Resisting the Edge-type Disturbance for Link Prediction in Heterogeneous Social Networks

Huan Wang
University of Macau, State Key Laboratory of Internet of Things for Smart City, Huazhong Agricultural University, College of Informatics, hwang@mail.hzau.edu.cn

Ruigang Liu
Huazhong Agricultural University, College of Informatics, liuruigang@webmail.hzau.edu.cn

Chuanqi Shi
Huazhong Agricultural University, College of Informatics, scqrandom42@gmail.com

Junyang Chen
Shenzhen University, College of Computer Science and Software Engineering, junyangchen@szu.edu.cn

Lei Fang
University of St Andrews, School of Computer Science, lf28@st-andrews.ac.uk

Shun Liu
University of Macau, Department of Electrical and Computer Engineering, slau.phys@gmail.com

Zhiguo Gong
University of Macau, State Key Laboratory of Internet of Things for Smart City, Macao Guangdong-Macau Joint Laboratory for Advanced and Intelligent Computing, fstzgg@um.edu.mo

The rapid development of heterogeneous social networks has proposed new challenges to the long-standing link prediction problem. Existing models trained on the verified edge samples from different types usually learn type-specific knowledge, and their type-specific predictions may be contradictory for unverified edge samples with uncertain types. This challenge is termed edge-type disturbance in link prediction in heterogeneous social networks. To address this challenge, we develop a disturbance-resilient prediction method (\textit{DRPM}) comprising a structural characterizer, a type differentiator, and a resilient predictor. The structural characterizer is responsible for learning edge representations for link prediction. Concurrently, the type differentiator distinguishes type-specific edge representations to generate diverse type experts while maximizing their link prediction performances on specific types. Further, the resilient predictor evaluates the reliability weights of different type experts to develop a resilient prediction mechanism to aggregate discriminable predictions. Extensive experiments conducted on various real-world datasets demonstrate the importance of the explainable introduction of the edge-type disturbance and the superiority of \textit{DRPM} over state-of-the-art methods.

CCS CONCEPTS: • Applied computing → Sociology; • Human centered computing→ Social networks; • Information systems → Data mining;

* Authors' Addresses: H. Wang, the State Key Laboratory of Internet of Things for Smart City, Department of Computer Information Science, University of Macau, China; and he is also the College of Informatics, Huazhong Agricultural University, Wuhan, 430070, China; E-mail: cswanghuan@whu.edu.cn; R. Liu, the College of Informatics, Huazhong Agricultural University, Wuhan, 430070, China; Email: liuruigang@webmail.hzau.edu.cn; C. Shi, College of Informatics, Huazhong Agricultural University, Wuhan, 430070, China; Email: scqrandom42@gmail.com; J. Chen, the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, 518060, China; E-mail: junyangchen@szu.edu.cn; L. Fang, the School of Computer Science, University of St Andrews, UK; E-mail: lf28@st-andrews.ac.uk; S. Liu, the Department of Electrical and Computer Engineering, University of Macau, China; E-mail: slau.phys@gmail.com; Z. Gong, the corresponding author, the State Key Laboratory of Internet of Things for Smart City, University of Macau, Macao Guangdong-Macau Joint Laboratory for Advanced and Intelligent Computing, China; Email: fstzgg@um.edu.mo.
Additional Key Words and Phrases: Heterogeneous social network, Link prediction, Resilient prediction mechanism, Edgetype disturbance

1 INTRODUCTION

With the rapid increase in social interactions globally via online services (e.g., LinkedIn, WeChat, and WhatsApp), social network analysis has attracted a lot of attentions in various fields, from noise corrected sampling [1] to anomaly detection [2] and from influence maximization [3] to social recommendation [4]. Link prediction is longstanding research area in social network analysis that aims to predict missing or future edges based on the collected network structure [5]. In particular, increased attention has been devoted toward research on link prediction in heterogeneous social networks. Heterogeneous social networks are usually represented as graphs that contain nodes and edges on different types, where the nodes and edges are used to represent the individuals and interactions, respectively [6]. Unlike homogeneous social networks, the formation processes of the edges on different types are usually driven by different evolitional mechanisms. It is difficult to simultaneously explain the formation processes of edges on different types by fitting their diverse evolitional mechanisms. For example, in Figure 1, the types of nodes contain “user,” “song,” and “singer.” The user–user edge type that represents friendship among users tends to follow a triadic closure mechanism [7], whereas the user–song edge type that represents users who like songs tends to follow a preferential attachment mechanism [8]. The difference in evolitional mechanisms makes it difficult to design a well-behaved link prediction method for both user–user and user–singer types. To shape correct and diverse social interactions, it is necessary to propose a general link prediction method for all edge types in heterogeneous social networks.

![Figure 1: Heterogeneous social network based on social music services.](image)

Thus far, numerous techniques have been proposed to tackle the issue of link prediction, mainly consisting of traditional calculation methods [9] and machine learning or deep learning-based methods [10]. The traditional calculation methods evaluate structural similarities, establish fitting models, or presuppose organizing principles to evaluate the connection likelihood of two nodes. Compared with traditional calculation methods, machine learning or deep learning-based methods have significantly improved performances because of their powerful automatic modeling capabilities. With sufficient
verified edge samples, machine learning or deep learning techniques are successfully applied in homogeneous social networks to build effective link prediction models [11, 12]. However, in heterogeneous social networks, it is difficult to collect sufficient verified edge samples on one type in real time and train a model for unverified edge samples on the same type. The edges on different types usually follow different evolutional mechanisms to generate type-specific features that are not sharable to characterize other edge types [13, 14]. Therefore, the edge-type disturbance has been provoked in heterogeneous social networks: existing models trained on the verified edge samples from different types tend to learn type-specific knowledge, and their type-specific predictions may be contradictory for unverified edge samples with uncertain types. We believe that resisting the edge-type disturbance is vital and beneficial to excerpting the power of machine learning or deep learning in link prediction. The target of this research is to resist the edge-type disturbance for link prediction in heterogeneous social networks. In addition to explaining the edge-type disturbance phenomenon, this research is beneficial to promoting the link prediction performances in actual applications, such as friend recommendation [15] and collaboration discovery on new research areas [16].

To resist the edge-type disturbance, the first issue is how to learn type-specific knowledge on different edge types during the training stage. The fusion of nontransferable type-specific representations from different types degrades the training effectiveness for link prediction on specific edge types, especially for new edge types. To learn type-specific representations, we are inspired to retain the type-specific features and remove the sharing features among different edge types. Although this is highly challenging, the learning of type-specific knowledge is the premise for resisting the edge-type disturbance. For example, in Figure 1, we need to capture the discriminable type-specific features to characterize the user–user, user–song, and song–singer types, respectively. The second issue involves how to aggregate the type-specific knowledge on different edge types during the prediction stage. The models trained on different edge types aim to maximize the link prediction performances on specific edge types, but their prediction conclusions remain beneficial to promote the link prediction performances on other edge types. Considering the importance difference of type-specific knowledge on different edge types, an effective mechanism is required to aggregate the discriminable predictions from different edge types to promote the link prediction performance. For example, in Figure 1, if a company aims to promote the new collaborations among singers, we need to predict the existence of singer–singer edges by aggregating the prediction conclusions from the user–user, user–song, and song–singer types.

To solve the issues above, we develop a disturbance-resilient prediction method (DRPM) for link prediction in heterogeneous social networks, which resists the edge-type disturbance to get performance improvements. The method consists of a structural characterizer, a type differentiator, and a resilient predictor. The structural characterizer automatically extracts the structural features of edges on different types. In combination with the structural characterizer, the type differentiator learns type-specific knowledge to generate the type experts that aim to maximize their link prediction performances on specific types. Further, the resilient predictor develops a unified resilient prediction mechanism to effectively aggregate the type-specific predictions from different type experts. Our main contributions are as follows.

- We are the first to recognize the challenge of the edge-type disturbance for link prediction in heterogeneous social networks. We propose DRPM to resist the edge-type disturbance to promote the link prediction performance in heterogeneous social networks.
- Our proposed DRPM not only learns type-specific knowledge to simultaneously generate diverse type experts during the training stage, but also develops a resilient prediction mechanism to aggregate type-specific predictions during the prediction stage. Finally, DRPM has a strong ability to resist the edge-type disturbance in link prediction during the training and prediction stages.
DRPM provides a general framework to resist the edge-type disturbance for link prediction in heterogeneous social networks. Its integrated structural characterizer can be substituted with alternative models specifically designed for extracting edge features in either single-model or multi-model scenarios.

The rest of this paper is structured as follows. Section 2 provides a summary of the research background. Section 3 defines the problem in this research. Section 4 presents our proposed DRPM to enhance link prediction performance by resisting the edge-type disturbance in heterogeneous social networks. In Section 5, we extensively evaluate the effectiveness of the DRPM through experiments. Finally, Section 6 concludes the research and suggests future directions.

2 BACKGROUND

In this section, we provide a brief introduction to the preliminary concepts about heterogeneous social networks and truth discovery, and then review work related to our research.

2.1 Preliminary Concepts

Heterogeneous social network. The social systems usually abstracted as heterogeneous social networks that contain the nodes and edges within different types. Heterogeneous social networks extract the complex network structures from actual social systems, and strictly distinguish the heterogeneity of nodes and edges [17]. Compared with homogeneous networks, heterogeneous social networks are able to integrate richer structural information and contain richer semantic information about nodes and edges. Heterogeneous social networks have been exploited in many data mining tasks by leveraging the semantic information of nodes and edges [18]. In particular, the diverse types of social interactions in heterogeneous social networks have brought new challenges for link prediction.

Truth discovery. With the increasing availability of data from multiple sources, the challenge arises of dealing with conflicting information for the same object. In response, truth discovery has emerged as a popular topic, aiming to integrate noisy information from multiple sources by estimating the reliability of each source [19]. Truth discovery methods operate without supervision, relying solely on the data to infer source reliability. These methods tightly integrate the estimation of source reliability and the discovery of truth from multi-source data, where source reliability is typically defined as the probability of a source providing true claims. The guiding principle is that the sources consistently providing accurate information are assigned higher reliability degrees, and the information supported by reliable sources is considered as truth. Various truth discovery methods have been proposed for different scenarios, successfully applied in diverse application domains [20].

2.2 Related Work

In recent years, online social networks have experienced tremendous growth and success. There are social networks that encompass multiple types of interacting components, resulting in a multitude of social interactions. These networks are referred to as heterogeneous social networks, characterized by different types of nodes and edges [6]. Heterogeneous social networks place emphasis on the heterogeneity of edge types, presenting a new challenge to the longstanding issue of link prediction [21].

The traditional calculation methods for link prediction have been developed for a long time. Although they have obtained excellent results on homogeneous social networks, their link prediction performances are degraded on heterogeneous social networks. The similarity-based methods calculate the structural similarity of nodes based on their structural information, but their effectiveness varies across different networks [22, 23]. In addition, the probabilistic models [24, 25] optimize a target function to fit the network properties and estimate the likelihood of unobserved edges using conditional probabilities. Further,
the maximum likelihood methods exploit network structure principles and maximize the likelihood of the observed structure, but they are difficult to handle large-scale networks [26, 27].

Thanks to the advancements in machine learning and deep learning technologies [28], numerous link prediction methods based on these approaches have emerged to address the link prediction problem in heterogeneous social networks. Compared to traditional calculation methods, the learning-based approaches have demonstrated significant performance improvements due to their superior capability for automatic modeling [29]. Negi and Chaudhury [30] proposed a multi-task, metric learning solution to tackle link prediction in heterogeneous networks. Their method took into account task correlations, robustness to non-informative features, and the non-stationary degree distribution across networks. Chen et al. [31] introduced a novel heterogeneous information network embedding model based on metric learning to capture both first-order and second-order proximities. They achieved this by initially projecting vertices from object spaces to corresponding relation spaces and then calculating the proximity between these projected vertices. Zhao et al. [32] presented a unique multi-view adversarial completion model, where each relation space was characterized from a single viewpoint to leverage the topological structural information in each view. Zhang and Chen [33] adopted a Graph Neural Network (GNN) to learn heuristics from local subgraphs surrounding target edges. They proposed captured general graph structure features from local enclosing subgraphs and defined a function that maps subgraph patterns to link existence. Additionally, Wang et al. [34] developed a framework that bridged static representation learning methods with global information from the entire graph, utilizing localized attention-driven mechanisms to learn contextual node representations. Cen et al. [35] proposed a unified method that tackles the embedding learning problem and introduced a highly effective link prediction approach for attributed multiplex heterogeneous networks. Additionally, Hu et al. [36] developed a novel method for embedding heterogeneous information networks for link prediction with a generator and a discriminator. The generator learned the node distribution to generate negative samples, while the discriminator worked in collaboration with the generator to capture the diverse heterogeneous semantics of the network.

Furthermore, there are numerous other heterogeneous graph learning approaches for user modeling that can be utilized for link prediction. Wang et al. [37] introduced a disentangled heterogeneous graph attention network for top-$N$ recommendation, which effectively learned disentangled user-item representations from various aspects. Long et al. [38] proposed a Self-Supervised Metagraph Informax Network, which explored the potential of jointly incorporating social and knowledge-aware relational structures into the user preference representation for recommendation purposes. Fan et al. [39] addressed the challenge of modeling complex objects and rich interactions in intent recommendation by leveraging a heterogeneous information network. They developed a metapath-guided heterogeneous GNN to learn the embeddings of objects in intent recommendation. Additionally, several recent heterogeneous graph learning models have demonstrated high relevance to our research, particularly in encoding relation heterogeneity for embedding generation. Ji et al. [40] discovered an important phenomenon of semantic confusion that arises with the growth of model depth, leading to performance degradation of heterogeneous GNNs. To address this issue, they explained the semantic confusion and proposed a novel Heterogeneous Graph Propagation Network to alleviate it. Furthermore, Zhao et al. [41] made the first attempt to learn an optimal heterogeneous graph structure for heterogeneous GNNs and introduced a novel framework for jointly performing Heterogeneous Graph Structure Learning and GNN parameter learning for classification tasks.

Unfortunately, the potential edge-type disturbance in heterogeneous social networks significantly impacts the performance of existing link prediction methods. Models trained on verified edge samples from different types tend to learn type-specific knowledge, leading to contradictory predictions for unverified edge samples with uncertain types. Currently, there is a lack of a satisfactory link prediction method that can effectively handle this edge-type disturbance and improve
performance in heterogeneous social networks. In this research, we aim to address this challenge by proposing a disturbance-
resilient link prediction method that leverages the ignored disadvantage of edge-type disturbance and turns it into an
advantage to enhance link prediction accuracy.

3 PROBLEM DEFINITION

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$G$</td>
<td>Heterogeneous social network.</td>
<td>$h_e$</td>
<td>Label of the edge sample $e$.</td>
</tr>
<tr>
<td>$V$</td>
<td>Node set in $G$.</td>
<td>$\varnothing$</td>
<td>$k$th type expert.</td>
</tr>
<tr>
<td>$E$</td>
<td>Edge set in $G$.</td>
<td>$J_e$</td>
<td>Representation vector of $e$.</td>
</tr>
<tr>
<td>$T_o$</td>
<td>Type set of nodes in $V$.</td>
<td>$J_{(k)}$ Representation vector for the $k$th edge type.</td>
<td></td>
</tr>
<tr>
<td>$r$</td>
<td>Sample ratio between test and training sets.</td>
<td>$b^\alpha$</td>
<td>Reliability weight of the type expert $\varnothing$.</td>
</tr>
<tr>
<td>$R_F$</td>
<td>Set of feature representations of edges.</td>
<td>$P(e)$</td>
<td>Existing likelihood of $e$.</td>
</tr>
<tr>
<td>$K$</td>
<td>Total number of type experts.</td>
<td>$J_e$</td>
<td>Representation vector of the edge sample $e$.</td>
</tr>
<tr>
<td>$U$</td>
<td>Set of unobserved edges.</td>
<td>$\xi$</td>
<td>Newly added edge sample ratio.</td>
</tr>
<tr>
<td>$S_k$</td>
<td>Set of the edges on the $k$th type.</td>
<td>$T_p$</td>
<td>Set of possible edge types.</td>
</tr>
<tr>
<td>$e$</td>
<td>Edge sample.</td>
<td>$T^n_p$</td>
<td>Set of new types.</td>
</tr>
<tr>
<td>$\eta$</td>
<td>Learning rate.</td>
<td>$T^h_p$</td>
<td>Set of historical types.</td>
</tr>
<tr>
<td>$\varphi(v)$</td>
<td>Type of node $v$.</td>
<td>$(t_1, t_2)$</td>
<td>Edge type.</td>
</tr>
<tr>
<td>$C^{(k)}(e)$</td>
<td>Prediction performance by the $k$th type expert for the edge sample $e$.</td>
<td>$r_e^{(k)}$</td>
<td>Type correlation between the edge sample $e$ and the $k$th edge type.</td>
</tr>
</tbody>
</table>

The common symbols in this research are listed in Table 1. A network graph is used to represent the heterogeneous social
network, denoted by $G = (V, E, T_o, \varphi)$. $V$ represents the set of nodes and $E$ represent the set of edges, respectively. $T_o$
represents the type set of nodes in $V$ and $|T_o| \geq 2$. $\varphi$ is a function that maps the node in $V$ to the node type in $T_o$. and $\varphi(v)$
represents the type of the node $v$ in $V$. To focus on the edge-type disturbance, the set of possible edge types in $G$ is denoted
as $T_p$ and $T_p = \{(t_1, t_2) \mid t_1 \in T_o \wedge t_2 \in T_o\}$. We use $(t_1, t_2)$ to denote an edge type. To develop a generalized method, an
unweighted and undirected network graph $G$ is defined, where $(t_1, t_2)$ and $(t_2, t_1)$ denote a same edge type. The formal
definitions about necessary standard terminologies are as follows. (1) The edges in $E$ are observed ones, and the other possible
edges among the nodes in $V$ are unobserved ones. The set of the unobserved edges in $G$ can be expressed as $U =
\{(v_1, v_2) \mid v_1, v_2 \in V \wedge v_1 \neq v_2 \wedge (v_1, v_2) \notin E\}$. (2) The predicted missing edges are the unobserved edges that are existent.
(3) Verified edge samples are the edge samples that are verified as missing edges or not, and unverified edge samples are the
edge samples that are not verified.

Formally, we define the link prediction problem in heterogeneous social networks as follows: Given a heterogeneous
social network $G = (V, E, T_o, \varphi)$, we are required to obtain a matching set $Ms = \{(e, \delta) \mid e \in U \wedge \delta \in [0, 1]\}$, where the
reasonable value $\delta$ is assigned to each unobserved edge $e$ to quantify its existent likelihood. To better explain the edge-type
disturbance for link prediction, we further collect the types of the edges in $E$ to construct the set of historical types, denotes
as $T^h_p$. The remaining edge types in $T_p$ are used to construct the new type set $T^n_p$, and $T_p = T^h_p \cup T^n_p$. Specifically, for the
homogeneous social network, it is obvious that $|T^h_p| = 1$ and $|T^n_p| = 0$. The unobserved edges in $U$ and the observed edges
in $E$ have a common edge type. However, the link prediction problem in heterogeneous social networks is more challenging.
On the one hand, the types of the observed edges in $E$ are usually multiple, and different historical edge types coexist in $T^h_p$. 

6
For example, in Figure 1, the historical types of the observed edges contain the user–user, user–song, and song–singer types. On the other hand, the type of an unobserved edge in $U$ may belong to a historical edge type in $T^h_p$ or a new edge type in $T^n_p$. For example, in Figure 1, the historical edge types contain the user–user, user–song, and song–singer types, and the singer–singer type can be considered as a new edge type. A missing edge we aim to predict can belong to an uncertain type, which has the possibility to become a user–user, user–song, song–singer, or singer–singer type.

4 DISTURBANCE-RESILIENT PREDICTION METHOD

This section introduces the proposed $DRPM$, which is outlined in Section 4.1. The structural characterizer, type differentiator, and resilient predictor are the primary modules of the $DRPM$. Section 4.2 explains the structural characterizer to learn edge representations for link prediction. Section 4.3 details the type differentiator to generate different type experts for maximizing the link prediction performances on different edge types. Section 4.4 details the resilient predictor that develops a unified resilient prediction mechanism in link prediction. In Section 4.5, the integration of the $DRPM$ is illustrated for resisting the edge-type disturbance.

4.1 Framework of $DRPM$

The framework of the $DRPM$ is shown in Figure 2. The structural characterizer, type differentiator, and resilient predictor cooperate in link prediction in heterogeneous social networks. The structural characterizer extracts the structural features of the edge samples on different types. The resistance of the edge-type disturbance for link prediction is developed from both the training and prediction stages. During the training stage, the type differentiator learns discriminative type-specific knowledge to simultaneously generate diverse type experts, which retains the type-specific features and removes the sharing features among different types. Each type expert focuses on maximizing the link prediction performance on one edge type to reduce the loss of the type-specific knowledge. During the prediction stage, the proposed resilient predictor develops a unified resilient prediction mechanism to aggregate the type-specific prediction conclusions from different type experts. In this mechanism, we not only enhance the importance of the credible type experts, but also
enhance the importance of the type experts on their familiar edge types. As a result, the integrated DRPM combines the structural characterizer, type differentiator, and resilient predictor to promote the link prediction performance in heterogeneous social networks in an edge-type disturbance manner.

4.2 Structural Characterizer

Numerous techniques have been proposed to generate low-dimensional vector embeddings for nodes in heterogeneous social networks, which includes random walk-based methods [42, 43] and graph neural network-based methods [44, 45]. The structural characterizer has a replaceable graph embedding module, and state-of-the-art graph embedding methods are used as the embedding module to promote our method performance.

Given $G = (V, E, T, \varphi)$, the feature representations of all nodes in $V$ are encoded into a matrix $X \in \mathbb{R}^{n \times d}$. $X_v$ represents the $d$-dimensional feature representation of the node $v \in V$. We extract the feature representations $X_u$ and $X_v$ of nodes $u$ and $v$, respectively. The feature representation of the edge sample $e = (u, v)$ via Hadamard operator [46] is represented as follows:

$$r_e = X_u \ast X_v$$ (1)

Here, we denote the structural characterizer as $M_f(S; \chi_f)$, where $S$ is the set of inputted edge samples, and $\chi_f$ contains the set of the learned parameters. $R_F$ represents the output set of the feature representations of all edge samples in $S$. It is expressed as:

$$R_F = \{(e, r_e) | e \in S\}$$ (2)

Unlike homogeneous social networks, the formation processes of the edges on different types are typically driven by different evolutionary mechanisms to express different type-specific features, resulting in the edge-type disturbance during the training stage. The fusion of these type-specific representations in $R_F$ is not beneficial to make most of type-specific knowledge or patterns in heterogeneous social networks, which restricts the performance improvement on some specific edge types. This requires us to be able to minimize the type correlations among the learned edge representations within different types. Thus, instead of direct extraction of $R_F$, we need to retain the type-specific features and remove the sharing features among different edge types during the training stage.

4.3 Type Differentiator

The type differentiator generates different type experts to resist the edge-type disturbance during the training stage. Cooperating with other type experts, each type expert uses a fully connected layer with a softmax function to predict the missing edges. The type differentiator takes the feature representations of edges from the structural characterizer as the input to each type expert.

We first generate $K$ type experts for selected historical types in the set $T_p^h$, and we set $K = |T_p^h|$. $M_d^{(k)}(\cdot; \chi_d^{(k)})$ represents the $k$th type expert. $1 \leq k \leq K$ and $\chi_d^{(k)}$ denotes the set of all included parameters. We introduce $\chi_f^{(k)}$ to denote the specific $\chi_f$ in the structural characterizer $M_f(S; \chi_f)$ for the $k$th type expert. The evaluation process of the existent likelihood of an edge sample $e$ by the $k$th type expert is as follow:

$$P^{(k)}(e) = M_d^{(k)}(M_f((e); \chi_f^{(k)}); \chi_d^{(k)})$$ (3)

In addition, the $k$th type expert utilizes the common cross entropy to evaluate its effectiveness in evaluating the existent likelihood of the edge sample $e$. This process is denoted as follows:

$$C^{(k)}(e) = -\left[ y_e \log(P^{(k)}(e)) + (1 - y_e) \log(1 - P^{(k)}(e)) \right]$$ (4)
Here, \( y_e \in \{0,1\} \). If the edge sample \( e \) is predicted to be a missing one by the \( k \)th type expert, \( y_e = 1 \); otherwise, \( y_e = 0 \). When the \( k \)th type expert obtains a larger \( C^{(k)}(e) \) value, it is more effective in evaluating the existent likelihood of the edge sample \( e \). We introduce the set \( S_k \) to contain the edge samples on the \( k \)th type. Considering that the \( k \)th type expert aims to maximize the link prediction performance on the \( k \)th historical type, its prediction loss is as follows:

\[
L^{(k)} \left( \hat{x}_f^{(k)}, \hat{x}_d^{(k)} \right) = \frac{1}{|S_k|} \sum_{e \in S_k} C^{(k)}(e)
\]  

(5)

For the \( k \)th type expert, we optimize the parameters \( \hat{x}_f^{(k)} \) and \( \hat{x}_d^{(k)} \) to reduce the prediction loss \( L^{(k)} \left( \hat{x}_f^{(k)}, \hat{x}_d^{(k)} \right) \), denoted as follows:

\[
\left( \hat{x}_f^{(k)}, \hat{x}_d^{(k)} \right) = \arg \min_{x_f^{(k)}, x_d^{(k)}} L^{(k)} \left( x_f^{(k)}, x_d^{(k)} \right)
\]

(6)

For an edge sample \( e \), the correlation loss between the \( k_1 \)th and \( k_2 \)th type experts is defined by cross entropy as follows:

\[
Q^{(k_1,k_2)}(e) = - \left[ p^{(k_1)}(e) \log p^{(k_2)}(e) + \left( 1 - p^{(k_1)}(e) \right) \log \left( 1 - p^{(k_2)}(e) \right) \right]
\]

(7)

\( Q^{(k_1,k_2)}(e) \) represents the knowledge correlation between the \( k_1 \)th and \( k_2 \)th type experts for the edge sample \( e \). When the value of \( Q^{(k_1,k_2)}(e) \) is larger, the difference between \( p^{(k_1)}(e) \) and \( p^{(k_2)}(e) \) is larger, and the learned knowledge between the \( k_1 \)th and \( k_2 \)th type experts are more discriminable. Considering that there are \( K \) type experts and \( |S| \) edge samples, the whole correlation loss function is further defined as follows:

\[
L_a \left( \hat{x}_f^{(c)}, \hat{x}_d^{(c)} \right) = \frac{2}{|K|(|K|-1)|S|} \sum_{k=1}^{K} \sum_{k'=k+1}^{K} \sum_{e \in S} Q^{(k,k')}(e)
\]

(8)

The maximization of the whole correlation loss is used to reduce the knowledge correlation among different type experts. To ensure each type expert to learn type-specific knowledge to maximize type-specific prediction performances [47], we need to minimize the prediction loss and maximize the correlation loss. The final loss function is defined as follows:

\[
L \left( \hat{x}_f^{(c)}, \hat{x}_d^{(c)} \right) = \frac{1}{|K|} \sum_{k=1}^{K} L^{(k)} \left( \hat{x}_f^{(k)}, \hat{x}_d^{(k)} \right) - \lambda L_a \left( \hat{x}_f^{(c)}, \hat{x}_d^{(c)} \right)
\]

(9)

Here, \( \lambda \) is the trade-off parameter between the prediction loss and the averaged correlation loss. The structural characterizer \( M_f(S; \hat{x}_f) \) simultaneously cooperates with each type expert \( M_d^{(k)}(\cdot; \hat{x}_d^{(k)}) \) to minimize the final loss \( L \left( \hat{x}_f^{(c)}, \hat{x}_d^{(c)} \right) \) by seeking the optimal parameter sets of \( \hat{x}_f^{(c)} \) and \( \hat{x}_d^{(c)} \). This process is expressed as:

\[
\left( \hat{x}_f^{(c)}, \hat{x}_d^{(c)} \right) = \arg \min_{x_f^{(c)}, x_d^{(c)}} L \left( x_f^{(c)}, x_d^{(c)} \right)
\]

(10)

The discovered parameter set represents the saddle point of the final loss function, with the learning rate denoted as \( \eta \). We also add a gradient reversal layer between the structural characterizer and type differentiator. We update \( \hat{x}_f^{(c)} \) and \( \hat{x}_d^{(c)} \) according to Equations (11) and (12) based on stochastic gradient descent.

\[
\hat{x}_f^{(c)} \leftarrow \hat{x}_f^{(c)} - \eta \frac{\partial L}{\partial \hat{x}_f^{(c)}}
\]

(11)

\[
\hat{x}_d^{(c)} \leftarrow \hat{x}_d^{(c)} - \eta \frac{\partial L}{\partial \hat{x}_d^{(c)}}
\]

(12)

Along with the type change of the unverified edge sample, the importance of the type-specific knowledge within different type experts requires suitable adjustments. Thus, we need to design an effective mechanism to aggregate the type-specific knowledge from different type experts to promote the link prediction performance. To achieve this goal, we need to consider the consistency between unverified edge samples and type experts in link prediction. In particular, we should dynamically measure the type correlation between each unverified edge sample and each type expert.
4.4 Resilient Predictor

After the type differentiator learns type-specific knowledge to generate different type experts, the resilient predictor develops a resilient prediction mechanism to fuse their type-specific prediction conclusions for performance improvement during the prediction stage. On one hand, we evaluate the reliability weight of each type expert to enhance the importance of the credible type experts. On the other hand, we analyze the type correlations among edges to enhance the importance of the type experts on their familiar types.

Inspired by the truth discovery manner in an iterative discovery manner [48], the reliability evaluation process of each type expert consists of the investment step and the collection step. These two interdependent steps are iteratively conducted until reaching the final convergence.

(1) In the investment step, the initialized reliability weight of each type expert is assumed to be fixed and equivalent. Each type expert invests its reliability weight into its prediction labels, and each prediction label collects the vote score from the invested reliability weights of all type experts. We use $a_e$ to represent a candidate prediction label of the edge sample $e$, which can be a positive or negative label. $\Psi_{a_e}$ denotes the set of type experts that provide the prediction label $a_e$. The reliability weight of the $k$th type expert is $w^k$, $A^k$ is the set of the prediction labels made by the $k$th type expert and $|A^k|$ represents the prediction number. Each prediction label $a_e$ collects vote scores though weighted voting in the following operation:

$$vote(a_e) = \left(\sum_{k \in \Psi_{a_e}} \frac{w^k}{|A^k|}\right)$$

(13)

Here, $\mu$ is an adjustment parameter to maintain the setting flexibility. $\frac{w^k}{|A^k|}$ means that the reliability weight of the $k$th type expert is equally invested into $|A^k|$ prediction labels. This reliability investment step follows the principle that the prediction labels from the credible type experts will be counted more in the weighted voting.

(2) In the collection step, each type expert collects proportional reliability weights back from their prediction labels and updates its reliability weight. This updating operation is expressed as follows:

$$w^k = \sum_{a_e \in A^k} \left(\frac{vote(a_e) \cdot \frac{w^k}{|A^k|}}{\sum_{a_e \in A^k} \frac{w^k}{|A^k|}}\right)$$

(14)

The reliability weight $w^k$ indicates the probability of the $k$th type expert providing reliable prediction labels. A higher $w^k$ value indicates that the $k$th type expert is more reliable, and its prediction labels are more likely to be accurate. In Equation (13), as the received vote values increase following a non-linear function, the reliable prediction labels receive higher vote values and contribute more to the evaluation of the reliability weight. After assigning an initialized reliability weight $w^k = 1$ for each type expert, the reliability investment step in Equation (13) and the reliability collection step in Equation (14) are iteratively conducted until the final convergence. The final output value of $w^k$ is denoted as $\bar{w}^k$. During the optimization process of the $w^k$ value, the type expert who consistently provides reliable prediction labels will receive more reliability weights and consequently have higher reliability weights assigned to them.

Further, according to the structural characterizer, we calculate the averaged representation vector of the edge on the $k$th edge type, denoted as $J(k)$. After obtaining the representation vector $J_e$ for the edge sample $e$, the type correlation $r_e^{(k)}$ based on cosine similarity is defined as follows:

$$r_e^{(k)} = \frac{J_e \cdot J(k)}{|J(k)| |J_e|}$$

(15)
\( r_{e}^{(k)} \) quantifies the correlation between the type of the unverified edge sample \( e \) and the \( k \)th edge type. A larger \( r_{e}^{(k)} \) value denotes a higher type correlation. The introduction of cosine similarity focuses on the vector difference between \( J_e \) and \( J_{(k)} \), which satisfies our needs. Other similarity calculation methods can also be considered.

Finally, when aggregating the prediction conclusions from different type experts, we not only enhance the importance of the type experts with large reliability weights but also enhance the importance of the type experts with large type correlations. After obtaining the existent likelihood \( P^{(k)}(e) \) on each \( k \)th type, the existent likelihood \( P(e) \) for the edge sample \( e \) is calculated as follows:

\[
P(e) = \sum_{k=1}^{K} \phi^k \cdot r_{e}^{(k)} \times P^{(k)}(e)
\] (16)

Considering the importance difference of different type experts, \( P(e) \) utilizes the reliability weight and the type correlation to fuse the existent likelihood of each unverified edge sample in a resilient prediction mechanism.

Generally, the resilient predictor coordinates the type specificity of unverified edge samples and type experts during the prediction stage. Based on the type differentiator, it evaluates the reliability weight of each type expert and analyzes the type correlations among edges. Finally, the resilient predictor develops a resilient prediction mechanism to optimize the existent likelihood of each unverified edge sample by aggregating nontransferable type-specific knowledge.

### 4.5 Method Integration

Based on the integration of the structural characterizer, type differentiator, and resilient predictor, the DRPM is proposed to resist the edge-type disturbance for link prediction in heterogeneous social networks. First, the structural characterizer is responsible for learning edge representations for link prediction. Second, the type differentiator learns discriminative type-specific knowledge to simultaneously generate different type experts during the training stage. Third, the resilient predictor develops a unified resilient prediction mechanism to aggregate the prediction labels from different type experts during the prediction stage.

Deploying the training process of link prediction models requires a large amount of both positive and negative samples. Given \( G = (V, E, T_\nu, \phi) \), the observed edges in \( E \) are usually considered as positive samples, and we randomly select \( |E| \) unobserved edges on the types in \( T_\nu^h \) as negative samples. Thus, we construct the training set \( S_{train} \) by combining these positive samples and negative samples, and \( Y_{train} \) denotes the corresponding label set. In addition, the unobserved edges in \( U \) are used to construct the test set, and the DRPM aims to evaluate the existent likelihoods of the edges in \( U \) to predict missing ones. With the help of the structural characterizer \( M_e(S_{train}; X^{(k)}_e) \), the type differentiator generates the \( M^{(k)}_d (\cdot; X^{(k)}_d) \) for each \( k \)th edge type in \( T_\nu^h \) by capturing type-specific features in \( R_\phi \). Second, the resilient predictor evaluates the reliability weight \( w^k \) of each \( k \)th type expert in a truth discovery manner and quantifies the type correlation \( r_{e}^{(k)} \) for each edge sample \( e \). Finally, we quantify the existent likelihood \( P(e) \) by Equation (16) and enlarge the matching set \( Ms \). The integrated procedure is processed as follows.

---

**METHOD: DRPM**

**INPUT:**
- \( G = (V, E, T_\nu, \phi) \) — A heterogeneous social network.
- \( U \) — Set of possible edges on the new types in \( T_\nu^h \).
- \( \eta \) — Learning rate.

**OUTPUT:**
- \( MS \) — Matching set.
1: $MS = \emptyset$.
2: For each training epoch do
3:     For $k = 1$ to $|T_p^b|$ do
4:         Update the parameters in the structural characterizer as:
5:             $\chi_f^{(s)} \leftarrow \chi_f^{(s)} - \eta \frac{\partial L}{\partial \chi_f^{(s)}}$
6:         Update the parameters in the type differentiator as:
7:             $\chi_d^{(s)} \leftarrow \chi_d^{(s)} - \eta \frac{\partial L}{\partial \chi_d^{(s)}}$
8:         Produce type expert $M_d(k)\left(\cdot; \chi_d^{(k)}\right)$ for the $k$th edge type in $T_p^b$.
9:     End for
10: End for
11: Calculate the reliability weight $w^k$ of each $k$th type expert by Equations (13) and (14).
12: For each edge $e$ in $U$ do
13:     For $k = 1$ to $|T_p^b|$ do
14:         Calculate the type correlation $t_e^{(k)}$ by Equations (15).
15:     End for
16: Calculate $P(e)$ by Equation (16).
17: $MS = MS + \{(e, P(e))\}$
18: End for
19: Return $MS$.

Since the feature extraction process in the structural characterizer mainly involves the dot product calculation for the nodes in the set $V$, we can approximate its cost as $O(|V|^2d)$, where $d$ represents the representation dimension. When the fully connected layer in each type expert contains $h$ hidden layers, the cost of the type differentiator to generate $K$ type experts can be denoted as $O(K|E|hd)$, where $|E|$ is the edge number. In addition, each type expert in the resilient predictor needs to go through each edge sample with the cost $O(K|E|d)$. Thus, the final cost of DRPM is $O(|V|^2d + K|E|hd + K|E|d)$. Compared with the setting of one type expert, the setting of multiple type experts increases the computation complexity. This can be considered as the limitation of our method. However, the common value setting of $K$ is usually small, the computation complexity of multiple type experts will not increase much. Additionally, during the training stage, when some edge types with small numbers of edge samples cannot provide sufficient type-specific knowledge, we can merge them as one integrated edge type and generate a type expert for this integrated edge type. Further, the calculation process in the training epoch of each type expert is relatively independent. Instead of serial computing on the CPU of a computer one by one, the training epochs of different type experts can be executed simultaneously on different CPUs of different computers in the parallel computing environment. With an increasing CPU number, the computation complexity of our method has the potential to be reduced as $O(|V|^2d + |E|hd + |E|d)$.

5 EXPERIMENTAL ANALYSIS

In Section 5.1, we introduce the experimental setup. In Section 5.2, we compare the DRPM with state-of-the-art methods to verify its performance. In Section 5.3, we investigate the influence of the setting of the sample ratio $r$, which is used to divide the test set and the training set in experiments. In addition, the DRPM resists the edge-type disturbance in link prediction during both training and prediction stages. Thus, we design comparison experiments to prove the significance of resisting the edge-type disturbance during both training and prediction stages in Section 5.4. By flexibly controlling the
degree of the edge-type disturbance in link prediction, we further conduct comparison experiments to insight into the influence of the edge-type disturbance in Section 5.5.

5.1 Experimental Setup

Datasets. The statistics of two public sampled datasets are summarized in Table 2.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#node</th>
<th>#node type</th>
<th>#edge</th>
<th>#edge type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Facebook</td>
<td>22470</td>
<td>politicians (P), governmental organizations (G), television shows (T), and companies (C)</td>
<td>171002</td>
<td>((G,T),(G,P),(G,C),(T,P),(T,C))</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>((P,C),(T,T),(G,G),(C,C),(P,P))</td>
</tr>
<tr>
<td>DBLP</td>
<td>37791</td>
<td>papers (P), authors (A), terms (T), and venues (V)</td>
<td>170794</td>
<td>((P,A),(P,V),(P,T))</td>
</tr>
</tbody>
</table>

- **Facebook** [49]. This is a page-page network of verified Facebook sites, where nodes represent official Facebook pages and links indicate mutual likes between sites. The node features are extracted from site descriptions provided by page owners, summarizing the purpose of each site. The network was collected in November 2017 using the Facebook Graph API and includes pages from four Facebook-defined categories: politicians (P), governmental organizations (G), television shows (T), and companies (C).

- **DBLP** [50]. The DBLP computer science bibliography serves as the primary online reference for bibliographic information. Four research lines are considered for its collected publications: database, data mining, machine learning, and information retrieval. It lists relevant studies in the field using author name and publication year. Their types of the network nodes contain papers (P), authors (A), terms (T), and venues (V).

Comparison methods. Five state-of-the-art prediction methods and three typical link prediction methods are as follows.

- **PME** [31]. Projected metric embedding (PME) is a novel embedding model that extracts the heterogenous information network embedding of links. This model is built on metric learning, effectively capturing both first-order and second-order proximities in a unified manner. We further use the heterogeneous softmax function to evaluate the existent likelihoods of edges. All the hyperparameters are tuned to perform the best. It sets batch size \(B = 480\), margin \(m = 2\), and learning rate \(\alpha = 0.001\).

- **SEAL** [33]. SEAL is a Graph Neural Network (GNN) framework designed for link prediction on heterogeneous social networks by learning heuristics from local subgraphs. It extracts local subgraphs to retain valuable information associated with link existence and employs a GNN to learn general graph structure features for accurate link prediction. The hop number, \(h\), is set as 1, which aligns with theoretical findings indicating that the most pertinent information resides within local structures.

- **SLiCE** [34]. The SLiCE automatically learns the composition of different metapaths that characterize the context for a specific task without the need for any pre-defined metapaths. The node embeddings are set to a dimension of 128. A skip-gram based random walk approach is used to encode context subgraphs with global node features. Both the pre-training and fine-tuning steps are trained for 10 epochs with a batch size of 128, employing the cross-entropy loss function. Both the number of contextual translation layers and number of self-attention heads are set to 4.

- **GATNE** [35]. GATNE is developed for representation learning in attributed multiplex heterogeneous networks. This model can support both transductive learning and embedding learning and handle large-scale heterogeneous social
networks in link prediction. The number of maximum epochs is set to 50. The coefficient $\alpha_r$ and $\beta_r$ are all set to 1 for every edge type $r$. The default setting of Adam optimizer is used and the learning rate is set to $\eta = 0.001$.

- **HeGAN** [36]. The heterogeneous general adversarial network (HeGAN) is a new framework that uses generative adversarial networks for heterogeneous network embedding. It learns the node distribution to generate better negative samples and perceive the relationship to capture the rich semantics on the heterogeneous network. It runs a grid search over $\{1, 5, 10, 15, 20\}^2$. The optimal performance is obtained when $d$ is 64 and 128. The model achieves optimal performance when the variance of Gaussian distribution $\sigma^2 = 1.0$.

- **SVM** [51]. Support vector machine (SVM) is a supervised binary classification model that distinguishes between positive and negative class samples by finding the maximum interval classification plane. The application of kernel functions allows SVM to be used efficiently in the nonlinearly differentiable case. Since SVM only needs to save support vectors, it consumes less memory and is faster in the prediction process. It adopts the linear kernel, and its penalty coefficient is 50.

- **LR** [52]. Logistic regression (LR) is a commonly used classification model, particularly for binary classification tasks. It establishes a relationship between predictor variables and a categorical response variable. Logistic regression can estimate the probability of belonging to a specific level of the categorical response based on a given set of predictors. It has the advantages of ease of implementation, good interpretation, and easy scalability. The parameter of solver in LR selects the optimization algorithm $lbfgs$.

- **RF** [53]. Random forest (RF) is a robust machine learning technique introduced for classification and regression tasks. It consists of an ensemble of decision trees, where each tree is built from a bootstrap sample of the input data. One practical advantage of RF is the ability to automatically predict the probability of a link sample belonging to a specific class.

**Parameter settings.** The settings of the comparison methods SEAL, GATNE, HeGAN, PME, and SLICE remain consistent with their original configurations. Based on extracted preliminary feature representations of links, SVM, LR, and RF are developed to distinguish positive and negative samples based on typical settings [51-53]. In the structural characterizer section of the DRPM, the embedding dimension is uniformly set as 64, and the metric learning method is used as an example to obtain preliminary node embeddings. For all the type experts, we use the same batch size of 32 instances in the training stages. The training epoch is 200. During the prediction stage, the iteration number is 300 and the weight parameter $\alpha$ is 2/3.

We set the learning rate $\eta$ as 0.001 for all models. The final loss function is set with $\lambda = 1$. The initialized reliability weight of each type expert is 1. The weighted or directed datasets are simplified as general networks by ignoring their weights and directions. In the Facebook dataset, according to the edge numbers of different types, we reserve four types with relatively close orders of magnitude, namely $(T,T),(G,G),(C,C),$ and $(P,P)$. The numbers of the edges on these types are 12991, 81367, 20145, 36909. In the DBLP dataset, the numbers of the edges on the types $(P,A),(P,V),$ and $(P,T)$ are 41794, 14376, and 114624, respectively. For comparisons with other link prediction methods in a unified manner, we use the Node2vec as the basic embedding module in the structural characterizer [46]. We utilize a bootstrapping approach to obtain initial edge representations by leveraging the connectivity structure between nodes in the underlying network. The settings of the embedding module with different graph embedding methods are further analyzed in Section 5.7.

**5.2 Performance Comparison**

To ensure a unified comparison, we begin by constructing the training and test sets. The sample ratio is denoted by $r$. We randomly select $r \times |E|$ edges from the observed edge set $E$ as positive samples in the training set, while the remaining observed edges are considered as positive samples in the test set. Additionally, for the training set, we randomly select
$r \times |E|$ unobserved edges from $U$ as negative samples. Similarly, for the test set, we randomly select $(1-r) \times |E|$ unobserved edges from the remaining unobserved edges in $U$ as negative samples. We introduce three acknowledged evaluation indices to verify the link prediction performances [34-36]: $AUC$, $Accuracy$, and $Precision$. $AUC$ is the primary metric to center on the whole prediction performance of all samples in the test set, while $Precision$ and $Accuracy$ focus on the prediction performance of top-ranked samples in the test set. In addition, our method aims to promote the whole link prediction performance on all edge types. The calculation of $AUC$, $Accuracy$, and $Precision$ ignores the type difference and considers the whole link prediction performances on all edge types. On the Facebook and DBLP datasets, the values of $AUC$, $Accuracy$, and $Precision$ obtained by the DRPM and comparison methods are shown in Table 3. The Facebook dataset has the edge types of $(T, T), (G, G), (C, C)$, and $(P, P)$, and the DBLP dataset has the edge types of $(P, A), (P, V)$, and $(P, T)$. The link prediction performance on each edge type is also presented. The highest values are emphasized in bold type.

### Table 3: Link prediction performance of deep learning and traditional methods on two datasets

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Method</th>
<th>$AUC$</th>
<th>$Accuracy$</th>
<th>$Precision$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$(T, T), (G, G), (C, C), (P, P)$</td>
<td>$(T, T), (G, G), (C, C), (P, P)$</td>
<td>$(T, T), (G, G), (C, C), (P, P)$</td>
</tr>
<tr>
<td>Facebook</td>
<td>DRPM</td>
<td>0.90</td>
<td>0.81</td>
<td>0.74</td>
</tr>
<tr>
<td></td>
<td>SEAL</td>
<td>0.83</td>
<td>0.79</td>
<td>0.76</td>
</tr>
<tr>
<td></td>
<td>HeGAN</td>
<td>0.58</td>
<td>0.57</td>
<td>0.47</td>
</tr>
<tr>
<td></td>
<td>GATNE</td>
<td>0.70</td>
<td>0.67</td>
<td>0.73</td>
</tr>
<tr>
<td></td>
<td>PME</td>
<td>0.50</td>
<td>0.61</td>
<td>0.66</td>
</tr>
<tr>
<td></td>
<td>SLICE</td>
<td>0.51</td>
<td>0.53</td>
<td>0.67</td>
</tr>
<tr>
<td></td>
<td>SVM</td>
<td>0.56</td>
<td>0.59</td>
<td>0.54</td>
</tr>
<tr>
<td></td>
<td>LR</td>
<td>0.52</td>
<td>0.51</td>
<td>0.45</td>
</tr>
<tr>
<td></td>
<td>RF</td>
<td>0.57</td>
<td>0.56</td>
<td>0.17</td>
</tr>
</tbody>
</table>

| DBLP    | DRPM   | 0.84  | 0.81       | 0.88        |
|         | SEAL   | 0.80  | 0.76       | 0.64        |
|         | HeGAN  | 0.55  | 0.53       | 0.45        |
|         | GATNE  | 0.67  | 0.59       | 0.58        |
|         | PME    | 0.43  | 0.41       | 0.73        |
|         | SLICE  | 0.54  | 0.49       | 0.53        |
|         | SVM    | 0.52  | 0.51       | 0.50        |
|         | LR     | 0.57  | 0.55       | 0.42        |
|         | RF     | 0.52  | 0.51       | 0.43        |

Among the nine prediction methods, the DRPM exhibits the best link prediction performances in most cases. In Table 3, in the $Precision$ evaluation, the bad link prediction performances on the edge types $(T, T)$ and $(C, C)$ degrade the whole link prediction performance of DRPM on Facebook. Generally, compared with SEAL, HeGAN, GATNE, SLICE, PME, SVM, LR, and RF, the DRPM has the higher values of $AUC$, $Accuracy$, and $Precision$ on the Facebook and DBLP datasets. Due to the heterogeneity of the edge types in the training and test sets, the inherent edge-type disturbance degrades the effectiveness of comparison methods during both the training and prediction stages. The comparison methods trained on the verified edge samples from different types tend to learn lots of type-specific knowledge in the training set, and their type-specific predictions may be contradictory for unverified edge samples with unknown types in
the test set. Different from the comparison methods, the DRPM resists the edge-type disturbance for link prediction during both the training and prediction stages. The DRPM not only learns type-specific knowledge to simultaneously generate type experts during the training stage, but also develops a resilient prediction mechanism to aggregate type-specific prediction conclusions during the prediction stage. Therefore, the DRPM has the ability to get improved performances than comparison methods.

In addition, the performances of SEAL, HeGAN, GATNE, SLICE, PME, SVM, LR, and RF are easy to fluctuate on the Facebook and DBLP datasets. For example, the Accuracy and Precision values of SEAL on the Facebook dataset is excellent, while these values on the DBLP dataset is degraded. Since the DRPM fuses the prediction conclusions from the type experts on different edge types, its comprehensive prediction process is robust to keep excellent link prediction performances on different datasets. Further, deep learning methods have outperformed traditional machine learning methods by leveraging their superior feature extraction capabilities. The link prediction performances of DRPM, SEAL, HeGAN, GATNE, SLICE, and PME are better than these of SVM, LR, and RF in most cases on the Facebook and DBLP datasets.

5.3 Parametric Analysis

![Graphs showing changes in Accuracy, AUC, and Precision values obtained by DRPM on the Facebook and DBLP datasets.](image)

Figure 3: Changes in Accuracy, AUC, and Precision values obtained by DRPM on the Facebook and DBLP datasets. The shadows in the background emphasize the increase in evaluation values. The faster the growth velocity, the darker the shadow region becomes.

The sample ratio, r, is a critical parameter for the DRPM. It determines the division between the test set and the training set. We analyze how different r settings affect the performance of the DRPM. In the construction of the training and test sets, a larger r value denotes a larger size of the training set. Specifically, more observed edges and unobserved edges are selected as positive samples and negative samples in the training set, respectively. We adjust the setting of r to analyze the performance change of the DRPM. With different r values, the corresponding performance changes of the DRPM on the Facebook and DBLP datasets are shown in Figure 3.

In Figure 3, the link prediction performance of the DRPM is positively correlated with the γ value, suggesting the prediction rationality of the DRPM. First, with the increasement in the r value, the link prediction performances of the DRPM increase. Enlarging the size of the training set enables that each type expert has a larger likelihood to learn good type-
specific knowledge. The Accuracy, AUC, and Precision values are increasing in most cases on the Facebook and DBLP datasets. Especially, the Accuracy, AUC, and Precision values obtained by the DRPM at $r = 0.9$ and $r = 0.95$ are obviously larger than these values at $r = 0.5$ and $r = 0.55$ on the Facebook and DBLP datasets. Second, on the Facebook and DBLP datasets, the Accuracy, AUC, and Precision values decrease in most cases, and the darker shadow region with a faster growth velocity occurs at a smaller $r$ value. When enough verified edge samples have been added to the training set, the increase of verified edge samples cannot effectively improve the link prediction performance. On the Facebook and DBLP datasets, the Accuracy, AUC, and Precision values obtained by the DRPM at $r = 0.9$ and $r = 0.95$ are obviously larger than these values at $r = 0.5$ and $r = 0.55$ on the Facebook and DBLP datasets.

5.4 Significance of Resisting the Edge-type Disturbance

This research recognizes the issue of the edge-type disturbance for link prediction in heterogeneous social networks. Due to the mixture of these nontransferable type-specific features from different edge types in the training and test sets, the edge-type disturbance is provoked during the training and prediction stages in the link prediction process. Our proposed DRPM resists the edge-type disturbance to promote the link prediction performance during both training and prediction stages. Therefore, we verify the significances of resisting the edge-type disturbance by the DRPM during both training and prediction stages in this section.

5.4.1 Significance of Resisting the Edge-type Disturbance during the Training stage

To demonstrate the significance of resisting the edge-type disturbance during the training stage, we design a DRPM based comparison method. To enable our proposed DRPM to resist the edge-type disturbance during the training stage, we require the type differentiator to minimize the prediction loss and maximize the correlation loss simultaneously. However, even without this maximization process to reduce the correlation among type experts, our method can still be trained to generate different type experts to distinguish between negative and positive samples in the test set. Thus, we design a DRPM based comparison method, named DRPM$^S$. Different from DRPM, the DRPM$^S$ removes the maximization process of correlation loss in the type differentiator by setting $\lambda = 0$ in the final loss function. We set $r = 0.95$ as an example. Figure 4 showcases the performance comparisons between DRPM and DRPM$^S$.

![Graphs showing performance comparisons between DRPM and DRPM$^S$ for Facebook and DBLP datasets.](a) Facebook (b) DBLP
In Figure 4, compared with $DRPM^S$, the $DRPM$ owns significant performance improvements in link prediction in most cases. Although the $DRPM$ has the degraded $Precision$ value on Facebook, it promotes the $Accuracy$ value by 5.02% and 4.88%, and the $AUC$ value by 4.57% and 1.19% on Facebook and DBLP, respectively. The $Precision$ value focuses the prediction result of positive edge samples, which may result in the performance uncertainty on Facebook in some cases. Without the maximization process of correlation loss in the type differentiator, the $DRPM^S$ only minimizes the prediction loss during the training stage, where each type expert learns general knowledge. By contrast, with the supplemental help of the maximization process of the correlation loss, the $DRPM$ enables each type expert to learn discriminative type-specific knowledge and keep the type independence among trained type experts. It requires the trained type experts to reduce their type correlations by removing the sharing features among all edge types, which enable $DRPM$ to resist the potential edge-type disturbance during the training stage. The comparison results prove the importance of the type differentiator in the $DRPM$, and resisting the edge-type disturbance during the training stage is beneficial to promoting the link prediction performances in heterogeneous social networks.

5.4.2 Significance of Resisting the Edge-type Disturbance during the Prediction Stage

To resist the edge-type disturbance during the prediction stage, our proposed $DRPM$ designs a resilient prediction mechanism to fuse the prediction conclusions from different type experts for each unverified edge sample in a truth discovery manner. Without this resilient prediction mechanism in the resilient predictor, our method can still be trained to distinguish between negative and positive samples in the test set. Thus, we further design a $DRPM$ based comparison method, which is termed $DRPM^D$. The $DRPM^D$ only removes the resilient prediction mechanism in the resilient predictor by assigning each type expert a fixed and equivalent reliability weight. The performance comparisons between the $DRPM$ and $DRPM^D$ are shown in Figure 5.

Figure 5 illustrates that the $DRPM$ has the improved link prediction performances on the Facebook and DBLP datasets, significantly outperforming the $DRPM^D$. Compared with the $DRPM^D$, the $DRPM$ exhibits the $Accuracy$ improvements of
6.79% and 7.32%, AUC improvements of 12.67% and 2.38%, and Precision improvements of 2.32% and 0.57% on the Facebook and DBLP datasets, respectively. This strongly indicates that resisting the edge-type disturbance during the prediction stage effectively improves the link prediction performance in heterogeneous social networks. After the type differentiator learns type-specific knowledge to generate different type experts, the resilient predictor in the DRPM fuses their type-specific prediction conclusions during the prediction stage though a resilient prediction mechanism. On one hand, the resilient predictor evaluates the reliability weight of each type expert to enhance the importance of the credible type experts. Each type expert invests its reliability weight into its prediction labels, and the prediction label collects its vote score from the invested reliability weights by the type experts. On the other hand, the resilient predictor analyzes the type correlations among edges to enhance the importance of the type experts on their familiar types. We calculate the type correlation between the unverified edge samples and the type experts to adaptively calculate the existent likelihood of each unverified edge sample. However, without the resilient prediction mechanism in the resilient predictor, the DRPM$^D$ loses the ability to differentiate between the reliability weights of different type experts, which severely degrades its link prediction performances. Therefore, the design of the resilient predictor to resist the edge-type disturbance during the prediction stage is important and beneficial in the DRPM.

### 5.5 Insight Analysis of Edge-type Disturbance

In this section, we design comparison experiments to illustrate the influence of the edge-type disturbance in the link prediction process. To flexibly control the degree of the edge-type disturbance, we first divide the edge types into the sets of historical types and new types according to their edge numbers. For the Facebook dataset, we use the edge types $(T, T), (G, C), (C, C)$ and $(P, P)$ as historical types, and the remaining edge types $(G, T), (G, P), (G, C), (T, P), (T, C)$ and $(P, C)$ are used as new types. For the DBLP dataset, $(P, A)$ and $(P, V)$ are used as historical types, and $(P, T)$ is used as the new type. Second, we create a balanced set $S'$ by treating observed links on the new types as positive samples and randomly selecting equivalent unobserved links within the same types as negative samples. Third, we set $r = 0.5$ and divide the samples on the historical types into the initial training and test sets. The corresponding types of the samples in $S'$ have not been covered in the initial training and test sets. To adjust the edge-type disturbance in the training and test sets, we randomly select $\xi|S'|$ samples from $S'$ and add them equally into the initial training set and the initial test set. The increase in the $\xi$ value promotes the heterogeneity of edge types, resulting in the degree increase of the edge-type disturbance in the training and test sets. The performance changes of the DRPM and state-of-the-art comparison methods on the Facebook and DBLP datasets are shown in Figure 6 and Figure 7, respectively.
In Figure 6 and Figure 7, the existence of the edge-type disturbance phenomenon has a negative influence on the link prediction performance in heterogeneous social networks, which further suggests the importance of resisting the edge-type disturbance in the DRPM. The existence of the edge-type disturbance severely restricts the link prediction performances of the comparison methods in heterogeneous social networks. As the $\xi$ value increases, more and more samples on new types are added to enlarge the training set to enable the comparison methods to achieve improved link prediction performances. However, the overall link prediction performances of the comparison methods SEAL, GATNE, HeGAN, PME, and SLiCE have no obvious improvements on the Facebook and DBLP datasets, and their Accuracy, AUC, and Precision values have no significant increases in most cases. The increase of the samples on new types improves the heterogeneity of edge types in the training and test sets. This leads to the degree increase in the edge-type disturbance to weaken the training advantage brought by the enlargement of the training set. Especially, the Accuracy, AUC, and Precision values of the comparison methods even fall back in some cases. Furthermore, resisting the edge-type disturbance is beneficial to promoting the performance robustness. The DRPM has a strong ability to resist the edge-type disturbance during the training and prediction stages. Along with the increase in $\xi$, the DRPM not only obtains better link prediction performances than the comparison methods SEAL, GATNE, HeGAN, PME, and SLiCE, but also has an obvious increase in the link prediction performance. Further, the performance advantage of the DRPM continuously increases. In terms of the AUC, Precision, and Accuracy values, there is an increasing performance gap between the DRPM and the optimal comparison method on the Facebook and DBLP datasets. When more samples on new types are added into the training and test sets, the DRPM has larger room to exert its advantage to resist the edge-type disturbance in link prediction.

### 5.6 Prediction Performance of Each Type Expert on Each Edge Type

Since our DRPM is proposed to learn for each edge type separately by training separated type experts, each type expert indeed learns efficiently for its respective edge type. To show this empirically, we analyze the prediction performance of each type expert on each edge type in this section. We use the experiment setting in Section 5.5 as an example. We consider the edge types $\langle T, T \rangle$, $\langle G, G \rangle$, $\langle C, C \rangle$ and $\langle P, P \rangle$ on the Facebook dataset and the edge types $\langle P, A \rangle$ and $\langle P, V \rangle$ on the DBLP dataset. During the training stage, four type experts are generated for the respective edge types $\langle T, T \rangle$, $\langle G, G \rangle$, $\langle C, C \rangle$ and $\langle P, P \rangle$ on the Facebook dataset, and two type experts are generated for the respective edge types $\langle P, A \rangle$ and $\langle P, V \rangle$ on.
the DBLP dataset. On the Facebook and DBLP datasets, the prediction performance of each type expert on each edge type is shown in Table 4.

Table 4: The prediction performance of each type expert on each edge type

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Type Expert</th>
<th>Prediction Type</th>
<th>AUC</th>
<th>Precision</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Facebook</td>
<td>Type Expert for ((T,T))</td>
<td>((T,T))</td>
<td>0.7859</td>
<td>0.6769</td>
<td>0.7367</td>
</tr>
<tr>
<td></td>
<td></td>
<td>((G,G))</td>
<td>0.6252</td>
<td>0.5992</td>
<td>0.6845</td>
</tr>
<tr>
<td></td>
<td></td>
<td>((C,C))</td>
<td>0.6168</td>
<td>0.4724</td>
<td>0.6225</td>
</tr>
<tr>
<td></td>
<td></td>
<td>((P,P))</td>
<td>0.5942</td>
<td>0.6215</td>
<td>0.5242</td>
</tr>
<tr>
<td></td>
<td>Type Expert for ((G,G))</td>
<td>((G,G))</td>
<td>0.7425</td>
<td>0.6436</td>
<td>0.6215</td>
</tr>
<tr>
<td></td>
<td></td>
<td>((C,C))</td>
<td>0.7144</td>
<td>0.6228</td>
<td>0.5852</td>
</tr>
<tr>
<td></td>
<td></td>
<td>((P,P))</td>
<td>0.6572</td>
<td>0.5452</td>
<td>0.5464</td>
</tr>
<tr>
<td></td>
<td>Type Expert for ((C,C))</td>
<td>((T,T))</td>
<td>0.6827</td>
<td>0.5957</td>
<td>0.6215</td>
</tr>
<tr>
<td></td>
<td></td>
<td>((G,G))</td>
<td>0.6254</td>
<td>0.6103</td>
<td>0.5927</td>
</tr>
<tr>
<td></td>
<td></td>
<td>((C,C))</td>
<td>0.7722</td>
<td>0.7247</td>
<td>0.7164</td>
</tr>
<tr>
<td></td>
<td></td>
<td>((P,P))</td>
<td>0.7154</td>
<td>0.6827</td>
<td>0.7027</td>
</tr>
<tr>
<td></td>
<td>Type Expert for ((P,P))</td>
<td>((T,T))</td>
<td>0.6324</td>
<td>0.6225</td>
<td>0.6517</td>
</tr>
<tr>
<td></td>
<td></td>
<td>((G,G))</td>
<td>0.6735</td>
<td>0.7063</td>
<td>0.6813</td>
</tr>
<tr>
<td></td>
<td></td>
<td>((C,C))</td>
<td>0.7019</td>
<td>0.6356</td>
<td>0.6948</td>
</tr>
<tr>
<td></td>
<td></td>
<td>((P,P))</td>
<td>0.7627</td>
<td>0.7458</td>
<td>0.7749</td>
</tr>
<tr>
<td>DBLP</td>
<td>Type Expert for ((P,A))</td>
<td>((P,A))</td>
<td>0.8214</td>
<td>0.7271</td>
<td>0.7428</td>
</tr>
<tr>
<td></td>
<td></td>
<td>((P,V))</td>
<td>0.7413</td>
<td>0.6963</td>
<td>0.7047</td>
</tr>
<tr>
<td></td>
<td>Type Expert for ((P,V))</td>
<td>((P,A))</td>
<td>0.7659</td>
<td>0.7193</td>
<td>0.6972</td>
</tr>
<tr>
<td></td>
<td></td>
<td>((P,V))</td>
<td>0.7938</td>
<td>0.7764</td>
<td>0.7528</td>
</tr>
</tbody>
</table>

According to Table 4, each type expert gets the best prediction performance on the respective one among all edge types. In cooperation with the structural characterizer, each type expert learns type-specific knowledge to maximize the prediction performance on the corresponding edge type. On the Facebook dataset, the type experts for the edge types \((T,T)\), \((G,G)\), \((C,C)\) and \((P,P)\) obtain the largest \textit{AUC}, \textit{Accuracy}, and \textit{Precision} values on \((T,T)\), \((G,G)\), \((C,C)\) and \((P,P)\), respectively. Meanwhile, on the DBLP dataset, the type experts for the edge types \((P,A)\) and \((P,V)\) obtain the largest \textit{AUC}, \textit{Accuracy}, and \textit{Precision} values on \((P,A)\) and \((P,V)\), respectively. In addition, the best prediction performance on an edge type is obtained by its corresponding one among all type experts. For example, compared with the type expert for the edge type \((P,V)\), the type expert for the edge type \((P,A)\) obtains a larger \textit{AUC}, \textit{Accuracy}, and \textit{Precision} values on the edge type \((P,A)\).

Further, the type experts remain beneficial to promoting the prediction performances on noncorresponding type edges. For example, on the Facebook dataset, the type expert for the edge type \((T,T)\) also has satisfied prediction performances on the edge types \((G,G)\), \((C,C)\) and \((P,P)\), which can provide beneficial indication to promote the link prediction performances on noncorresponding type edges. Therefore, the resilient predictor in our \textit{DRPM} develops a resilient prediction mechanism to fuse the type-specific prediction conclusions to promote the link prediction performance.
### 5.7 Performances of DRPM under different Graph Embedding Methods

Table 5: Performances of DRPM under different graph embedding methods in the embedding module

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Model</th>
<th>AUC</th>
<th>Precision</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Facebook</td>
<td>$\text{DRPM}_{\text{Node2vec}}$</td>
<td>0.90</td>
<td>0.81</td>
<td>0.70</td>
</tr>
<tr>
<td></td>
<td>$\text{DRPM}_{\text{GNN}}$</td>
<td>0.48</td>
<td>0.58</td>
<td>0.16</td>
</tr>
<tr>
<td></td>
<td>$\text{DRPM}_{\text{PME}}$</td>
<td>0.52</td>
<td>0.08</td>
<td>0.16</td>
</tr>
<tr>
<td>DBLP</td>
<td>$\text{DRPM}_{\text{Node2vec}}$</td>
<td>0.84</td>
<td>0.84</td>
<td>0.82</td>
</tr>
<tr>
<td></td>
<td>$\text{DRPM}_{\text{GNN}}$</td>
<td>0.50</td>
<td>0.29</td>
<td>0.57</td>
</tr>
<tr>
<td></td>
<td>$\text{DRPM}_{\text{PME}}$</td>
<td>0.44</td>
<td>0.21</td>
<td>0.42</td>
</tr>
</tbody>
</table>

In our proposed DRPM, the structural characterizer has a replaceable embedding module, and we use the typical random walk based method Node2vec [46] and the graph neural network based methods GNN [46] and PME [31] in the embedding module. The initial settings in Section 5.2 are used in this section. Table 5 shows the performances of DRPM under different graph embedding methods in the embedding module. The performance of $\text{DRPM}_{\text{Node2vec}}$ is better than the performances of $\text{DRPM}_{\text{GNN}}$ and $\text{DRPM}_{\text{PME}}$, which is suitable to be used in this experiment. $\text{DRPM}_{\text{Node2vec}}$ obtains the larger $\text{AUC}$, $\text{Accuracy}$, and $\text{Precision}$ values than these of $\text{DRPM}_{\text{GNN}}$ and $\text{DRPM}_{\text{PME}}$ on the datasets of Facebook and DBLP. The representation learning process of Node2vec treats each link type equally, which provides the unbiased representations for each type expert to maximize the corresponding link prediction performance.

### 6 CONCLUSIONS AND FUTURE WORK

In this research, we introduce the issue of the edge-type disturbance for link prediction in heterogeneous social networks and propose a novel disturbance-resilient prediction method to solve it. Our method not only explains the edge-type disturbance phenomenon, but also exploits the edge-type disturbance to promote the link prediction performances. It integrates a structural characterizer, type differentiator, and resilient predictor in a disturbance-resilient manner. First, the structural characterizer is responsible for extracting the structural features of edges for link prediction. With the help of the structural characterizer, the type differentiator learns discriminative type-specific knowledge to generate different type experts. Further, the resilient predictor develops a unified resilient prediction mechanism to aggregate the prediction labels from different type experts. To resist the edge-type disturbance in link prediction, our method not only enhances the importance of the credible type experts, but also enhances the importance of the type experts on their familiar edge types. Our investigation on various real-world datasets proves the importance of the edge-type disturbance in link prediction, and its further application method outperforms state-of-the-art methods.

Our work serves as a basis for exploring additional methods for resisting edge-type disturbance and improving link prediction performance in heterogeneous social networks. There are several promising directions for further exploration. One possibility is to extend our study to directed or weighted heterogeneous social networks by considering weight and direction information. Also, we can attempt to leverage multi-modal attribute information of nodes and links, such as textual and image descriptions, to resist edge-type disturbance. Another possibility is to analyze the performance fluctuation in the unbalanced multi-class setup to further enrich the application scenarios of our method. In addition to link prediction, the research line of graph learning involves various other subtasks including node classification, graph classification, and community detection. Our research presents a promising approach to address the issue of edge-type disturbance in these subtasks in heterogeneous networks.
ACKNOWLEDGMENTS
This work is supported by the National Natural Science Foundation of China (62006089), Macau Young Scholars Program, National Key D&R Program of China (2019YFB1600704), Science and Technology Development Fund, Macau SAR (0068/2020/AGJ, SKL-IOTSC(UM)-2021-2023), GDST (2020B1212030003), MYRG2022-00192-FST, National Natural Science Foundation of China (62102265), Natural Science Foundation of Guangdong Province of China (2022A1515011474), Open Research Fund from Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ) (GML-KF-22-29).

REFERENCES