Be Causal: De-biasing Social Network Confounding in Recommendation

QIAN LI*, Curtin University, Australia
XIANGMENG WANG*, University of Technology Sydney
ZHICHAO WANG, University of New South Wales
GUANDONG XU†, University of Technology Sydney, Australia

In recommendation systems, the existence of the missing-not-at-random (MNAR) problem results in the selection bias issue, degrading the recommendation performance ultimately. A common practice to address MNAR is to treat missing entries from the so-called "exposure" perspective, i.e., modeling how an item is exposed (provided) to a user. Most of the existing approaches use heuristic models or re-weighting strategy on observed ratings to mimic the missing-at-random setting. However, little research has been done to reveal how the ratings are missing from a causal perspective. To bridge the gap, we propose an unbiased and robust method called DENC (*De-bias Network Confounding in Recommendation*) inspired by confounder analysis in causal inference. In general, DENC provides a causal analysis on MNAR from both the inherent factors (e.g., latent user or item factors) and auxiliary network's perspective. Particularly, the proposed exposure model in DENC can control the social network confounder meanwhile preserve the observed exposure information. We also develop a deconfounding model through the balanced representation learning to retain the primary user and item features, which enables DENC generalize well on the rating prediction. Extensive experiments on three datasets validate that our proposed model outperforms the state-of-the-art baselines.

CCS Concepts: • **Information systems**; • **Collaborative filtering**; • **Computer systems organization** → Robotics; • **Networks** → Network;

Additional Key Words and Phrases: Recommendation; Missing-Not-At-Random; Causal Inference; Bias; Propensity

ACM Reference Format:
Qian Li, Xiangmeng Wang, Zhichao Wang, and Guandong Xu. 2018. Be Causal: De-biasing Social Network Confounding in Recommendation. In *Woodstock '18: ACM Symposium on Neural Gaze Detection, June 03–05, 2018, Woodstock, NY*. ACM, New York, NY, USA, 23 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Recommender systems aim to handle information explosion meanwhile to meet users' personalized interests, which have received extensive attention from both research communities and industries [15, 18, 22]. The power of a recommender system highly relies on whether the observed user feedback on items "correctly" reflects the users' preference or not. The feedback can be categorised into explicit feedback (e.g., users' numerical ratings) or implicit feedback (e.g., purchases, views and clicks). However, such implicit or explicit feedback suffers from the missing issue that needs to be resolved to achieve high-quality recommendations [14, 45, 52]. To handle the partially observed feedback, a common assumption

*Equal contribution.
†Corresponding author: guandong.xu@uts.edu.au

for model building is that the feedback is missing at random (MAR), i.e., the probability of a rating to be missing is independent of the value. When the observed data follows the MAR, using only the observed data via statistical analysis methods can yield "correct" prediction without introducing bias [25, 31]. However, this MAR assumption usually does not hold in reality and the missing pattern exhibits *missing not at random* (MNAR) phenomenon. Generally speaking, prior study offers compelling evidence to show MNAR can be attributed to selection bias for explicit feedback [31] or exposure bias for implicit feedback [4]. These findings shed light on the origination of bias from MNAR explicit [44]. In our work, we focus on address the MNAR issue in explicit feedback to mitigate the selection bias. Particularly, selection bias occurs because users are free to choose which items to rate, so that the observed ratings are not the representative population of all ratings. That might because users are only exposed to a part of specific items so that unobserved interactions do not always represent negative preference. How to model the missing data mechanism and debias the rating performance forms up the main motivation of this research.

Existing MNAR-aware Methods

There are abundant methods for addressing the MNAR problem on the implicit or explicit feedback. For implicit feedback, traditional methods [15] take the uniformity assumption that assigns a uniform weight to down-weight the missing data, assuming that each missing entry is equally likely to be negative feedback. This is a strong assumption and limits models' flexibility for real applications. Recently, researchers tackle MNAR data directly through simulating the generation of the missing pattern under different heuristics [14]. Of these works, probabilistic models are presented as a proxy to relate missing feedback to various factors, e.g., item features. For explicit feedback, a widely adopted mechanism is to exploit the dependencies between rating missingness and the potential ratings (e.g., 1-5 star ratings) [19]. That is, high ratings are less likely to be missing compared to items with low ratings. However, these paradigm methods involve heuristic alterations to the data, which are neither empirically verified nor theoretically proven [40].

A couple of methods have recently been studied for addressing MNAR [14, 23, 42] by treating missing entries from the so-called "exposure" perspective, i.e., indicating whether or not an item is exposed (provided) to a user. For example, ExpoMF resorts modeling the probability of *exposure* [14], and up-weighting the loss of rating prediction with high *exposure* probability. However, ExpoMF can lead to a poor prediction accuracy for rare items when compared with popular items. Likewise, recent works [23, 42] resort to *propensity score* to model *exposure*. The *propensity score* introduced in causal inference indicates *the probability that a subject receiving the treatment or action*. Exposing a user to an item in a recommendation system is analogous to exposing a subject to a treatment. Accordingly, they adopt *propensity score* to model the *exposure* probability and re-weight the prediction error for each observed rating with the inverse *propensity score*. The ultimate goal is to calibrate the MNAR feedbacks into missing-at-random ones that can be used to guide unbiased rating prediction.

Whilst the state-of-the-art propensity-based methods are validated to alleviate the MNAR problem for recommendation somehow, they still suffer from several major drawbacks: 1) they merely exploit the user/item latent vectors from the ratings for mitigating MNAR, but fail to disentangle different causes for MNAR from a causal perspective; 2) technically, they largely rely on propensity score estimation to mitigate MNAR problem; the performance is sensitive to the choice of propensity estimator [52], which is notoriously difficult to tune.

The proposed approach

To overcome these obstacles, in contrast, we aim to address the fundamental MNAR issue in recommendation from a novel causal inference perspective, to attain a robust and unbiased rating prediction model. From a causal perspective, we argue that the selection bias (i.e., MNAR) in the recommendation system is attributed to the presence of *confounders*. As explained in Figure 1, *confounders* are factors (or variables) that affect both the treatment assignments (exposure) and

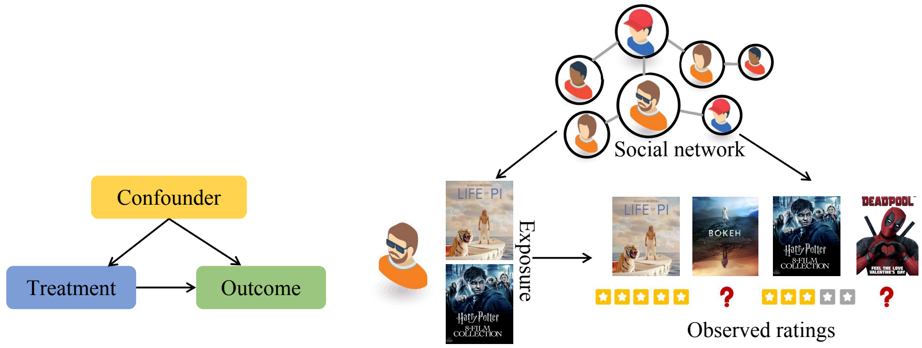


Fig. 1. The causal view for MNAR problem: *treatment* and *outcome* are terms in the theory of causal inference, which denote an action taken (e.g.,*exposure*) and its result (e.g., *rating*), respectively. The *confounder* (e.g., *social network*) is the common cause of treatment and outcome.

the outcomes (rating). For example, friendships (or social network) can influence both users' choice of movie watching and their further ratings. Users tend to consume and rate the items that they like and the items that have been consumed by their friends. So, *the social network is indeed a confounding factor that affects which movie the user is exposed to and how the user rates the movie*. The confounding factor results in a *distribution discrepancy between the partially observed ratings and the complete ratings* as shown in Figure 2. Without considering the distribution discrepancy, the rating model trained on the observed ratings fails to generalize well on the unobserved ratings. With this fact in mind, our idea is to analyze the confounder effect of social networks on rating and exposure, and in turn, fundamentally alleviate the MNAR problem to predict valid ratings.

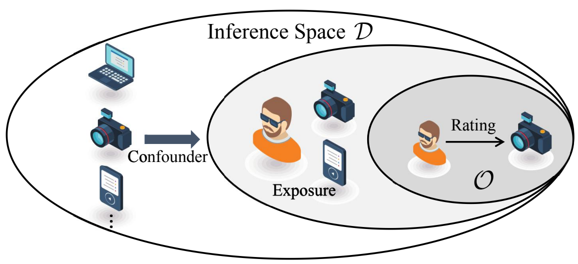


Fig. 2. The training space of conventional recommendation models is the observed rating space $\mathcal{O}$, whereas the inference space is the entire exposure space $\mathcal{D}$. The discrepancy of data distribution between $\mathcal{O}$ and $\mathcal{D}$ leads to selection bias in conventional recommendation models.

In particular, we attempt to study the MNAR problem in recommendation from a causal view and propose an unbiased and robust method called DENC (*De-bias Network Confounding in Recommendation*). To sufficiently consider the selection bias in MNAR, we model the underlying factors (i.e., inherent user-item information and social network) that can generate observed ratings. In light of this, as shown in Figure 4, we construct a causal graph based recommendation framework by disentangling three determinants for the ratings, i.e., *inherent factors*, *confounder* and *exposure*. Each determinant accordingly corresponds to one of three specific components in DENC: *deconfonder model*, *social network confounder* and *exposure model*, all of which jointly determine the rating outcome.

In summary, the key contributions of this research are as follows:

- Fundamentally different from previous works, DENC is the first method for the unbiased rating prediction through disentangling determinants of selection bias from a causal view.
- The proposed *exposure model* is capable of revealing the exposure assignment and accounting for the confounder factors derived from the *social network confounder*, which thus remedies selection bias in a principled manner.
- We develop a *deconfonder model* via the balanced representation learning that embeds inherent factors independent of the exposure, therefore mitigating the distribution discrepancy between the observed rating and inference space.
- We conduct extensive experiments to show that our DENC method outperforms state-of-the-art methods. The generalization ability of our DENC is also validated by verifying different degrees of confounders.

2 RELATED WORK

In this section, we discuss the relationship between missing mechanisms and bias, as well as some recommendation methods to address this issue.

2.1 MNAR Assumption and Bias Issue

To analyze data with missing values, it is imperative to understand the missing mechanisms. Missing data mechanisms are categorised into three categories: missing completely at random (MCAR), missing at random (MAR), and missing not at random (MNAR) [39]. In the recommendation scenario, MCAR refers to the missingness that the probability of a rating to be missing is completely random; the missingness is MAR if the probability of not observing a rating is independent of the value of that rating but related to some of the observed data; and the mechanism is MNAR if it is neither MCAR nor MAR. A critical assumption behind collaborative filtering (CF) is that the missing ratings are MAR [15, 18, 18, 19, 37], i.e., the missingess of user's feedback is independent of user's preference [15]. Following this MAR assumption, numerous approaches have been developed, including matrix factorization based-recommenders [18, 37], SVD++ [19] and timeSVD [18]. However, this MAR assumption does not hold because real-world recommender systems are subject to bias [27, 31], including but not limited to selection bias in explicit feedback [14, 27, 29, 31] and exposure bias in implicit feedback [4, 14, 28, 57]. These biases make the observed feedback deviate from reflecting user true preference, which are theoretically and empirically proved by several studies [42, 45, 56]. Hence, without considering the biases, naively fitting feedback would lead to suboptimal prediction. In our work, we focus on the explicit user rating data and achieve a high prediction accuracy using MNAR feedback.

2.2 MNAR-aware Methods

Given the wide existence of data biases, we investigate the related work of addressing the bias for the MNAR feedback, including data imputation-based and propensity-based methods.

2.2.1 Data Imputation-based. Note that the main reason for the selection bias in the observed rating data is that users are free to deliberately choose which items to rate. Early works adopt a direct manner for mitigating selection bias, which jointly integrate rating prediction and missing data model (i.e. 'which items the user select to rate') via sophisticated approximate inference [14, 24, 26, 31, 46, 55]. The basic assumption behind these methods is that the probability of users' selection on items depends on users' rating values for that item. For example, Marlin and Zemel [31] model the missing probability of a user-item pair dependent on the user rating values through a mixture of Multinomials. Alternatively, a probabilistic matrix factorization is proposed to characterize the missing probability of a user-item pair [26, 47] to improve the flexibility of MM model. Hernandez et al. [14] use a new probabilistic matrix factorization model with hierarchical priors for ordinal rating data, which increases robustness to the selection of hyper-parameters. Recently, Ohsawa et al. [34] further extend probabilistic matrix factorization to a Gated PMF by considering the dependency between why a user consumes an item and how that affects the rating value. Chen et al. [5] model user's consumption with social influence for better estimating user's preference on items. In summary, the data-imputation based approach often has a large bias due to imputation inaccuracy, which would be propagated into training a prediction model and easily mislead the prediction [7, 22, 52].

2.2.2 Propensity-based. To remedy the selection bias in evaluation, another kind of methods considers a recommendation as an intervention analogous to treating a patient with a specific medicine [23, 42, 48, 53]. The propensity score for a user-item pair is computed as the marginal probability of observing a rating value for the user-item pair, which can offset the selection bias when training a recommendation model. Particularly, they directly re-weight the prediction error for each observed rating with the inverse propensity score of observing that rating. For example, Schnabel et al. [42] compute the propensity from user ratings or indirectly through user and item covariates, and propose an empirical risk minimization approach to learning the unbiased estimators from biased rating data. Alternatively, Liang et al. [23] capture the propensity score using user exposure (what the user sees). Then, the inverse propensity score is leveraged to train a click model (what the user click on) via a Bayesian model to correct exposure bias. These works re-weight the observational click data as though it came from an "experiment" where users are randomly shown items. For MNAR implicit feedback, Saito et al. [41] construct an unbiased estimator for the loss function of interest using only biased implicit feedback. However, most of these methods are sensitive to the choice of propensity score estimators and can suffer from high variance of the propensities [8, 52, 54]. Accordingly, Wang et al. integrate the propensity score estimation and the data imputation model in a theoretically sophisticated manner [52] such that the performance is less affected by the mis-specification of the models. In general, although propensity-based methods outperform the state-of-the-art traditional recommendation methods, they do not take social information into consideration.

3 PROBLEM FORMULATION

In this section, we first introduce the notations of causal inference so as to prepare readers with the basics. Following this, we analyze the confounding bias of conventional recommender system from a causal view.

3.1 Notations of Causal Inference

Causal inference aims to estimate the counterfactual outcome that is the outcome if the unit had taken another treatment or action [36, 38, 39]. However, estimating counterfactual outcome from the observable data is challenging due to the presence of confounders [3, 4, 53]. To understand this issue, we present the some key definitions in causal inference.

Definition 1 (Treatment). *Treatment refers to the action or intervention that applies to a sample.*

Definition 2 (Potential outcome). *For each unit-treatment pair, the outcome of that treatment when applied on that unit is the potential outcome.*

Since a unit can only take one treatment, only one potential outcome can be observed (i.e., factual outcome), and the remaining unobserved potential outcomes are the counterfactual outcome.

Definition 3 (Confounder). *Given a pair of treatment and outcome, we say a variable is a confounder iff it affects both treatment and outcome.*

Confounder is a common causes of the treatment and outcome, which leads to the confounding bias when we estimate counterfactual outcome from observational data [38, 39]. Confounding bias in causal inference is equivalent to a domain adaptation scenario where a model is trained on a "source" (observed) data distribution, but should perform well on a "target" (counterfactual) one [1, 23]. Handing confounding bias is the essential part of causal inference [3, 36], which makes estimating counterfactual outcome from observational data feasible.

3.2 A Causal Inference Perspective on Recommendation

Viewing recommendation from a causal inference perspective, we argue that exposing a user to an item in recommendation is analogous to exposing a patient to a treatment in a medical study. In both tasks, we have only partial observations of how much certain users (patients) prefer (benefit from) certain items (treatments). We are interested in the counterfactual question "if user (patient) had exposed (adopted) to other items (treatments), how much would the user (patient) prefer (benefits from)?". Following this principle, we aim to answer such a counterfactual prediction in recommendation. Prior to that, we first give the notations. We assume that $Y \in \mathbb{R}^{m \times n} = [\dot{y}_{ui}]$ is the user-item rating matrix, in which $\dot{y}_{ui}$ is the rating given by user u to item i. In addition, for every user-item pair (u, i), we have a binary exposure $a_{ui} \in \{1, 0\}$ indicates that the item i is exposed to user u or not. Let G denote user-user social graph among users where $G_{kj} = 1$ if u_k has a relation to u_j and zero otherwise. Let $N_s(u)$ be the set of users whom u directly connected with. Based on these notations, we give a formal problem definition as below.

Problem 1 (Causal View for Recommendation). *Given the social network G and partially observed ratings Y, for every user-item pair (u, i) with $a_{ui} = 0$, we aim to* estimating the ratings had these items been exposed by all users.

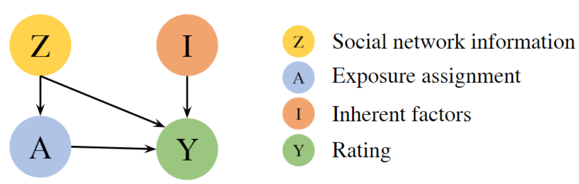


Fig. 3. The causal graph in recommendation.

Inspired by causal inference theory [36, 38, 39], we resort to causal graph that provides potentials to answer this question. As a directed acyclic graph, causal graph can describe the generation mechanism of recommendation results and guide the design of recommendation methods. In our work, we investigate social network as a confounder that is a common cause of item exposure A and rating Y. In particular, we abstract a structural causal graph, as shown in Figure 3, to explicitly analyze the causal relations in the conventional recommender system. The causal graph consists of four variables: confounder Z, exposure A, inherent factor I and rating Y. Every directed edge represents a causal relation between two variables. The rationality of causal relations in Figure 3 can be explained as follows.

- $Z \rightarrow A$: the social network information of users affects users' choice of movie. For example, a user's social network might affect the movies he is exposed to.
- $Z \rightarrow Y$: a users' social network can affect users' preference on items. Similarly, the social network can affect how much the user likes movies he watched.
- $(Z, A) \rightarrow Y$: observed ratings are generated as results of *which items are exposed to user* and *the user's preference for each of those items.*
- $I \rightarrow Y$: inherent factors I affects the recommendation outcome Y. For example, I refers to the inherent factors that are acquired from demographic features of users and items. For example, user ID and item genre.

In recommendation scenario, the social network is a confounder variable affects both user's exposure to items and the user's rating. Recall that our interest is to estimate counterfactual ratings of the unexposed user-item pair (i.e., $a_{u,i} = 0$) in which if the user had been exposed to the item. According to causal inference [36], the confounder in recommendation scenario leads to the selection bias.

Definition 4 (Selection Bias). *The observed ratings in the user-item pair (i.e., $a_{u,i} = 1$) is not representative to the unexposed user-item pair (i.e., $a_{u,i} = 0$) we are interested in.*

Selection bias indicates the observed ratings are not representative samples of the whole population, since users in different social networks have different selection preferences. Consequently, without handling the selection bias, counterfactual rating model is trained to over-recommend the majority population and amplify the imbalance, thus would work poorly in rating estimation. Thus, eliminating the impact of the confounder is the necessary to attain an unbiased counterfactual rating prediction.

4 METHODOLOGY

To resolve the impact of the confounder, we propose a novel approach called DENC to disentangle determinants on rating outcome guided by the causal graph in Figure 3. The overall framework of our DENC is shown in Figure 3 includes three components: *social network confounder*, *exposure model* and *deconfonder model*. In the following, we will elaborate on each component and the debiasing process for rating prediction.

4.1 Exposure Model

To cope with the selection bias caused by users or the external social relations, we build on the causal inference theory and propose an effective exposure model. Guided by the treatment assignment mechanism in causal inference, we propose a novel exposure model that computes the probability of exposure variable specific to the user-item pair. This model is beneficial to understand the generation of the *Missing Not At Random* (MNAR) patterns in ratings, which thus remedies selection biases in a principled manner. For example, user goes to watch the movie because of his friend's strong recommendation. Thus, we propose to mitigate the selection bias by exploiting the network connectivity information that indicating *to which extent the exposure for a user will be affected by its neighbors.*

4.1.1 Social Network Confounder. To control the selection bias arisen from the external social network, we propose a confounder representation model that quantifies the common biased factors affecting both the exposure and rating.

We now discuss the method of choosing and learning exposure. Let G present the social relationships among users U, where an edge denotes there is a friend relationship between users. We resort to node2vec [11] method and learn network embedding from diverse connectivity provided by the social network. More details about node2vec method can

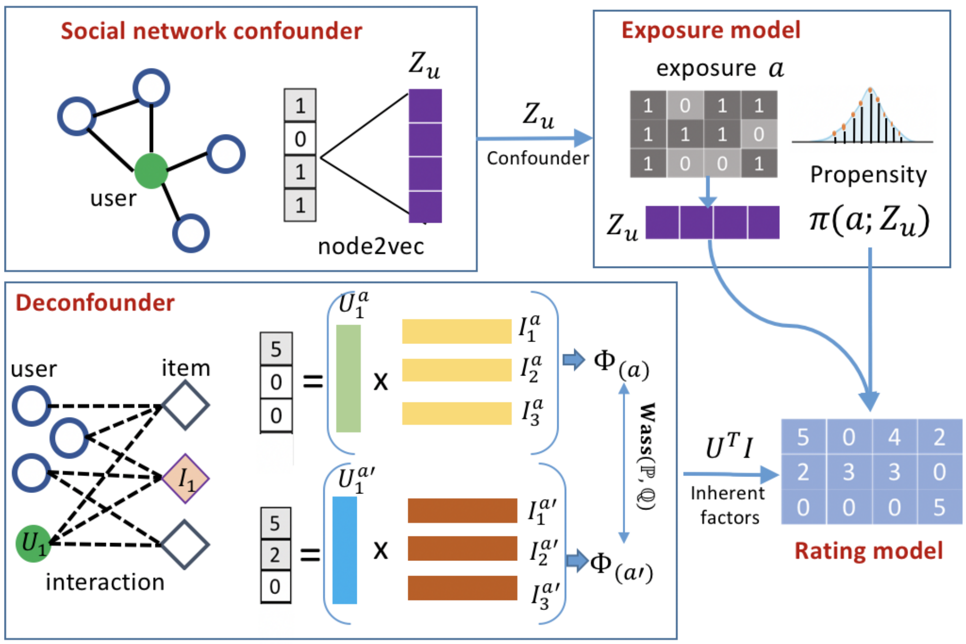


Fig. 4. Our DENC method consists of *Social network confounder*, *exposure model*, *deconfonder model* and *rating model*.

be found in Section A.4 in the appendix. To mine the deep social structure from G, for every source user u, node2vec generates the network neighborhoods $N_s(u) \subset G$ of node u through a sampling strategy to explore its neighborhoods in a breadth-first sampling as well as a depth-first sampling manner. The representation Z_u for user u can be learned by minimizing the negative likelihood of preserving network neighborhoods $N_s(u)$:

$$\mathcal{L}_z = -\sum_{u \in G} \log P(N_s(u)|Z_u) = \sum_{u \in G} \left[\log \sum_{v \in G} \exp(Z_v \cdot Z_u) - \sum_{u_i \in N_s(u)} Z_{u_i} \cdot Z_u \right] \quad (1)$$

The final output $Z_u \in \mathbb{R}^d$ sufficiently explores diverse neighborhoods of each user, which thus represents to what extent the exposure for a user is influenced by his friends in graph G.

4.1.2 Exposure Assignment Learning. The exposure under the recommendation scenario is not randomly assigned. Users in social networks often express their own preferences over the social network, which therefore will affect their friends' exposure policies. In this section, to characterize the *Missing Not At Random* (MNAR) pattern in ratings, we resort to causal inference [36] to build the exposure mechanism influenced by social networks.

To begin with, we are interested in the binary exposure a_{ui} that defines whether the item i is exposed ($a_{ui} = 1$) or unexposed ($a_{ui} = 1$) to user u, i.e., $a_{ui} = 1$. Based on the informative confounder learned from social network, we propose the notation of *propensity* to capture the exposure from the causal inference language.

Definition 5 (Propensity). *Given an observed rating $y_{ui} \in$ rating and confounder Z_u in (1), the propensity of the corresponding exposure for user–item pair (u, i) is defined as*

$$\pi(a_{ui}; Z_u) = P(a_{ui} = 1 | y_{ui} \in rating; Z_u) \quad (2)$$

In view of the foregoing, we model the exposure mechanism by the probability of a_{ui} being assigned to 0 or 1.

$$P(a_{ui}) = \prod_{u,i} P(a_{ui}) = \prod_{(u,i)\in O} P(a_{ui} = 1) \prod_{(u,i)\notin O} P(a_{ui} = ?) \tag{3}$$

where O is an index set for the observed ratings. The case of $a_{ui} = 1$ can result in an observed rating or unobserved rating: 1) for the observed rating represented by $y_{ui} \in$ rating, we definitely know the item i is exposed, i.e., $a_{ui} = 1$; 2) an unobserved rating $y_{ui} \notin$ rating may represent a negative feedback (i.e., the user is not reluctant to rating the item) on the exposed item $a_{ui} = 1$. In light of this, based on (2), we have

$$\begin{aligned} P(a_{ui} = 1) &= P(a_{ui} = 1, y_{ui} \in \text{rating}) + P(a_{ui} = 1, y_{ui} \notin \text{rating}) \\ &= \pi(a_{ui}; Z_u) P(y_{ui} \in \text{rating}) + W_{ui} P(y_{ui} \notin \text{rating}) \end{aligned} \tag{4}$$

where $W_{ui} = P(a_{ui} = 1 | y_{ui} \notin \text{rating})$. The exposure a_{ui} that is unknown follows the distributions as

$$P(a_{ui} = ?) = 1 - P(a_{ui} = 1) \tag{5}$$

By substituting Eq. (4) and Eq. (5) for Eq. (3), we attain the exposure assignment for the overall rating data as

$$P(a_{ui}) = \prod_{(u,i)\in O} \pi(a_{ui}; Z_u) \prod_{(u,i)\notin O} (1 - W_{ui}) \tag{6}$$

Inspired by [35], we assume uniform scheme for W_{ui} when no side information is available. According to most causal inference methods [36, 43], a widely-adopted parameterization for $\pi(a_{ui}; Z_u)$ is a logistic regression network parameterized by $\Theta = \{W_0, b_0\}$, i.e.,

$$\pi(a_{ui}; Z_u, \Theta) = \mathbb{I}_{y \in \text{rating}} \cdot \left[1 + e^{-(2a_{ui}-1)(Z_u^\top \cdot W_0 + b_0)}\right]^{-1} \tag{7}$$

Based on Eq. (7), the overall exposure $P(a_{ui})$ in Eq. (6) can be written as the function of parameters $\Theta = \{W_0, b_0\}$ and Z_u, i.e.,

$$\mathcal{L}_a = \sum_{u,i} - \log P(a_{ui}; Z_u, \Theta) \tag{8}$$

where social network confounder Z_u is learned by the pre-trained node2vec algorithm. Similar to supervised learning, Θ can be optimized through minimization of the negative log-likelihood.

4.2 Deconfounder Model

Traditional recommendation learns the latent factor representations for user and item by minimizing errors on the observed ratings, e.g., matrix factorization. Due to the existence of selection bias, such a learned representation may not necessarily minimize the errors on the unobserved rating prediction. Inspired by [43], we propose to learn a balanced representation that is independent of exposure assignment such that it represents inherent or invariant features in terms of users and items. The invariant features must also lie in the inference space shown in Figure 2, which can be used to consistently infer unknown ratings using observed ratings. This makes sense in theory: if the learned representation is hard to distinguish across different exposure settings, it represents invariant features related to users and items.

According to Figure 3, we can define two latent vectors $U \in \mathbb{R}^{k_d}$ and $I \in \mathbb{R}^{k_d}$ to represent the inherent factor of a user and a item, respectively. Recall that different values for W_{ui} in Eq. (6) can generate different exposure assignments for the observed rating data. Following this intuition, we construct two different exposure assignments a and $\hat{a}$ corresponding two settings of W_{ui}. Accordingly, $\Phi_{(a)}$ and $\Phi_{(\hat{a})}$ are defined to include inherent factors of users and items, i.e., $\Phi_{(a)} = \left[U_1^{(a)}, \cdots, U_M^{(a)}, I_1^{(a)}, \cdots, I_M^{(a)}\right] \in \mathbb{R}^{k_d \times 2M}$, $\Phi_{(\hat{a})} = \left[U_1^{(\hat{a})}, \cdots, U_M^{(\hat{a})}, I_1^{(\hat{a})}, \cdots, I_M^{(\hat{a})}\right] \in \mathbb{R}^{k_d \times 2M}$. Figure 3 also indicates that the inherent factors of user and item would keep unchanged even if the exposure variable is altered from 0 to 1, and vice versa. That means $U \in \mathbb{R}^{k_d}$ and $I \in \mathbb{R}^{k_d}$ should be independent of the exposure assignment, i.e., $U^{(a)}U^{(\hat{a})}$ or $I^{(a)}I^{(\hat{a})}$. Accordingly, minimizing the discrepancy between $\Phi_{(a)}$ and $\Phi_{(\hat{a})}$ ensures that the learned factors embeds no information about the exposure variable and thus reduce selection bias. The penalty term for such a discrepancy is defined as

$$\mathcal{L}_d = \text{disc}\left(\Phi_{(\hat{a})}, \Phi_{(a)}\right) \tag{9}$$

Inspired by [33], we employ *Integral Probability Metric* (IPM) to estimate the discrepancy between $\Phi_{(\hat{a})}$ and $\Phi_{(a)}$. $\text{IPM}_{\mathcal{F}}(\cdot, \cdot)$ is the (empirical) integral probability metric defined by the function family $\mathcal{F}$. Define two probability distributions $\mathbb{P} = P(\Phi_{(\hat{a})})$ and $\mathbb{Q} = P(\Phi_{(a)})$, the corresponding IPM is denoted as

$$\text{IPM}_{\mathcal{F}}(\mathbb{P}, \mathbb{Q}) = \sup_{f \in \mathcal{F}} \left| \int_S f d\mathbb{P} - \int_S f d\mathbb{Q} \right| \tag{10}$$

where $\mathcal{F} : S \rightarrow \mathbb{R}$ is a class of real-valued bounded measurable functions. We adopt $\mathcal{F}$ as 1-Lipschitz functions that lead IPM to the Wasserstein-1 distance, i.e.,

$$Wass(\mathbb{P}, \mathbb{Q}) = \inf_{f \in \mathcal{F}} \sum_{\mathbf{v} \in \text{col}_i(\Phi_{(\hat{a})})} \|f(\mathbf{v}) - \mathbf{v}\| \mathbb{P}(\mathbf{v}) d\mathbf{v} \tag{11}$$

where $\mathbf{v}$ is the i-th column of $\Phi_{(\hat{a})}$ and the set of push-forward functions $\mathcal{F} = \left\{ f \mid f : \mathbb{R}^d \rightarrow \mathbb{R}^d \text{ s.t. } \mathbb{Q}(f(\mathbf{v})) = \mathbb{P}(\mathbf{v}) \right\}$ can transform the representation distribution of the exposed $\Phi_{(\hat{a})}$ to that of the unexposed $\Phi_{(a)}$. Thus, $\|f(\mathbf{v}) - \mathbf{v}\|$ is a pairwise distance matrix between the exposed and unexposed user-item pairs. Based on the discrepancy defined in (12), we define $C(\Phi) = \|f(\mathbf{v}) - \mathbf{v}\|$ and reformulate penalty term in (9) as

$$\mathcal{L}_d = \inf_{\gamma \in \Pi(\mathbb{P}, \mathbb{Q})} \mathbb{E}_{(\mathbf{v}, f(\mathbf{v})) \sim \gamma} C(\Phi) \tag{12}$$

We adopt the efficient approximation algorithm proposed by [43] to compute the gradient of (12) for training the deconfounder model. In particular, a mini-batch with l exposed and l unexposed user-item pairs is sampled from $\Phi_{(\hat{a})}$ and $\Phi_{(a)}$, respectively. The element of distance matrix $C(\Phi)$ is calculated as $C_{ij} = \|\text{col}_i(\Phi_{(\hat{a})}) - \text{col}_j(\Phi_{(a)})\|$. After computing $C(\Phi)$, we can approximate f and the gradient against the model parameters [1]. In conclusion, the learned latent factors generated by the deconfounder model embed no information about exposure variable. That means all the confounding factors are retained in social network confounder Z_u.

4.3 Learning

4.3.1 Rating prediction. Having obtained the final representations U and I by the deconfounder model, we use an inner product of $U^\top I$ as the inherent factors to estimate the rating. As shown in the causal structure in Figure 4, another component affecting the rating prediction is the social network confounder. A simple way to incorporate these components into recommender systems is through a linear model as follows.

$$\hat{y}_{ui} = \sum_{u,i \in O} U^\top I + {W_u}^\top Z_u + \epsilon_{ui}, \quad \epsilon_{ui} \sim \mathcal{N}(0, 1) \tag{13}$$

[1] For a more detailed calculation, refer to Algorithm 2 in the appendix of prior work [43]

where W_u is a coefficient that describes how much the confounder Z_u contributes to the rating. To define the unbiased loss function for the biased observations y_{ui}, we leverage the IPS strategy [42] to weight each observation with *Propensity*. By Definition 5, the intuition of the inverse propensity is to down-weight the commonly observed ratings while up-weighting the rare ones.

$$\mathcal{L}_y = \frac{1}{|O|} \sum_{u,i \in O} \frac{(y_{ui} - \hat{y}_{ui})^2}{\pi(a_{ui}; Z_u)} \tag{14}$$

4.3.2 Optimization. To this end, the objective function of our DENC method to predict ratings could be derived as:

$$\mathcal{L} = \mathcal{L}_y + \lambda_a \mathcal{L}_a + \lambda_z \mathcal{L}_z + \lambda_d \mathcal{L}_d + \mathcal{R}(\Omega) \tag{15}$$

where Ω represents the trainable parameters and $\mathcal{R}(\cdot)$ is a squared l_2 norm regularization term on Ω to alleviate the overfitting problem. λ_a, λ_z and λ_d are trade-off hyper-parameters. To optimize the objective function, we adopt Stochastic Gradient Descent(SGD) [2] as the optimizer due to its efficiency.

5 EXPERIMENTS

To more thoroughly understand the nature of MNAR issue and the proposed unbiased DENC, experiments are conducted to answer the following research questions:

- (**RQ1**) How confounder bias caused by the social network is manifested in real-world recommendation datasets?
- (**RQ2**) Does our DENC method achieve the state-of-the-art performance in debiasing recommendation task?
- (**RQ3**) How does the embedding size of each component (e.g., social network confounder and deconfounder model) in our DENC method impact the debiasing performance?
- (**RQ4**) How do the missing social relations impact the debiasing performance of our DENC method?

5.1 Setup

5.1.1 Evaluation Metrics. We adopt two popular metrics including Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) to evaluate the performance. Since improvements in MAE or RMSE have a significant impact on the quality of the Top-K recommendations [17], we also evaluate our DENC with Precision@K and Recall@K for the ranking performance[2].

5.1.2 Datasets. We conduct experiments on three datasets including one semi-synthetic dataset and two benchmark datasets `Epinions` [3] and `Ciao` [49] [4]. We maintain all the user-item interaction records in the original datasets instead of discarding items that have sparse interactions with users.[5] The semi-synthetic dataset is generated by incorporating the social network into `MovieLens`[6] dataset. The details of these datasets are given in Section A.1 in the appendix.

5.1.3 Baselines. We compare our DENC against three groups of methods for rating prediction: (1) **Traditional methods**, including NRT [20] and PMF [32]. (2) **Social network-based methods**, including GraphRec [9], DeepFM+ [12], SocialMF [16], SREE [21] and SoReg [30]. (3) **Propensity-based methods**, including CausE [1] and D-WMF [53]. More implementation details of baselines and parameter settings are included in Section A.2 in the appendix.

[2]We consider items with a rating greater than or equal to 3.5 as relevant
[3]http://www.cse.msu.edu/ tangjili/trust.html
[4]http://www.cse.msu.edu/ tangjili/trust.html
[5]Models can benefit from the preprocessed datasets in which all items interact with at least a certain amount of users, for such preprocessing will reduce the dataset sparsity.
[6]https://grouplens.org/datasets/movielens

Table 1. Statistics of Datasets. Density for rating (density-R) is $\#ratings/(\#users \cdot \#items)$, Density for social relations (density-SR) is $\#relations/(\#users \cdot \#users)$.

	Epinions	Ciao	MovieLens-1M
# users	22,164	7,317	6,040
# items	296,277	104,975	3,706
# ratings	922,267	283,319	1000,209
density-R (%)	0.0140	0.0368	4.4683
# relations	355,754	111,781	9,606
density-SR (%)	0.0724	0.2087	0.0263

Table 2. Performance comparison: bold numbers are the best results. Strongest baselines are highlighted with underlines.

		Traditional		Social network-based					Propensity-based		Ours		
Dataset	Metrics	PMF	NRT	SocialMF	SoReg	SREE	GraphRec	DeepFM+	CausE	D-WMF	**DENC**	**improv.**	***p*-value**
Epinions	MAE	0.9505	0.9294	0.8722	0.8851	0.8193	0.7309	0.5782	0.5321	0.3710	**0.2684**	38.2%	5.73e-5
	RMSE	1.2169	1.1934	1.1655	1.1775	1.1247	0.9394	0.6728	0.7352	0.6299	**0.5826**	8.1%	3.96e-3
Ciao	MAE	0.8868	0.8444	0.7614	0.7784	0.7286	0.6972	0.3641	0.4209	0.2808	**0.2487**	12.9%	3.62e-4
	RMSE	1.1501	1.1495	1.0151	1.0167	0.9690	0.9021	0.5886	0.8850	0.5822	**0.5592**	4.1%	7.32e-5
MovieLens-1M	MAE	0.8551	0.8959	0.8674	0.9255	0.8408	0.7727	0.5786	0.4683	0.3751	**0.2972**	26.2%	3.31e-5
$\Delta(Z_u) = -0.35$	RMSE	1.0894	1.1603	1.1161	1.1916	1.0748	0.9582	0.6730	0.8920	0.6387	**0.5263**	21.4%	6.11e-4
MovieLens-1M	MAE	0.8086	0.8801	0.8182	0.8599	0.7737	0.7539	0.5281	0.4221	0.3562	**0.2883**	23.4%	8.21e-6
$\Delta(Z_u) = 0$	RMSE	1.0034	1.1518	1.0382	1.1005	0.9772	0.9454	0.6477	0.8333	0.6152	**0.5560**	10.6%	1.75e-5
MovieLens-1M	MAE	0.7789	0.7771	0.7969	0.8428	0.7657	0.7423	0.3672	0.4042	0.3151	**0.2836**	11.1%	3.61e-3
$\Delta(Z_u) = 0.35$	RMSE	0.9854	0.9779	1.0115	1.0792	0.9746	0.9344	0.5854	0.8173	0.5962	**0.5342**	9.6%	4.38e-4

5.1.4 Parameter Settings. We implement all baseline models on a Linux server with Tesla P100 PCI-E 16GB GPU. [7] Datasets for all models except CausE [8] are split as training/test sets with a proportion of 80/20, and 20% of the training set are validation set.

We optimize all models with Stochastic Gradient Descent(SGD) [2]. For fair comparisons, a grid search is conducted to choose the optimal parameter settings, e.g., dimension of user/item latent vector k_{MF} for matrix factorization-based models and dimension of embedding vector d for neural network-based models. The embedding size is initialized with the Xavier [10] and searched in $[8, 16, 32, 64, 128, 256]$. The batch size and learning rate are searched in $[32, 64, 128, 512, 1024]$ and $[0.0005, 0.001, 0.005, 0.01, 0.05, 0.1]$, respectively. The maximum epoch N_{epoch} is set as 2000, an early stopping strategy is performed. Moreover, we employ three hidden layers for the neural components of NRT, GraphRec and DeepFM+. Like our DENC method, DeepFM+ uses node2vec to train the social network embeddings. Hence, the embedding size of its node2vec is set as the same as in our DENC for a fair comparison.

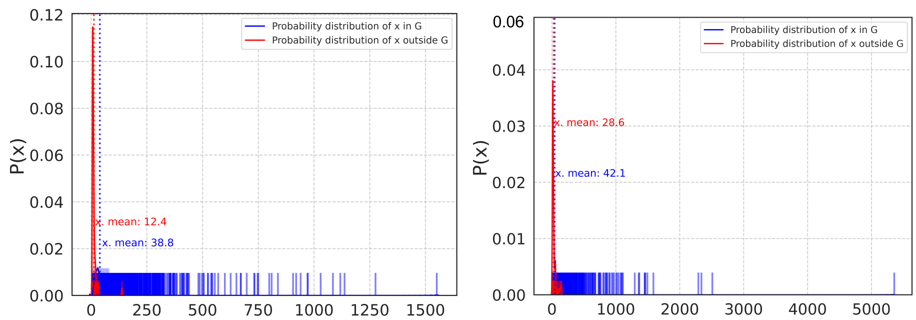
Without specification, unique hyperparameters of DENC are set as: three coefficients λ_a, λ_z and λ_d are tuned in $[0.2, 0.4, 0.6, 0.8, 1]$. The dimension of node2vec embedding size k_a and the dimension of inherent factor k_d are tuned in $[8, 16, 32, 64, 128, 256]$, and their influences are reported in Section 5.4.

[7] Our code is currently shared on Github, we leave the link void now but promise to activate it after paper acceptance.

[8] As in CausE, we sample 10% of the training set to build an additional debiased dataset (mandatory in model training), where items are sampled to be uniformly exposed to users.

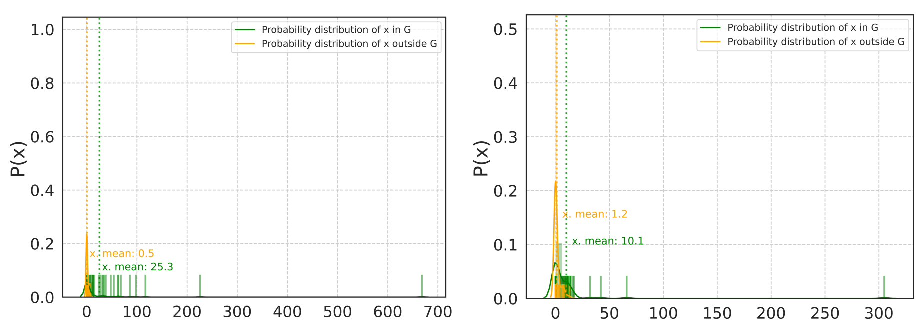
5.2 Understanding Social Confounder (RQ1)

We initially conduct an experiment to understand to what extent the confounding bias caused by social networks is manifested in real-world recommendation datasets. We claim that the social network as a confounder bias the interactions between the user and items. We aim to verify two kinds of scenarios: (1) User in the social network interacts with more items than users outside the social network. (2) The pair of user-neighbor in the social network has more common interacted items than the pair of user-neighbor outside the social network. Intuitively, an unbiased platform should expect users to interact with items broadly, which indicates that interactions are likely to be evenly distributed. Thus, we investigate the social confounder bias by analyzing the statistics of interactions in these two scenarios in `Epinions` and `Ciao` dataset.



(a) Distribution on `Ciao`. (b) Distribution on `Epinions`.

Fig. 5. Scenario (1): the distribution of x (the number of items interacted by a user). The smooth probability curves visualize how the number of items is distributed.



(a) Distribution on `Ciao`. (b) Distribution on `Epinions`.

Fig. 6. Scenario (2): the distribution of x (the number of items commonly interacted by a user-pair).

For the first scenario, we construct two user sets within or outside the social network, i.e., $\mathcal{U}_G$ and $\mathcal{U}_{\setminus G}$. Specially, $\mathcal{U}_G$ is constructed by randomly sampling a set of users in social network G, and $\mathcal{U}_{\setminus G}$ is randomly sampled out of G. The size of $\mathcal{U}_G$ and $\mathcal{U}_{\setminus G}$ is the same and defined as n. Following the above guidelines, we sample $n = 7,000$ users for $\mathcal{U}_G$ and $\mathcal{U}_{\setminus G}$. Figure 5 depicts the distributions of the interacted items by users in $\mathcal{U}_G$ and $\mathcal{U}_{\setminus G}$. The smooth curves are continuous distribution estimates produced by the kernel density estimation. Apparently, the distribution for $\mathcal{U}_{\setminus G}$ is significantly skewed: most of the users interact with few items. For example, on Ciao, more than 90% of users interact with fewer than 40 items. By contrast, most users in the social network tend to interact with items more frequently. In general, the distribution curve of $\mathcal{U}_G$ is quite different from $\mathcal{U}_{\setminus G}$, which reflects that the social network influences the interactions between users and items. In addition, the degree of bias varies across different datasets: Epinions is less biased than Ciao.

For the second scenario, based on $\mathcal{U}_G$ and $\mathcal{U}_{\setminus G}$, we further analyze the number of commonly interacted items by the user-pair. Particularly, we randomly sample four one-hop neighbours for each user in $\mathcal{U}_G$ to construct user-pairs. Since users in $\mathcal{U}_{\setminus G}$ have no neighbours, for each of them, we randomly select another four users[9] in $\mathcal{U}_{\setminus G}$ to construct four user-pairs. Recall that $\mathcal{U}_G$ and $\mathcal{U}_{\setminus G}$ both have 7,000 users, then we totally have $4 \times 7,000$ user-pairs for $\mathcal{U}_{\setminus G}$ and $\mathcal{U}_{\setminus G}$, respectively. Figure 6 represents the distribution of how many items are commonly interacted by the users in each pair.[10] Figure 6 indicates most user-neighbour pairs in the social network have fewer than 20 items in common. However the user-pairs outside the social network nearly have no items in common, i.e., less than 1. We can conclude that social networks can encourage users to share more items with their neighbours, compared with users who are not connected by any social networks.

5.3 Performance Comparison (RQ2)

We compare the rating prediction of DENC with nine recommendation baselines on three datasets including Epinions, Ciao and MovieLens-1M. Table 2 demonstrates the performance comparison, where the confounder $\Delta(Z_u)$ in MovieLens-1M is assigned with three different settings, i.e., -0.35, 0 and 0.35. The improvements and statistical significance test are performed between DENC and the strongest baselines (highlighted with underline). Analyzing Table 2, we have the following observations.

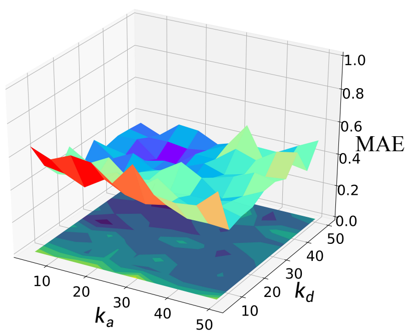
- Overall, our DENC consistently yields the best performance among all methods on five datasets. For instance, DENC improves over the best baseline model w.r.t. MAE/RMSE by 38.2%/8.1%, 12.9%/4.1%, and 26.2%/21.4% on Epinions, Ciao and MovieLens-1M ($\Delta(Z_u)$=-0.35) datasets, respectively. We can conclude that the improvements of our DENC are statistically significant with all $p < 0.01$. These results indicate the effectiveness of DENC on the task of rating prediction, which has adopted a principled causal inference way to leverage both the inherent factors and auxiliary social network information for improving recommendation performance.
- Among the three kinds of baselines, propensity-based methods serves as the strongest baselines in most cases. This justifies the effectiveness of exploring the missing pattern in rating data by estimating the propensity score, which offers better guidelines to identify the unobserved confounder effect from ratings. However, propensity-based methods perform worse than our DENC, as they ignore the social network information. It is reasonable that exploiting the social network is useful to alleviate the confounder bias to rating outcome. The importance of social networks can be further verified by the fact that most of the social network-based methods consistently outperform PMF on all datasets.
- All baseline methods perform better on Ciao than on Epinions, because Epinions is significantly sparser than Ciao with 0.0140% and 0.0368% density of ratings. Besides this, DENC still achieves satisfying performance on Epinions and its performance is competitive with the counterparts on Ciao. This demonstrates that its exposure model of DENC has an outstanding capability of identifying the missing pattern in rating prediction, in which

[9] According to the statistics, we discover that 90% of users have at least four one-hop neighbours in Ciao and Epinions
[10] For example, $\{user1, user2, user3, user4\}$ are one-hop neighbours of $user5$. If the number of commonly items interacted by $user1$ and $user5$ is 3, then $x = 3$ in the x-axis of Figure 6 is nonzero.

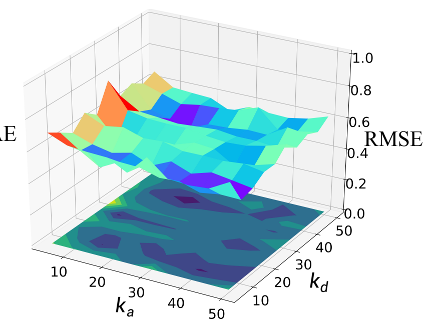
biased user-item pairs in `Epinions` can be captured and then alleviated. In addition, the performance of DENC on three `Movielens-1M` datasets is stable w.r.t. different levels of confounder bias, which verifies the robust debiasing capability of DENC.

5.4 Ablation Study (RQ3)

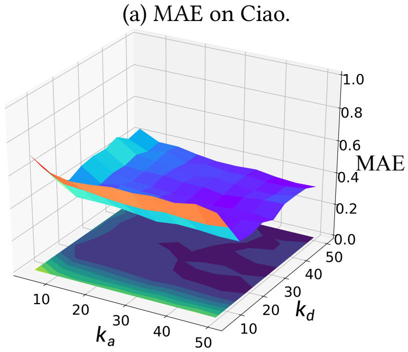
In this section, we conduct experiments to evaluate the parameter sensitivity of our DENC method. We have five important hyperparameters: k_a and k_d that correspond to the embedding size in loss function $\mathcal{L}_a$ and $\mathcal{L}_d$, respectively; λ_a, λ_z and λ_d that correspond to the trade-off parameters for $\mathcal{L}_a$, $\mathcal{L}_z$ and $\mathcal{L}_d$, respectively. Based on the hyperparameter setup in Section 5.1.4, we vary the value of one hyperparameter while keeping the others unchanged.



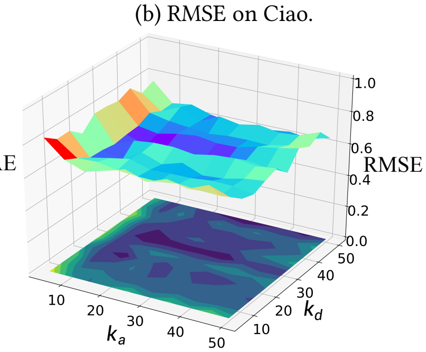
(a) MAE on Ciao.



(b) RMSE on Ciao.



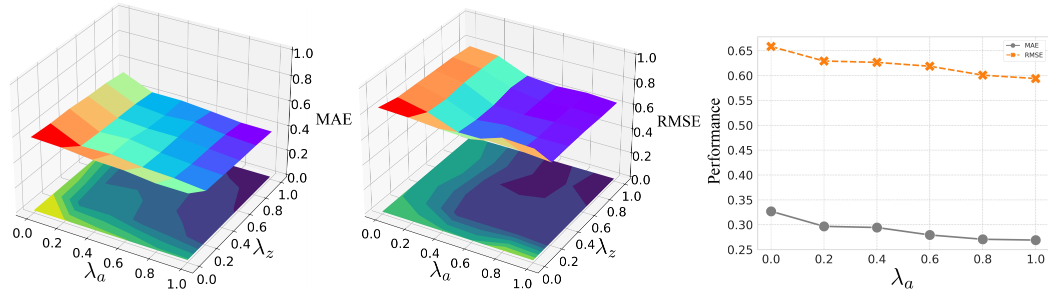
(a) MAE on Epinions.



(b) RMSE on Epinions.

Fig. 7. Our DENC: Parameter sensitivity of k_a and k_d against (a) MAE (b) RMSE on `Ciao` and `Epinions` dataset.

Figure 7 lays out the performance of DENC with different embedding sizes. For both datasets, the performance of our DENC is stable under different hyperparameters k_a and k_d. The performance of DENC increases while the embedding size increase from approximately 0-15 for k_d; afterwards, its performance decreases. It is clear that when the embedding size is set to approximately k_a=45 and k_d=15, our DENC method achieves the optimal performance. Our DENC is less sensitive to the change of k_a than k_d, since MAE/RMSE values change with a obvious concave curve along k_d=0 to 50 in Figure 7, while MAE/RMSE values only change gently with a downward trend along k_a=0 to 50. It is reasonable since k_d controls the embedding size of disentangled user-item representation attained by the deconfounder model, i.e., the inherent factors, while social network embedding size k_a serves as the controller for auxiliary social information, the former can influence the essential user-item interaction while the latter affects the auxiliary information.



(a) MAE on λ_a and λ_z. (b) RMSE on λ_a and λ_z. (c) Performance on λ_d.

Fig. 8. Our DENC's sensitivity to λ_a, λ_z and λ_d on Epinions dataset.

5.5 Sensitivity to Trade-off Parameter

As defined in objective function (15), the three most important trade-off parameters λ_a, λ_z and λ_d balance the contributions of exposure model loss, confounder loss and discrepancy loss, respectively. We evaluate our DENC's sensitivity to these three parameters on Epinions dataset. As shown in Figure 8, the values of trade-off parameters are chosen from $[0, 0.2, 0.4, 0.6, 0.8, 1]$. Figure 8 (a) and (b) present the performance of our model in terms of MAE and RMSE, which are generated by fixing the discrepancy loss weight λ_d and varying the trade-off between the other two parameters. Apparently, our performance is significantly improved compared with the model without λ_a and λ_z, i.e., the errors are reduced. Also, the overall performance on different combinations of hyperparameters of λ_a and λ_z is stable over a large parameter range, which confirms the effectiveness and robustness of debiasing in DENC approach. This conclusion is consistent with our model evaluation results.

Figure 8 (c) indicates that adding the discrepancy loss to account for the selection bias can improve the performance in terms of MAE and RMSE compared with only having the estimation of confounder and exposure assignment. This is the main reason why our method performs well when debiasing rating, but propensity-based method with logistic regression predicting the exposure assignment cannot accurately estimate rating.

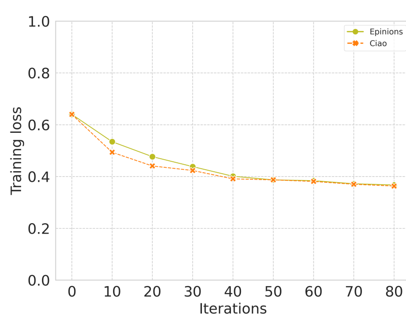


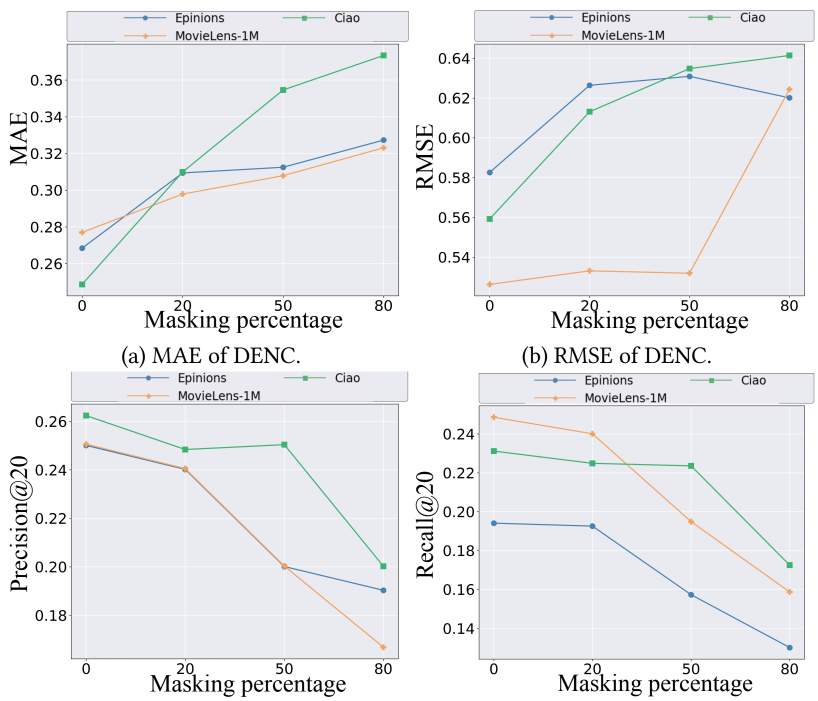
Fig. 9. Our DENC's loss convergence curves on Epinions and Ciao datasets.

5.6 Convergence Analysis

In Figure 9, we plot the convergence of objective loss (15) on the training set of `Epinions` and `Ciao`. One can see that the overall loss decreases as the epoch increases on both datasets. Note that the rates of convergences are different in different dataset. For example, the red curve starts to decrease significantly at epoch 10 and converges at epoch 40. While the green curve first converges a bit more slowly and then become stable at around epoch 40.

5.7 Case Study (RQ4)

We first investigate how the missing social relations affect the performance of DENC. We randomly mask a percentage of social relations to simulate the missing connections in social networks. For `Epinions`, `Ciao` and `MovieLens` dataset, we fix the social network confounder as $\Delta(Z_u) = 0$. Meanwhile, we exploit different percentages of missing social relations including {20%, 50%, 80%}. Note that we do not consider the missing percentage of 100%, i.e., the social network information is completely unobserved. Considering that the social network is viewed as a proxy variable of the confounder, the social network should provide partially known information. Following this guideline, we firstly investigate how the debias capability of our DENC method varies under the different missing percentages. Secondly, we also report the ranking performance of DENC (percentages of missing social relations is set to 0%) under Precision@K and Recall@K with $K = \{10, 15, 20, 25, 30, 35, 40\}$ to evaluate our model thoroughly.



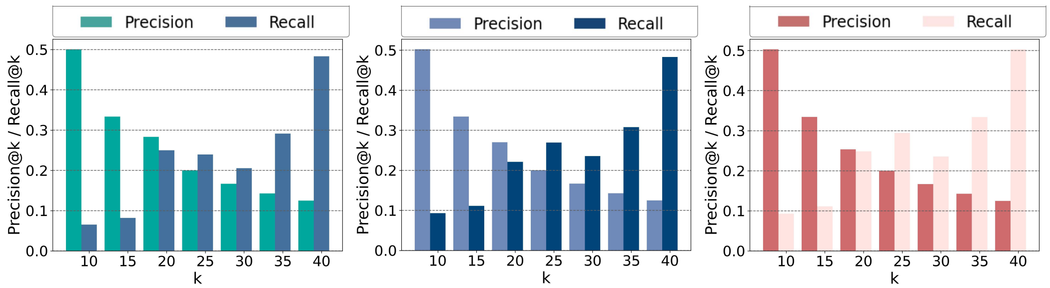
(a) MAE of DENC. (b) RMSE of DENC.

(c) Precision@20 of DENC. (d) Recall@20 of DENC.

Fig. 10. Our DENC: debias performance w.r.t. different missing percentages of social relation.

Figure 10 illustrates our debias performance w.r.t. different missing percentages of social relations on three datasets. As shown in Figure 10, the missing social relations can obviously degrade the debias performance of DENC method.

The performance evaluated by four metrics in Figure 10 consistently degrades when the missing percentage increases from 0% to 80%, which is consistent with the common observation. This indicates that the underlying social network can play a significant role in a recommendation, because it can capture the preference correlations between users and their neighbours.



(a) `Epinions`. (b) `Ciao`. (c) `MovieLens-1M` ($\Delta(Z_u) = 0$).

Fig. 11. Performance of DENC in terms of Precision@K and RecallG@K under difference K

Based on the evaluation on Precision@K and Recall@K, Figure 11 shows that DENC achieves stable performance on Top-K recommendation when K (i.e., the length of ranking list) varies from 10 to 40. Our DENC can recommend more relevant items within top K positions when the ranking list length increases.

6 CONCLUSION AND FUTURE WORK

In this paper, we have researched the missing-not-at-random problem in the recommendation and addressed the confounding bias from a causal perspective. Instead of merely relying on inherent information to account for selection bias, we developed a novel social network embedding based de-bias recommender for unbiased rating, through correcting the confounder effect arising from social networks. We evaluate our DENC method on two real-world and one semi-synthetic recommendation datasets, with extensive experiments demonstrating the superiority of DENC in comparison to state-of-the-arts. In future work, we will explore the effect of different exposure policies on the recommendation system using the intervention analysis in causal inference. In addition, another promising further work is to explore the selection bias arisen from other confounder factors, e.g., user demographic features. This can be explained that a user's nationality affects which restaurant he is more likely to visit (i.e., exposure) and meanwhile affects how he will rate the restaurant (i.e., outcome).

ACKNOWLEDGMENTS

This work is partially supported by the Australian Research Council (ARC) under Grant No. DP200101374, LP170100891, DP220103717 and LE220100078.

REFERENCES

[1] Stephen Bonner and Flavian Vasile. 2018. Causal embeddings for recommendation. In *Proceedings of the 12th ACM Conference on Recommender Systems*. 104–112.

[2] Léon Bottou. 2010. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*. Springer, 177–186.

[3] Denis Charles, Max Chickering, and Patrice Simard. 2013. Counterfactual reasoning and learning systems: The example of computational advertising. *Journal of Machine Learning Research* 14 (2013).

[4] Jiawei Chen, Hande Dong, Xiang Wang, Fuli Feng, Meng Wang, and Xiangnan He. 2020. Bias and debias in recommender system: A survey and future directions. *arXiv preprint arXiv:2010.03240* (2020).

[5] Jiawei Chen, Can Wang, Martin Ester, Qihao Shi, Yan Feng, and Chun Chen. 2018. Social recommendation with missing not at random data. In *2018 IEEE International Conference on Data Mining (ICDM)*. IEEE, 29–38.

[6] Peng Cui, Xiao Wang, Jian Pei, and Wenwu Zhu. 2018. A survey on network embedding. *IEEE Transactions on Knowledge and Data Engineering* 31, 5 (2018), 833–852.

[7] Miroslav Dudík, John Langford, and Lihong Li. 2011. Doubly robust policy evaluation and learning. *arXiv preprint arXiv:1103.4601* (2011).

[8] Tri Dung Duong, Qian Li, and Guandong Xu. 2021. Prototype-based Counterfactual Explanation for Causal Classification. *arXiv preprint arXiv:2105.00703* (2021).

[9] Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. 2019. Graph neural networks for social recommendation. In *The World Wide Web Conference*. 417–426.

[10] Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. 249–256.

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

[12] H Guo, R Tang, Y Ye, Z Li, and X DeepFM He. [n.d.]. a factorization-machine based neural network for CTR prediction. arXiv 2017. *arXiv preprint arXiv:1703.04247* ([n. d.]).

[13] Keith Henderson, Brian Gallagher, Tina Eliassi-Rad, Hanghang Tong, Sugato Basu, Leman Akoglu, Danai Koutra, Christos Faloutsos, and Lei Li. 2012. Rolx: structural role extraction & mining in large graphs. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. 1231–1239.

[14] José Miguel Hernández-Lobato, Neil Houlsby, and Zoubin Ghahramani. 2014. Probabilistic matrix factorization with non-random missing data. In *International Conference on Machine Learning*. PMLR, 1512–1520.

[15] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative filtering for implicit feedback datasets. In *2008 Eighth IEEE International Conference on Data Mining*. Ieee, 263–272.

[16] Mohsen Jamali and Martin Ester. 2010. A matrix factorization technique with trust propagation for recommendation in social networks. In *Proceedings of the fourth ACM conference on Recommender systems*. 135–142.

[17] Yehuda Koren. 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. 426–434.

[18] Yehuda Koren. 2009. Collaborative filtering with temporal dynamics. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. 447–456.

[19] Yehuda Koren and Robert Bell. [n.d.]. Advances in collaborative filtering. In *Recommender systems handbook*. Springer, 77–118.

[20] Piji Li, Zihao Wang, Zhaochun Ren, Lidong Bing, and Wai Lam. 2017. Neural rating regression with abstractive tips generation for recommendation. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*. 345–354.

[21] Wentao Li, Min Gao, Wenge Rong, Junhao Wen, Qingyu Xiong, Ruixi Jia, and Tong Dou. 2017. Social recommendation using Euclidean embedding. In *2017 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 589–595.

[22] Zongxi Li, Haoran Xie, Guandong Xu, Qing Li, Mingming Leng, and Chi Zhou. 2021. Towards purchase prediction: A transaction-based setting and a graph-based method leveraging price information. *Pattern Recognition* 113 (2021), 107824.

[23] Dawen Liang, Laurent Charlin, and David M Blei. 2016. Causal inference for recommendation. In *Causation: Foundation to Application, Workshop at UAI*. AUAI.

[24] Dawen Liang, Laurent Charlin, James McInerney, and David M Blei. 2016. Modeling user exposure in recommendation. In *Proceedings of the 25th international conference on World Wide Web*. 951–961.

[25] Daryl Lim, Julian McAuley, and Gert Lanckriet. 2015. Top-n recommendation with missing implicit feedback. In *Proceedings of the 9th ACM Conference on Recommender Systems*. 309–312.

[26] Guang Ling, Haiqin Yang, Michael R Lyu, and Irwin King. 2012. Response aware model-based collaborative filtering. *arXiv preprint arXiv:1210.4869* (2012).

[27] Roderick JA Little and Donald B Rubin. 2019. *Statistical analysis with missing data*. Vol. 793. John Wiley and Sons.

[28] Dugang Liu, Pengxiang Cheng, Zhenhua Dong, Xiuqiang He, Weike Pan, and Zhong Ming. 2020. A general knowledge distillation framework for counterfactual recommendation via uniform data. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 831–840.

[29] Ling Luo, Haoran Xie, Yanghui Rao, and Fu Lee Wang. 2019. Personalized recommendation by matrix co-factorization with tags and time information. *expert systems with applications* 119 (2019), 311–321.

[30] Hao Ma, Dengyong Zhou, Chao Liu, Michael R Lyu, and Irwin King. 2011. Recommender systems with social regularization. In *Proceedings of the fourth ACM international conference on Web search and data mining*. 287–296.

[31] Benjamin M Marlin and Richard S Zemel. 2009. Collaborative prediction and ranking with non-random missing data. In *Proceedings of the third ACM conference on Recommender systems*. 5–12.

[32] Andriy Mnih and Russ R Salakhutdinov. 2008. Probabilistic matrix factorization. In *Advances in neural information processing systems*. 1257–1264.

[33] Alfred Müller. 1997. Integral probability metrics and their generating classes of functions. *Advances in Applied Probability* (1997), 429–443.

[34] Shohei Ohsawa, Yachiko Obara, and Takayuki Osogami. 2016. Gated probabilistic matrix factorization: learning users' attention from missing values. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*. 1888–1894.

[35] Rong Pan, Yunhong Zhou, Bin Cao, Nathan N Liu, Rajan Lukose, Martin Scholz, and Qiang Yang. 2008. One-class collaborative filtering. In *2008 Eighth IEEE International Conference on Data Mining*. IEEE, 502–511.

[36] Judea Pearl. 2009. *Causality*. Cambridge university press.

[37] Steffen Rendle and Lars Schmidt-Thieme. 2008. Online-updating regularized kernel matrix factorization models for large-scale recommender systems. In *Proceedings of the 2008 ACM conference on Recommender systems*. 251–258.

[38] Donald B Rubin. 1974. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of educational Psychology* 66, 5 (1974), 688.

[39] Donald B Rubin. 1976. Inference and missing data. *Biometrika* 63, 3 (1976), 581–592.

[40] Yuta Saito. 2020. Asymmetric Tri-training for Debiasing Missing-Not-At-Random Explicit Feedback. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 309–318.

[41] Yuta Saito, Suguru Yaginuma, Yuta Nishino, Hayato Sakata, and Kazuhide Nakata. 2020. Unbiased Recommender Learning from Missing-Not-At-Random Implicit Feedback. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. 501–509.

[42] Tobias Schnabel, Adith Swaminathan, Ashudeep Singh, Navin Chandak, and Thorsten Joachims. 2016. Recommendations as treatments: Debiasing learning and evaluation. In *international conference on machine learning*. PMLR, 1670–1679.

[43] Uri Shalit, Fredrik D Johansson, and David Sontag. 2017. Estimating individual treatment effect: generalization bounds and algorithms. In *International Conference on Machine Learning*. PMLR, 3076–3085.

[44] Aude Sportisse, Claire Boyer, and Julie Josse. 2020. Imputation and low-rank estimation with Missing Not At Random data. *Statistics and Computing* 30, 6 (2020), 1629–1643.

[45] Harald Steck. 2010. Training and testing of recommender systems on data missing not at random. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. 713–722.

[46] Harald Steck. 2011. Item popularity and recommendation accuracy. In *Proceedings of the fifth ACM conference on Recommender systems*. 125–132.

[47] Harald Steck. 2013. Evaluation of recommendations: rating-prediction and ranking. In *Proceedings of the 7th ACM conference on Recommender systems*. 213–220.

[48] Adith Swaminathan and Thorsten Joachims. 2015. The self-normalized estimator for counterfactual learning. *advances in neural information processing systems* 28 (2015).

[49] J. Tang, H. Gao, and H. Liu. 2012. mTrust: Discerning multi-faceted trust in a connected world. In *Proceedings of the fifth ACM international conference on Web search and data mining*. ACM, 93–102.

[50] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. Line: Large-scale information network embedding. In *Proceedings of the 24th international conference on world wide web*. 1067–1077.

[51] Daixin Wang, Peng Cui, and Wenwu Zhu. 2016. Structural deep network embedding. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. 1225–1234.

[52] Xiaojie Wang, Rui Zhang, Yu Sun, and Jianzhong Qi. 2019. Doubly robust joint learning for recommendation on data missing not at random. In *International Conference on Machine Learning*. 6638–6647.

[53] Yixin Wang, Dawen Liang, Laurent Charlin, and David M Blei. 2018. The deconfounded recommender: A causal inference approach to recommendation. *arXiv preprint arXiv:1808.06581* (2018).

[54] Guandong Xu, Tri Dung Duong, Qian Li, Shaowu Liu, and Xianzhi Wang. 2020. Causality Learning: A New Perspective for Interpretable Machine Learning. *arXiv preprint arXiv:2006.16789* (2020).

[55] Haiqin Yang, Guang Ling, Yuxin Su, Michael R Lyu, and Irwin King. 2015. Boosting response aware model-based collaborative filtering. *IEEE Transactions on Knowledge and Data Engineering* 27, 8 (2015), 2064–2077.

[56] Longqi Yang, Yin Cui, Yuan Xuan, Chenyang Wang, Serge Belongie, and Deborah Estrin. 2018. Unbiased offline recommender evaluation for missing-not-at-random implicit feedback. In *Proceedings of the 12th ACM Conference on Recommender Systems*. 279–287.

[57] Baolin Yi, Xiaoxuan Shen, Hai Liu, Zhaoli Zhang, Wei Zhang, Sannyuya Liu, and Naixue Xiong. 2019. Deep matrix factorization with implicit feedback embedding for recommendation system. *IEEE Transactions on Industrial Informatics* 15, 8 (2019), 4591–4601.

A APPENDIX

A.1 Datasets

The statistics of baseline datasets are given in Table 1. In `Epinions` and `Ciao`, the rating values are integers from 1 (like least) to 5 (like most). Since observed ratings are very sparse (rating density 0.0140% for `Epinions` and 0.0368% for `Ciao`), thus the rating prediction on these two datasets is challenging.

In addition, we also simulate a semi-synthetic dataset based on `MovieLens`. It is well-known that `MovieLens` is a benchmark dataset of user-movie ratings without social network information. For `MovieLens-1M`, we first need to construct a social network G by placing an edge between each pair of users independently with a probability 0.5 depending on whether the nodes belong to G. Recall that the social network is viewed as the confounder (common cause) which affects both exposure variables and ratings. We generate the exposure assignment by the confounder Z_u of three levels $\Delta(Z_u) \in \{-0.35, 0, 0.35\}$. Then, the exposure a_{ui} and rating outcome y_{ui} are simulated as follows.

$$
\begin{aligned}
& a_{ui} \sim \text{Bern}\left(\Delta\left(Z_u\right)\right) & \\
& y_{ui} = a_{ui} \cdot (y_{ui}^{\text{mov}} + \beta_u \Delta\left(Z_u\right) + \varepsilon) & \varepsilon \sim N(0,1), \quad u \in G \\
& y_{ui} = y_{ui}^{\text{mov}} & u \notin G
\end{aligned}
$$

where y_{ui}^{mov} is the original rating in `MovieLens` and the parameter β_u controls the amount of social network confounder. The exposure a_{ui} indicating whether item i being exposed to user u is given by a Bernoulli distribution parameterized by the confounder Z_u. The non-zero a_{ui} is used to simulate the semi-synthetic rating y_{ui} by the second equation. The third equation indicates that the ratings of user will keep unchanged if s/he is not connected by G.

A.2 Baselines

We compare our DENC against three groups of methods, covering matrix factorization method, social network-based method, and propensity-based method. For each group, we select its representative baselines with details as follows.

- **PMF** [32]: The method utilizes user-item rating matrix and models latent factors of users and items by Gaussian distributions;
- **NRT** [20]: A deep-learning method that adopts multi-layer perceptron network to model user-item interactions for rating predictions.
- **SocialMF** [16]: It considers the social information by adding the propagation of social relation into the matrix factorization model.
- **SoReg** [30]: It models social information as regularization terms to constrain the Matrix Factorization framework.
- **SREE** [21]: It models users and items embeddings into a Euclidean space as well as users' social relations.
- **GraphRec** [9]: This is a state-of-the-art social recommender that models social information with Graph Neural Network, it organizes user behaviors as a user-item interaction graph.
- **DeepFM** [12]+: DeepFM is a state-of-the-art recommender that integrates Deep Neural Networks and Factorization Machine (FM). To incorporate the social information into DeepFM, we change the output of FM in DeepFM+ to the linear combination of the original FM function in [12] and the pre-trained *node2vec* user embeddings. We also change the task of DeepMF from click-through rate (CTR) to rating prediction.
- **CausE** [1]: It firstly fits exposure variable embedding with Poisson factorization, then integrates the embedding into PMF for rating prediction.

Table 3. Experimental results of DENC-α and DENC-β.

Dataset	Models	MAE	RMSE
Epinions	DENC-α	0.4725	0.8234
	DENC-β	0.4294	0.7876
	DENC	0.2684	0.5826
Ciao	DENC-α	0.4380	0.8026
	DENC-β	0.3870	0.6723
	DENC	0.2487	0.5592

- **D-WMF** [53]: A propensity-based model which uses Poisson Factorization to infer latent confounders then augments Weighted Matrix Factorization to correct for potential confounding bias.

A.3 Model Variants Configuration

To get a better understanding of our DENC method, we further evaluate the key components of DENC including *Exposure model* and *Social network confounder*. We evaluate the performance of DENC on the condition that if a specific component is removed, and then compare the performance of the intact DENC method. In the following, we define two variants of DENC as (1) DENC-α that removes *Exposure model*; (2) DENC-β that removes *Social network confounder*. Note that we do not consider the evaluation of removing *Deconfounder* in DENC, since *Deconfounder* models the inherent factors of user-item information, removing user-item information in a recommender can result in poor performance. We record evaluation results in Table 3 and have the following findings:

- By comparing DENC with DENC-α, we find that *Exposure model* is important for capturing missing patterns and thus boosting the recommendation performance. Removing *Exposure model* can lead a drastic degradation of MAE/RMSE by 20.41%/24.08% on Epinions and 18.93%/24.34% on Ciao, respectively.
- We observe that without *Social network confounder*, the performance of DENC-β is deteriorated significantly, with the degradation of MAE/RMSE by 16.10%/20.50% on Epinions and 13.83%/11.31% on Ciao, respectively.
- *Exposure model* has a greater impact on DENC compared with *Social network confounder*. It is reasonable since *Exposure model* simulates the missing patterns, then *Social network confounder* can consequently debias the potential confounding bias under the guidance of missing patterns.

A.4 Investigation on Different Network Embedding Methods

We construct network embedding with node2vec [11] that has the capacity of learning richer representations by adding flexibility in exploring neighborhoods of nodes. Besides, by adjusting the weight of the random walk between breadth-first and depth-first sampling, embeddings generated by node2vec can balance the trade-off between homophily and structural equivalence [13], both of which are essential feature expressions in recommendation systems. The key characteristic of node2vec is its scalability and efficiency as it scales to networks of millions of nodes.

By comparison, we further investigate how different network embedding methods impact the performance of DENC, i.e., LINE [50], SDNE [51].

- **LINE** [50] preserves both first-order and second-order proximities, it suits arbitrary types of information networks and can easily scale to millions of nodes.
- **SDNE** [51] is a Deep Learning-based network embedding method, like LINE, it exploits the first-order and second-order proximity jointly to preserve the network structure.

We train the three embedding methods with embedding size d=10 while the batch size and epochs are set to 1024 and 50, respectively. The experimental results are given in Table 4.

Table 4. Experimental results of DENC under node2vec, LINE, SDNE.

Dataset	Embedding	MAE	RMSE	Precision@20	Recall@20
Epinions	node2vec	0.2684	0.5826	0.2832	0.2501
	LINE	0.4241	0.6307	0.1736	0.1534
	SDNE	0.4021	0.6137	0.1928	0.1837
Ciao	node2vec	0.2487	0.5592	0.2703	0.2212
	LINE	0.5218	0.7605	0.1504	0.1209
	SDNE	0.4538	0.6274	0.2082	0.1594

The results show that under the same experimental settings, DENC performs worse with embeddings trained by LINE and SDNE compared with node2vec on both datasets. Although LINE considers the higher-order proximity, unlike node2vec, it still cannot balance the representation between homophily and structural equivalence [13], in which connectivity information and network structure information can be captured jointly. The results show that our DENC benefits more from the balanced representation that can learn both the connectivity information and network structure information. Based on higher-order proximity, SDNE develops a deep-learning representation method. However, compared with node2vec, SDNE suffers from higher time complexity. The deep architecture of SDNE framework mainly causes the high time complexity of SDNE, the input vector dimension can expand to millions for the auto-encoder in SDNE [6]. Thus, we consider it reasonable that our DENC with SDNE embedding cannot outperform the counterpart with node2vec embedding under the same training epochs, since it requires more iterations for SDNE to get finer representation.