CausalRD: A Causal View of Rumor Detection via Eliminating Popularity and Conformity Biases

Weifeng Zhang*, Ting Zhong*, Ce Li*, Kunpeng Zhang†, and Fan Zhou*§
*University of Electronic Science and Technology of China, Chengdu, China
†Robert H. Smith School of Business, University of Maryland, College Park, USA
§Corresponding author: fan.zhou@uestc.edu.cn

***Abstract*—A large amount of disinformation on social media has penetrated into various domains and brought significant adverse effects. Understanding their roots and propagation becomes desired in both academia and industry. Prior literature has developed many algorithms to identify this disinformation, particularly rumor detection. Some leverage the power of deep learning and have achieved promising results. However, they all focused on building predictive models and improving forecast accuracy, while two important factors - popularity and conformity biases - that play critical roles in rumor spreading behaviors are usually neglected.**

To overcome such an issue and alleviate the bias from these two factors, we propose a rumor detection framework to learn debiased user preference and effective event representation in a causal view. We first build a graph to capture causal relationships among users, events, and their interactions. Then we apply the causal intervention to eliminate popularity and conformity biases and obtain debiased user preference representation. Finally, we leverage the power of graph neural networks to aggregate learned user representation and event features for the final event type classification. Empirical experiments conducted on two real-world datasets demonstrate the effectiveness of our proposed approach compared to several cutting-edge baselines.

***Index Terms*—Rumor detection, Causal Inference, Graph neural networks**

I. Introduction

The increasing use and conveniences of social media platforms, such as Twitter, Weibo, and Facebook, impulse the collection of information. More and more people are involved in online social networks (OSNs), which allows for social connectedness in a time of social distancing. For example, Twitter has about 396 million active global users, and according to the 2021 Pew Research Center Survey [1], around half of U.S. users (46%) visit the site daily. This provides an opportunity for the exposure of all kinds of content, which also contributes to the rapid diffusion of unconfirmed information, e.g., rumors. Social media platforms and the Internet are argued as a fertile ground for the propagation of this information [2]. Recent years have witnessed several risky instances, e.g., financial market security [3], presidential election [4] and COVID-19 pandemic[1]. This prolonged false rumor crisis has intensified the need to understand rumors in a more nuanced way.

In the literature [5], [6], rumors are defined as items of information that are unverified at the time of posting. Researchers observe that people are more likely to share false information, which is more novel than true news [7], and tend to stop spreading a rumor if it is revealed as false [8]. Thus, detecting rumors and strangling their dissemination in an early stage can effectively mitigate possible harmful effects. Rumor detection, which aims to identify the credibility of the information from social media platforms, has attracted a great deal of attention [9], [10].

Issues and Challenges. Prior sociological and psychological studies have shown the correlation between user preference and online media diets [11]. For example, filter bubbles [12] would make people become isolated intellectually and tend to spread disinformation. Inspired by this phenomenon, a number of disinformation detection methods based on implicit or explicit user preference have been proposed and have shown promising results in terms of detecting fake news [13] and sarcasm [14]. Existing works generally take historical posts of a user as a proxy and leverage this information to represent the user's preference. However, we notice that two factors are largely ignored by these approaches, which makes the learned user preference inconsistent with reality.

One factor is the *popularity bias* aroused by the recommendation strategy of social media platforms, which we argue is also present at a rumor detection task. Specifically, events on social media are not exposed to users at an equal frequency. Popular events have a higher probability of being recommended to users [15]. Another factor is the *conformity bias*. Research on human behavior [16] and social psychology [17] have shown that conformity is a ubiquitous phenomenon when spreading disinformation in social media. Users might engage in events with high popularity, which might not match their media diets, though.

Due to these two biases, it is inappropriate to obtain users' preference directly from their historical behaviors or posts. For example, during the U.S. election, some social media users who do not care about politics might join in related discussions because they are heavily recommended and exposed to such events (popularity bias), and accordingly, many users are involved in these events (conformity bias). Since the effects of these two biases vary with events, we consider extracting the user preference without being affected by popularity and conformity biases, namely *debiased user preference*.

This goal motivates us to answer a counterfactual question

[1]https://www.bbc.com/news/world-53755067

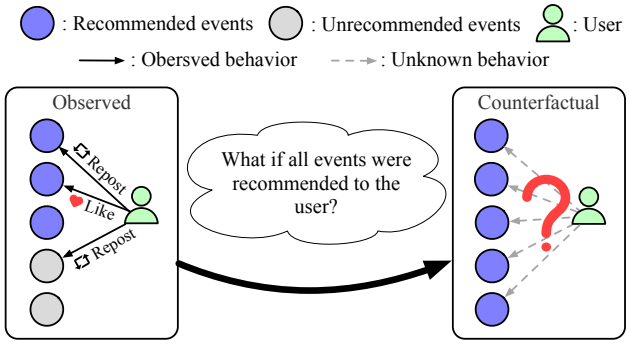


Fig. 1: The illustration of the counterfactual question about popularity and conformity biases.

as shown in Fig. 1: if all events were equally popular and exposed to users with the same probabilities, which events would they choose to participate in (e.g., reply or retweet)? In this way, user behaviors are not subjected to popularity and conformity biases. That is to say, answering this question means the debiased user preference needs to be obtained.

Our Approach. To address the aforementioned challenges and answer the essential counterfactual question, we first analyze the causality of rumor propagation and construct a novel causal graph for eliminating the popularity and conformity biases embedded in the user-event interactions. Next, we propose our CausalRD framework with two phases: the debias phase and the inference phase. Specifically, in the debias phase, we define and refine a causal graph with proper interventions and learn debiased user preference along with an effective positive and negative sampling strategy. Then, the obtained user representation and the post-level textual features are aggregated respectively with the propagation graph structure by two GNN encoders for rumor inference. The main contributions of this work are as follows:

- We abstract a causal graph of user-event interactions and analyze the popularity and conformity biases that prevent us from obtaining inherent user preference through historical data. To the best of our knowledge, this is the first study of rumor detection based on debiased user preference in a causal view.
- We introduce causal inference into the rumor detection task to eliminate popularity and conformity biases and propose the CausalRD model, which leverages debiased user preference embeddings for detecting rumors.
- We conduct extensive experiments on two real-world datasets, demonstrating the effectiveness of our approach compared to state-of-the-art benchmark baselines. The results have shown that debiased user preference can improve the accuracy of rumor detection.

II. Related Work

A. Rumor Detection

Terms like rumor, misinformation, disinformation, and fake news are closely related and are often mixed up. However, one major difference among them is the originator's intention of posting events. Disinformation or fake news is intentionally fabricated, while rumors are unverified information propagated to mislead the public without intention. We refer readers to elaborated surveys [6], [18], [19] for deep characterization about the rumor definition and its detection systems.

Rumor detection leverages various features, mainly categorized as: temporal, structural, and linguistic. Temporal features aim to capture how rumors spread over time, while structural features learn the connectivity among users who post the rumor. Finally, linguistic features represent the content information for posts and responses and can be embedded as vectors by pre-training language modeling, e.g., BERT [20].

Feature-based methods [21], [22] study the factors affecting social information diffusion [23], including text content, user comments, and temporal-structural features. Researchers manually collect and select features to train supervised classifiers, e.g., decision tree, random forest, and SVM. However, this usually requires extensive professional domain knowledge to analyze and find out the most significant and relevant features. Thus, they are also criticized for the lack of generalizability, i.e., hard to be transferred to new domains.

Inspired by the power of Recurrent and Convolutional neural networks, e.g., RNN [24] and CNN [25], for modeling the sequential data, recent studies adopt deep learning modules to extract temporal-structural features for an effective rumor representation. Since Ma et al. [26] first proposed an RNN method to learn both the temporal and textual representations, various RNN-based architectures have been studied for rumor detection. Among them, Recursive Neural Networks (RvNN) [27] refines a *bottom-up* and a *top-down* neural network which are capable of exploiting tree-structured rumor propagation. Subsequently, Yu et at. [28] introduced CNN to flexibly extract key features scattered among one input sequence propagation of signals. More recently, there has been an emergence of graph representation learning approaches [29], [30], [31] for aggregating the linguistic feature with the propagation structure. The nodes in a directed event graph denote users who participate in the event, and edges between nodes depict paths of information diffusion. By leveraging powerful GNNs [32], a high-level event-graph representation can be preserved in the embedding space.

B. Machine Learning in Causality

Our work is also related to causality with machine learning. In prior literature, there exist many approaches to help researchers understand the causal effect of a given treatment (e.g., an event), such as randomized control experiments and natural experiments (e.g., an exogenous shock) [33], [34]. One major challenge, however, is the cost of experiments, which tend to be expensive and occasionally harmful to participants or platforms. Additionally, randomization is impossible in many situations due to unethical issues. Therefore, researchers have been seeking to develop new techniques to estimate causal effects using the collected observational data, primarily including regression, matching, and weighting.

Matching and weighting methods are then developed to tackle endogeneity issues in observational studies by balancing covariates in treatment and control groups. Current weighting methods for treatment effect estimation are often built upon the idea of propensity scores or covariate balance [35], [36]. To estimate the propensity score, researchers often assume a simply specified functional form to model the relationship of covariates to the treatment assignment. However, this simple assumption can cause an issue of model misspecification [37] which in turn leads to biased treatment effect estimation. Therefore, many researchers have proposed using machine learning methods such as Lasso, boosting regression, bagged CART, and random forest in order to improve propensity score estimation by modeling the complicated non-linearity [38].

Inspired by the successful results of machine learning, many research combining machine learning and causal inference are proposed. Guo et al. [39] utilize graph neural networks to learn the individual causal effects from networked observational data. Cheng et al. [40] analyze the causality between user attributes and user behavior in fake news dissemination. In recommendation system, causal inference can be leveraged to mitigate the clickbait issue [41] and popularity bias [42]. In this work, we leverage the idea of causal inference to eliminate the biases for better machine learning performance in the task of rumor detection, specifically focusing on how to leverage causal intervention to alleviate popularity and conformity biases and how to obtain debiased user preference.

III. Preliminaries

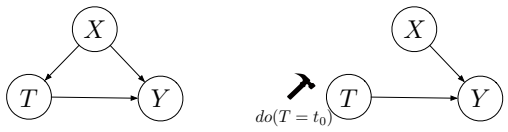
We now define notations, formulate the problem of rumor detection, and introduce several primary concepts for the causal graph.

A. Problem Statement & Notation

In this study, lowercase characters (e.g., x) denote scalars, uppercase characters (e.g., X) denote random variables, lowercase bold characters (e.g., $\mathbf{x}$) denote vectors, uppercase bold characters (e.g., $\mathbf{X}$) denote matrices, and uppercase calligraphic characters (e.g., $\mathcal{C}$) denote entities such as events, posts, and users, unless specified otherwise.

We define $\mathcal{D} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_{|\mathcal{D}|}\}$ as a rumor detection dataset, where $\mathcal{C}_i = \{\mathcal{P}_i^0, \mathcal{P}_i^1, \mathcal{P}_i^2, \ldots \mathcal{P}_i^{m_i}, \mathcal{G}_i\}$ is the i-th event which consists of a source post $\mathcal{P}_i^0$, responsive posts $\mathcal{P}_i^j, j \geq 1$ and a propagation tree structure $\mathcal{G}_i$, note that the number of all posts $m_i + 1$ varies with i. $\mathcal{G}_i$ can also be defined as a graph $\mathcal{G}_i = (\mathcal{V}_i, \mathcal{E}_i)$, which regards the source post and its corresponding responsive posts as nodes, i.e., $\mathcal{V}_i = \{\mathcal{P}_i^0, \mathcal{P}_i^1, \mathcal{P}_i^2, \ldots \mathcal{P}_i^{m_i}\}$, and $\mathcal{E}_i$ represents the reply or reposting relationships among the posts of $\mathcal{C}_i$, e.g., for $\mathcal{P}_i^0, \mathcal{P}_i^1 \in \mathcal{V}_i$, $\mathcal{P}_i^0 \rightarrow \mathcal{P}_i^1$ exists if $\mathcal{P}_i^1$ responses to $\mathcal{P}_i^0$ [43], [44]. Additionally, each post $\mathcal{P}_i^*$ is associated with the content $\mathcal{T}_i^*$ and a promulgator (user) $\mathcal{U}_{\mathcal{P}_i^*}$. Note that some users have participated in more than one events, and even within the same event one user could have edited more than one posts.

Each event $\mathcal{C}_i$ is associated with one of four finer-grained classes: Non-rumor, False Rumor, True Rumor and Unverified Rumor which are denoted by NR, FR, TR and UR, respectively [44], [45]. The rumor detection task is to learn a classifier f in a supervised manner, predicting the label of the event, i.e., $f : \mathcal{C}_i \rightarrow y_i$, where $y_i \in \{\text{NR,FR,TR,UR}\}$.



(a) Without intervention. (b) With intervention.

Fig. 2: An illustration example of the causal graph. X, Y, and T usually denote covariates, outcome, and treatment, respectively.

B. Causal Graph

The Bayesian network is a probabilistic model which leverages the graph to represent random variables and conditional independence by vertices and edges, respectively. It is associated with a directed acyclic graph and its joint probability density function could be factored into a product of individual probabilities conditional on their parent variables:

$$P(X_1, X_2, \ldots, X_n) = \prod_{i=1}^{n} P(X_i | Pa_{\mathcal{G}}(X_i)), \tag{1}$$

where $Pa_{\mathcal{G}}(X_i)$ denotes the parents of node X_i, i.e., the nodes pointing to X_i directly by a direct edge. However, the Bayesian network does not imply causation, since "no causation without manipulation" [46], the manipulation or intervention in causality refers to fixing the value of a variable and observing the result. Based on Bayesian network, causal graph is proposed to represent causality [47]. The intervention in a causal graph has two equivalent forms: (1) mathematically, *do-operation* $\text{do}(X = x)$ denotes that the variable X is manipulated and fixed to the value of x; (2) graphically, the aforementioned intervention means removing all edges pointing to X and setting X to x.

Fig. 2a depicts the causal graph without any interventions, whose joint probability density function is:

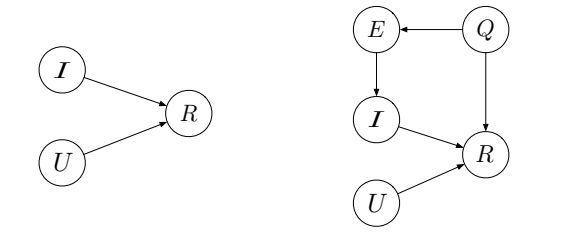
$$P(X, T, Y) = P(X)P(T|X)P(Y|X, T). \tag{2}$$

And Fig. 2b illustrates the causal graph with intervention denoted by $\text{do}(T = t_0)$. Thus the probability function is updated as:

$$P(X, \text{do}(T = t_0), Y) = P(X)P(Y|X, T = t_0), \tag{3}$$

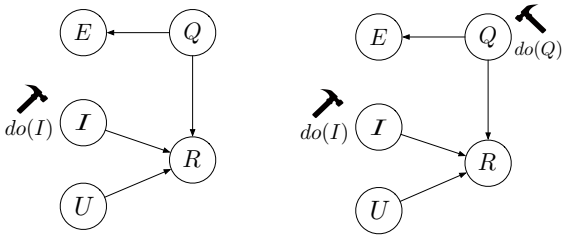
what shall be emphasized is that

$$\begin{aligned} P(X,&T = t_0, Y) \\ &= P(X)P(T = t_0|X)P(Y|X, T = t_0) \\ &\neq P(X, \text{do}(T = t_0), Y). \end{aligned} \tag{4}$$



(a) The causal graph of conventional methods.

(b) The causal graph of this work.



(c) The causal graph without popularity bias.

(d) The causal graph without popularity and conformity bias.

Fig. 3: The causal graphs for conventional rumor detection and our proposed model.

IV. Methodology

In this section, we first analyze the causal graph of user-event interaction in two views: a conventional view, and the view designed for user-event interaction. We then propose our CausalRD framework that consists of two components: debias phase and inference phase. The overview architecture of CausalRD is shown in Fig.4.

A. Causal Graph of Event Propagation

Informed by the power of causal graphs which can represent causality, we construct a causal graph which characterizes user-event interactions on social media. We take Twitter, the most popular microblogging and social networking service, as an example to demonstrate the causal relations, and the data generating process of user-event interactions. In this section, we intend to answer the following two research questions:
(1) What are the differences between the conventional and our proposed causal graphs?
(2) What makes observational data incapable of representing user preference accurately?

For the first question, existing misinformation detection efforts [13], [14] tend to utilize user features (e.g., user profile and historical posts) besides textual features to gain promising performance. Nevertheless, we argue that these studies are based on a problematic assumption, i.e., all events that one user has participated in reflect his/her authentic preference. They directly learn the preference of one user from his/her historical behaviors. As demonstrated in Fig. 3a, conventional methods consider three random variables I, U, R, which denote the tweet related to the event, user preference, and the probability of retweet/reply, respectively. And they assume R is only influenced by I and U.

Back to the aforementioned popularity and conformity biases, we construct a real-world causal graph for this task as shown in Fig. 3b, where I, U, R denote the same. Additionally, we take the popularity of a tweet (denoted as Q) and the exposure probability (denoted as E) into consideration. The direct path $Q \rightarrow R$ indicates the conformity effect, i.e., users prefer to engage in and share the tweets with high popularity. The indirect path $Q \rightarrow E \rightarrow I$ reflects the popularity bias which indicates that more popular tweets tend to be exposed to more people because of the simple and widely used recommendation policy by social media. And $\{I, U\} \rightarrow R$ denote the same causal mechanism embedded in Fig. 3a.

For the second question, user preference embeddings are obtained from U, however, we notice that Q affects both I and R, leading to an issue of *selection bias* [48] and Missing Not At Random (MNAR) [49]. Specifically, popular events have higher probability being exposed to users, and users tend to participate in popular events, which makes it hard to extract user preference from users' behaviors. For example, suppose we have observed user $\mathcal{U}$ who has engaged in 7 rumors with high popularity and 3 non-rumors with low popularity. We cannot easily assume $\mathcal{U}$ prefers to forward rumors because of the popularity and conformity biases. If all events are exposed to $\mathcal{U}$ and they are equally popular, the behavior of $\mathcal{U}$ should reflect his/her real preference on the event. Thus, to extract debiased user preference among user-event interactions, we propose to add a debias phase based on causal intervention.

B. Debias Phase

Causal Intervention. Based on our proposed causal graph, we are able to intervene data with a causality method: *do-operation*. Since popularity bias arising from path $Q \rightarrow E \rightarrow I$ and node E is not observed, we cut off the path between E and I as depicted in Fig. 3c, and this is denoted by $P(R|\text{do}(I), U)$. In this way, the popularity of a tweet cannot cause an exposure probability of a tweet, in other words, tweets are exposed to users randomly, indicating the popularity bias has been eliminated. Note that $P(R|\text{do}(I), U) \neq P(R|I, U)$ mentioned in Section 3, this makes impossible to evaluate $P(R|\text{do}(I), U)$ from observational data $\{I, U, R\}$ directly. Thanks to the backdoor criterion[47] in causal inference, we are able to deal with intervention under certain conditions.

We intervene the model input based on the back-door adjustment, and combining Bayes' rule on the new casual graph, we have:

$$P(R|U, \text{do}(I)) = \sum_{q \in Q} P(R|U, I, q)P(q), \tag{5}$$

where $P(q)$ is the prior of popularity, the equation holds because Q is not the descendant of I, and the path $I \leftarrow E \leftarrow Q \rightarrow R$ are blocked by Q, that is, Q satisfies the backdoor criterion relative to (I, R).

Theorem 1. *Backdoor Criterion. A set of variables Z satisfies the backdoor criterion relative to order pair(X, Y) iff both two conditions hold: No variables in Z are descendants of X*

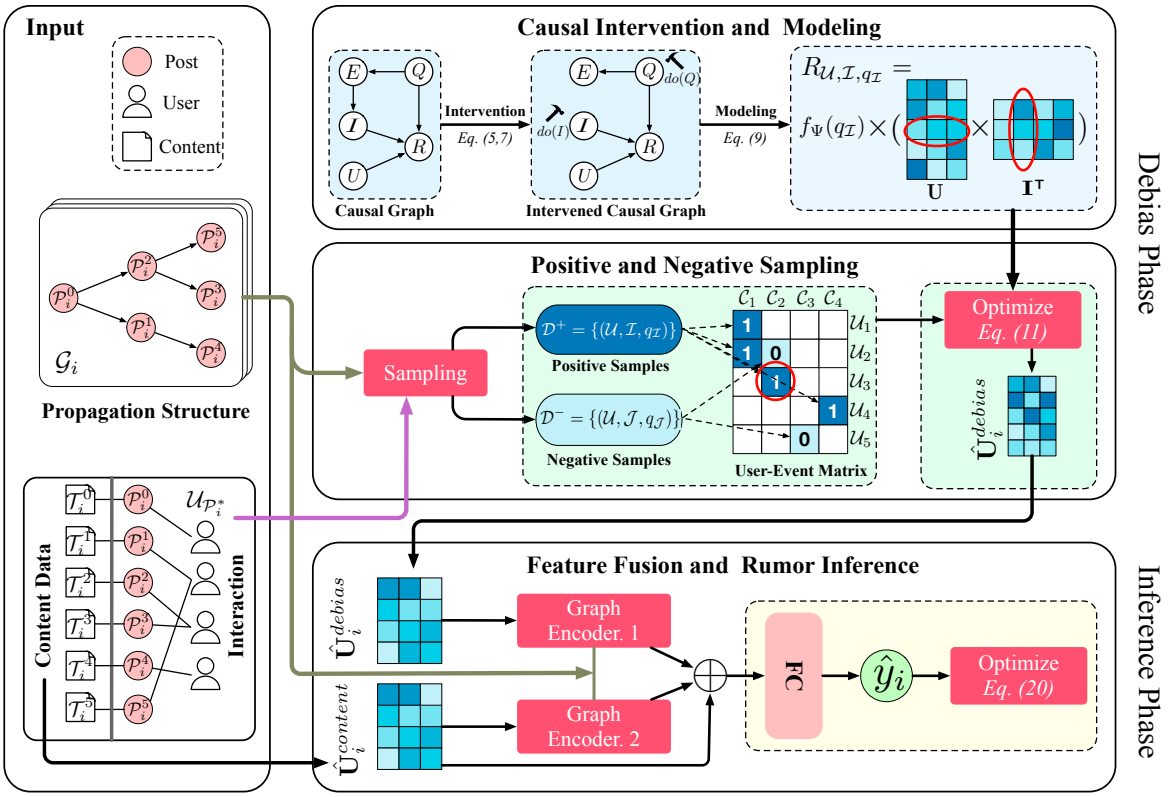


Fig. 4: The overview architecture of CausalRD that contains two phases. (1) The debias phase first models the event propagation with causal intervention. It obtains debiased user preference embeddings $\hat{\mathbf{U}}^{debias}$ with combining the structural negative sampling; (2) In the inference phase, the debiased user preference embeddings $\hat{\mathbf{U}}_i^{debias}$ and content embeddings $\hat{\mathbf{U}}_i^{content}$ of each event are leveraged for identifying the veracity of the event.

,and Z blocks every path from X to Y that has an arrow to X. If the backdoor criterion is satisfied, then the causal effect of X on Y is given by:

$$P(Y|\text{do}(X)) = \sum_z P(Y|X, Z=z)P(Z=z). \tag{6}$$

Eq. 5 demonstrates the event propagation whose popularity bias has been removed, yet there exists conformity bias mentioned before because of $Q \rightarrow R$. In order to eliminate the conformity bias for user preference, we need to make sure the popularity Q of each event is equal. Thus we set Q as a constant ζ as shown in Fig. 3d, and Eq. 5 can be rewritten as:

$$P(R|U, \text{do}(I), \text{do}(Q=\zeta)) = P(R|U, I, Q=\zeta), \tag{7}$$

since $P(Q) = 1$ only when $Q = \zeta$, $P(Q) = 0$ otherwise.

Now our target is learning the conditional distribution $P(R|Q, U, I)$, note that we need to obtain embeddings representing the user's preference which is similar to recommendation system, thus we choose matrix factorization to model the distribution, besides,

$$P(R=1|U, I, Q) = f_\Psi(Q) * f_\Omega(U, I), \tag{8}$$

where $f_\Psi(*)$ is an non-linear function parameterized by Ψ, and $f_\Omega(*)$ is the function of matrix factorization. For a specific user $\mathcal{U}$, tweet $\mathcal{I}$ and its popularity $q_\mathcal{I}$,

$$P(R=1|\mathcal{U}, \mathcal{I}, q_\mathcal{I}) = f_\Psi(q_\mathcal{I}) * (\mathbf{U}\mathbf{I}^\intercal)_{\mathcal{U},\mathcal{I}}, \tag{9}$$

where $\mathbf{U} \in \mathbb{R}^{n_u \times h}$ and $\mathbf{I} \in \mathbb{R}^{n_i \times h}$ are user preference embedding matrix and event topic embedding matrix respectively, n_u and n_i denote the total number of users and tweets, we compute the popularity $q_\mathcal{I}$ by adding the number of retweets and comments when user $\mathcal{U}$ participates in the event $\mathcal{I}$. $(\mathbf{U}\mathbf{I}^\intercal)_{\mathcal{U},\mathcal{I}}$ is the matching score between user $\mathcal{U}$ and tweet $\mathcal{I}$, the larger the value, the more user $\mathcal{U}$ prefer the event $\mathcal{I}$. When $q_\mathcal{I}$ is set as a constant, $P(R=1|\mathcal{U}, \mathcal{I}, q_\mathcal{I}) \propto (\mathbf{U}\mathbf{I}^\intercal)_{\mathcal{U},\mathcal{I}}$, thereby $\mathbf{U}$ is the debiased user preference embedding.

Structural Negative Sampling. Prior studies have shown that negative sampling is beneficial in many domains [50]. For each user, we define its positive samples are the events preferred by the user, while negative samples are those that the user does not want to retweet or reply to. It is obvious that positive samples can be easily collected, e.g., events in which the user has participated in. Nevertheless, negative samples are usually implicit in rumor detection datasets. A common solution in recommendation is uniform sampling proposed by Bayesian Personalized Ranking [51], which is limited in our task. Since events cannot be exposed to every user in social media, the events that users do not participate in are not necessarily those

that users do not like, i,e., not all unseen events are negative samples.

To address this problem, we make use of the propagation structure of tweets to obtain negative samples named as structural negative sampling. The intuition behind this is that, if user $\mathcal{U}_1$ has retweeted/replied to a tweet posted by another user $\mathcal{U}_2$ who is called the source poster, then other tweets of $\mathcal{U}_2$ are accessible to $\mathcal{U}_1$. We regard these tweets which are not interacted with $\mathcal{U}_1$ as the negative samples to $\mathcal{U}_1$. For the users who do not have any source posters, we randomly sample negative samples from events they have not participated in.

Optimizing. When training, Eq. 9 should reach to 1 for positive samples and 0 for negative samples, thus we define the loss function as follows,

$$\mathcal{L}_r = - \sum_{\substack{(\mathcal{U},\mathcal{I},q_\mathcal{I})\in\mathcal{D}^+ \\ (\mathcal{U},\mathcal{J},q_\mathcal{J})\in\mathcal{D}^-}} \log \sigma(r_{\mathcal{U},\mathcal{I},q_\mathcal{I}} - r_{\mathcal{U},\mathcal{J},q_\mathcal{J}}), \tag{10}$$

where $r_{\mathcal{U},\mathcal{I},q_\mathcal{I}}$ is short for $P(R=1|q_\mathcal{I},\mathcal{U},\mathcal{I})$, $\mathcal{D}^+$ and $\mathcal{D}^-$ are the positive and negative samples sets, respectively. $\sigma(x) = \frac{1}{1+\exp(-x)}$ is a logistic function which transforms the input into range $(0,1)$. For user $\mathcal{U}$, $\mathcal{I}$ is the positive sample and $\mathcal{J}$ is the negative sample. To prevent over-fitting, we adopt L_2 regularization in the model and the objective function is,

$$\mathcal{L}_d = \mathcal{L}_r + \|\Psi\|_2^2 + \|\Omega\|_2^2, \tag{11}$$

$$\hat{\mathbf{U}}^{debias} = \arg\min_{\mathbf{U}} \mathcal{L}_d. \tag{12}$$

We minimize the objective function in the debias phase as demonstrated in Eq. 11 and obtain user embeddings $\hat{\mathbf{U}}^{debias}$ named as a debiased preference embedding matrix.

C. Inference Phase

The embeddings generated from above *debias phase* represent the preference that users' historical behaviors reveal in the social platform. We take the learned user preference embedding as a part of features for the entire rumor event representation. Since, the inherent textual content and propagation structure are also critical and play important roles in representing the features of rumor event. To take both these information and the obtained implicit user preference into consideration, we leverage graph encoders [32] which allow aggregating the node embeddings while, most importantly, fusing the post textual data, user preference and propagation structure with the entire event representation.

User Preference and Content Feature Fusion. For each event $\mathcal{C}_i = \{\mathcal{P}_i^0, \mathcal{P}_i^1, \mathcal{P}_i^2, \dots \mathcal{P}_i^{m_i}, \mathcal{G}_i\}$, we think the contents of $\mathcal{P}_i^j$'s responsive post $\mathcal{T}_i^j$ record the user' attitude toward the specific event $\mathcal{C}_i$. Thus, following previous works [27], [29], [31] we embed the textual data by leveraging models of natural language processing,

$$\mathbf{t}_i^j = f_{emb}(\mathcal{T}_i^j), \quad j \in [0, m_i] \tag{13}$$

where, $f_{emb}(*)$ is the model for embedding textual data, here TF-IDF[52], [53] is chosen, and features are extracted with top-5000 words in terms of TF-IDF values. $\mathbf{t}_i^j \in \mathbb{R}^{1\times d}$ is the content feature of j-th post in i-th event. We then concatenate all content features $\mathbf{t}_i^j$ to obtain the content embedding matrix $\hat{\mathbf{U}}_i^{content}$ for event $\mathcal{C}_i$,

$$\hat{\mathbf{U}}_i^{content} = \text{CONCATE}(\mathbf{t}_i^0, \mathbf{t}_i^1, \dots, \mathbf{t}_i^{m_i}). \tag{14}$$

With the event propagation graph structure $\mathcal{G}_i = (\mathcal{V}_i, \mathcal{E}_i)$, we adopt two GNN encoders $\mathbf{ENC}(\cdot)$, i.e., the same framework but not sharing weights, to preserve the debiased user preference $\hat{\mathbf{U}}_i^{debias}$ and post content $\hat{\mathbf{U}}_i^{content}$ information into the graph-level embeddings $\mathbf{h}(\mathcal{G}_i)$ and $\mathbf{o}(\mathcal{G}_i)$, respectively. We now take user preference fusion for example. The graph encoder is leveraged to encapsulate each node's representation by aggregating user preference information from its neighbors. Following by the encoder is a pooling function $\mathbf{READOUT}(\cdot) : \mathbb{R}^{n\times d} \rightarrow \mathbb{R}^d$, to aggregate node features from the final layer. Thus the entire event graph's representation can been obtained by:

$$\mathbf{h}_i^{j(k)} = \mathbf{ENC}(\mathbf{h}_i^{j(k-1)}) \tag{15}$$

$$\mathbf{h}(G_i)^{(k)} = \mathbf{READOUT}(\{\mathbf{h}_i^{j(k)}\}_{j=0}^{|\mathcal{V}_i|}) \tag{16}$$

$$\mathbf{h}(G_i) = \mathbf{CONCAT}(\{\mathbf{h}(G_i)^{(k)}\}_{k=1}^{K}) \tag{17}$$

where $h_i^{j(k)}$ is the feature vector of post $\mathcal{P}_i^j$ at the k-th graph encoder layer, $\{h_i^{j(0)}\}_{j=0}^{\mathcal{V}_i}$ is initialized with $\hat{\mathbf{U}}_i^{debias}$, $\mathbf{h}(\mathcal{G}_i) \in \mathbb{R}^d$ is the graph embedding of i-th event and d is the dimension of GNN layer. For the post content fusion, we initialize another GNN encoder and take $\hat{\mathbf{U}}_i^{content}$ as the input feature. Thus, the output $\mathbf{o}(\mathcal{G}_i)$ represents the post content graph embedding of event $\mathcal{C}_i$. We choose GIN [54] as the graph-level encoder and multi-layer perceptrons (MLPs) as the pooling function. Note that the encoder is alternative and can be replaced with other GNNs, e.g., GCN [55]and GAT [56] (cf. Table II).

Prediction. It is known that source posts of events contain much more information than retweets or replies [29], [31]. To emphasize the source post, we reuse the content feature $\mathbf{t}_i^0$ of a source post $\mathbf{x}_i^0$. Concating $\mathbf{h}(\mathcal{G}_i), \mathbf{o}(\mathcal{G}_i)$ and $\mathbf{t}_i^0$, we then get the i-th event embedding $\mathbf{e}_i$, and the prediction of event $\hat{y}_i$ is made via a fully connected layer and a softmax layer:

$$\mathbf{e}_i = \mathbf{h}(\mathcal{G}_i) \oplus \mathbf{o}(\mathcal{G}_i) \oplus \mathbf{t}_i^0 \tag{18}$$

$$\hat{y}_i = \text{Softmax}(\text{FC}(\mathbf{e}_i)) \tag{19}$$

where $\oplus$ is the concatenation operation and $\hat{y}$ is the predicted value which indicates the probability of the tweet associated with one of four given labels. For each tweet, our goal is to minimize the cross-entropy loss:

$$\mathcal{L} = \sum_{i=1}^{\|\mathcal{C}\|} y_i \log \hat{y}_i + \eta \|\Theta\|_2^2, \tag{20}$$

where Θ denotes the model parameters, and η is a regularization factor.

V. Experiments

In this section, we conduct experiments to evaluate the performance of our proposed model using real world social media datasets and answer the following three research questions (RQ):

TABLE I: Descriptive statistics of two datasets

Statistic	Twitter15	Twitter16
# of user	276,663	173,487
# of source tweets	1,490	818
# of non-rumors	374	205
# of false rumors	370	205
# of true rumors	372	205
# of unverified rumors	374	203

- **RQ1**: How effective is CausalRD in detecting rumors in social media based on debiased user preference embeddings?
- **RQ2**: How do different components affect the performance of CausalRD?
- **RQ3**: Is debiased user preference more effective than biased user preference in rumor detection?

A. Experiment Settings, Baselines & Metrics

Dataset: To examine CausalRD and its real world applications, we use two publicly available datasets - *Twitter15* and *Twitter16* as our experimental basis. These two datasets are collected from Twitter platform released by Ma et al. [44], and contain 1,490 and 818 source tweets, respectively. The source tweets are annotated by referring to the labels of events where they are from, specifically, the labels of events are verified by the rumor debunking websites, e.g., snopes.com. Each source tweet is associated with textual content and corresponding participants, upon which, a user-event (source tweets) bipartite graph can be built. We show the statistics of two datasets in Table I.

Implementation Details. For all experiments of CausalRD and its variants, unless specified, we uniformly adopt the following settings for comparison. In the debias phase, we optimize the model via Adam [57] to perform module training with the learning rate of $lr^{debias} = 1e-3$, batch size of $4,096$ and L2 weight decay of $2e-2$. We implement $f_\Psi(Q)$ with a 7-layer multi-layer perceptron (MLP). In the inference phase, detection performance is reported under 5-fold cross-validation. We also leverage Adam to optimize the inference model with a learning rate of $lr^{infer} = 1e{-}4$ and a batch size of 128. In addition, for all hidden layers in classification subnet, dropout with a ratio of 0.5 and a L2 weight decay of $1e-3$ are used. The prediction model is trained with a cross entropy loss. Early stop is performed by monitoring the validation loss.

Baselines. We compare the proposed method with several state-of-the-art baselines:

• *DTC* [21]: implements several conventional machine learning techniques, e.g., decision trees, SVM, decision rules and Bayes networks, to exploit the implicit credibility of an event based on a set of handcrafted features.

• *SVM-TS* [22]: develops a linear SVM classifier and time-series model to extract the handcrafted social context features for rumor detection.

• *RvNN* [27]: conducts tree-structured recursive neural networks to learn rumor representations with the propagation structure.

• *PCC_RNN+CNN* [58]: addresses the rumor detection task as a multivariate time series problem and inputs the partial propagation path into RNN and CNN, respectively. Then it concatenates two outputs for early rumor detection.

• *Bi-GCN* [29]: introduces a Bi-directional GCN, i.e., a combination of Top-Down GCN and Bottom-Up GCN, to extract both propagation and dispersion patterns of rumors.

• *RDEA* [31]: designs three event augmentation strategies with the permutation on both content features and propagation structures. The whole training process follows an integrated contrastive self-supervised learning manner.

Metrics: Following existing works [27], [29], we evaluate Accuracy (Acc.) over the four categories and F1-score (F_1) on each class, i.e., NR, FR, TR and UR.

B. Performance Comparison (RQ1)

Table II summarizes experimental results of our method, its variants, and baselines on two datasets *Twitter15* and *Twitter16*. We have the following observations: (a) Deep learning methods achieve better performance comparing to all hand-crafted feature-based methods, which indicates learning high-level and effective representation is essential for rumor detection; (b) RvNN only uses hidden feature vectors of all leaf nodes which is heavily relied on latest posts. However, source posts contain abundant information, which plays a leading role in identifying the veracity of a rumor. Bi-GCN performs better than RvNN since it pays more attention on the source posts by applying root feature enhancement operation; (c) Propagation structure modeling is the research hotspot of this task. PCC_RNN+CNN simply transforms the diffusion progress of event into a sequence by time, which ignores some structural dependencies, and gains the worst results among deep learning methods. RvNN surpasses PCC_RNN+CNN, as it naturally conforms to the tree-structured propagation layout of tweets. Bi-GCN and RDEA adopt GNNs to learn a high-level of representation of the event. So far, graph encoder is the optimal and flexible module to aggregate structural, textual, and temporal features together via neural networks; (d) We notice that all baselines do not utilize user behaviors to obtain their preference. CausalRD performs the best, which can be attributed to two effects: (1) CausalRD leverages user behaviors to capture global preference of users in a causal view, which eliminates aforementioned biases; (2) both content features and context features are utilized for detecting rumors.

C. Ablation Study (RQ2)

To better analyze the key components in our model, we design and implement following variants:

• *CausalRD-Biased* - in which we remove the causal intervention in the debias phase, and extract user preference embeddings via the conventional causal graph.

• *CausalRD-noPreference* - in which we remove the graph encoder that leverages debiased user preference embeddings,

TABLE II: The rumor detection performance of baselines, our model and its variants. Bold-face indicates the best score, and underlined denotes the second best. Non-Rumor (NR), False Rumor (FR), True Rumor (TR), Unverified Rumor (UR).

Method	*Twitter15*					*Twitter16*				
	Acc.	**NR**	**FR**	**TR**	**UR**	**Acc.**	**NR**	**FR**	**TR**	**UR**
DTC	0.454	0.733	0.355	0.317	0.415	0.464	0.643	0.393	0.419	0.403
SVM-TS	0.544	0.796	0.472	0.404	0.483	0.574	0.755	0.420	0.571	0.526
RvNN	0.723	0.682	0.758	0.821	0.654	0.737	0.662	0.743	0.835	0.708
PPRC_RNN+CNN	0.697	0.689	0.760	0.696	0.645	0.702	0.608	0.711	0.816	0.664
Bi-GCN	0.836	0.791	0.842	0.887	0.801	0.864	0.788	0.859	0.932	0.864
RDEA	0.855	0.831	0.857	0.903	0.816	0.880	0.823	0.878	0.937	0.875
CausalRD-Biased	0.849	0.818	0.848	0.901	0.802	0.861	0.811	0.855	0.901	0.859
CausalRD-noPreference	0.851	0.828	0.854	0.901	0.816	0.868	0.823	0.863	0.926	0.866
CausalRD-Random	0.853	0.822	0.858	0.905	0.815	0.876	0.819	0.862	0.929	0.871
CausalRD-mlp3	0.850	0.821	0.852	0.899	0.816	0.872	0.825	0.867	0.924	0.869
CausalRD-mlp5	0.858	0.850	0.858	0.903	0.820	0.880	0.827	0.882	0.933	0.875
CausalRD-mlp9	0.861	0.863	0.857	0.907	0.834	0.887	0.837	0.889	**0.940**	0.880
CausalRD-GCN	0.855	0.856	0.852	0.898	0.825	0.878	0.829	0.875	0.931	0.871
CausalRD-GAT	0.860	0.865	0.857	0.905	0.835	0.885	0.836	0.885	0.934	0.880
CausalRD	**0.862**	**0.866**	**0.861**	**0.910**	**0.837**	**0.891**	**0.839**	**0.894**	0.939	**0.883**

i.e., neither debiased nor biased user preference information is used.

• *CausalRD-Random* - in which we adopt random negative sampling strategy instead of structural negative sampling.

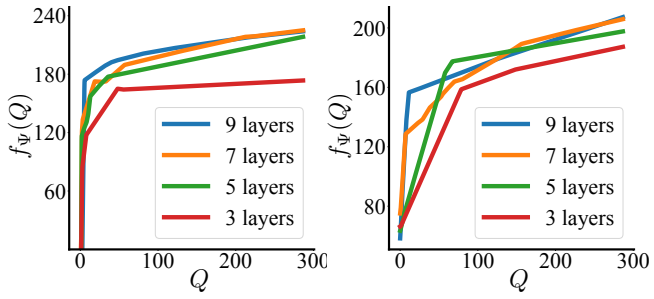
• *CausalRD-mlp3, CausalRD-mlp5* and *CausalRD-mlp9* - are the different versions of CausalRD that implement the $f_\Psi(Q)$ with 3, 5 or 9 hidden layers. Note that the default setting of $f_\Psi(Q)$ is a 7-layer MLP.

• *CausalRD-GCN* and *CausalRD-GAT* - are the variants that use different GNN encoders to aggregate both the debiased user preference and content information in the inference phase.

CausalRD vs. CausalRD-Biased. We first perform an ablation study on biased and debiased user preference to empirically study the impact of causal intervention. In the debias phase, we remove the popularity and conformity bias as Eq. (9), where $f_\Psi(q_\mathcal{I})$ denotes the influence on user behavior through popularity of event $q_\mathcal{I}$. When the term $f_\Psi(q_\mathcal{I})$ is removed, Eq. (9) is the same as conventional methods, thereby the preference embeddings of users are biased and the model is named as CausalRD-Biased. The empirical results between CausalRD and CausalRD-Biased are summarized in Table II, it is not surprising that the CausalRD outperforms CausalRD-Biased, indicating the popularity and conformity biases indeed affect the detection of rumors.

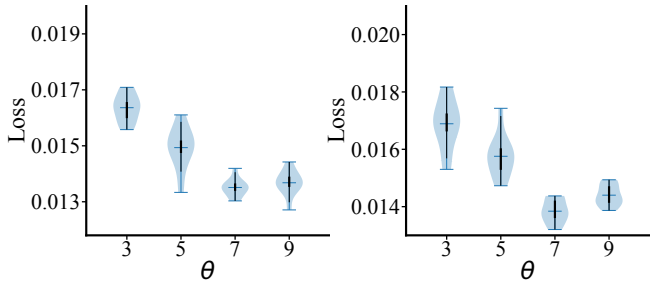
CausalRD vs. CausalRD-noPreference. To evaluate the effectiveness of debiased user preference generated from the debias phase of our method, we compare the performance of CausalRD and CausalRD-noPreference. The results show that our proposed method has 1.2% and 2.6% improvement over the variants in both datasets.

Comparing to the results based on aforementioned biased user preference, we draw some critical inference, user preference is beneficial to rumor detection only when the popularity and conformity biases are eliminated, i.e., debiased user preference works whereas biased user preference do harm to prediction. Additional, it is apparent that the interaction data including biases is not able to represent users' preference correctly.



(a) *Twitter15*. (b) *Twitter16*.

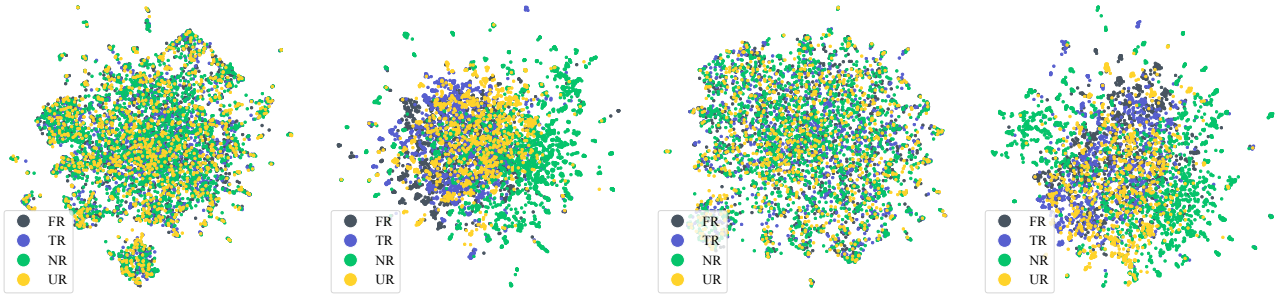
Fig. 5: The learned influence of popularity with different number of hidden layers.



(a) *Twitter15*. (b) *Twitter16*.

Fig. 6: Training losses under different θ's in the debias phase.

CausalRD vs. CausalRD-Random. To understand the role of the structural negative sampling via propagation structure in our method, we perform an ablation study to test these two types of negative sampling strategies. In both real world datasets, the performance shows structural negative sampling strategy is more effective than random sampling. This discrepancy could be attributed to the fact that social media platforms provide numerous information for users. User could only view a small part of it, thereby it does not mean that users dislike those events that users are not involved in. Rather it is more



(a) Biased preference of *Twitter15*. (b) Debiased preference of *Twitter15*. (c) Biased preference of *Twitter16*. (d) Debiased preference of *Twitter16*.

Fig. 7: Visualization of biased and debiased user preference embeddings for *Twitter15* and *Twitter16* with 2-D Umap.

likely that they might not see these posts at all.

If we take negative samples randomly, the learned embeddings will fail to capture the user preference, by contrast, structural negative sampling makes full use of the propagation structures of events and takes the negative samples which are more likely to be seen but not liked for users, we propose the sampling strategy out of an assumption that if two users are connected in a propagation structure, e.g., $\mathcal{U}_1 \rightarrow \mathcal{U}_2$, then $\mathcal{U}_2$ has a greater chance of seeing $\mathcal{U}_1$'s posts than those who have never interacted with $\mathcal{U}_1$. This assumption is consistent with reality, e.g., $\mathcal{U}_2$ is the follower of $\mathcal{U}_1$, and $\mathcal{U}_2$ has access to all tweets posted by $\mathcal{U}_1$, whether or not $\mathcal{U}_2$ engages in these events is a more accurate reflection of its preference.

The effect of different GNN encoders. The selection of graph encoder is not limited in our architecture. To verify that the performance improvement is not attributed to the encoder part, in addition to GIN, we implement two GNN encoders including GCN and GAT. The results show that GIN is the most expressive GNN encoder, and GNN encoder is crucial to fully represent the rumor event.

The effect of $f_\Psi(Q)$. When eliminating the bias of popularity, we implement a neural network to capture the influence of popularity. In order to analyze the effect of $f_\Psi(Q)$ with different number of hidden layers θ, we let $\theta = \{3, 5, 7, 9\}$ and optimize Eq. (11) in the debias phase, then we extract the learned $f_\Psi(Q)$ from the model, the four functions learned on both Twitter datasets are depicted in Fig. 5. We can observe that the overall influence is monotonically increasing with popularity, which is consistent with our aforementioned discussion and the growth of influence slows down as popularity increases. To select the best θ for each dataset, we compare the loss in the debias phase with different θ's, and the training losses of two datasets are presented in Fig. 6. Combining their performance in rumor detection shown in Table II, we choose $\theta = 7$ as our proposed model's parameter for efficiency.

D. Visualization of User Preference (RQ3)

To understand how well debiased phase with causal intervention captures the debiased user preference among data, we first extract the debiased user preference embeddings from our CausalRD framework which are denoted as $\hat{\mathcal{U}}^{debias}$, and the biased user preference embeddings $\hat{\mathcal{U}}^{bias}$ based on conventional causal graph. Then we compare and visualize these two embedding vectors in a 2D space using Umap[59]. We count the number of rumors in each user's engagement in four categories (i.e., NR, FR, TR and UR) and select the rumor with the largest number as the user's label, visualization of both datasets are presented in Fig. 7.

We observe that the biased preference embeddings $\hat{\mathcal{U}}^{bias}$ of four classes are mixed up as shown in Fig. 7a and 7c, indicating that it is impossible to distinguish users by the biased embeddings. For debiased user preference embeddings in Fig. 7b and 7d, the users who prefer non-rumors (e.g., information from official institutions) are distinguishable among all users, thereby benefiting the performance of detecting rumors. This can be further verified by the quantitative results in Table II. Users who prefer engaging in non-rumors (NR) are more distinguishable than others. A possible explanation is that these Twitter users prefer following the credible channels whose information source are more reliable. If the popularity and conformity biases are eliminated, i.e., the received tweet events are equal in popularity, the ability to identify the information veracity of NR users is more prominent. Thus, user preference representations after eliminating popularity and conformity biases, tend to be more distinguishable.

VI. Conclusion

In this study, we investigate the popularity bias and conformity bias in rumor detection. To eliminate these biases, and extract implicit and debiased user preference from the historical user-event interactions, we construct a novel causal graph and apply causal interventions. In addition, we also improve the effectiveness of the debiasing process by a particular structural negative sampling strategy. Extensive experimental evaluation and model component analyses validate the effectiveness of CausalRD on eliminating biases of observational data and improving identification ability.

Acknowledgment

This work was supported by National Natural Science Foundation of China (NSFC) Grant No. 62176043 and Grant No. 62072077.

REFERENCES

[1] P. R. Center, "Social media use in 2021," 2021.

[2] S. Vosoughi, M. N. Mohsenvand, and D. Roy, "Rumor gauge: Predicting the veracity of rumors on twitter," *ACM Trans. Knowl. Discov. Data*, vol. 11, no. 4, pp. 50:1–50:36, 2017.

[3] K. Rapoza, "Can 'fake news' impact the stock market?" 2017.

[4] A. Bovet and H. Makse, "Influence of fake news in twitter during the 2016 us presidential election," *Nature Communications*, vol. 10, 2019.

[5] N. DiFonzo and P. Bordia, "Rumor, gossip and urban legends," *Diogenes*, vol. 54, pp. 19 – 35, 2007.

[6] A. Zubiaga, A. Aker, K. Bontcheva, M. Liakata, and R. Procter, "Detection and resolution of rumours in social media: A survey," *ACM Comput. Surv.*, vol. 51, no. 2, pp. 32:1–32:36, 2018.

[7] S. Vosoughi, D. Roy, and S. Aral, "The spread of true and false news online," *Science*, vol. 359, pp. 1146 – 1151, 2018.

[8] A. Zubiaga, G. W. S. Hoi, M. Liakata, R. Procter, and P. Tolmie, "Analysing how people orient to and spread rumours in social media by looking at conversational threads," *PLoS ONE*, vol. 11, 2016.

[9] J. Choi, S. Moon, J. Woo, K. Son, J. Shin, and Y. Yi, "Rumor source detection under querying with untruthful answers," in *INFOCOM*, 2017, pp. 1–9.

[10] S. Qu, Z. Zhao, L. Fu, X. Wang, and J. Xu, "Joint inference on truth/rumor and their sources in social networks," in *INFOCOM*, 2020, pp. 924–933.

[11] K. Shu, H. R. Bernard, and H. Liu, "Studying fake news via network analysis: detection and mitigation," in *Emerging research challenges and opportunities in computational social network analysis and mining*. Springer, 2019, pp. 43–65.

[12] E. Pariser, *The filter bubble: How the new personalized web is changing what we read and how we think*. Penguin, 2011.

[13] Y. Dou, K. Shu, C. Xia, P. S. Yu, and L. Sun, "User preference-aware fake news detection," in *SIGIR*, 2021, pp. 2051–2055.

[14] A. Khattri, A. Joshi, P. Bhattacharyya, and M. J. Carman, "Your sentiment precedes you: Using an author's historical tweets to predict sarcasm," in *WASSA@EMNLP*, 2015, pp. 25–30.

[15] H. Abdollahpouri, M. Mansoury, R. Burke, and B. Mobasher, "The unfairness of popularity bias in recommendation," *arXiv preprint arXiv:1907.13286*, 2019.

[16] J. Colliander, ""this is fake news": Investigating the role of conformity to other users' views when commenting on and spreading disinformation in social media," *Comput. Hum. Behav.*, vol. 97, pp. 202–215, 2019.

[17] H. A. Teunissen, R. Spijkerman, M. J. Prinstein, G. L. Cohen, R. C. Engels, and R. H. Scholte, "Adolescents' conformity to their peers' pro-alcohol and anti-alcohol norms: The power of popularity," *Alcoholism: Clinical and experimental research*, vol. 36, no. 7, pp. 1257–1267, 2012.

[18] S. Kumar and N. Shah, "False information on web and social media: A survey," *CoRR*, vol. abs/1804.08559, 2018.

[19] A. Bondielli and F. Marcelloni, "A survey on fake news and rumour detection techniques," *Inf. Sci.*, vol. 497, pp. 38–55, 2019.

[20] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[21] C. Castillo, M. Mendoza, and B. Poblete, "Information credibility on twitter," in *WWW*, 2011, pp. 675–684.

[22] J. Ma, W. Gao, Z. Wei, Y. Lu, and K. Wong, "Detect rumors using time series of social context information on microblogging websites," in *CIKM*, 2015, pp. 1751–1754.

[23] F. Zhou, X. Xu, G. Trajcevski, and K. Zhang, "A survey of information cascade analysis: Models, predictions, and recent advances," *ACM Computing Surveys*, vol. 54, no. 2, 2021.

[24] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[25] Y. LeCun, Y. Bengio, and G. E. Hinton, "Deep learning," *Nat.*, vol. 521, no. 7553, pp. 436–444, 2015.

[26] J. Ma, W. Gao, P. Mitra, S. Kwon, B. J. Jansen, K. Wong, and M. Cha, "Detecting rumors from microblogs with recurrent neural networks," in *IJCAI*, 2016, pp. 3818–3824.

[27] J. Ma, W. Gao, and K. Wong, "Rumor detection on twitter with tree-structured recursive neural networks," in *ACL*, 2018, pp. 1980–1989.

[28] F. Yu, Q. Liu, S. Wu, L. Wang, and T. Tan, "A convolutional approach for misinformation identification," in *IJCAI*, 2017, pp. 3901–3907.

[29] T. Bian, X. Xiao, T. Xu, P. Zhao, W. Huang, Y. Rong, and J. Huang, "Rumor detection on social media with bi-directional graph convolutional networks," in *AAAI*, 2020, pp. 549–556.

[30] X. Yang, Y. Lyu, T. Tian, Y. Liu, Y. Liu, and X. Zhang, "Rumor detection on social media with graph structured adversarial learning," in *IJCAI*, 2020, pp. 1417–1423.

[31] Z. He, C. Li, F. Zhou, and Y. Yang, "Rumor detection on social media with event augmentations," in *SIGIR*, 2021, pp. 2020–2024.

[32] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *ICLR*, 2017.

[33] J. Pearl, "The seven tools of causal inference, with reflections on machine learning," *Communications of the ACM*, vol. 62, no. 3, pp. 54–60, 2019.

[34] B. Schölkopf, "Causality for machine learning," *arXiv preprint arXiv:1911.10500*, 2019.

[35] J. K. Lunceford and M. Davidian, "Stratification and weighting via the propensity score in estimation of causal treatment effects: a comparative study," *Statistics in medicine*, vol. 23, no. 19, pp. 2937–2960, 2004.

[36] G. W. Imbens, "Nonparametric estimation of average treatment effects under exogeneity: A review," *Review of Economics and statistics*, vol. 86, no. 1, pp. 4–29, 2004.

[37] C. Drake, "Effects of misspecification of the propensity score on estimators of treatment effect," *Biometrics*, pp. 1231–1236, 1993.

[38] S. Wager and S. Athey, "Estimation and inference of heterogeneous treatment effects using random forests," *Journal of the American Statistical Association*, vol. 113, no. 523, pp. 1228–1242, 2018.

[39] R. Guo, J. Li, and H. Liu, "Learning individual causal effects from networked observational data," in *WSDM*, 2020, pp. 232–240.

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

[41] W. Wang, F. Feng, X. He, H. Zhang, and T.-S. Chua, "Clicks can be cheating: Counterfactual recommendation for mitigating clickbait issue," in *SIGIR*, 2021, pp. 1288–1297.

[42] Y. Zhang, F. Feng, X. He, T. Wei, C. Song, G. Ling, and Y. Zhang, "Causal intervention for leveraging popularity bias in recommendation," *arXiv preprint arXiv:2105.06067*, 2021.

[43] K. Wu, S. Yang, and K. Q. Zhu, "False rumors detection on sina weibo by propagation structures," in *ICDE*, 2015, pp. 651–662.

[44] J. Ma, W. Gao, and K. Wong, "Detect rumors in microblog posts using propagation structure via kernel learning," in *ACL*, 2017, pp. 708–717.

[45] A. Zubiaga, A. Aker, K. Bontcheva, M. Liakata, and R. Procter, "Detection and resolution of rumours in social media: A survey," *ACM Comput. Surv.*, vol. 51, no. 2, pp. 32:1–32:36, 2018.

[46] P. W. Holland, "Statistics and causal inference," *Journal of the American statistical Association*, vol. 81, no. 396, pp. 945–960, 1986.

[47] J. Pearl, "Causality," 2009.

[48] F. Vella, "Estimating models with sample selection bias: a survey," *Journal of Human Resources*, pp. 127–169, 1998.

[49] R. J. Little and D. B. Rubin, *Statistical analysis with missing data*. John Wiley & Sons, 2019, vol. 793.

[50] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *ICLR*, 2013.

[51] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "BPR: bayesian personalized ranking from implicit feedback," in *UAI*, 2009, pp. 452–461.

[52] H. P. Luhn, "A statistical approach to mechanized encoding and searching of literary information," *IBM Journal of research and development*, vol. 1, no. 4, pp. 309–317, 1957.

[53] K. S. Jones, "A statistical interpretation of term specificity and its application in retrieval," *Journal of documentation*, 1972.

[54] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?" in *ICLR*, 2019.

[55] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *ICLR*, 2017.

[56] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *ICLR*, 2018.

[57] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2015.

[58] Y. Liu and Y. B. Wu, "Early detection of fake news on social media through propagation path classification with recurrent and convolutional networks," in *AAAI*, 2018, pp. 354–361.

[59] L. McInnes, J. Healy, and J. Melville, "Umap: Uniform manifold approximation and projection for dimension reduction," *arXiv preprint arXiv:1802.03426*, 2018.