} \pi^t(\boldsymbol{x}_i) \hat{y}_i(t), \tag{4.4}$$

where $\hat{y}_i(t)$ is the estimated outcome of instance i under treatment t. However, relying on the strong ignorability assumption makes direct methods suffer from the bias caused by hidden confounders (Beygelzimer *et al.*, 2008).

Alternatively, the weighted estimators (Bottou *et al.*, 2013; Swaminathan and Joachims, 2015a) are proposed to achieve counterfactual evaluation:

$$\hat{\tau}(\pi) = \frac{1}{N}\sum_{i=1}^{N} \hat{w}(\boldsymbol{x}_i, t_i) y_i, \tag{4.5}$$

where $\hat{w}(\boldsymbol{x}_i, t_i)$ maps the observed features and treatment of an instance to its weight. In weighted estimators, the utility of a treatment assignment function is estimated by a weighted average of factual outcomes. As a result, weighted estimators do not need to bother with counterfactual outcomes. Weighted estimators often adopt the inverse propensity scoring (IPS) weights (Kitagawa and Tetenov, 2018):

$$\hat{w}_{IPS}(\boldsymbol{x}_i, t_i) = \frac{\pi^{t_i}(\boldsymbol{x}_i)}{P(t = t_i|\boldsymbol{x})}, \tag{4.6}$$

where $P(t = t_i|\boldsymbol{x}_i)$ denotes the true probability of instance i to receive treatment t_i in the observational data. However, we often have to estimate $P(t = t_i|\boldsymbol{x}_i)$ as how the treatments are assigned in the observational data is unknown. To avoid the extreme values of estimated $P(t = t_i|\boldsymbol{x}_i)$, techniques including normalization and clipping have been introduced (Bottou *et al.*, 2013; Swaminathan and Joachims, 2015a,b).

The doubly robust estimator (Dudík *et al.*, 2011) estimates utility of treatment assignment functions based on estimated counterfactual outcomes and IPS weights:

$$\hat{\tau}(\pi) = \frac{1}{N}\sum_{i=1}^{N}[\sum_{t} \pi^t(\boldsymbol{x}_i)\hat{y}_i(t) + \hat{w}_{IPS}(\boldsymbol{x}_i, t_i)(y_i - \hat{y}_i(t_i))]. \tag{4.7}$$

We can see the three types of counterfactual evaluation methods cannot utilize the network information. The success of using network information to handle hidden confounders has been demonstrated in other tasks of causal inference (Veitch *et al.*,

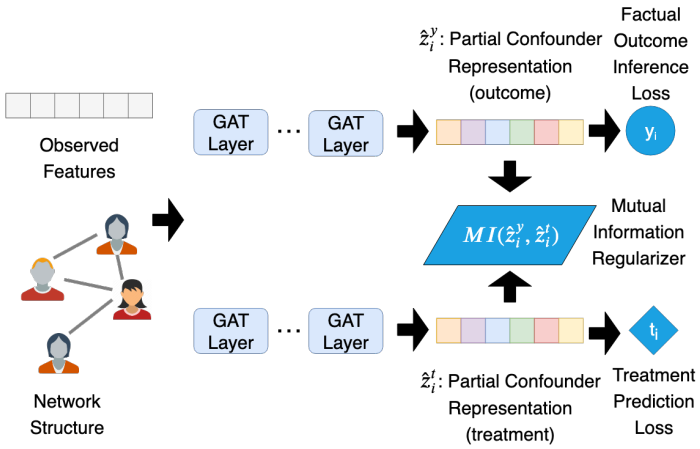


Figure 4.1: An overview of learning partial representations of latent confounders in the proposed framework CONE.

2019; Guo *et al.*, 2020c) (e.g., causal effect estimation). Motivated by the success, we investigate incorporating network information in counterfactual evaluation.

4.3 Proposed Framework – Counterfactual Network Evaluator (CONE)

In this section, we present the proposed framework to tackle the counterfactual evaluation problem with networked observational data. shows an overview of the proposed framework's training phase. As shown in Fig. 4.1, the goal of the training phase is to learn two partial representations of latent confounders with the supervision of the factual outcome and the observed treatment, respectively. Then in the inference phase, the learned partial representations would be utilized to infer the utility of a treatment assignment function. Then we cover the detailed description of the proposed framework in the rest of this section.

4.3.1 Learning Partial Repretations of Latent Confounders.

To leverage the network information in the procedure of learning latent confounders' partial representations, for each partial representation, a representation

learning function $g : \mathbb{R}^M \times \mathcal{A} \to \mathbb{R}^D$ maps the observed features along with the network information to the D-dimensional space of partial latent confounders. We let g^t and g^y denote the partial representation learning functions supervised by the observed treatment and the factual outcome, respectively. In this work, we approximate the functions, g^t and g^y, with Graph Attentional (GAT) layers (Veličković *et al.*, 2018) to capture the unknown edge weights in the real-world networked observational data. Intuitively, each GAT layer maps a feature vector and the network information to a partial representation vector. In this work, each GAT layer employs multi-head graph attention. To compute a partial representation vector of instance i, it concatenates the multiple heads' outputs. Each head outputs a weighted aggregation of information from the neighbors of instance i in the network $\boldsymbol{A}$ (Veličković *et al.*, 2018). An arbitrary number of GAT layers can be stacked to approximate the functions g^y and g^t. Here, for notation simplicity, each partial representation learning function is formulated by a single GAT layer as:

$$\begin{aligned} \hat{\boldsymbol{z}_i^t} = g^t(\boldsymbol{x}_i, \boldsymbol{A}) = \|_{k=1}^{K} \delta(\sum_{j \in \mathcal{N}_i} \alpha_{ij}^k \boldsymbol{W}^k \boldsymbol{x}_j) \\ \hat{\boldsymbol{z}_i^y} = g^y(\boldsymbol{x}_i, \boldsymbol{A}) = \|_{k=1}^{K} \delta(\sum_{j \in \mathcal{N}_i} \beta_{ij}^k \boldsymbol{U}^k \boldsymbol{x}_j), \end{aligned} \tag{4.8}$$

where $\|$ denotes concatenation. $\mathcal{N}_i$ signifies the set of neighbors of the i-th instance in the network $\boldsymbol{A}$. K is the number of attention heads. $\boldsymbol{W}^k, \boldsymbol{U}^k$ are the weight matrices of the k-th attention head. δ is the ELU activation unit. α_{ij}^k and β_{ij}^k are the normalized attention coefficients which represent the importance of the edge between instance i and j in the inference of the observed treatment and outcome, respectively. We compute them as:

$$\begin{aligned} \alpha_{ij}^k = \frac{\exp(\delta'(\boldsymbol{a}^T[\boldsymbol{W}^k \boldsymbol{x}_i \parallel \boldsymbol{W}^k \boldsymbol{x}_j]))}{\sum_{l \in \mathcal{N}_i} \exp(\delta'(\boldsymbol{a}^T[\boldsymbol{W}^k \boldsymbol{x}_i \parallel \boldsymbol{W}^k \boldsymbol{x}_l]))}, \\ \beta_{ij}^k = \frac{\exp(\delta'(\boldsymbol{b}^T[\boldsymbol{U}^k \boldsymbol{x}_i \parallel \boldsymbol{U}^k \boldsymbol{x}_j]))}{\sum_{l \in \mathcal{N}_i} \exp(\delta'(\boldsymbol{b}^T[\boldsymbol{U}^k \boldsymbol{x}_i \parallel \boldsymbol{U}^k \boldsymbol{x}_l]))}, \end{aligned} \tag{4.9}$$

where δ' denotes the LeakyReLU unit (Xu *et al.*, 2015) and $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^{2M}$ denotes the weight vectors. One can interpret $\delta'(\boldsymbol{a}^T[\boldsymbol{W}^k\boldsymbol{x}_i \parallel \boldsymbol{W}^k\boldsymbol{x}_j])$ and $\delta'(\boldsymbol{b}^T[\boldsymbol{U}^k\boldsymbol{x}_i \parallel \boldsymbol{U}^k\boldsymbol{x}_j])$ as the unnormalized attention coefficients of the edge between the instances i and j. Then these coefficients are normalized by applying the softmax function.

Factual Outcome Inference Loss. First, the supervision of factual outcomes is leveraged to learn the partial representation of latent confounders corresponding to the factual outcome, i.e., $\hat{\boldsymbol{z}^y}$. This partial representation contains information that is useful in the inference of factual outcomes. Specifically, we aim to learn a function $f^y : \mathbb{R}^D \rightarrow \mathbb{R}$ that maps the partial representation to the factual outcome. We implement the function f^y with a neural network with fully connected layers and ELU activation units (except the last layer). Therefore, we introduce a penalty term which minimizes the mean squared error on the inferred factual outcomes as:

$$\mathcal{L}^y = \frac{1}{N}\sum_{i=1}^{N}(f^y(\hat{\boldsymbol{z}_i^y}) - y_i)^2. \tag{4.10}$$

Observed Treatment Prediction Loss. Then, we utilize the observed treatment as the label to supervise the learning process of the latent confounders' treatment partial representation, i.e., $\hat{\boldsymbol{z}^t}$. Here, an observed treatment prediction function $f^t : \mathbb{R}^D \rightarrow (0, 1)$ that maps the partial representation to the estimated propensity score $\hat{P}(t = 1|\hat{\boldsymbol{z}}_i^t)$ is parameterized by a fully connected layer with a sigmoid activation as

$$\hat{P}(t = 1|\hat{\boldsymbol{z}^t}) = f^t(\hat{\boldsymbol{z}^t}) = \sigma(\mathbf{v}^T\hat{\boldsymbol{z}}_i^t + c), \tag{4.11}$$

where σ is the sigmoid function, $\mathbf{v}$ and c are the weight vector and the bias. Then we train the partial representation $\hat{\boldsymbol{z}^t}$ by minimizing the cross-entropy loss on predicting the observed treatment:

$$\mathcal{L}^t = -\frac{1}{N}\sum_i t_i \log(\hat{P}(t = 1|\hat{\boldsymbol{z}^t})) + (1 - t_i)\log(\hat{P}(t = 0|\hat{\boldsymbol{z}^t})). \tag{4.12}$$

Maximizing the Mutual Information between Partial Representations. Intuitively, both partial representations are learned to approximate part of the information contained in the latent confounders. Therefore, we propose to let the two partial representations agree with each other by maximizing the mutual information between the distributions of the two partial representations of latent confounders, i.e., $P(\hat{\boldsymbol{z}^t})$ and $P(\hat{\boldsymbol{z}^y})$. Mutual information is a measure of dependence between two random variables which gauges how much the uncertainty in one variable can be reduced by knowing the value of the other one. We know that mutual information is equivalent to the Kullback-Leibler (KL) divergence between the joint distribution and the product of the marginals (Belghazi *et al.*, 2018). It is often quite difficult to compute mutual information with multi-dimensional continuous random variables. Here, the Donsker-Varadhan representation of KL divergence (Donsker and Varadhan, 1983) is adopted to compute a tight lower bound of the mutual information between the distribution of the treatment partial latent confounders $P(\hat{\boldsymbol{z}^t})$ and that of the the outcome partial latent confounders $P(\hat{\boldsymbol{z}^y})$ as:

$$\begin{aligned} &MI(\hat{\boldsymbol{z}^t}, \hat{\boldsymbol{z}^y}) = D_{KL}(P(\hat{\boldsymbol{z}^t}, \hat{\boldsymbol{z}^y})||P(\hat{\boldsymbol{z}^y}) \otimes P(\hat{\boldsymbol{z}^y})) = \\ &\sup_{h \in \mathcal{H}} \mathbb{E}_{P(\hat{\boldsymbol{z}^t}, \hat{\boldsymbol{z}^y})}[h(\hat{\boldsymbol{z}^t}, \hat{\boldsymbol{z}^y})] - \log(\mathbb{E}_{P(\hat{\boldsymbol{z}^y}) \otimes P(\hat{\boldsymbol{z}^y})}[e^{h(\hat{\boldsymbol{z}^t}, \hat{\boldsymbol{z}^y})}]), \end{aligned} \tag{4.13}$$

where $h : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}$ is a function that maps the two partial representations to a real number and $\mathcal{H}$ denotes the space of such functions. We can confirm that given the function h, the lower bound of the mutual information can be efficiently computed by sampling $(\hat{\boldsymbol{z}^t}, \hat{\boldsymbol{z}^y})$ from the empirical joint distribution $P(\hat{\boldsymbol{z}^t}, \hat{\boldsymbol{z}^y})$ and sampling $\hat{\boldsymbol{z}^t}$ and $\hat{\boldsymbol{z}^y}$ separately from the empirical marginals $P(\hat{\boldsymbol{z}^t})$ and $P(\hat{\boldsymbol{z}^y})$. Here, we parameterize the function h with a neural network with fully connected layers and ELU activation (except the last layer). Then we can formulate the penalty term that maximizes the lower bound of the mutual information between the two partial

representations as:

$$\mathcal{L}^{MI} = -\mathbb{E}_{P(\hat{\boldsymbol{z}}^t, \hat{\boldsymbol{z}}^y)}[h(\hat{\boldsymbol{z}}^t, \hat{\boldsymbol{z}}^y)] + \log(\mathbb{E}_{P(\hat{\boldsymbol{z}}^y)\otimes P(\hat{\boldsymbol{z}}^y)}[\exp^{h(\hat{\boldsymbol{z}}^t, \hat{\boldsymbol{z}}^y)}]). \tag{4.14}$$

Finally, we can formulate the training objective as

$$\arg\min_{\boldsymbol{\theta}_{-h}, \boldsymbol{\theta}_h} \mathcal{L} = \mathcal{L}^y + \gamma\mathcal{L}^t + \zeta\mathcal{L}^{MI}, \tag{4.15}$$

where $\boldsymbol{\theta}_{-h}$ denotes the parameters of those components implementing the functions g^t, g^y, f^t and f^y; while $\boldsymbol{\theta}_h$ signifies the parameters of the component implementing the function h. The hyperparameters γ and ζ are non-negative scalars that control the trade-off between the three penalty terms.

4.3.2 Optimization

Here, we describe the optimization algorithm with which the proposed framework learns the partial representations. Algorithm 1 exhibits an overview of this optimization algorithm. In each epoch, the framework first computes the loss function $\mathcal{L}$ (Step 2-6 in Algorithm 1) in the feed-forward direction. Then, in Step 7 and 8 of Algorithm 1, the two sets of parameters $\boldsymbol{\theta}_{-h}$ and $\boldsymbol{\theta}_h$ are updated by applying one step of gradient descent where the gradients are computed by the Adam optimizer (Kingma and Ba, 2014).

4.3.3 Counterfactual Evaluation

With the functions trained, we can compute the partial representations of any instance in the networked observational data. However, the counterfactual outcomes and the propensity scores inferred by the functions f^y and f^t can be suboptimal because each of them only uses one of the partial representations. To overcome this issue, we propose to combine the partial representations to estimate the utility of

Algorithm 1 Learning the partial representations of latent confounders

Input: learning rate η; hyperparameters γ and ζ; number of iterations E; the functions g^t, g^y, f^t, f^y and h; networked observational data $\left\{(\boldsymbol{x}_i, t_i, y_i)_{i=1}^N, \boldsymbol{A}\right\}$;.

Output: Partial representations: $\hat{\boldsymbol{z}}_i^t$ and $\hat{\boldsymbol{z}}_i^y$.

Init : Let iteration counter $e = 0$; Initialize model parameters $\boldsymbol{\theta}_{-h}$ and $\boldsymbol{\theta}_h$ with Xavier initialization.

1: **while** $e \leq E$ **do**

2: compute the partial representations $\hat{\boldsymbol{z}}_i^t$ and $\hat{\boldsymbol{z}}_i^y$ with Eq. (4.8).

3: compute $\mathcal{L}^y$ and $\mathcal{L}^t$ with Eq. (4.10) and Eq. (4.11).

4: use $(\hat{\boldsymbol{z}}_i^t, \hat{\boldsymbol{z}}_i^y)_{i=1}^N$ as samples of the joint distribution $P(\hat{\boldsymbol{z}}^t, \hat{\boldsymbol{z}}^y)$;

5: use $(\hat{\boldsymbol{z}}_i^t)_{i=1}^N, (\hat{\boldsymbol{z}}_{j(i)}^y)_{i=1}^N$ as the samples from the two marginals $P(\hat{\boldsymbol{z}}^t)$ and $P(\hat{\boldsymbol{z}}^y)$, where $j(i)$ is the i-th element of the permuted index vector $permute([1, ..., N])$.

6: compute $\mathcal{L}^{MI}$ and $\mathcal{L}$ with Eq. (4.14) and (4.15).

7: update $\boldsymbol{\theta}_{-h} \leftarrow Adam(\mathcal{L}, \boldsymbol{\theta}_{-h})$.

8: update $\boldsymbol{\theta}_h \leftarrow Adam(\mathcal{L}, \boldsymbol{\theta}_h)$.

9: **end while**

a treatment assignment function. In particular, we take the concatenation of them to form the representation of latent confounders as $\hat{\boldsymbol{z}}_i = concat([\hat{\boldsymbol{z}}_i^y, \hat{\boldsymbol{z}}_i^t])$. Then the representation of latent confounders is used to train a doubly robust estimator. First, to infer counterfactual outcomes, we follow the direct naive method in (Bennett and Kallus, 2019). In particular, for each treatment group, we simply train a neural network with fully connected layers and ELU activation (expect the last layer) with $(\hat{\boldsymbol{z}}_i)_{i=1}^N$ as the input and the factual outcomes $(y_i)_{i=1}^N$ as the label. Second, to estimate the propensity scores, a logistic regression model is trained with $(\hat{\boldsymbol{z}}_i)_{i=1}^N$ as input and the observed treatments $(t_i)_{i=1}^N$ as the label. Then with the two trained models

and the latent confounder representations $\hat{\boldsymbol{z}}_i$, we adapt the original doubly robust estimator (Eq. (4.7)) to infer the utility of a treatment assignment function π with networked observational data as:

$$\begin{aligned}\hat{\tau}(\pi) = \frac{1}{N}\sum_{i=1}^{N}[\sum_{t}\pi^t(\boldsymbol{x}_i, \boldsymbol{A})\hat{y}_i(\hat{\boldsymbol{z}}_i, t) \\ + \hat{w}_{SNIPS}(\hat{\boldsymbol{z}}_i, t_i)(y_i - \hat{y}_i(\hat{\boldsymbol{z}}_i, t_i))],\end{aligned} \tag{4.16}$$

where $\hat{y}_i(\hat{\boldsymbol{z}}_i, t_i)$ is the outcome inferred by the simple direct method. In terms of the sample weights, we adopt the self-normalized inverse propensity scoring ($\hat{w}_{SNIPS}$) to avoid the extreme values and reduce variance (Swaminathan and Joachims, 2015b). The self-normalized inverse propensity scoring weights are computed as:

$$\hat{w}_{SNIPS}(\hat{\boldsymbol{z}}_i, t_i) = \frac{\hat{w}_{IPS}(\hat{\boldsymbol{z}}_i, t_i)}{\sum_{i=1}^{N}\hat{w}_{IPS}(\hat{\boldsymbol{z}}_i, t_i)}, \tag{4.17}$$

where $\hat{w}_{IPS}(\hat{\boldsymbol{z}}_i, t_i) = \frac{\pi^{t_i}(\boldsymbol{x}_i, \boldsymbol{A})}{\hat{P}(t=t_i|\hat{\boldsymbol{z}}_i)}$. The probability for instance i to receive the treatment t_i, $\hat{P}(t = t_i|\hat{\boldsymbol{z}}_i)$, is inferred by the logistic regression model.

4.4 Experimental Evaluation

In this section, we investigate whether network information among observational data can help improve counterfactual evaluation through extensive experiments.

4.4.1 Dataset Description

In real-world situations, only the factual outcome of each instance is observable. For example, we can observe the potential outcome y_i^1 of the i-th instance iff $t_i = 1$. As a result, it is extremely challenging to collect data with ground truth of counterfactual outcomes. Therefore, we follow (Veitch *et al.*, 2019; Guo *et al.*, 2020c) to synthesize the treatments and outcomes based on the observed features and network information which are extracted from two real-world datasets. Specifically, we introduce two

networked observational datasets for evaluating the utility of treatment assignment functions. Based on the observed features and the network structures, we introduce the data generating process which synthesizes treatments and outcomes. To reflect real-world situations, we consider hidden confounders and unknown edge weights. We fully cover the steps to reproduce the semi-synthetic datasets from the publicly available datasets, BlogCatalog and Flickr.

BlogCatalog (BC) is a social media website where users post blogs. Each instance is a blogger. Each edge presents the friendship between two bloggers. The features are the bag-of-words representation of the a blogger's keywords. Here, the task is to learn a treatment assignment function which determines to promote a blogger's article more on mobile devices or desktops such that users' opinion is optimized. We extend the original BC dataset (Li *et al.*, 2019b) by synthesizing (a) the outcomes – readers' opinions on bloggers; and (b) the treatments – readers' device preference. Similar to the News dataset (Johansson *et al.*, 2016; Schwab *et al.*, 2018) that are widely used in causal inference literature, the following assumptions are made: (1) Readers either read on mobile devices or desktops. We say a blogger get treated (controlled) if her blogs are more popular on mobile devices (desktops). (2) Readers prefer to read certain topics from mobile devices, others from desktops. (3) The latent confounders of a blogger are determined by her and her neighbors' topics. (4) The latent confounders of a blogger influence both readers' preference of devices (treatment) and readers' opinion (outcome). Based on these assumptions, we train a topic model on a large set of documents to synthesize treatments and outcomes. Then we define the centroid of each treatment group with topics: (i) we randomly sample a blogger and let her topic distribution be the centroid of the treated, denoted by $\bar{r}^1$; (ii) we let the centroid of the controlled, $\bar{r}^0$, be the average topic distribution of all bloggers. Then the treatments and outcomes are synthesized based on the similarity

between the topic distributions of bloggers and the two centroids. Let $r(\boldsymbol{x}_i)$ be the topic distribution of the i-th blogger's description, we model the readers' preference of browsing devices on the blogger's content:

$$P(t=1|\boldsymbol{x}_i, \tilde{\boldsymbol{A}}) = \frac{\exp(p_i^1)}{\exp(p_i^1)+\exp(p_i^0)}, \tag{4.18}$$

where $p_t^i = \kappa_1 r(\boldsymbol{x}_i)^T \bar{r}^t + \kappa_2 (\tilde{\boldsymbol{A}} r(\boldsymbol{x}_j))^T \bar{r}^t$.

where $t \in \{0,1\}$. For blogger i, the first term on RHS represents the confounding bias caused by the topics of herself. The second term on RHS signifies that caused by the topics of her neighbors. $\kappa_1 \geq 0$ and $\kappa_2 \geq 0$ control the strength of these two terms. When $\kappa_1 = \kappa_2 = 0$ the treatment assignment is random and the greater the value κ_1 and κ_2 are, the less the treatment assignment is, and therefore, the more significant the confounding bias is. We let $\tilde{\boldsymbol{A}}$ denote the normalized weighted adjacency matrix, where each entry $\tilde{\boldsymbol{A}}_{ij}$ denotes the importance of an edge with related to the influence on confounding bias. In social networks, the edge weights are unknown, so only the unweighted adjacency matrix $\boldsymbol{A}$ is observable in the data. However, the unobserved weighted adjacency matrix $\tilde{\boldsymbol{A}}$ is the one that influences the treatments and outcomes. Thus, an ideal causal inference approach needs to catch the weights of each edge. If $\boldsymbol{A}_{ij} = 1$, then we sample $\tilde{\boldsymbol{A}}_{ij} = \tilde{\boldsymbol{A}}_{ji} \sim \text{Uniform}(0.1, 1)$; otherwise, we set $\tilde{\boldsymbol{A}}_{ij} = \tilde{\boldsymbol{A}}_{ji} = 0$. Then the raw outcomes of the i-th blogger are simulated as:

$$y_i^{raw}(t) = (1-t)p_0^i + t p_1^i + \epsilon. \tag{4.19}$$

The noise ϵ is sampled from a zero-mean Gaussian distribution with standard deviation 0.01. Then we normalize the raw outcomes $y_i(t)$ with:

$$y_i(t) = \frac{y_i^{raw}(t) - \mu(\boldsymbol{y}^{raw})}{\sigma'(\boldsymbol{y}^{raw})}, \tag{4.20}$$

where $\mu(\boldsymbol{y}^{raw})$ and $\sigma'(\boldsymbol{y}^{raw})$ signify the mean and standard deviation of all raw outcomes. In this work, we set $\kappa_1 = 1$ and vary $\kappa_2 \in \{1, 2\}$. Meanwhile, 50 LDA topics

Table 4.1: Statistics of the datasets

Dataset	Instances	Edges	Features	κ_2	Treated Instances	Instances with $y_i^1 > y_i^0$
BC	5,196	173,468	8,189	1	2579.5 ± 29.891	1030.1 ± 331.31
				2	2448.6 ± 539.687	2031.1 ± 1149.696
Flickr	7,575	239,738	12,047	1	3700.8 ± 156.873	2708.3 ± 745.03
				2	3859.4 ± 218.072	3182.1 ± 588.958

are learned from the training corpus. Then we reduce the vocabulary by taking union of the most probable 100 words from each topic, which results in 2,173 bag-of-word features.

Flickr is an online community where users share images and videos. Each instance is a user and each edge is the friendship between two users. The features of each user are the tags of interest. We adopt the same settings and assumptions as we do for the BC datasets. Thus, we aim to evaluate treatment assignment functions that determine which device is more proper to promote a user's images. We learn 50 topics from the training corpus using LDA and concatenate the top 25 words of each topic which reduces the feature dimension to 1,210. We set the parameters the same as the BC dataset.

Table 4.1 presents the statistics of the two semi-synthetic datasets under various settings. The average and standard deviation of the number of treated instances and the number of instances that satisfy $y_i^1 > y_i^0$ are calculated over the 10 simulations under each setting of parameters. They vary because the true edge weights are randomly sampled from the uniform distribution Uniform$(0.1, 1)$.

4.4.2 Experimental Settings

We randomly split each dataset into training (60%), validation (20%) and test sets (20%) and report the results of the test sets for 10 runs on each simulation. Grid

search is applied to find optimal hyperparameters. The detailed setup of parameter study is as follows: learning rate is 10^{-3}, the number of epochs is 100, the number of GAT layer is 1 and the numbers of fully connected layers of function f^y and f^t are 1, the number of attention head is 2, the number of hidden units of each attention head and each fully connected layer are 128 and 16. For evaluation, we adapt those in (Bennett and Kallus, 2019; Zou *et al.*, 2019) to a class of the treatment assignment functions which take both observed features and network information as input. The treatment assignment functions with random weights are considered:

$$\pi_{rw}^t(\boldsymbol{x}_i, \boldsymbol{A}) = \frac{\exp(\boldsymbol{\psi}^{tT}\boldsymbol{x}_i + \frac{1}{|\mathcal{N}(i)|}\sum_{j\in\mathcal{N}(i)}\boldsymbol{\delta}^{tT}\boldsymbol{x}_j)}{\sum_t \exp(\boldsymbol{\psi}^{tT}\boldsymbol{x}_i + \frac{1}{|\mathcal{N}(i)|}\sum_{j\in\mathcal{N}(i)}\boldsymbol{\delta}^{tT}\boldsymbol{x}_j)}, \tag{4.21}$$

where $\mathcal{N}(i)$ is the set of neighbors of instance i in the network $\boldsymbol{A}$. The random weights are obtained as $\boldsymbol{\psi}^1, \boldsymbol{\delta}^1 \sim 2Bern(\mathbf{0.5}) - \mathbf{1}$ and $\boldsymbol{\psi}^0 = -\boldsymbol{\psi}^1, \boldsymbol{\delta}^0 = -\boldsymbol{\delta}^1$. For these treatment assignment functions, the ground truth utilities are obtained with Eq. (4.3).

To corroborate the effectiveness of CONE, it is evaluated against the state-of-the-art kernel based methods, neural network based methods, and classic methods:

Optimal Kernel Balancing (OKB) (Bennett and Kallus, 2019) is the state-of-the-art kernel based weighted estimator which minimizes an adversarial balance objective.

Inverse Propensity Scoring (IPS-X) (Bottou *et al.*, 2013) is a weighted estimator which fits a propensity scoring model using observed features. Specifically, a logistic regression model is trained with supervision of the observed treatments.

Self-normalized Inverse Propensity Scoring (SNIPS-X) (Swaminathan and Joachims, 2015b) is a variant of the weighted estimator IPS-X where self-normalized weights are employed.

The Direct Method (Qian and Murphy, 2011) estimates the utility of a treatment assignment function through the inference of counterfactual outcomes (Eq (4.4)). We

consider three models that can infer counterfactual outcomes: OLS1, OLS2 (Louizos *et al.*, 2017), and the simple direct method using a neural network model (DM-X) (Bennett and Kallus, 2019).

Doubly Robust Estimators (Dudík *et al.*, 2011) combine direct methods and the inverse propensity scoring (Eq. (4.7)). Here, we consider the combination of each aforementioned direct method (OLS1, OLS2 or DM-X) and the IPS-X method. We call them DR-OLS1, DR-OLS2, and DR-DM-X.

To the best of our knowledge, this is the first work utilizing network information for counterfactual evaluation. So, there is no baseline that naturally incorporates network information. We also tried to concatenate the adjacency matrix to the original features to allow baselines utilize the network information for a fair comparison. However, such an approach cannot improve the performance of baselines due to the high dimensionality and sparsity of the network information.

Then, we formally present the two evaluation metrics, root mean squared error (RMSE) and mean absolute error (MAE) as

$$\begin{aligned} RMSE &= \sqrt{\frac{1}{K}\sum_{k=1}^{K}(\hat{\tau}_k(\pi) - \tau_k(\pi))^2} \\ MAE &= \frac{1}{K}\sum_{k=1}^{K}|\hat{\tau}_k(\pi) - \tau_k(\pi)|, \end{aligned} \tag{4.22}$$

where K is the number of simulations.

4.4.3 Results

Effectiveness. Experimental results corroborate the effectiveness of the proposed framework. Table 4.2 shows the empirical results evaluated on the BC and Flickr datasets with $\kappa_1 = 1$ and $\kappa_2 \in \{1, 2\}$. The following observations are made from these experimental results: (1) The proposed framework CONE results in better per-

Table 4.2: Experimental results corroborating the effectiveness of CONE

	BlogCatalog				Flickr			
	$\kappa_2 = 1$		$\kappa_2 = 2$		$\kappa_2 = 1$		$\kappa_2 = 2$	
	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
CONE (ours)	**0.034**	**0.026**	**0.037**	**0.027**	**0.014**	**0.011**	**0.014**	**0.012**
OKB	0.141	0.135	0.150	0.143	0.073	0.063	0.093	0.083
IPS-X	0.042	0.039	0.089	0.074	0.018	0.016	0.030	0.027
SNIPS-X	0.042	0.038	0.089	0.074	0.018	0.017	0.029	0.027
DM-X	0.229	0.229	0.241	0.239	0.099	0.097	0.117	0.114
OLS1	0.302	0.301	0.347	0.346	0.144	0.143	0.168	0.167
OLS2	0.275	0.274	0.308	0.304	0.139	0.139	0.162	0.161
DR-DM-X	0.041	0.034	0.071	0.060	0.019	0.018	0.028	0.026
DR-OLS1	0.042	0.039	0.089	0.074	0.018	0.016	0.030	0.027
DR-OLS2	0.047	0.041	0.090	0.078	0.019	0.017	0.031	0.028

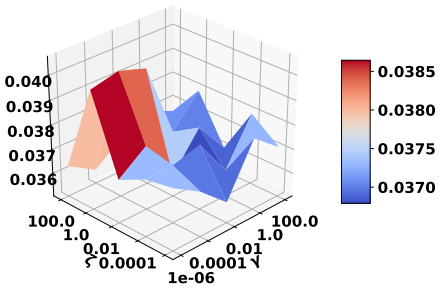
formance than the state-of-the-art baseline methods consistently on both datasets under various settings. One-tailed T-tests show that the results of CONE are significantly better with a significance level of 0.05. (2) Measured by the increase in both error metrics, the performance of CONE worsens less than other methods when the hidden confounding effect grows (from $\kappa_2 = 1$ to $\kappa_2 = 2$). This verifies that capturing the patterns of hidden confounders from network structures with the combined partial representations helps counterfactual evaluation of treatment assignment functions.

Parameter Study. Here, we investigate the influence of the hyperparameters γ and ζ on the performance of CONE. Note that γ controls the penalty on the predictions of observed treatments based on the related partial representation. And ζ determines to what extent the two partial representations agree with each other.

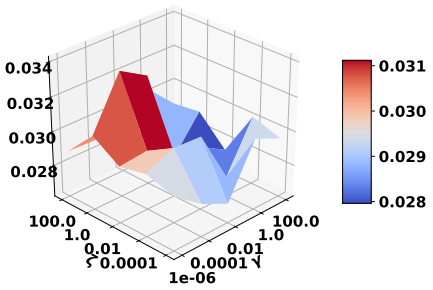
The detailed setup of the parameter is as follows. We search the learning rate in $\{10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}\}$, the number of fully connected layers of f^y, f^t and h in $\{1, 2, 3\}$, the number of hidden units in the GAT layers in $\{8, 16, 32\}$, the number of attention heads in $\{2, 4, 8\}$, γ and ζ in $\{10^{-6}, 10^{-4}, 10^{-2}, 1, 100\}$. We set γ and ζ in the range of $\{10^{-6}, 10^{-4}, 10^{-2}, 1, 100\}$. Fig. 4.2 presents the experimental results on the BC and Flickr datasets with $\kappa_1 = 1$ and $\kappa_2 = 2$. We can observe that when $\gamma \in [1, 100]$ and $\zeta \in [0.01, 1]$, CONE achieves the best performance on the BC dataset ($\kappa_2 = 2$). For the Flickr dataset ($\kappa_2 = 2$), CONE consistently performs well when $\gamma \in [10^{-6}, 1]$ and $\zeta \in [10^{-6}, 100]$. Similar results can be obtained with other settings. We can conclude that CONE maintains stable performance by varying the hyperparameters in a wide range, which is often desired in real-world applications.

4.5 Summary

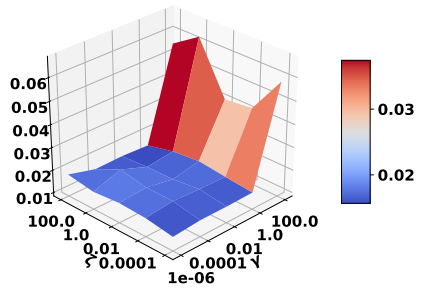
In this chapter, we study the problem of counterfactual evaluation in networked observational data. In particular, we investigate the hypothesis that utilizing network information will help handle hidden confounders in counterfactual evaluation. We propose a novel framework, CONE, which leverages the network information along with the observed features to mitigate hidden confounding effects for counterfactual evaluation. Empirical results from extensive experiments show the effectiveness of CONE and verify that incorporating network information indeed helps us control hidden confounders in the task of counterfactual evaluation. Related future work includes counterfactual evaluation and optimization of treatment assignment functions in various types of network data (e.g., dynamic networks (Sarkar *et al.*, 2019; Marin *et al.*, 2017; Shakarian *et al.*, 2015a)).



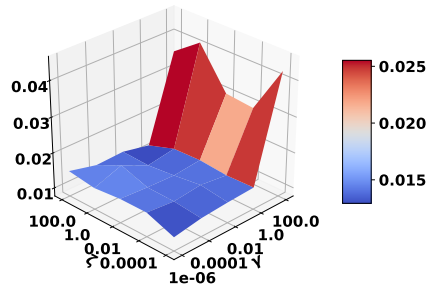
(a) RMSE of BC ($\kappa_2 = 2$)



(b) MAE of BC ($\kappa_2 = 2$)



(c) RMSE of Flickr ($\kappa_2 = 2$)



(d) MAE of Flickr ($\kappa_2 = 2$)

Figure 4.2: Parameter study results

Chapter 5

DEBIASING GRID-BASED SEARCH IN E-COMMERCE

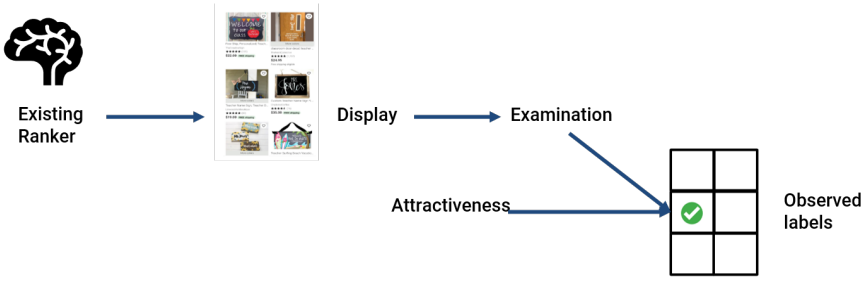


Figure 5.1: Causal diagram describing the data generating process of search log data in e-commerce.

The widespread usage of e-commerce websites in daily life and the resulting wealth of implicit feedback data form the foundation for systems that train and test e-commerce search ranking algorithms. While convenient to collect, implicit feedback data inherently suffers from various types of bias since user feedback is limited to products they are exposed to by existing search ranking algorithms and impacted by how the products are displayed. Fig. 5.1 shows the causal diagram of the data generating process of search log data for e-commerce. We can observe that there is a path through which the selection bias generated by the existing ranker is propagated to the observed labels. This is not desired as it makes the observed labels result from both the selection bias and the attractiveness of the item to the user, but our goal is to model attractiveness (user preference).

In the literature, a vast majority of existing methods have been proposed towards unbiased learning to rank for list-based web search scenarios. However, such methods cannot be directly adopted by e-commerce websites mainly for two reasons. First,

in e-commerce websites, search engine results pages (SERPs) are displayed in 2-dimensional grids. We can consider such grids as a network, where each position is a node and its first-hop neighbors are the four positions surrounding it. The existing methods have not considered the difference in user behavior (probability to examine a certain position) between list-based web search and grid-based product search. Second, there can be multiple types of labels (e.g., clicks and purchases) on e-commerce websites. We aim to utilize all types of implicit feedback as the supervision signals.

In this chapter, we extend the methodology of unbiased learning to rank to the problem of e-commerce search. In particular, we consider a grid-based product search scenario where a SERP is considered as a grid, a special type of network, and each position is considered as a node in the network. We propose a novel framework which (1) forms the theoretical foundations to allow multiple types of implicit feedback in unbiased learning to rank and (2) incorporates the *row skipping* and *slower decay* click models to capture unique user behavior patterns in grid-based search for inverse propensity scoring. Through extensive experiments on real-world e-commerce search log datasets across browsing devices and product taxonomies, we show that the proposed framework outperforms the state of the art unbiased learning to rank algorithms. These results also reveal important insights on how user behavior patterns vary in e-commerce SERPs across browsing devices and product taxonomies.

5.1 Problem Statement

In this section, we introduce the technical preliminaries and present the problem statement. We start with an introduction of the technical preliminaries. Then we introduce the settings of unbiased learning to rank in grid-based product search.

5.1.1 Technical Preliminaries

Generally, boldface uppercase letters (e.g., $\boldsymbol{X}$), boldface lowercase letters (e.g., $\boldsymbol{x}$) and normal lowercase (e.g., x) letters denote matrices, vectors and scalars, respectively. Let $\boldsymbol{x}_q^i$ denote the feature vector of the query-product pair in the i-th position and $\boldsymbol{X}_q$ signify the feature matrix of all query-product pairs in the SERP of the query q. $\bar{\boldsymbol{y}}_q$ signify the vector of product indexes in the search session corresponding to the query q in the observed search log data. $\boldsymbol{o}_q$ denotes the binary vector corresponding to whether a product in q is examined. For example, $o_q^i = 1$ (0) means the product ranked in the i-th position has been examined (not examined). $\boldsymbol{c}_q$ and $\boldsymbol{p}_q$ are the vectors of clicks and purchases of the products $\bar{\boldsymbol{y}}_q$ in the SERP of the query q. $c_q^i = 0, 1$ means the i-th product is not clicked and clicked. $p_q^i = 0, 1$ means the product is not purchased and purchased, respectively. Then the training set containing n queries and their search result sessions can be denoted by $\{\boldsymbol{X}_q, \bar{\boldsymbol{y}}_q, \boldsymbol{c}_q, \boldsymbol{p}_q\}_{q=1}^n$. We define a ranker as a function $f : \mathcal{X} \to \mathbb{R}$ mapping the features of a query-product pair to a real number standing for its ranking score.

5.1.2 Problem Statement

In this work, we focus on the offline setting where randomized experiments are not available. In contrast to (Wang *et al.*, 2016; Joachims *et al.*, 2017) where randomized experiments are performed, we can neither obtain user feedback to SERPs with randomized ranking nor ground truth of propensity scores. This requires us to estimate propensity scores along with train the ranker as in (Ai *et al.*, 2018; Hu *et al.*, 2019).

Definition 4. ***Unbiased Learning to Rank for Grid-based Product Search.*** *Given search log data* $\{\boldsymbol{X}_q, \bar{\boldsymbol{y}}_q, \boldsymbol{c}_q, \boldsymbol{p}_q\}_{q=1}^n$ *and the number of columns and rows of*

	Challenge	Treatment space	Treatment policy	Use of network information
This work	Selection bias	Large	Ranker	Propensity model
CONE	Confounding bias	Small	Node classifier	Latent confounders

Table 5.1: Comparison between unbiased learning to rank for grid-based product search and counterfactual evaluation in network data.

e-commerce SERPs, we aim to learn the propensity score model(s) which would be used to reweigh products for unbiased estimate of rankers' loss and train unbiased rankers with inverse propensity scoring to maximize e-commerce search metrics (e.g., purchase NDCG@K) on held-out test data.

Table 5.1 illustrates the connections and differences between the problem of unbiased learning to rank and that of counterfactual evaluation with networked observational data. Note that the use of network information in this work is simplified. In particular, we considering the position, i.e., the row and column index, of a certain item determines its probability of examination by users.

5.2 Inverse Propensity Scoring for Grid-based Product Search

In this section, we start with a brief introduction of background knowledge. Then we provide descriptions of the proposed framework including the loss function and the propensity score models. In particular, we propose propensity models based on data analysis results. A propensity model infers the probability to examine a certain item in the SERP based on the position of it, which reflects user behavior patterns in grid-based displayed SERPs.

5.2.1 Background

Cascade Click Models. Click models have been used to connect user behavior patterns (e.g., click rate) to the evaluation metrics of learning to rank algorithms (Moffat

and Zobel, 2008). The cascade model (Craswell *et al.*, 2008) is one of the most widely adopted click models which can quantify the probabilities of multiple types of users' behaviors (e.g., click, stop and examination) in list-based web search SERPs. In particular, let α describes the how likely users continue to browse the next product, then the probability that users stop and leave the search results page at position i can be formulated as $\beta(i) = (1 - \alpha) \prod_{j=0}^{i-1} \alpha$. In a series of randomized controlled trial (Craswell *et al.*, 2008), the cascade click model has been shown to outperform others in click prediction tasks.

Propensity Score Estimation from Observational Data. Generally, unbiased estimation of propensity scores requires randomized experiments (Joachims *et al.*, 2017; Wang *et al.*, 2018). However, randomized experiments can be expensive, time consuming and can hurt users' experience. In (Wang *et al.*, 2018; Ai *et al.*, 2018; Hu *et al.*, 2019), Expectation Maximization (EM) style optimization algorithms have been proposed to learn propensity models without randomized experiments. These methods are based on the intuition that the joint optimum of the ranker and the propensity model leads to unbiased estimates of propensity scores. But these algorithms can be trapped in local joint optimum. Based on the same intuition, in our proposed framework, we aim to find the joint optimum of the two models by minimizing the loss function through grid search on hyperparameters.

5.2.2 Pairwise Unbiased Learning to Rank for Multiple Types of Feedback

Joint Examination Hypothesis

The *examination hypothesis* is a widely adopted assumption in the literature of unbiased learning to rank (Joachims *et al.*, 2017; Ai *et al.*, 2018; Hu *et al.*, 2019), which postulates that a user clicks a document iff the document is examined and relevant.

Only considering click and the attractiveness of products (similar to relevance of documents), we can rewrite the straightforward counterpart of the original examination hypothesis in the context of e-commerce as:

$$P(c_q^i = 1|\boldsymbol{x}_q^i) = P(o_q^i = 1|\boldsymbol{x}_q^i)P(a_q^i = 1|\boldsymbol{x}_q^i), \tag{5.1}$$

where a_q^i is the binary variable representing attractiveness of the product at position i of the search results page of query q. We define attractiveness of a product as how attractive it appears in SERPs.

However, in the context of e-commerce search, we need to adapt this hypothesis such that we can take multiple types of user feedback into consideration. For simplicity, in this work, we only consider two types of feedback: clicks and purchases. Nevertheless, the proposed hypothesis as well as the other components of the proposed framework can be extended to account for more types of feedback (e.g., favorite and add-to-cart). To consider both clicks and purchases, we propose the *joint examination hypothesis*, a novel extension of the examination hypothesis, which is defined as:

Joint Examination Hypothesis. No matter if a user eventually does purchase or not purchase a product, she clicks a product iff the product is examined and attractive. The joint examination hypothesis can be formulated as:

$$P(p_q^i, c_q^i = 1|\boldsymbol{x}_q^i) = P(o_q^i = 1|\boldsymbol{x}_q^i)P(p_q^i, a_q^i = 1|\boldsymbol{x}_q^i) \tag{5.2}$$

In short, the joint examination hypothesis extends the examination hypothesis to the context of e-commerce where multiple types of feedback exist. We are aware of that the joint examination hypothesis is a stronger assumption than the original examination hypothesis as we can recover the original examination hypothesis (Eq. (5.1)) by marginalizing the joint examination hypothesis (Eq. (5.2)) over $P(p_q^i)$. Note that

this assumption can be relaxed when noisy clicks are taken into consideration, which is similar to that in (Joachims *et al.*, 2017). We do *not* model purchase as a function of attractiveness because we define attractiveness of a product as how attractive it appears in SERPs for a user to start engaging (i.e, click). It is natural to consider users' shopping journey as a two-stage process illustrated in (Wu *et al.*, 2018), where at first users search for a query and decide to click on a product displayed by SERPs when found it attractive. Then, the user makes purchase decision after examining the detail catalog on the product landing page.

Less Clicks for Less Attractive Products. We add a mild assumption

$$P(a_q^i = 0|\boldsymbol{x}_q^i) = \zeta P(c_q^i = 0|\boldsymbol{x}_q^i), \tag{5.3}$$

where we let $\zeta \in (0, 1]$ such that the assumption is coherent with Eq. (5.1). Intuitively, this means a less attractive product would receive less clicks.

The Loss Function

Let $\mathcal{I}_q$, $\mathcal{I}'_q$ and $\mathcal{I}''_q$ denote three types of pairs: *(click, no feedback), (purchase, no feedback) and (purchase, click)*, respectively. Then, the loss function of mis-ranking (as well as the gradients) can be reduced to an aggregation of losses defined over these three types of pairs. Note that the main task of e-commerce search engines is to maximize purchase or revenue of the website. But users would *unlikely* be able to make purchase decisions based on product images (and limited information) displayed on SERPs, instead, the product images shown on SERPs need to first attract them to click on products first, which then lead them to the product landing pages and help them to inform purchase decision after examining the product details. Therefore, in SERPs, we also aim to maximize the attractiveness of products shown in top positions such that purchase decisions can be triggered later after clicking. Based on this

intuition, we first formulate the loss function based on purchases and attractiveness by adopting the fashion of pairwise ranking algorithms and then propose an unbiased estimate of it using implicit feedback data as:

$$\begin{aligned}\mathcal{L} = &\int \mathbb{1}(p_q^i = 0)\ L\ dP(\boldsymbol{x}_i, a_q^i = 1, \boldsymbol{x}_j, a_q^j = 0) \\ &+ A \int \mathbb{1}(p_q^i = 1)\ L\ dP(\boldsymbol{x}_i, a_q^i = 1, \boldsymbol{x}_j, a_q^j = 0) \\ &+ B \int L'\ dP(\boldsymbol{x}_i, p_q^i = 1, a_q^i = 1, \boldsymbol{x}_j, p_q^j = 0, a_q^j = 1),\end{aligned} \tag{5.4}$$

where the function $L = L(\boldsymbol{x}_i, a_q^i, \boldsymbol{x}_j, a_q^j)$ denotes the pairwise loss penalizing mis-ranking of *(click, no feedback)* or *(purchase, no feedback)* pairs. Similarly, the function $L' = L'(\boldsymbol{x}_i, a_q^i, p_q^i, \boldsymbol{x}_j, a_q^j, p_q^j)$ signifies the penalty for mis-ranking on *(purchase, click)* pairs. Note that the parameterization of the functions L and L' can be flexible. The details of how the loss functions L and L' are defined and optimized can be found in Section 5.3. Note that under Assumption Eq. (5.2), both click and purchase imply attractiveness. $\mathbb{1}(\cdot)$ is the indicator function. The hyperparameters $A, B \geq 0$ control the trade-off of penalizing the mis-ranking (purchase, no feedback) and (purchase, click) with respect to the pairs on (click, no feedback). Therefore, the loss of (purchase, no feedback) and (purchase, click) pairs are multiplied with A and B, respectively.

Unbiased Estimate of the Loss Function

However, we are not capable to evaluate this loss function (Eq. (5.4)) with implicit feedback data because the ground truth of attractiveness cannot be observed. Alternatively, we aim to infer the attractiveness through the observed user feedback including clicks and purchases. This can be done by replacing the loss functions and probabilities relevant to attractiveness with the counterparts of user feedback with

the following assumptions:

$$L(\boldsymbol{x}_q^i, \mathrm{a}_\mathrm{q}^\mathrm{i}, \boldsymbol{x}_q^j, \mathrm{a}_\mathrm{q}^\mathrm{j}) = L(\boldsymbol{x}_q^i, \mathrm{c}_\mathrm{q}^\mathrm{i}, \boldsymbol{x}_q^j, \mathrm{c}_\mathrm{q}^\mathrm{j}) \tag{5.5}$$

$$L'(\boldsymbol{x}_q^i, \mathrm{a}_\mathrm{q}^\mathrm{i}, p_q^i, \boldsymbol{x}_q^j, \mathrm{a}_\mathrm{q}^\mathrm{j}, p_q^i) = L'(\boldsymbol{x}_q^i, \mathrm{c}_\mathrm{q}^\mathrm{i}, p_q^i, \boldsymbol{x}_q^j, \mathrm{c}_\mathrm{q}^\mathrm{j}, p_q^j) \tag{5.6}$$

$$L(\boldsymbol{x}_q^i, \mathrm{c}_\mathrm{q}^\mathrm{i}, \boldsymbol{x}_q^j, \mathrm{c}_\mathrm{q}^\mathrm{j}) \neq 0 \;\; iff \;\; \mathrm{c}_\mathrm{q}^\mathrm{i} \neq \mathrm{c}_\mathrm{q}^\mathrm{j}. \tag{5.7}$$

$$L'(\boldsymbol{x}_q^i, \mathrm{c}_\mathrm{q}^\mathrm{i}, p_q^i, \boldsymbol{x}_q^j, \mathrm{c}_\mathrm{q}^\mathrm{j}, p_q^j) \neq 0 \;\; iff \;\; (c_q^i = c_q^j = 1) \cap (p_q^i \neq p_q^j). \tag{5.8}$$

We propose a loss function $\mathcal{L}_{imp}$ that can be evaluated on implicit feedback data. The subscript *imp* means implicit feedback. With inverse propensity scoring, we show below in Theorem 1 that the new loss function $\mathcal{L}_{imp}$ is an unbiased estimate of the original loss. In particular, the proposed loss can be formulated as:

$$\begin{aligned}
\mathcal{L}_{imp} = & \int \mathbb{1}(p_q^i = 0)\; L\; \frac{dP(\boldsymbol{x}_q^i, \boldsymbol{c}_q^i = 1, \boldsymbol{x}_q^j, \boldsymbol{c}_q^j = 0)}{P(o_q^i = 1 | \boldsymbol{x}_q^i)} \\
& + A' \int \mathbb{1}(p_q^i = 1)\; L\; \frac{dP(\boldsymbol{x}_q^i, \boldsymbol{c}_q^i = 1, \boldsymbol{x}_q^j, \boldsymbol{c}_q^j = 0)}{P(o_q^i = 1 | \boldsymbol{x}_q^i)} \\
& + B' \int L' \frac{dP(\boldsymbol{x}_q^i, \boldsymbol{c}_q^i = 1, \boldsymbol{x}_q^j, \boldsymbol{c}_q^j = 1)}{P(o_q^i = 1 | \boldsymbol{x}_q^i) P(o_q^j = 1 | \boldsymbol{x}_q^j)},
\end{aligned} \tag{5.9}$$

where $A' = \zeta A$ and $B' = B$.

Theorem 1. *With the assumptions in Eq.* (5.2) *and Eq.* (5.5)-(5.8), $\mathcal{L}_{imp}$ *is an unbiased estimate of the original loss function* $\mathcal{L}$.

Then, we present the proof for Theorem 1 based on the assumptions made earlier in this Section including Eq. (5.2), Eq. (5.3) and Eq. (5.5)-(5.8). Note that although it is a proof, but it directly guides the design of the loss function of the proposed framework. Therefore, we believe it is closely related to the reproducibility of this work. Here, we present the proof for Theorem 3.1 based on the assumptions made earlier including Eq. (5.2) and Eq. (5.5)-(5.8).

Proof. First, we show the first (second) term of $\mathcal{L}_{imp}$ is equivalent to the first (second) term of $\mathcal{L}$. Essentially, the goal is to replace the infeasible variables, a_q^i and a_q^j, in the

original loss (Eq. (5.4) with the feasible implicit feedback.

$$\begin{aligned}
&\int L(a_q^i, a_q^j) dP(\boldsymbol{x}_i, a_q^i = 1, \boldsymbol{x}_j, a_q^j = 0) \\
&= \int L(a_q^i, a_q^j) dP(a_q^i = 1|\boldsymbol{x}_i) dP(a_q^j = 0|\boldsymbol{x}_j) P(\boldsymbol{x}_q^i, \boldsymbol{x}_q^j) d\boldsymbol{x}_q^i d\boldsymbol{x}_q^j \\
&= \zeta \int L(c_q^i, c_q^j) \frac{dP(c_q^i = 1|\boldsymbol{x}_q^i)}{P(o_q^i = 1|\boldsymbol{x}_q^i)} dP(c_q^j = 0|\boldsymbol{x}_q^j) P(\boldsymbol{x}_q^i, \boldsymbol{x}_q^j) d\boldsymbol{x}_q^i d\boldsymbol{x}_q^j
\end{aligned} \tag{5.10}$$

The second equality comes from the assumption of Less Clicks for Less Attractive Products.

We then show that the last term on the RHS of Eq. (5.4) is equivalent to the last term in RHS of Eq. (5.9).

$$\begin{aligned}
&\int L' \frac{dP(\boldsymbol{x}_q^i, p_q^i = 1, c_q^i = 1, \boldsymbol{x}_q^j, p_q^j = 0, c_q^j = 1)}{P(o_q^i = 1|\boldsymbol{x}_q^i) P(o_q^j = 1|\boldsymbol{x}_q^j)} \\
= &\int L' P(p_q^i = 1|\boldsymbol{x}_q^i, c_q^i = 1) \frac{P(c_q^i = 1|\boldsymbol{x}_q^i)}{P(o_q^i = 1|\boldsymbol{x}_q^i)} \\
&\times P(p_q^j = 0|\boldsymbol{x}_q^j, c_q^j = 1) \frac{P(c_q^j = 1|\boldsymbol{x}_q^j)}{P(o_q^j = 1|\boldsymbol{x}_q^j)} P(\boldsymbol{x}_q^i, \boldsymbol{x}_q^j) d\boldsymbol{x}_q^i d\boldsymbol{x}_q^j.
\end{aligned} \tag{5.11}$$

This only requires us to show $P(p_q^i|\boldsymbol{x}_q^i, c_q^i = 1) = P(p_q^i|\boldsymbol{x}_q^i, a_q^i = 1)$, which can be proved as follows for both $p_q^i = 0$ and $p_q^i = 1$:

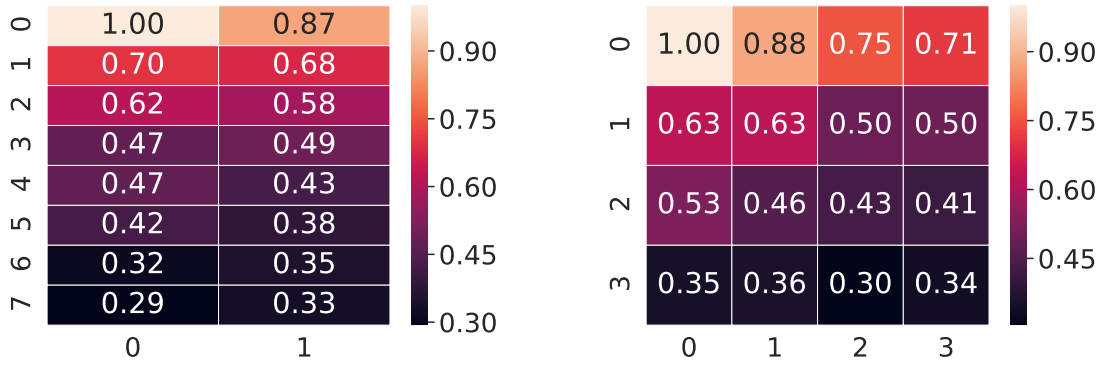
$$\begin{aligned}
P(p_q^i|\boldsymbol{x}_q^i, c_q^i = 1) &= \frac{P(p_q^i, c_q^i = 1|\boldsymbol{x}_q^i)}{P(c_q^i = 1|\boldsymbol{x}_q^i)} = \frac{P(p_q^i, c_q^i = 1|\boldsymbol{x}_q^i)}{P(a_q^i = 1|\boldsymbol{x}_q^i) P(o_q^i = 1|\boldsymbol{x}_q^i)} \\
&= \frac{P(p_q^i, a_q^i = 1|\boldsymbol{x}_q^i) P(o_q^i = 1|\boldsymbol{x}_q^i)}{P(a_q^i = 1|\boldsymbol{x}_q^i) P(o_q^i = 1|\boldsymbol{x}_q^i)} = P(p_q^i|\boldsymbol{x}_q^i, a_q^i = 1)
\end{aligned} \tag{5.12}$$

where the second equality is from the original examination hypothesis (Eq. (5.1)) which can be recovered from the joint examination hypothesis (Eq. (5.2)). The third equality is directly from the joint examination hypothesis (Eq. (5.2)).

With what mentioned above, we let $A' = \zeta A$, $B' = B$ and the proof is complete.

□

With Theorem 1, we now know that the proposed loss function (Eq. (5.9)) provides unbiased estimate of the original loss function (Eq. (5.4)) given biased implicit



(a) NCTR on mobile devices

(b) NCTR on desktops

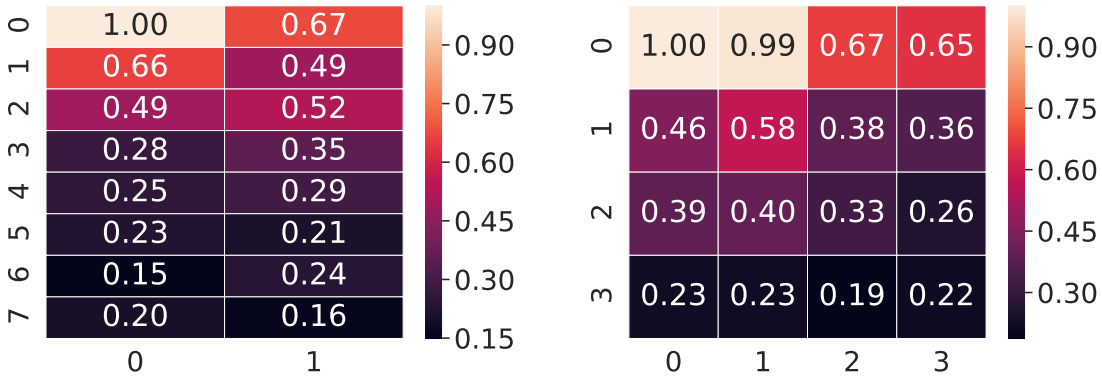
Figure 5.2: Normalized click through rate (NCTR) in the top 16 positions of the H&L dataset.

feedback data. Following existing work (Joachims *et al.*, 2017; Wang *et al.*, 2018), we simplify the problem with the following assumption: The probability of examination only depends on the position, which can be formulated as $P(o_q^i = 1|\boldsymbol{x}_q^i) = P(o^i)$. As the focus of this work is to handle multiple types of user feedback and incorporate the unique user behavior patterns in grid-based product search for unbiased learning to rank, we leave modeling position bias with richer information (e.g., query-product features) as future work.

5.2.3 Propensity Score Models for Grid-based Product Search

We Here, we motivate to use two click models as propensity models by data analysis results which verify that they can capture unique user behavior patterns in grid-based product search. Then descriptions of the two propensity models are given below.

In the literature (Xie *et al.*, 2019), variants of the cascade click model (Craswell *et al.*, 2008) have been proposed to capture the unique patterns of users' behaviors



(a) NPR on mobile devices

(b) NPR on desktops

Figure 5.3: Normalized purchase rate (NPR) in the top 16 positions of the H&L dataset.

in grid-based search. These models can provide consistent probabilities of users' behaviors (e.g., examination, continuing to browse the next product and skipping a row) in such context. In eye tracking experiments of (Xie *et al.*, 2019), three unique phenomena have been observed in grid-based search with images: row skipping, slower decay and middle bias. In this work, we propose to utilize the click models capturing the *row skipping* and *slower decay* phenomena as propensity score models for unbiased learning to rank. We also provide reasons why middle bias is not considered in this work through data analysis below. To motivate the usage of the two propensity models, we show a series of data analysis results on real-world e-commerce search log data here while the detail description of data and experiment are further explained in Section 5.4. Limited by space, we only show the results on the *Home and Living* datasets, similar observations are also made on the *Paper and Party Supplies* datasets (see Section 5.4.1 for dataset description). In Fig. 5.2-5.3, we show the normalized click through rate (NCTR) and purchase rate (NPR) of the top 16 positions for data collected from both mobile devices and desktops. These NCTR and NPR are the click

through rate and purchase rate of each position divided by those of the first position. Different from the previous work (Xie *et al.*, 2019) which focused on the development of novel evaluation metric for learning to rank, our target is to capture users' behavior patterns in grid-based product search for more accurate and interpretable propensity score modeling.

The *middle bias* model (Xie *et al.*, 2019) is not considered in this work for two reasons: (1). The number of columns in the SERPs of our data is small. Specifically, the SERPs show products in 2 columns for mobile devices and 4 columns for desktops. (2). Further evident in our empirical analysis (Fig. 5.2-5.3), we also do not observe the middle bias phenomenon. In particular, the NCTR and NPR of products in the middle for desktops (4-column display) are not significantly higher than those of the other products.

Row Skipping. In our datasets, similar to (Xie *et al.*, 2019), we observe the row skipping phenomena where users can skip some rows before they click, purchase or leave SERPs. As shown in Fig. 5.2-5.3, we can see that the click through rate and purchase rate are not monotonically decreasing from top to the bottom. For example, the last position of Fig. 5.2(a) has higher NCTR than the forth last position. Based on this observation, let $r(i)$ be the row number of the i-th product. We use the *row skipping* cascade model as a propensity score model to quantify $P(o^i)$:

$$P(o^i = 1) = \prod_{k=0}^{r(i)-1} \Big\{(1-\gamma) \prod_{j=S(k)}^{S(k)+N(k)-1} \alpha + \gamma\Big\} \prod_{j=S(r(i))}^{i-1} \alpha$$

γ models the trend to skip a row. $S(k)$ and $N(k)$ are the number of items before and in the k-th row. Intuitively, in the row skipping cascade model, if a user reached position i, she must have gone through the k-th row before the row of position i ($k < r(i)$). There are two possible situations: she either skipped the k-th row with the row skipping probability γ or decided to continue browsing on every single position

on that row with probability $\prod_{j=S(k)}^{S(k)+N(k)-1} \alpha$.

Slower Decay. Similar to what has been discovered by previous study (Xie *et al.*, 2019), in grid-based product search, the decay of users' attention from top to bottom in each SERP is slower than that in list-based web search. In Fig. 5.2(a) and 5.2(b), we can observe that the NCTRs on mobile devices and desktop take 10 positions to drop to 43% and 46% of the NCTR of the first positions, which is much slower than the drop of attention in list-based web search shown in Fig. 3 of (Xie *et al.*, 2019). We can specify the probability of examination at position i as:

$$P(o^i = 1) = \prod_{j=0}^{i-1} \min(\beta^{row(j)}\alpha, 1.0), \tag{5.13}$$

where $\beta \geq 1$ models the increased patience of users in grid-based product search compared to that in the original cascade model. When $\beta = 1.0$, $P(o^i = 1)$ of the slower decay model is the same as that in the cascade model.

Besides the these models, we encourage practitioners to design models of $P(o^i)$ based on a combination of domain knowledge and propensity scores estimated from online experiments.

5.3 Optimization

Without randomized experiments, we aim to achieve a joint optimum of both the propensity score models and the ranker with the implicit feedback data. Due to the simplicity of the propensity models, we consider parameters of the propensity models (α, γ, and β) as hyperparameters and adopt grid search along with minimizing the loss function $\mathcal{L}_{imp}$ to reach the joint optimum. Different from the existing ones (Hu *et al.*, 2019; Joachims *et al.*, 2017; Ai *et al.*, 2018; Wang *et al.*, 2018), the proposed propensity model leverages the user behavior patterns in grid-based product search.

Given propensity scores ($P(o^i)$) computed from either the row skipping or the

slower decay model based on hyperparameters α, γ and β, we aim to learn a ranker f based on the unbiased loss function $\mathcal{L}_{imp}$. In particular, we adopt LambdaMART (Wu *et al.*, 2010) where the ranker is the gradient boosting trees (GBDT) or MART (Friedman, 2001). In LambdaMART, instead of using a explicit loss function, we directly define the gradients of an implicit loss function, which are known as lambda gradients (Burges, 2010). Toward unbiased learning to rank, similar to (Hu *et al.*, 2019), we directly apply inverse propensity scoring to the lambda gradients. In addition, in e-commerce search, we need to consider multiple types of user feedback. We also assign different weights, A' and B', to the gradient components corresponding to trade-off in mis-ranking loss among three types of pairs. Therefore, we propose an extension of the original lambda gradient (Burges, 2010). In particular, the lambda gradient of the k-th product (λ_k) can be written as:

$$\begin{aligned}\frac{\partial \mathcal{L}_{imp}}{\partial f(\boldsymbol{x}_k)} = \lambda_k = \sum_q \left(\sum_{\overline{y}_q^i = k \cap (i,j) \in \mathcal{I}_q} \frac{\lambda_{ij}}{P(o^i)} - \sum_{\overline{y}_q^i = k \cap (j,i) \in \mathcal{I}_q} \frac{\lambda_{ij}}{P(o^j)} \right) \\ + A' \sum_q \left(\sum_{\overline{y}_q^i = k \cap (i,j) \in \mathcal{I}'_q} \frac{\lambda_{ij}}{P(o^i)} - \sum_{\overline{y}_q^i = k \cap (j,i) \in \mathcal{I}'_q} \frac{\lambda_{ij}}{P(o^j)} \right) \\ + B' \sum_q \left(\sum_{\overline{y}_q^i = k \cap (i,j) \in \mathcal{I}''_q} \frac{\lambda_{ij}}{P(o^i)P(o^j)} - \sum_{\overline{y}_q^i = k \cap (j,i) \in \mathcal{I}''_q} \frac{\lambda_{ij}}{P(o^i)P(o^j)} \right),\end{aligned}$$

where $\overline{y}_q^i = k$ means product k is at the i-th position in the SERP of query q. In addition, λ_{ij} is defined as:

$$\lambda_{ij} = \frac{-2}{1 + \exp(2(f(\boldsymbol{x}_q^i) - f(\boldsymbol{x}_q^j)))} |\Delta_{ij}|, \tag{5.14}$$

where $|\Delta_{ij}|$ denotes the absolute value of difference in a predefined metric (e.g., NDCG@K) if the ranking of item i and j are swapped. Note that product price is not directly involved in the lambda gradient to prevent bias towards expensive products.

5.4 Experiment

In this section, we start with data description followed by experimental settings. Then, through extensive experimental results, we aim to answer the following key research questions: (1). How effective is the proposed framework compared to the baselines in the task of reranking products in grid-based search? (2). How does user behavior patterns in grid-based product search vary across browsing devices and product taxonomies?

5.4.1 Dataset Description

Here, we provide a brief description of the product search log data used in experimental studies of this work. In addition to an introduction and a summary of statistics of the dataset, we also include details of the feature engineering procedure. Datasets are collected from the e-commerce website at Etsy, which is an international online marketplace for small businesses selling vintages, hand-crafted products and supplies. In particular, we pick two of the most popular product taxonomies: *Paper and Party Supplies* (PPS) and *Home and Living* (H&L). For understanding the difference in users' behaviors when they are browsing the search sessions via different devices, search logs from both desktop and mobile devices are collected. Therefore, we obtain 4 datasets: Desktop PPS, Mobile PPS, Desktop H&L and Mobile H&L. For each dataset, we only include the search result sessions with at least a click of those queries which triggered at least a click or a purchase. The statistics for these four datasets are then shown in Table 5.2. The number of features per dataset may vary because we remove the columns with incomplete values due to missing information in the data. For example, some of new sellers may not be familiar with the platform and therefore might forget to provide tags for some products.

Table 5.2: Data Statistics

Dataset	Sessions	Products	Clicks	Purchases	Features
Desktop PPS	15,360	734,289	19,241	1,913	213
Mobile PPS	12,777	611,304	14,861	1,446	215
Desktop H&L	24,905	1,184,454	29,446	2,436	195
Mobile H&L	24,208	1,148,804	26,851	2,287	195

Feature Engineering. The search log datasets are preprocessed to fit the format of $(\boldsymbol{X}_q, \overline{\boldsymbol{y}}_q, \boldsymbol{c}_q, \boldsymbol{p}_q)$ using the feature engineering tool Buzzsaw (Stanton *et al.*, 2018). We summarize the features into the following four categories based on which subject they are related to: product, shop, query and interaction. In terms of how the features are computed, similar to (Haldar *et al.*, 2019), we consider features including raw features (e.g., content similarity matching between query and product, product or shop attributes such as price, title, materials, shipping time), ratio statistics (e.g., domestic sales ratio, the ratio of a product's contribution to a shop's sale), mean values over time windows (e.g., average CTR, purchase rate of the product or shop in search results in last x days) and composition features (e.g., the difference between product price and average clicked price for the query). Further descriptions of example features can be found in Table 5.3. Note that, in this work, price is one of the features in the product category. Given the fact the price of a product can change (e.g., discount), in each SERP, we use the exact price of the product at the time when the query is performed. Price can also influence the probability to examine a product, we leave the investigation on this direction to future work.

Table 5.3: Feature Description

Feature Category	Examples
Product	Average historical rates of the product in last x days
	Price of the product
	Average processing and shipping time after purchases
Shop	Average rating of the products from the shop
	Decile of the shop's sale
	Top categories sold in the shop
Query	Average price of clicked products from the query
	Logarithm of purchase count from the query over x days
	Top buyer taxonomy purchased for the query
Interaction	BM25 of product's listing title and tags with query
	Ratio of a product's contribution to a shop's sale
	Difference in query average purchase price and product price

5.4.2 Experimental Settings

Here, we report the experimental settings. For unbiased learning to rank algorithms, in the offline settings, the most commonly adopted way of evaluation is via the task of reranking the products in SERPs of a hold-out test set (Ai *et al.*, 2018; Hu *et al.*, 2019). We randomly split the search sessions of each dataset into training (70%), validation (10%) and test sets (20%). We set $A = B = 50$ in accordance to the approximated ratio between clicks and purchases in our data after a global smoothing. We perform grid search to find optimal hyperparameter settings for the propensity score models. We search α in $\{0.8, 0.825, ..., 0.975\}$, β in $\{1.05, 1.1, 1.15, 1.2\}$ and γ in $\{0.8, 0.825, ..., 0.975\}$ to keep $P(o^i = 1)$ in a reasonable range. Algorithms that can achieve a global optimal w.r.t. parameters of both the ranker and the propensity model can also be used to obtain values of α, β and γ. For the LambdaMART ranker

of the proposed framework, we search the number of leaves in $\{31, 127, 511\}$ for each tree and the number of trees in $\{100, 200, ..., 1,000\}$. Other parameters are adopted from the default setting of unbiased LambdaMART, while similar settings are also used for the baselines.

Baselines. We consider 6 baseline methods, including *four* classic learning to rank algorithms and *two* state-of-the-art unbiased learning to rank algorithms that can work without propensity scores estimated by randomized experiments. Because the implementation of Regression EM (Wang *et al.*, 2018) is not available and empirical results in (Hu *et al.*, 2019) also show that Unbiased LambdaMART outperform Regression EM, it is valid to skip Regression EM as a baseline in this work. Similar to the proposed model, we also enable every single baseline to handle multiple types of user feedback, by aggregating the loss function across different types with importance weights, i.e., 50:1 ratio between purchases and clicks. By doing so when compare performance, we can eliminate the influence of utilizing multiple types of feedback and safely claim the differences are caused by (1) the proposed propensity score estimation models and (2) the underlying learning to rank models. The baselines are:

MART (M) (Wu et al.*, 2010)* is a gradient boosting algorithm leveraging multiple additive regression trees as weak learners. It minimizes pairwise loss functions (e.g., cross entropy loss).

RankBoost (RB) (Freund et al.*, 2003)* is a pairwise algorithm based on AdaBoost, which minimizes cross entropy loss.

LambdaMART (LM) (Wu et al.*, 2010)* is an extension of MART which reweights each pair to optimize listwise ranking measures (e.g., NDCG@K).

Random Forests (RF) (Breiman, 2001) is a variant of the classic machine learning algorithm which minimizes cross entropy loss.

Unbiased LambdaMART (ULM) (Hu et al.*, 2019)* is a variant of LambdaMART

where each pair is reweighted by the product of their inverse propensity scores. Two propensity scores are estimated for each position along with the ranker: one for products that are clicked and purchased, and the other one for products with no feedback.

Dual Learning (DL) (Ai et al.*, 2018)* performs joint optimization of two models. The first model is a neural network trained to maximize a listwise ranking measure. The second model is a neural network learned to optimize the likelihood of examinations on clicked products.

Evaluation Metrics. We then describe the evaluation metrics. In this work, we perform experiments in the offline setting. In particular, we evaluate the proposed framework and the baselines on the organic search logs obtained from the e-commerce website on Etsy. In organic search (non-sponsored search), the target is to maximize purchase and revenue of e-commerce websites, therefore, we adopt the three widely used metrics purchase NDCG@K, revenue NDCG@K and purchase mean average precision (MAP) (Wu *et al.*, 2018):

$$NDCG_{pur}@K = \frac{1}{IDCG_{pur}@K} \sum_{i=1}^{K} \frac{2^{p_q^i} - 1}{\log_2(i+1)}$$

$$NDCG_{rev}@K = \frac{1}{IDCG_{rev}@K} \sum_{i=1}^{K} \left(\frac{2^{p_q^i} - 1}{\log_2(i+1)} price_q^i \right),$$

$$MAP_{pur}@K = \frac{1}{K} \sum_{i=1}^{K} |\{j | p_q^j = 1, j = 1, ..., i\}|/i,$$

where $IDCG_{pur}@K$ and $IDCG_{rev}@K$ are the normalizers. Revenue NDCG@K is a variant of purchase NDCG@K by weighting the gain of each product with price. To consider slow decay of user attention, we set $K = 1, 2, 5, 10, 20$ for the NDCGs and $K = 20$ for MAP.

In the offline setting, we are not able to perform randomized experiments to obtain ground truth of propensity scores. Therefore, unlike the previous work relying

on simulated propensity scores and relevance labels (Ai *et al.*, 2018; Hu *et al.*, 2019), we could not obtain the attractiveness of products that received no feedback. To the best of our ability, we apply these evaluation metrics on hold-out test sets of search logs. This may not be the theoretically optimal strategy and we understand that there can exist attractive products which comes without user feedback. However, because of the unavailability of ground truth of attractiveness of products, we leave handling the attractive products without user feedback as a future work.

5.4.3 Experimental Results

Effectiveness. Here, we report the experimental results to show (1) how effective the proposed framework is in terms of improving e-commerce search results and (2) how users behavior patterns vary across different browsing devices and taxonomies. We show the results in Table 5.4 and make the following observations:

- At least one of the two proposed methods outperforms the baselines in almost all of the cases. This demonstrates the effectiveness of our proposed unbiased ranker, which is able to capture unique user behavior patterns in grid-based product search with these two simple propensity models.

- The proposed framework shows superior performance to unbiased LambdaMART. This corroborates the efficacy of the proposed propensity score models. This is because unbiased LambdaMART relies on a different pairwise inverse propensity scoring strategy but shares the same underlying ranker (LambdaMART). This observation can be attributed to incorporating prior knowledge of users' behavior patterns to guide the learning process of propensity score models.

- Row skipping performs better in the H&L datasets, this can be caused by the fact that users have more specific intent when they browse SERPs in this

Table 5.4: Results on the four datasets. Best results are highlighted in boldface.

	$NDCG_{pur}$					$NDCG_{rev}$					MAP_{pur}
	@1	@2	@5	@10	@20	@1	@2	@5	@10	@20	@20
Desktop PPS (Paper and Party Supplies)											
M	0.082	0.121	0.181	0.232	0.291	0.078	0.126	0.184	0.234	0.289	0.079
RB	0.087	0.117	0.184	0.243	0.303	0.087	0.110	0.182	0.241	0.303	0.084
LM	0.101	0.128	0.194	0.249	0.305	0.100	0.130	0.194	0.248	0.308	0.097
RF	0.096	0.128	0.192	0.239	0.295	0.088	0.117	0.185	0.233	0.287	0.096
ULM	0.109	0.142	0.201	0.251	0.308	0.109	0.142	0.201	0.250	0.307	0.106
DL	0.098	0.136	0.211	0.277	0.327	0.098	0.136	0.211	0.277	0.327	0.094
RS	0.111	0.141	0.196	0.256	0.312	0.110	0.141	0.196	0.256	0.312	0.106
SD	**0.144**	**0.173**	**0.232**	**0.281**	**0.340**	**0.143**	**0.173**	**0.232**	**0.281**	**0.339**	**0.139**
Mobile PPS (Paper and Party Supplies)											
M	0.154	0.197	0.236	0.294	0.347	0.148	0.184	0.227	0.289	0.343	0.154
RB	0.067	0.116	0.181	0.232	0.286	0.085	0.135	0.201	0.252	0.300	0.067
LM	0.111	0.148	0.216	0.262	0.322	0.119	0.159	0.225	0.272	0.335	0.111
RF	0.138	0.177	0.232	0.286	0.339	0.131	0.176	0.244	0.298	0.343	0.136
ULM	0.151	0.192	0.254	0.293	0.345	0.150	0.192	0.253	0.292	0.344	0.149
DL	0.102	0.144	0.235	0.291	0.340	0.100	0.143	0.235	0.290	0.339	0.101
RS	0.148	0.182	0.243	0.298	0.351	0.164	0.203	0.265	0.318	0.370	0.155
Slower Decay	**0.166**	**0.208**	**0.281**	**0.321**	**0.371**	**0.176**	**0.223**	**0.293**	**0.332**	**0.383**	**0.165**
Desktop H&L (Home and Living)											
M	0.114	0.152	0.212	0.265	0.318	0.119	0.151	0.215	0.265	0.319	0.116
RB	0.085	0.127	0.193	0.240	0.297	0.096	0.131	0.194	0.239	0.297	0.070
LM	0.107	0.135	0.210	0.266	0.323	0.109	0.138	0.213	0.263	0.321	0.101
RF	0.109	0.164	0.228	0.275	0.325	0.102	0.145	0.212	0.260	0.307	0.112
ULM	0.148	0.184	0.243	0.284	0.340	0.148	0.185	0.243	0.284	0.340	0.145
DL	0.097	0.138	0.211	0.266	0.322	0.097	0.138	0.211	0.266	0.322	0.096
RS	**0.165**	**0.199**	**0.252**	**0.300**	**0.354**	**0.165**	**0.200**	**0.252**	**0.301**	**0.354**	**0.163**
SD	0.141	0.182	0.242	0.290	0.347	0.139	0.182	0.242	0.290	0.346	0.135
Mobile H&L (Home and Living)											
M	0.147	0.200	0.261	0.306	0.350	0.159	0.216	0.274	0.322	0.369	0.147
RB	0.084	0.117	0.169	0.227	0.291	0.094	0.131	0.181	0.234	0.295	0.083
LM	0.119	0.155	0.23	0.281	0.322	0.124	0.161	0.239	0.29	0.33	0.117
RF	0.125	0.187	0.250	0.296	0.341	0.137	0.194	0.263	0.307	0.354	0.123
ULM	**0.181**	**0.237**	0.285	0.322	0.367	**0.181**	**0.237**	0.285	0.322	0.367	**0.180**
DL	0.116	0.154	0.221	0.288	0.331	0.116	0.154	0.221	0.288	0.331	0.114
RS	0.172	0.222	**0.287**	0.324	0.372	0.172	0.222	**0.287**	0.324	0.371	0.173
SD	**0.181**	0.233	0.282	**0.329**	**0.377**	**0.181**	0.233	0.282	**0.329**	**0.377**	**0.180**

taxonomy, which means they would more likely to skip rows of products that do not look attractive. In addition, the price of products in this taxonomy has larger variance, users may skip those rows showing expensive products.

- On the mobile datasets, the performance of the best baseline, i.e. unbiased LambdaMART, is closer to the proposed framework than that on desktop

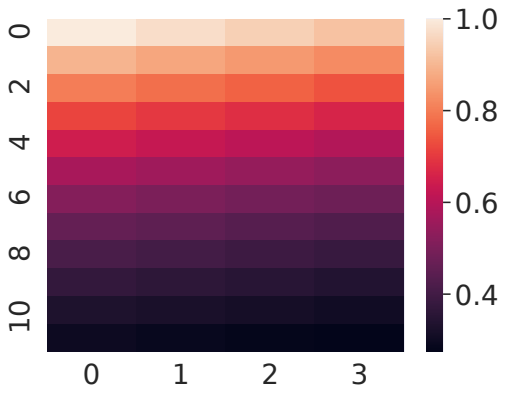
datasets. This is because list-based web search is a better proxy for mobile devices with products displayed in 2 columns as comparing to those on desktops (4 columns).

We train separate models for different taxonomies to show that modeling different user behavior patterns across product taxonomies can be beneficial. In practical deployments, a single ranker is often trained and tested across all taxonomies. The model with highest purchases or revenue across taxonomies in randomized online experiments may be preferred in such a case.

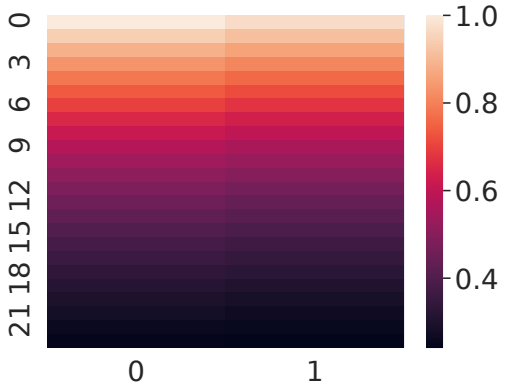
Propensities. Next, we report the values of propensity scores and the hyperparameters α, β and γ that achieve the optimal performance for the proposed framework. For Desktop and Mobile PPS, the slower decay models with $\alpha = 0.95, \beta = 1.1$ and $\alpha = 0.925, \beta = 1.15$ outperform others. For Desktop and Mobile H&L, the row skipping model with $\alpha = 0.95, \gamma = 0.975$ and the slower decay model with $\alpha = 0.925, \beta = 1.1$. We show propensity scores estimated for the H&L datasets in Fig. 5.4 to draw connection with the earlier empirical results (Fig. 5.2-5.3). Although we cannot perfectly reconstruct the non-monotonically decreasing patterns in Fig. 5.2-5.3, in Fig. 5.4(a), we can observe that positions at the left bottom can have higher estimated $P(o^i)$ than some positions. We can regard these propensity scores as upper bounds of the NCTR values observed in Fig. 5.2. This is because the NCTR values result from a combination of position bias (propensity scores) and effectiveness of the ranking algorithm(s) that generated the search logs. It can also be observed that users are more patient when they browse with desktops.

5.5 Summary

In this chapter, we study the novel problem unbiased learning to rank algorithms in grid-based product search for e-commerce. It is the first step toward handling the



(a) desktop H&L



(b) mobile H&L

Figure 5.4: Propensity scores obtained through grid search that achieve the optimal performance.

special challenges in this problem. In particular, the proposed framework utilizes multiple types of feedback and leverages users' behavior patterns in grid-based product search for propensity score modeling. We prove that the proposed loss function evaluated on implicit feedback data provides unbiased estimate of the ideal loss. We then motivate the usage of the row skipping and slower decay models for inverse propensity scoring justified through empirical evidence from data analysis. Finally, extensive ex-

perimental results show the effectiveness of the proposed framework across browsing devices and product taxonomies in datasets collected from a real-world e-commerce website.

Chapter 6

CONCLUSION AND FUTURE WORK

This chapter summaries the key contributions of this dissertation prospectus and present a brief discussion on promising future research directions.

6.1 Conclusion

Side information such as network structures that showing relationships among instances naturally appear in many types of observational data. Such data not only create new chances and but also pose new challenges for learning causality with observational data. This dissertation is dedicated to the development of principled frameworks that explore novel applications of such topological information toward a solution to mitigating confounding and selection bias in both causal inference and machine learning tasks. The main thrusts of the presented work are summarized as follows:

- **Learning Individual-level Causal Effects with Networked Observational Data.** The first part of the dissertation prospectus focuses on developing a novel framework for estimating individual-level causal effects with networked observational data. The research efforts mainly explore how to effectively utilize network information to compensate for the unobserved confounders toward unbiased treatment effect estimation. The proposed framework – network deconfounder (Chapter 3) learns representations of hidden confounders from a combination of observed features and network information through graph convolutional layers. The representations are trained to infer observed outcomes in order to achieve accurate inference in the space of treatment effects. At

the same time, these representations are balanced between the treatment group and the control group using a Wasserstein-1 distance based penalty term for mitigating confounding bias.

- **Counterfactual Evaluation of Novel Treatment Assignment Functions with Networked Observational Data.** The second part of this dissertation prospectus develops a novel framework for evaluating novel treatment assignment functions with networked observational data. The main focus of this research is to answer the question: How do we learn good representation of latent confounders to accurately evaluate treatment assignment functions with networked observational data? In the proposed framework, CONE (Chapter 4), the partial representations of latent confounders are learned by predicting the observed outcomes and treatments separately. A neural mutual information estimator (Belghazi *et al.*, 2018) is applied in the training process to ensure the two partial representations agree with each other. This design is based on the definition of confounders (Pearl, 2009) – the variables causally influence both treatment assignments and observed outcomes. Then the latent confounder representations can be used in a series of counterfactual evaluation tools such as the IPS estimator (Kitagawa and Tetenov, 2018) and the doubly robust estimators (Dudík *et al.*, 2011) to accomplish the task.

- **Debiasing Grid-based Search in E-commerce.** The third part of the dissertation formulates and investigates a novel problem of unbiased learning to rank algorithms in the context of grid-based product search for e-commerce. In this problem, we consider the display of items in a 2-dimensional SERP. In particular, a SERP can be considered as a network connecting neighboring items. In the proposed methodology, we simplify this network information to

only consider the position of each item – the row and column index of an item in SERPs. In addition, the proposed framework is developed in a way such that it can utilize multiple types of feedback and leverage prior knowledge of special users' behavior patterns in grid-based product search for propensity score modeling. We propose an unbiased estimator of the true loss function with two types of implicit feedback. We then perform data analysis of NCTR and NPR to justify the usage of the row skipping and slower decay models as the propensity models. Finally, extensive experiments are performed. Their results supports our claim of the effectiveness of the proposed framework across two browsing devices and two product taxonomies in real-world datasets collected from the e-commerce website Etsy.

6.2 Future Work

Causal Inference under Interference. A series of existing methods for causal inference with networked observational data are based on the assumption that there does not exist interference (Guo *et al.*, 2020c,b; Veitch *et al.*, 2019). However, in real-world applications, one's decision may influence others' outcomes. For example, in the context of e-commerce, service providers such as Etsy and Amazon and sellers can only perform interventions (e.g., displaying ads and discount) on items. However, the interesting outcomes are buyers' decisions (Doudchenko *et al.*, 2020), which are causally influenced by interventions on the items. In such situations, the fundamental assumption used to identify causal effects, i.e., the Stable Unit Treatment Value Assumption (SUTVA), does not hold. This results in new challenges of causal identification. Current solutions of interference can be categorized into two classes: (1) redefining units (Holtz *et al.*, 2020) and (2) generalized propensity score (GPS) (Doudchenko *et al.*, 2020). Redefining units relies on the assumption that

we can find units (subpopulations) where there is little interference among different redefined units. This approach often overwhelmingly relies on heuristics to find units satisfying this assumptions. GPS methods rely on the unconfoundedness assumption conditioning on propensity scores, which may not lead to robust estimates when propensity score models are misspecified.

Therefore, one future research direction of great potential is to relax the assumptions needed to the problem of interference with the help of machine learning models. Alternatively, one can work on the improvement of traditional methods with advanced machine learning models. For example, one can use graph neural networks to learn representations of nodes, and then redefine units as clusters in the representation space to work around the challenge of interference (Holtz *et al.*, 2020). In many real-world cases, the outcomes vary with time (Cheng *et al.*, 2021; Ma *et al.*, 2021). To handle these temporal variations, one can extend quasi-experiments such as Difference in Difference and Synthetic Control to the situation of interference for causal identification. Such ideas can also be extended to resolve causal inference problems in various downstream applications such as understanding user intent in fake news spreading (Cheng *et al.*, 2020) and cyberbullying (Cheng *et al.*, 2019).

Fairness of Treatment Assignment Functions. Due to the nature of observational data, we only have the access to one of the potential outcomes. Various existing work (Bennett and Kallus, 2019; Guo *et al.*, 2020b; Zou *et al.*, 2019) has been done toward solving the challenge in evaluating the utility of novel treatment assignment functions. In fact, this special characteristic of observational data also leads to an interesting open problems relevant to treatment assignment functions. One of them is how to assess the fairness of a novel treatment assignment function with observational data. This problem is of vital importance to study when we rely on data-driven approaches to make high-stakes decisions that can impact individuals' critical outcomes

such as health status and school/job admissions. Due to the lack of counterfactual outcomes, some fairness metrics designed for regular machine learning models such as Equalized Odds (Zafar *et al.*, 2017) may not be directly identifiable from observational data. It remains an open problem to connect existing counterfactual evaluation and optimization methods to measuring fairness of novel treatment assignment functions with observational data. One research direction is to perform a comprehensive study of the identifiability of fairness notions. First, we can consider group-level notions such as Demographic Parity, Equality of Opportunity and Predictive Quality Parity (Du *et al.*, 2020). Furthermore, one can also study the identification and estimation problem of individual-level notions such as individual fairness (Mukherjee *et al.*, 2020) and counterfactual fairness (Kusner *et al.*, 2017). A recent work for out-of-domain generalization, i.e., Invariant Risk Minimization (IRM) (Arjovsky *et al.*, 2019; Guo *et al.*, 2021), has been shown to be effective in optimizing a series of group-level fairness metrics (Adragna *et al.*, 2020). IRM essentially imposes the conditional independence between prediction and sensitive groups given learned representations. Based on this observation, another potential research direction is to learn fair features to mitigate unfairness in observational data and treatment assignment policies.

Debiasing Interactive Machine Learning. Besides unbiased learning to rank in e-commerce, there are many interactive machine learning problems where labels are generated by interactions between users and an existing machine learing algorithm. For example, recommendation systems (Chen *et al.*, 2020) and interactive NLP systems (Sokolov *et al.*, 2016).

We can extend the idea of unbiased learning to rank in e-commerce to more realistic scenarios. Given other information shown in the SERPs, what causally influences users' examination behaviors would also include meta information from SERPs including prices, product photos, and product ratings etc. Therefore, the first di-

rection is to explore propensity models with meta information. It is also possible to consider personalized propensity models where user attributes are taken as input of the propensity model. Second, for optimizing long-term marketplace-level objective metrics such as the number of active users, it is important to ensure the fairness and diversity among sellers. For example, novel inductive bias such as Determinantal Point Process based regularization or exploration algorithms such as UCB and Thompson sampling can be developed to stimulate rankers to explore novel products that have never been shown at top of the SERPs. Development of other methods to consider products with low or no feedback in evaluation metrics are also a potential research direction.

REFERENCES

Adragna, R., E. Creager, D. Madras and R. Zemel, "Fairness and robustness in invariant learning: A case study in toxicity classification", arXiv preprint arXiv:2011.06485 (2020).

Agarwal, A., I. Zaitsev and T. Joachims, "Counterfactual learning-to-rank for additive metrics and deep models", arXiv preprint arXiv:1805.00065 (2018).

Ai, Q., K. Bi, C. Luo, J. Guo and W. B. Croft, "Unbiased learning to rank with unbiased propensity estimation", in "The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval", pp. 385–394 (ACM, 2018).

Ai, Q., Y. Zhang, K. Bi, X. Chen and W. B. Croft, "Learning a hierarchical embedding model for personalized product search", in "Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval", pp. 645–654 (ACM, 2017).

Arbour, D., D. Garant and D. Jensen, "Inferring network effects from observational data", in "KDD", pp. 715–724 (ACM, 2016).

Arjovsky, M., L. Bottou, I. Gulrajani and D. Lopez-Paz, "Invariant risk minimization", arXiv preprint arXiv:1907.02893 (2019).

Athey, S. and S. Wager, "Efficient policy learning", arXiv preprint arXiv:1702.02896 (2017).

Bang, H. and J. M. Robins, "Doubly robust estimation in missing data and causal inference models", Biometrics **61**, 4, 962–973 (2005).

Bareinboim, E. and J. Tian, "Recovering causal effects from selection bias", in "Proceedings of the AAAI Conference on Artificial Intelligence", vol. 29 (2015).

Bareinboim, E., J. Tian and J. Pearl, "Recovering from selection bias in causal and statistical inference", in "Proceedings of the AAAI Conference on Artificial Intelligence", vol. 28 (2014).

Belghazi, M. I., A. Baratin, S. Rajeshwar, S. Ozair, Y. Bengio, D. Hjelm and A. Courville, "Mutual information neural estimation", in "ICML", pp. 530–539 (2018).

Bennett, A. and N. Kallus, "Policy evaluation with latent confounders via optimal balance", arXiv preprint arXiv:1908.01920 (2019).

Beygelzimer, A., S. Dasgupta and J. Langford, "Importance weighted active learning", arXiv preprint arXiv:0812.4952 (2008).

Blei, D. M., A. Y. Ng and M. I. Jordan, "Latent dirichlet allocation", Journal of machine Learning research **3**, Jan, 993–1022 (2003).

Bottou, L., J. Peters, J. Quiñonero-Candela, D. X. Charles, D. M. Chickering, E. Portugaly, D. Ray, P. Simard and E. Snelson, "Counterfactual reasoning and learning systems: The example of computational advertising", JMLR **14**, 1, 3207–3260 (2013).

Breiman, L., "Random forests", Machine learning **45**, 1, 5–32 (2001).

Bruna, J., W. Zaremba, A. Szlam and Y. LeCun, "Spectral networks and locally connected networks on graphs", arXiv preprint arXiv:1312.6203 (2013).

Burges, C. J., "From ranknet to lambdarank to lambdamart: An overview", Learning **11**, 23-581, 81 (2010).

Cao, Z., T. Qin, T.-Y. Liu, M.-F. Tsai and H. Li, "Learning to rank: from pairwise approach to listwise approach", in "Proceedings of the 24th international conference on Machine learning", pp. 129–136 (ACM, 2007).

Cartwright, N. *et al.*, "Nature's capacities and their measurement", OUP Catalogue (1994).

Chen, J., H. Dong, X. Wang, F. Feng, M. Wang and X. He, "Bias and debias in recommender system: A survey and future directions", arXiv preprint arXiv:2010.03240 (2020).

Cheng, L., R. Guo and H. Liu, "Long-term effect estimation with surrogate representation", in "Proceedings of the 14th ACM International Conference on Web Search and Data Mining", pp. 274–282 (2021).

Cheng, L., R. Guo, K. Shu and H. Liu, "Towards causal understanding of fake news dissemination", arXiv preprint arXiv:2010.10580 (2020).

Cheng, L., R. Guo, Y. Silva, D. Hall and H. Liu, "Hierarchical attention networks for cyberbullying detection on the instagram social network", in "Proceedings of the 2019 SIAM international conference on data mining", pp. 235–243 (SIAM, 2019).

Chernozhukov, V., D. Chetverikov, M. Demirer, E. Duflo, C. Hansen, W. Newey and J. Robins, "Double/debiased machine learning for treatment and structural parameters", (2018).

Cook, T. D., D. T. Campbell and W. Shadish, *Experimental and quasi-experimental designs for generalized causal inference* (Houghton Mifflin Boston, 2002).

Craswell, N., O. Zoeter, M. Taylor and B. Ramsey, "An experimental comparison of click position-bias models", in "Proceedings of the 2008 international conference on web search and data mining", pp. 87–94 (ACM, 2008).

Cuturi, M. and A. Doucet, "Fast computation of wasserstein barycenters", in "International Conference on Machine Learning", pp. 685–693 (2014).

Defferrard, M., X. Bresson and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering", in "Advances in neural information processing systems", pp. 3844–3852 (2016).

Dehejia, R. H. and S. Wahba, "Causal effects in nonexperimental studies: Reevaluating the evaluation of training programs", Journal of the American statistical Association **94**, 448, 1053–1062 (1999).

Derr, T., Y. Ma and J. Tang, "Signed graph convolutional networks", in "2018 IEEE International Conference on Data Mining (ICDM)", pp. 929–934 (IEEE, 2018).

Ding, K., J. Li, R. Bhanushali and H. Liu, "Deep anomaly detection on attributed networks", in "Proceedings of the 2019 SIAM International Conference on Data Mining", pp. 594–602 (SIAM, 2019).

Donsker, M. D. and S. S. Varadhan, "Asymptotic evaluation of certain markov process expectations for large time. iv", Communications on Pure and Applied Mathematics **36**, 2, 183–212 (1983).

Doudchenko, N., M. Zhang, E. Drynkin, E. Airoldi, V. Mirrokni and J. Pouget-Abadie, "Causal inference with bipartite designs", arXiv preprint arXiv:2010.02108 (2020).

Du, M., F. Yang, N. Zou and X. Hu, "Fairness in deep learning: A computational perspective", IEEE Intelligent Systems (2020).

Dudík, M., J. Langford and L. Li, "Doubly robust policy evaluation and learning", in "ICML", pp. 1097–1104 (Omnipress, 2011).

Freund, Y., R. Iyer, R. E. Schapire and Y. Singer, "An efficient boosting algorithm for combining preferences", Journal of machine learning research **4**, Nov, 933–969 (2003).

Friedman, J. H., "Greedy function approximation: a gradient boosting machine", Annals of statistics pp. 1189–1232 (2001).

Glorot, X., A. Bordes and Y. Bengio, "Deep sparse rectifier neural networks", in "Proceedings of the fourteenth international conference on artificial intelligence and statistics", pp. 315–323 (2011).

Goswami, A., P. Mohapatra and C. Zhai, "Quantifying and visualizing the demand and supply gap from e-commerce search data using topic models", in "Companion Proceedings of The 2019 World Wide Web Conference", pp. 348–353 (ACM, 2019).

Goswami, A., C. Zhai and P. Mohapatra, "Towards optimization of e-commerce search and discovery", in "The 2018 SIGIR Workshop On eCommerce", (2018).

Gulrajani, I., F. Ahmed, M. Arjovsky, V. Dumoulin and A. C. Courville, "Improved training of wasserstein gans", in "NeurIPS", pp. 5767–5777 (2017).

Guo, R., L. Cheng, J. Li, P. R. Hahn and H. Liu, "A survey of learning causality with data: Problems and methods", ACM Computing Surveys (CSUR) **53**, 4, 1–37 (2020a).

Guo, R., J. Li and H. Liu, "Counterfactual evaluation of treatment assignment functions with networked observational data", in "Proceedings of the 2020 SIAM International Conference on Data Mining", pp. 271–279 (SIAM, 2020b).

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

Guo, R., P. Zhang, H. Liu and E. Kiciman, "Out-of-distribution prediction with invariant risk minimization: The limitation and an effective fix", arXiv preprint arXiv:2101.07732 (2021).

Haldar, M., M. Abdool, P. Ramanathan, T. Xu, S. Yang, H. Duan, Q. Zhang, N. Barrow-Williams, B. C. Turnbull, B. M. Collins *et al.*, "Applying deep learning to airbnb search", in "Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining", pp. 1927–1935 (ACM, 2019).

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

Hirano, K., G. W. Imbens and G. Ridder, "Efficient estimation of average treatment effects using the estimated propensity score", Econometrica **71**, 4, 1161–1189 (2003).

Holtz, D., R. Lobel, I. Liskovich and S. Aral, "Reducing interference bias in online marketplace pricing experiments", arXiv preprint arXiv:2004.12489 (2020).

Hu, Z., Y. Wang, Q. Peng and H. Li, "Unbiased lambdamart: An unbiased pairwise learning-to-rank algorithm", in "The World Wide Web Conference", pp. 2830–2836 (ACM, 2019).

Joachims, T., "Optimizing search engines using clickthrough data", in "Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining", pp. 133–142 (ACM, 2002).

Joachims, T., A. Swaminathan and T. Schnabel, "Unbiased learning-to-rank with biased feedback", in "Proceedings of the Tenth ACM International Conference on Web Search and Data Mining", pp. 781–789 (2017).

Johansson, F., U. Shalit and D. Sontag, "Learning representations for counterfactual inference", in "ICML", pp. 3020–3029 (2016).

Kallus, N., "Balanced policy evaluation and learning", in "NeurIPS", pp. 8895–8906 (2018).

Kallus, N. and A. Zhou, "Confounding-robust policy improvement", arXiv preprint arXiv:1805.08593 (2018).

Karmaker Santu, S. K., P. Sondhi and C. Zhai, "On application of learning to rank for e-commerce search", in "Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval", pp. 475–484 (ACM, 2017).

Kingma, D. P. and J. Ba, "Adam: A method for stochastic optimization", arXiv preprint arXiv:1412.6980 (2014).

Kipf, T. N. and M. Welling, "Semi-supervised classification with graph convolutional networks", arXiv preprint arXiv:1609.02907 (2016).

Kitagawa, T. and A. Tetenov, "Who should be treated? empirical welfare maximization methods for treatment choice", Econometrica **86**, 2, 591 (2018).

Kuroki, M. and J. Pearl, "Measurement bias and effect restoration in causal inference", Biometrika **101**, 2, 423–437 (2014).

Kusner, M. J., J. R. Loftus, C. Russell and R. Silva, "Counterfactual fairness", arXiv preprint arXiv:1703.06856 (2017).

LaLonde, R. J., "Evaluating the econometric evaluations of training programs with experimental data", The American economic review pp. 604–620 (1986).

Li, J., R. Guo, C. Liu and H. Liu, "Adaptive unsupervised feature selection on attributed networks", in "Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery and data mining", pp. 92–100 (ACM, 2019a).

Li, J., X. Hu, J. Tang and H. Liu, "Unsupervised streaming feature selection in social media", in "Proceedings of the 24th ACM International on Conference on Information and Knowledge Management", pp. 1041–1050 (ACM, 2015).

Li, J., L. Wu, R. Guo, C. Liu and H. Liu, "Multi-level network embedding with boosted low-rank matrix approximation", in "ASONAM", pp. 49–56 (2019b).

Louizos, C., U. Shalit, J. M. Mooij, D. Sontag, R. Zemel and M. Welling, "Causal effect inference with deep latent-variable models", in "NeurIPS", pp. 6446–6456 (2017).

Ma, J., R. Guo, C. Chen, A. Zhang and J. Li, "Deconfounding with networked observational data in a dynamic environment", in "Proceedings of the 14th ACM International Conference on Web Search and Data Mining", WSDM '21, p. 166–174 (Association for Computing Machinery, New York, NY, USA, 2021), URL `https://doi.org/10.1145/3437963.3441818`.

Makar, M., A. Swaminathan and E. Kıcıman, "A distillation approach to data efficient individual treatment effect estimation", in "Proceedings of the AAAI Conference on Artificial Intelligence", vol. 33, pp. 4544–4551 (2019).

Marcus, G. and E. Davis, *Rebooting AI: building artificial intelligence we can trust* (Pantheon, 2019).

Marin, E., R. Guo and P. Shakarian, "Temporal analysis of influence to predict users' adoption in online social networks", in "SBP", pp. 254–261 (Springer, 2017).

Miao, W., Z. Geng and E. J. Tchetgen Tchetgen, "Identifying causal effects with proxy variables of an unmeasured confounder", Biometrika **105**, 4, 987–993 (2018).

Moffat, A. and J. Zobel, "Rank-biased precision for measurement of retrieval effectiveness", ACM Transactions on Information Systems (TOIS) **27**, 1, 2 (2008).

Monti, F., K. Otness and M. M. Bronstein, "Motifnet: a motif-based graph convolutional network for directed graphs", in "2018 IEEE Data Science Workshop (DSW)", pp. 225–228 (IEEE, 2018).

Mukherjee, D., M. Yurochkin, M. Banerjee and Y. Sun, "Two simple ways to learn individual fairness metrics from data", in "International Conference on Machine Learning", pp. 7097–7107 (PMLR, 2020).

Nabi, R., J. Pfeiffer, M. A. Bayir, D. Charles and E. Kıcıman, "Causal inference in the presence of interference in sponsored search advertising", arXiv preprint arXiv:2010.07458 (2020).

Pearl, J., "Causal inference in statistics: An overview", Statistics surveys **3**, 96–146 (2009).

Pearl, J., "On measurement bias in causal inference", arXiv preprint arXiv:1203.3504 (2012).

Pearl, J. and D. Mackenzie, *The book of why: the new science of cause and effect* (Basic Books, 2018).

Peters, J., D. Janzing and B. Schölkopf, *Elements of causal inference: foundations and learning algorithms* (MIT press, 2017).

Qian, M. and S. A. Murphy, "Performance guarantees for individualized treatment rules", Annals of statistics **39**, 2, 1180 (2011).

Rakesh, V., R. Guo, R. Moraffah, N. Agarwal and H. Liu, "Linked causal variational autoencoder for inferring paired spillover effects", in "CIKM", pp. 1679–1682 (2018).

Robins, J. M., A. Rotnitzky and L. P. Zhao, "Estimation of regression coefficients when some regressors are not always observed", Journal of the American statistical Association **89**, 427, 846–866 (1994).

Rubin, D. B., "Bayesian inference for causal effects: The role of randomization", The Annals of statistics pp. 34–58 (1978).

Rubin, D. B., "Causal inference using potential outcomes: Design, modeling, decisions", JASA **100**, 469, 322–331 (2005).

Saint-Jacques, G., M. Varshney, J. Simpson and Y. Xu, "Using ego-clusters to measure network effects at linkedin", arXiv preprint arXiv:1903.08755 (2019).

Sarkar, S., R. Guo and P. Shakarian, "Using network motifs to characterize temporal network evolution leading to diffusion inhibition", SNAM **9**, 1, 14 (2019).

Schlichtkrull, M., T. N. Kipf, P. Bloem, R. Van Den Berg, I. Titov and M. Welling, "Modeling relational data with graph convolutional networks", in "European Semantic Web Conference", pp. 593–607 (Springer, 2018).

Schnabel, T., A. Swaminathan, A. Singh, N. Chandak and T. Joachims, "Recommendations as treatments: Debiasing learning and evaluation", arXiv preprint arXiv:1602.05352 (2016).

Schölkopf, B., F. Locatello, S. Bauer, N. R. Ke, N. Kalchbrenner, A. Goyal and Y. Bengio, "Toward causal representation learning", Proceedings of the IEEE (2021).

Schwab, P., L. Linhardt, S. Bauer, J. M. Buhmann and W. Karlen, "Learning counterfactual representations for estimating individual dose-response curves", arXiv preprint arXiv:1902.00981 (2019).

Schwab, P., L. Linhardt and W. Karlen, "Perfect match: A simple method for learning representations for counterfactual inference with neural networks", arXiv preprint arXiv:1810.00656 (2018).

Shakarian, P., A. Bhatnagar, A. Aleali, E. Shaabani and R. Guo, "Evolutionary graph theory", in "Diffusion in Social Networks", pp. 75–91 (Springer, 2015a).

Shakarian, P., A. Bhatnagar, A. Aleali, E. Shaabani, R. Guo *et al.*, *Diffusion in social networks* (Springer, 2015b).

Shalit, U., F. D. Johansson and D. Sontag, "Estimating individual treatment effect: generalization bounds and algorithms", in "Proceedings of the 34th International Conference on Machine Learning-Volume 70", pp. 3076–3085 (JMLR. org, 2017).

Shalizi, C. R. and E. McFowland III, "Estimating causal peer influence in homophilous social networks by inferring latent locations", arXiv preprint arXiv:1607.06565 (2016).

Shalizi, C. R. and A. C. Thomas, "Homophily and contagion are generically confounded in observational social network studies", Sociological methods & research **40**, 2, 211–239 (2011).

Sokolov, A., J. Kreutzer, C. Lo and S. Riezler, "Learning structured predictors from bandit feedback for interactive nlp", in "Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)", pp. 1610–1620 (2016).

Sorokina, D. and E. Cantu-Paz, "Amazon search: The joy of ranking products", in "Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval", pp. 459–460 (ACM, 2016).

Stanton, A., L. Hong and M. Rajashekhar, "Buzzsaw: A system for high speed feature engineering", in "Proceedings of the 1st SysML conference", (2018).

Swaminathan, A. and T. Joachims, "Counterfactual risk minimization: Learning from logged bandit feedback", in "ICML", pp. 814–823 (2015a).

Swaminathan, A. and T. Joachims, "The self-normalized estimator for counterfactual learning", in "NeurIPS", pp. 3231–3239 (2015b).

Toulis, P., A. Volfovsky and E. M. Airoldi, "Propensity score methodology in the presence of network entanglement between treatments", arXiv preprint arXiv:1801.07310 (2018).

van der Maaten, L. and G. Hinton, "Visualizing data using t-SNE", Journal of Machine Learning Research **9**, 2579–2605 (2008).

Van Gysel, C., M. de Rijke and E. Kanoulas, "Learning latent vector spaces for product search", in "Proceedings of the 25th ACM International on Conference on Information and Knowledge Management", pp. 165–174 (ACM, 2016).

Veitch, V., Y. Wang and D. M. Blei, "Using embeddings to correct for unobserved confounding", arXiv preprint arXiv:1902.04114 (2019).

Veličković, P., G. Cucurull, A. Casanova, A. Romero, P. Liò and Y. Bengio, "Graph Attention Networks", ICLR (2018).

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

Wang, X., M. Bendersky, D. Metzler and M. Najork, "Learning to rank with selection bias in personal search", in "Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval", pp. 115–124 (ACM, 2016).

Wang, X., N. Golbandi, M. Bendersky, D. Metzler and M. Najork, "Position bias estimation for unbiased learning to rank in personal search", in "Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining", pp. 610–618 (ACM, 2018).

Wang, X., X. He, M. Wang, F. Feng and T.-S. Chua, "Neural graph collaborative filtering", arXiv preprint arXiv:1905.08108 (2019).

Wang, Y. and D. M. Blei, "The blessings of multiple causes", arXiv preprint arXiv:1805.06826 (2018).

Wu, L., D. Hu, L. Hong and H. Liu, "Turning clicks into purchases: Revenue optimization for product search in e-commerce", in "The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval", pp. 365–374 (ACM, 2018).

Wu, Q., C. J. Burges, K. M. Svore and J. Gao, "Adapting boosting for information retrieval measures", Information Retrieval **13**, 3, 254–270 (2010).

Xia, F., T.-Y. Liu, J. Wang, W. Zhang and H. Li, "Listwise approach to learning to rank: theory and algorithm", in "Proceedings of the 25th international conference on Machine learning", pp. 1192–1199 (ACM, 2008).

Xie, X., J. Mao, Y. Liu, M. de Rijke, Y. Shao, Z. Ye, M. Zhang and S. Ma, "Grid-based evaluation metrics for web image search", (2019).

Xu, B., N. Wang, T. Chen and M. Li, "Empirical evaluation of rectified activations in convolutional network", arXiv preprint arXiv:1505.00853 (2015).

Yao, L., S. Li, Y. Li, M. Huai, J. Gao and A. Zhang, "Representation learning for treatment effect estimation from observational data", in "NeurIPS", pp. 2634–2644 (2018).

Ying, R., R. He, K. Chen, P. Eksombatchai, W. L. Hamilton and J. Leskovec, "Graph convolutional neural networks for web-scale recommender systems", in "Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining", pp. 974–983 (ACM, 2018).

Zafar, M. B., I. Valera, M. Gomez Rodriguez and K. P. Gummadi, "Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment", in "Proceedings of the 26th international conference on world wide web", pp. 1171–1180 (2017).

Zafarani, R., M. A. Abbasi and H. Liu, *Social media mining: an introduction* (Cambridge University Press, 2014).

Zou, H., K. Kuang, B. Chen, P. Chen and P. Cui, "Focused context balancing for robust offline policy evaluation", in "KDD", pp. 696–704 (ACM, 2019).

BIOGRAPHICAL SKETCH

Ruocheng Guo is a Ph.D. candidate of Computer Engineering (Computer Systems) at Arizona State University under the supervision of Professor Huan Liu. His research lies in causal inference, machine learning, and data mining. He was the recipient of the 2020 ASU CIDSE Doctoral Fellowship. He has published more than 30 innovative works in highly ranked journals such as ACM CSUR and top conference proceedings such as KDD, WSDM, CIKM, IJCAI, and SDM. He was a research intern at Microsoft Research AI, an AI resident at Google X, and a research intern at Etsy. He received his MSc degree from the Hong Kong University of Science and Technology and his BEng degree from Huazhong University of Science and Technology. More can be found at `http://www.public.asu.edu/~rguo12`.