

Automatic Self-supervised Learning for Social Recommendations

Xin He(hexin20@mails.jlu.edu.cn)^a, Wenqi Fan(wenqi.fan@polyu.edu.hk)^b,
Ying Wang(wangying2010@jlu.edu.cn)^c,
Mingchen Sun(mcsun20@mails.jlu.edu.cn)^c, Xin Wang^{a,*}

^a *The Department of Computer Science and Technology, the School of Artificial Intelligence (SAI), at Jilin University, 2699 Qianjin Street, Changchun, Jilin, China*

^b *The Hong Kong Polytechnic University, Kowloon, Hong Kong, Hong Kong, China*

^c *The Department of Computer Science and Technology, College of Computer Science and Technology, at Jilin University, 2699 Qianjin Street, Changchun, Jilin, China*

Abstract

In recent years, researchers have leveraged social relations to enhance recommendation performance. However, most existing social recommendation methods require carefully designed auxiliary social tasks tailored to specific scenarios, which depend heavily on domain knowledge and expertise. To address this limitation, we propose Automatic Self-supervised Learning for Social Recommendations (AusRec), which integrates multiple self-supervised auxiliary tasks with an automatic weighting mechanism to adaptively balance their contributions through a meta-learning optimization framework. This design enables the model to automatically learn the optimal importance of each auxiliary task, thereby enhancing representation learning in social recommendations. Extensive experiments on several real-world datasets demonstrate that AusRec consistently outperforms state-of-the-art baselines

*Corresponding author

Email address: xinwang@jlu.edu.cn (Xin Wang)

by 3.3%–10.7% in Recall@10 and 1.4%–7.1% in NDCG@10, validating its effectiveness and robustness across different recommendation scenarios. The code is in: <https://github.com/hexin5515/AusRec>.

Keywords: Graph Neural Networks, Meta Learning, Recommendation Systems, Self-supervised Learning

1. Introduction

In the age of information overload, people are increasingly overwhelmed by the vast amount of information available online, which motivates intelligent information systems to help select the relevant information that can match users’ personal interests from the huge amount of data. As representative intelligent information systems, recommendation systems aim to
5 mitigate information overload and are widely applied across various online platforms [1, 2, 3], such as online shopping (e.g., Amazon¹ and Taobao²) and social media (e.g., Facebook³, Twitter⁴, and Weibo⁵). More specifically, recommendation systems perform to suggest a list of items to users that match
10 users’ interests and are more likely to be interacted by users. As one of the most widely used technologies in existing advanced recommendation systems, collaborative filtering (CF) is based on the assumption that users who have similar preferences in the history are likely to have similar preferences in the future [4]. In general, user-item historical interactions are key for expressing

¹<https://www.amazon.cn>

²<https://www.taobao.com>

³<http://www.facebook.com>

⁴<http://twitter.com>

⁵<https://weibo.com>

15 users’ preferences and can be encoded to learn the representations of both users and items in recommendation systems [5, 6, 7].

Over the past decade, the development of online social networking services (e.g., Facebook, Weibo, Douban) has been significant, recent studies show that social relations between users can be incorporated into enhancing the representation learning of users for recommendation systems [8, 2, 9, 10, 11],
20 which is motivated by the observation that users’ preferences are likely to be influenced by their socially connected friends (e.g., classmates, colleagues, family members, etc.) [12]. For example, SocialMF [13] and GraphRec [9] incorporate first-order social neighbors to enhance user representation learning.
25 DiffNet [10] proposes an influence diffusion method to stimulate social influence propagation by considering high-order social information. ESRF [11] proposes to utilize triangle relations among users in social recommendations. We illustrate some representative social relations tasks in most existing social recommendations in Figure 1, such as one-hop relations among users, higher-
30 order relations among users, triangular social relations, meta-path-based social relations, etc. Intuitively, in social networks, users can share more similar preferences towards items among social triangle relations (with strong ties) than meta-path-based social relations (with weak ties). In summary, most existing social recommendation methods are required to carefully develop
35 various auxiliary social relation tasks (e.g., high-order, social triangle, etc.) to enhance user representation learning by considering the characteristics of social networks/graphs for improving social recommendations [14].

In recent years, Self-Supervised Learning (SSL) techniques have become an emerging learning paradigm that exploits rich unlabeled data in en-

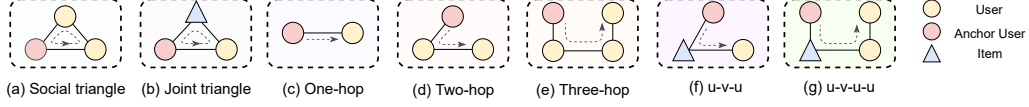


Figure 1: Examples of various social relations in social networks. u indicates a user, and v indicates an item.

hancing representation learning via designing appropriate auxiliary tasks in various research fields, such as CV and NLP [15, 16]. Owing to their powerful capabilities in representation learning with unlabeled data, self-supervised learning has been successfully incorporated into recommendation systems [17, 18, 19, 20]. More specifically, most existing SSL-based recommendation methods propose to design a few SSL auxiliary tasks to assist the primary (target) task (i.e., item recommendations). For example, SGL [17] presents a general framework for self-supervised graph learning with stochastic augmentations on user-item graphs (e.g., node/edge dropout) for recommendations. DHCN [18] integrates self-supervised learning into network training by maximizing the mutual information from two channels. MHCN [19] uses social triangle relations to conduct contrastive learning auxiliary tasks for improving social recommendation.

Despite the aforementioned successes, they do not fully take advantage of self-supervision signals to enhance representation learning under the self-supervised learning paradigm in social recommendations due to their fundamental challenges. **First**, the adopted Self-Supervised Auxiliary tasks (SS-A tasks for short) must be carefully selected, requiring substantial domain knowledge and expertise in primary task (e.g., complicated recommendation scenarios and datasets’ characteristics), since different SS-A tasks vary in im-

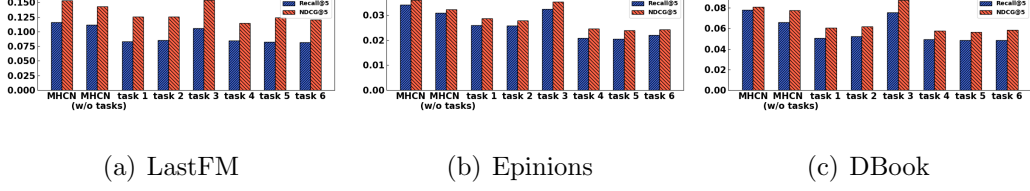


Figure 2: Performance of multiple self-supervised auxiliary tasks for the advanced social recommendation method MHCN [19] under Recall@5 and NDCG@5 metrics over three datasets (i.e., LastFM, Epinions, and DBook). MHCN(w/o tasks) denotes the MHCN model without any auxiliary tasks enhanced. The versions of MHCN model with single self-supervised auxiliary task enhanced are represented by ssl 1 to 6, including social triangle relation, joint triangle relation, meta-path relation ($u_i \rightarrow v_j \rightarrow u_{i'}$), 1-hop neighbors, 2-hop neighbors, 3-hop neighbors.

portance depending on the specific datasets and downstream tasks. **Second**, inappropriate combinations with multiple SS-A tasks can result in **negative transfer**, in which various SS-A tasks may dominate training and hinder representation learning in the primary recommendation task. Worse still, balancing multiple SS-A tasks to assist the primary task is immensely challenging. However, most existing self-supervised enhanced recommendation methods only incorporate a few SS-A tasks, assign equal weights to various SS-A tasks, or heavily tune hyper-parameters to control the contribution of different SS-A tasks [17, 18, 19].

In order to verify the aforementioned weaknesses, we conduct preliminary studies to examine how a variety of SS-A tasks perform on an advanced SSL-based social recommendation method MHCN over three datasets, including LastFM, Epinions, and DBook (refer to Section 4.1.1 for more details.). The experimental results are presented in Figure 2, where it can be observed that

different SS-A tasks have a distinct influence on the primary recommendation
 75 task across different datasets. What’s worse, MHCN with various SSL tasks
 (except task 3) imposes a negative effect on MHCN (w/o tasks), where these
 methods fail to transfer knowledge from multiple SS-A tasks to the primary
 recommendation task. From these observations, we can conclude that the
 construction of SS-A tasks in social recommendation strongly depends on
 80 the datasets and domain knowledge. Therefore, an important research ques-
 tion is **how to automatically take advantage of multiple SS-A tasks
 to enhance representation learning for improving social recommen-
 dations**. To the best of our knowledge, the studies to automatically leverage
 various SS-A tasks for social recommendations remain rarely explored.

85 In this paper, to tackle the aforementioned challenges, we aim to build an
Automatic self-supervised learning for Social Recommendations (AusRec),
 which can automatically leverage SS-A tasks for assisting the primary rec-
 ommendation task. More specifically, we propose an automatic weighting
 mechanism to learn weights for various SS-A tasks, so as to balance the con-
 90 tribution of such SS-A tasks for enhancing representation learning in social
 recommendations. What’s more, the proposed automatic weighting mecha-
 nism is formulated as a meta learning optimization problem with an auto-
 matic weighting network. The main contributions are summarized as follows:

- We study a novel problem of taking advantage of multiple self-supervised
 95 social auxiliary tasks by adjusting their weights in an end-to-end man-
 ner for social recommendations;
- We introduce a principled way to incorporate multiple SS-A tasks, so as
 to transfer informative knowledge to enhance representation learning

by utilizing abundant unlabeled data in social relations;

- 100 • We propose a novel self-supervised learning framework for social recommendations (**AusRec**), which automatically balance the importance of various SS-A tasks to improve the primary social recommendations;
- We conduct comprehensive experiments and ablation studies on various real-world datasets to demonstrate the superiority of AusRec.

105 The remainder of this paper is organized as follows. Section 2 reviews related work on recommendation systems, social recommendations, self-supervised learning, and meta-learning for recommendation systems. Section 3 presents the proposed AusRec in detail, including the formulation of self-supervised auxiliary tasks, the automatic weighting mechanism, and the
110 model training process. Section 4 reports the experimental setup, datasets, baseline comparisons, and ablation studies. Section 5 discusses the limitations and future work of our approach, and Section 6 concludes the paper.

2. Related Work

In this section, we review related work of recommendation systems, social
115 recommendation methods, self-supervised learning, and meta-learning approaches for recommendation systems.

2.1. Recommendation Systems

Modern recommendation systems generally fall into three categories: content-based, collaborative filtering, and hybrid systems. Collaborative filtering
120 (CF) is one of the most widely adopted techniques in these systems, known

for its strong performance across various recommendation scenarios. For example, MF [5] represents each user and item ID as a vectorized representation, and calculates user’s preferences for items by conducting inner product between user and item. In practice, instead of expressing preferences directly in terms of ratings, users also express their preferences for items implicitly by clicking or browsing. Later on, Bayesian Personalized Ranking (BPR) [21] is developed to handle implicit feedback between users and items, optimizing a pairwise ranking loss to ensure that items liked by a user rank higher than items disliked by the user in their recommendation list. In recent years, with the success of representation learning, graph neural networks (GNNs) [22, 23, 24] have been increasingly adopted in recommendation systems [25]. LightGCN [26] simplifies the GNNs-based recommendation model by removing feature transformation and nonlinear activation, enhancing training efficiency and achieving better performance. Thus, it is often considered as a backbone method for new GNNs-based recommendation methods or a baseline method for comparison with new GNNs-based recommendation methods.

2.2. Social Recommendations

With the rise of social media, leveraging social relationships for recommendations has gained significant attention in recent years [13, 10, 9, 11, 27]. In particular, Mohsen Jamali et al. propose a novel matrix factorization-based recommendation method SocialMF [13], which incorporates social influence into the model so that each user’s features depend on the feature vectors of their direct neighbours in social networks. As cold-start users are more dependent on social networks than the general users, the impact of using

trust propagation on cold start users becomes even more important. What's more, recent deep learning techniques have advanced classic social recommendation methods [28, 11]. More specifically, due to the great success in learning meaningful representations, graph neural networks techniques have significantly improved the performance of social recommendations [9, 10, 11]. For example, Fan et al. design the very first method GraphRec[9] to employ the power of graph neural networks for improving the learning of user and item representations in social recommendations. Wu et al. propose to model the process of social influence propagation via GNNs model to obtain improved representations of users and items [10]. Yu et al. develop a GCN-based auto-encoder to enhance relational data by encoding more intricate higher-order connection patterns [11]. Note that most of the existing social recommendation methods introduce one or a few types of social relations into the recommendation models. In this work, we explore incorporating multiple social relations into recommendations and automatically leveraging them for the primary recommendation task.

2.3. Self-supervised Learning

In recent year, Self-supervised learning (SSL) is proposed to enhance representation learning by training with ground-truth samples derived from the raw data. Recent advances in SSL have significantly broadened its applications across various domains, including computer vision [29, 30], natural language processing [31, 32], and graph representation learning [33, 34, 35, 36, 37, 38, 39, 40]. These studies mainly design SS-A tasks in the view of the data structure to mark the unlabeled data [40]. For instance, TimesURL [29] enhances model performance by introducing a frequency-temporal-based aug-

mentation strategy and designing double universums as high-quality hard negatives to improve contrastive learning (A type of SSL). data2vec [31] improves model performance by introducing a unified self-supervised learning framework that predicts contextualized latent representations of the full input from a masked view across modalities. InfoMotif [37] utilizes correlations among attributes in motifs to regularize graph neural networks with mutual information maximization.

As an emerging paradigm, SSL achieves great success in recommendation systems [41, 17, 19, 42]. SSL in recommendation systems can be divided into four categories: contrastive-based, predictive-based, generative-based, and hybrid [43, 39]. Contrastive Learning (CL)-based methods treat a single user or item as an instance, then generate multiple views based on the same instance via different data augmentation methods, and finally bring views of the same instance closer together while distancing views of different instances in the embedding space. For example, SGL [17] adopts various data augmentation methods to construct SS-A tasks for improving the performance of recommendation systems. DHCN [18] integrates SSL into the model training by optimizing mutual information among the representations of sessions. MHCN [19] employs social triangle relations to construct contrastive SS-A tasks to improve social recommendations. The generative methods design a SS-A task to reconstruct the original user/item information, making the model learn to predict part of the data based on another part of the data, which is inspired by the masked language models [44, 45, 46, 47]. Different from generative methods, predictive methods [48] can be seen as a self-prediction rather than predicting the missing parts of raw data. The

original data is used to produce new samples and labels. The hybrid methods integrate the SS-A tasks constructed based on the three different methods mentioned above into a recommendation model [39]. These methods achieve a more comprehensive self-supervision by integrating different types of self-supervised signals, and further improve the performance of recommendation systems. Despite the compelling successes, most existing SSL-based recommendation methods only incorporate a few SS-A tasks, assign equal weights to various SS-A tasks, or heavily tune hyper-parameters to regulate the contribution of various SS-A tasks. Meanwhile, they are unable to automatically balance multiple SS-A tasks for assisting target social recommendations across datasets. In this work, we make the first attempt to fill this gap by introducing an automatic self-supervised learning framework for social recommendations.

2.4. Meta Learning for Recommendation Systems

Meta learning has achieved remarkable successes on a variety of deep learning tasks in recent years [49, 50, 51]. For example, meta learning plays an important role in few-shot learning, learning a base model that generates the parameters of other models or learning optimizers that can generalize to the new tasks. Besides, meta learning also aims to improve the performance of existing algorithms via metadata [52, 53]. Recently, there are some studies [54, 55, 56, 57] have investigated meta learning techniques to recommendation systems. MetaCS-DNN [54] is trained in a meta learning manner, so as to achieve the needs of a wide range of users, and the model can also be quickly adapted to the specific user after a few update steps. MeLU [55] attempts to solve the user cold-start problem in recom-

mendation systems from the view of few-shot via meta learning. $\lambda\mathbf{OPT}$ [56] learns finer-grained regularization parameters for recommendation via meta learning. SML [57] improves the performance of recommendation systems by leveraging meta learning to guide the process of re-training. In our paper, we adopt meta learning techniques to learn automatic weights towards various self-supervised tasks and transfer knowledge in assisting the primary recommendation task.

3. The Proposed Method

This section presents the definitions and notations used throughout the paper, followed by an introduction to our proposed automatic self-supervised learning framework for social recommendations (AusRec). Finally, we detail the model training process and parameter learning approach.

3.1. Definitions and Notations

Let $U = \{u_1, u_2, \dots, u_m\}$ and $V = \{v_1, v_2, \dots, v_n\}$ be the sets of users and items, respectively, where m is the number of users and n is the number of items. We denote $\mathbf{R} \in \mathbb{R}^{m \times n}$ as a user-item interaction matrix, where $r_{ij} = 1$ if user u_i has interacted with item v_j , otherwise $r_{ij} = 0$. We use $\mathbf{S} \in \mathbb{R}^{m \times m}$ to denote the user-user social matrix, where $s_{ii'} = 1$ if user u_i has a social relation with user $u_{i'}$, otherwise $s_{ii'} = 0$. In this work, since data in social recommendations can be naturally represented as graph structure $\mathcal{G} = (\mathcal{G}^{\mathbf{R}}, \mathcal{G}^{\mathbf{S}})$, the user-item interaction matrix (\mathbf{R}) and user-user social network (\mathbf{S}) can be represented as $\mathcal{G}^{\mathbf{R}}$ and $\mathcal{G}^{\mathbf{S}}$, respectively. Mathematically, the adjacency matrix of the graph \mathcal{G} which combines user-item graph $\mathcal{G}^{\mathbf{R}}$ and user-user social graph $\mathcal{G}^{\mathbf{S}}$ can be defined as:

$$\mathbf{A} \in \mathbb{R}^{(m+n) \times (m+n)} = \begin{pmatrix} \mathbf{S} & \mathbf{R} \\ \mathbf{R}^T & \mathbf{0} \end{pmatrix}. \quad (1)$$

245 Given \mathbf{R} and \mathbf{S} , our goal is to predict users' unknown preferences towards items (i.e., the unobserved entries in \mathbf{R}) by learning user and item representations.

3.2. An Overview of the Proposed Method

Figure 3 illustrates the architecture of the proposed model, comprising
 250 two primary components: (1) **primary recommendation task and self-supervised auxiliary tasks**, which aims to construct multiple SS-A tasks to assist the primary social recommendations; (2) **automatic weighting mechanism**, which is introduced to balance multiple SS-A tasks by learning the weights towards different SS-A tasks via meta learning. More specifically,
 255 the SS-A tasks assist the primary recommendation task by predicting whether there is a specific social relation between users, aiming to transfer knowledge from the SS-A tasks to the primary recommendation task. At the same time, an automatic weighting network is developed to automatically adjust the weights of different SS-A tasks via meta learning. The main goal
 260 of our proposed framework can be summarized as:

- Learning the representations of users and items from user-item interactions and social networks to make recommendations by constructing various SS-A tasks.
- Automatically weighting various SS-A tasks, balancing primary and

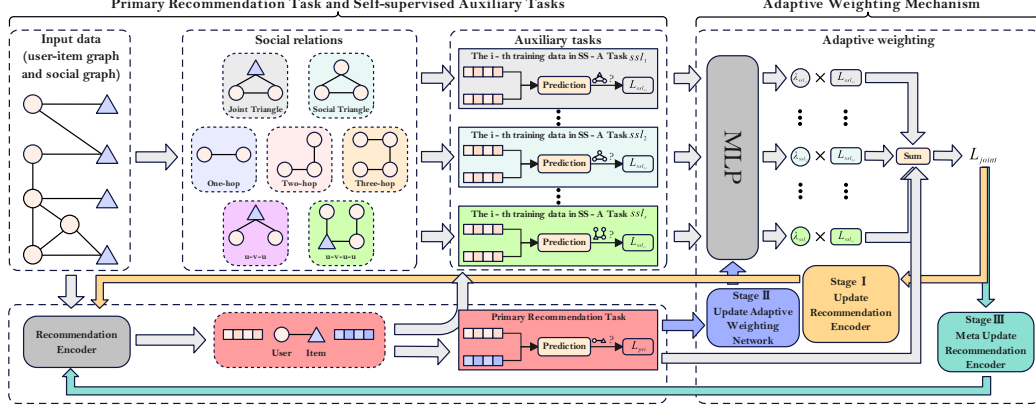


Figure 3: The overall framework of the proposed method. The parameters updating process consists of three main stages in the automatic weighting mechanism. Stage I performs a hypothetical update of the recommendation encoder ($\hat{\mathbf{W}}$) using the current task weights. Stage II evaluates these weights on a meta dataset and updates the weighting network (Θ) accordingly. Stage III then updates the encoder (\mathbf{W}) with the newly optimized weights.

265 SS-A tasks to avoid negative knowledge transfer and further improve
the performance of the primary recommendation task.

Next, we will detail each model component.

3.3. Primary Recommendation Task and Self-supervised Auxiliary Tasks

270 This component aims to construct multiple SS-A tasks to assist the pri-
mary social recommendation task. First, we introduce the primary recom-
mendation task to model social recommendations. Second, we detail how to
take advantage of multiple SS-A tasks for enhancing user and item represen-
tation learning. Finally, we provide the overall optimization objective of our
proposed framework.

275 3.3.1. Primary Recommendation Task: Social Recommendations

Due to the success of graph neural networks (GNNs) [58] in representation learning, we adopt the GNNs-based recommendations to encode user-item interactions and social relations for user and item representation learning [26, 10]. Here, LightGCN [26], as one of the most representative methods, is
 280 adopted as our target encoder to learn user and item representation in user-item graph $\mathcal{G}^{\mathbf{R}}$ and user-user social graph $\mathcal{G}^{\mathbf{S}}$ for social recommendations. More specifically, given the adjacency matrix $\mathbf{A} \in \mathbb{R}^{(m+n) \times (m+n)}$ (in Eq. 1) for modeling graph data $\mathcal{G} = (\mathcal{G}^{\mathbf{R}}, \mathcal{G}^{\mathbf{S}})$ in social recommendations, node u 's representation (i.e., user or item) can be learned by aggregation operation as
 285 follows:

$$\mathbf{e}_u^{k+1} = \text{AGG}(\{\mathbf{e}_v^k : v \in \mathcal{N}_u\}), \quad (2)$$

where \mathbf{e}_u^{k+1} denotes the representation of the user node after aggregating the representations \mathbf{e}_v^k of the item nodes connected to it at the k -th layer. AGG denotes the GNNs aggregation function to learn node u 's representation through the local neighbors \mathcal{N}_u in \mathcal{G} . To be more specific, if u_* represents a
 290 user node, \mathcal{N}_{u_i} contains the set of items (with which u_i has interacted with in the user-item interaction network \mathbf{R}) and the set of social neighbors (whom u_i is directly connected with in the user-user social network \mathbf{S}); Otherwise we use \mathcal{N}_{v_j} to represent the set of users who have interacted with item v_j .

With the final node representations $\mathbf{e}_u/\mathbf{e}_v$, the goal of our primary social
 295 recommendation task is to predict how likely user u_i would interact with item v_j through an inner product:

$$\hat{y}_{u_i v_j} = \mathbf{e}_{u_i}^T \mathbf{e}_{v_j}. \quad (3)$$

where $\hat{y}_{u_i v_j}$ denotes the predicted rating of user u_i on item v_j generated by the model.

3.3.2. Self-supervised Auxiliary Tasks (SS-A Tasks)

300 In order to take advantage of rich unlabeled data in social recommendations, we propose to construct multiple self-supervised auxiliary (SS-A) tasks to provide self-supervision signals for assisting the training of primary recommendation task without using manual annotations. According to the social correlation theories [59, 60, 61], we explore multiple social SS-A tasks
305 based on graph structure $\mathcal{G} = (\mathcal{G}^{\mathbf{R}}, \mathcal{G}^{\mathbf{S}})$.

- **Triangle Relations** (Social Triangle, Joint Triangle): Triangle relations can reflect users preferences towards items and their social behaviors in social network [59, 62], such as friendship, community of interest, as shown in Figure 1 (a) and (b). Here, we introduce two representative
310 types of triangle relations, namely social triangle $((\mathbf{SS}) \odot \mathbf{S})$ and joint triangle $((\mathbf{RR}^T) \odot \mathbf{S})$, where \odot denotes the element-wise product.
- **N -hop Neighbors** (1, 2, 3, ..., n -hop neighbors): The N -hop neighbors auxiliary tasks explore the best way to propagate messages among users, where nodes in social networks can reach others via the shortest
315 path between them [60].
- **Meta-paths**: Meta-paths [61] have been widely used to capture diverse, long-range, and meaningful relations between entities in heterogeneous graph, where a path represents a sequence of nodes connected with heterogeneous edges. Here, in social recommendation, we consider

320 two types of meta paths to model social behaviors, i.e., $u_i \rightarrow v_j \rightarrow u_{i'}$
 (uvu) and $u_i \rightarrow u_{i'} \rightarrow v_j \rightarrow u_{i''}$ ($uvvu$), where $u_i, u_{i'}, u_{i''} \in U$ and
 $v_j \in V$. More meta paths will be explored in the future.

For simplicity, the set of SS-A tasks can be denoted as $\text{SSL} = \{\text{ssl}_1, \text{ssl}_2, \dots, \text{ssl}_s\}$, and the number of such tasks is $|\text{SSL}|$. The training data for SS-A tasks
325 can be obtained from raw data without using any manual annotations. For
instance, for $u_i \rightarrow v_j \rightarrow u_{i'}$ in meta- uvu , we need to distinguish whether a
pair of users (i.e., $(u_i, u_{i'})$) exists such relation belonging to ssl_s (i.e., meta-
 uvu), where the self-supervised learning label $y_{u_i, u_{i'}}^{\text{ssl}_s} = 1$ if there exists such
relation belonging to ssl_s , otherwise $y_{u_i, u_{i'}}^{\text{ssl}_s} = 0$. Then, the SS-A tasks' pre-
330 diction can be formulated via inner product on users' final representations
as follows:

$$\hat{y}_{u_i, u_{i'}}^{\text{ssl}_s} = \mathbf{e}_{u_i}^T \mathbf{e}_{u_{i'}}. \quad (4)$$

where $\hat{y}_{u_i, u_{i'}}^{\text{ssl}_s}$ denotes the model-predicted probability of a social relation be-
tween user u_i and user $u_{i'}$.

3.3.3. Optimization Loss for Primary Recommendation and SS-A Tasks

335 With the supervised labels from the primary recommendation task and
self-supervised labels from SS-A tasks, we employ the pairwise BPR loss [21]
to encourage the prediction of an observed entry to be higher than its unob-
served counterparts:

$$\mathcal{L}_{\text{pri}} = \sum_{(u_i, v_j, v_{j'}) \in O_{\text{pri}}} -\ln \sigma(\hat{y}_{u_i v_j} - \hat{y}_{u_i v_{j'}}) + \delta \|\mathbf{E}^{(0)}\|^2, \quad (5)$$

$$\mathcal{L}_{\text{ssl}_s} = \sum_{(u_i, u_{i'}, u_j) \in O_{\text{ssl}_s}} -\ln \sigma(\hat{y}_{u_i u_{i'}}^{\text{ssl}_s} - \hat{y}_{u_i u_j}^{\text{ssl}_s}) + \delta \|\mathbf{E}^{(0)}\|^2, \quad (6)$$

where a hyper-parameter δ is used to control the contribution of \mathcal{L}_2 regularization $\|\mathbf{E}^{(0)}\|^2$ so as to reduce the risk of overfitting. σ is the sigmoid function. As for primary recommendation task, the set of $O_{\text{pri}} = \{(u_i, v_j, v_{j'}) | (u_i, v_j) \in \mathcal{E}_{\text{pri}}, (u_i, v_{j'}) \notin \mathcal{E}_{\text{pri}}\}$ denotes the pairwise training data in primary recommendation task, where $(u_i, v_j) \in \mathcal{E}_{\text{pri}}$ indicates that the interaction between user u_i and item v_j is observed (positive) pair, while $(u_i, v_{j'}) \notin \mathcal{E}_{\text{pri}}$ indicates that the interaction between user u_i and item $v_{j'}$ is not observed (negative). Similarly, as for BPR loss in SS-A task ssl_s (i.e., $\mathcal{L}_{\text{ssl}_s}$), the set of $O_{\text{ssl}_s} = \{(u_i, u_{i'}, u_j) | (u_i, u_{i'}) \in \mathcal{E}_{\text{ssl}_s}, (u_i, u_j) \notin \mathcal{E}_{\text{ssl}_s}\}$ denotes the pairwise training data in SS-A task ssl_s , where $(u_i, u_{i'}) \in \mathcal{E}_{\text{ssl}_s}$ indicates that the interaction between user u_i and user $u_{i'}$ is observed (positive) pair, while $(u_i, u_j) \notin \mathcal{E}_{\text{ssl}_s}$ indicates that the interaction between user u_i and user u_j is not observed (negative). To optimize SS-A tasks, we apply negative sampling by randomly selecting an unobserved user/item and pairing it with the self-supervised social relations as a negative instance.

3.4. Automatic Weighting Mechanism

To effectively learn user and item representations, we present a joint optimization objective that combines the primary recommendation task with various SS-A tasks, which can be formally defined as:

$$\mathcal{L}_{\text{joint}} = \sum_{i=1}^N \frac{1}{N} \mathcal{L}_{\text{pri},i}^{\text{train}} + \sum_{s=1}^{|\text{SSL}|} \lambda_{\text{ssl}_s} \sum_{i=1}^{N_{\text{ssl}_s}} \frac{1}{N_{\text{ssl}_s}} \mathcal{L}_{\text{ssl}_s,i}, \quad (7)$$

where $\mathcal{L}_{\text{pri},i}^{\text{train}}$ and $\mathcal{L}_{\text{ssl}_s,i}$ are the i -th instance of training set in primary recommendation task and SS-A task ssl_s , respectively. N and N_{ssl_s} denote the size of training set in primary recommendation task and SS-A task ssl_s . λ_{ssl_s}

denotes the hyper-parameters to control the contribution on SS-A task ssl_s to primary recommendation task, which can have a significant influence on model's performance in various recommendation scenarios and over different datasets. However, most existing works assign equal weight to λ_{ssl_s} or heavily
 365 tune them in experiments, which can result in sub-optimal recommendation performance due to negative transfer among various SS-A tasks.

In our work, to address the above issue, AusRec aims to learn the primary recommendation task by automatically leveraging multiple SS-A tasks with flexible weights. More specifically, an automatic weighting mechanism
 370 is introduced to automatically balance various SS-A tasks via learning (automatic) weights λ_{ssl_s} for assisting representation learning in social recommendations. This automatic weighting mechanism is formulated as a meta-learning task using bi-level optimization [63], where the joint optimization objective of the primary social recommendation task with various SS-A tasks
 375 can be defined as:

$$\min_{\mathbf{W}, \Theta} \sum_{i=1}^{|D^{\text{pri-meta}}|} \frac{1}{|D^{\text{pri-meta}}|} \mathcal{L}_{\text{pri},i}^{\text{meta}}(\mathbf{W}^*(\Theta)), \quad (8)$$

$$\begin{aligned} \text{s.t. } \mathbf{W}^*(\Theta) = \arg \min_{\mathbf{W}} & \left(\sum_{i=1}^{|D^{\text{pri}}|} \frac{1}{|D^{\text{pri}}|} \mathcal{L}_{\text{pri},i}^{\text{train+meta}}(\mathbf{W}) + \right. \\ & \left. \sum_{s=1}^{|SSL|} \sum_{i=1}^{N_{\text{ssl}_s}} \frac{1}{N_{\text{ssl}_s}} V([\mathcal{L}_{\text{ssl}_s,i}; e_{\text{ssl}_s}]; \Theta) \mathcal{L}_{\text{ssl}_s,i}(\mathbf{W}) \right), \end{aligned} \quad (9)$$

where the automatic weights λ_{ssl_s} can be learned by an automatic weighting network $V(\cdot; \Theta)$ which can be modeled as a two-layer MLP network with trainable parameters Θ . \mathbf{W} denotes the parameters of the recommendation encoder in both the primary task and the auxiliary tasks. $\mathcal{L}_{\text{pri},i}^{\text{meta}}$ is loss

function of the i -th instance of meta data $D^{\text{pri-meta}}$ in the primary recommendation task. $D^{\text{pri-meta}}$ is a small portion of the primary task's training data $D^{\text{pri}} = D^{\text{train+meta}} = \{D^{\text{pri-train}}, D^{\text{pri-meta}}\}$ to guide the learning of the automatic weighting network. $\mathcal{L}_{\text{pri},i}^{\text{train+meta}}$ and $\mathcal{L}_{\text{ssl},i}$ are loss functions of the i -th instance of training set in the primary recommendation task and SS-A tasks. D^{SSL} denotes the training set for SS-A task. $|D^{\text{pri}}|$ is the size of whole training set in the primary recommendation task, $|D^{\text{pri-meta}}|$ is the size of meta data from the primary recommendation task and much less than the size of the whole training set.

In each training epoch under meta learning paradigm [63, 64], the automatic weighting network $V([\mathcal{L}_{\text{ssl},i}; e_{\text{ssl},i}]; \Theta)$ takes the concatenation of $\mathcal{L}_{\text{ssl},i}$ and $e_{\text{ssl},i}$ as input, where $e_{\text{ssl},i}$ is the one-hot vector of task ssl_i to distinguish which task the loss of input to the automatic weighting network comes from. Note that the formulation in Eq. 9 enables learning to support the primary task by optimizing Θ through meta-learning. The optimal parameters Θ^* can be obtained by minimizing the following loss:

$$\Theta^* = \arg \min_{\Theta} \mathcal{L}_{\text{pri}}^{\text{meta}}(\mathbf{W}^*(\Theta)) \quad (10)$$

$$\triangleq \sum_{i=1}^{|D^{\text{pri-meta}}|} \frac{1}{|D^{\text{pri-meta}}|} \mathcal{L}_{\text{pri},i}^{\text{meta}}(\mathbf{W}^*(\Theta)). \quad (11)$$

3.5. Model Training

The process to optimize our proposed AusRec is shown in the right part of Figure 3, which includes three stages to update the parameters of recommendation encoder \mathbf{W} and automatic weighting network Θ . Specially, stage I is a hypothetical update step for the main recommendation encoder based on the current task weights. Stage II uses the performance of this

hypothetical encoder on a meta dataset to evaluate how good the current weights are, and updating the weighting network accordingly. And stage III performs the actual update of the recommendation encoder using the newly improved weights. The overall training procedure of our proposed AusRec can be summarized in Algorithm 1.

3.5.1. Stage I: Updating Parameters of Recommendation Encoder

At each training epoch, we begin by sampling mini-batches of data from the primary recommendation task and SS-A tasks. Then, the loss $\mathcal{L}_{\text{ssl}_s}$ of the SS-A tasks are weighted by the automatic weighting network and summed up with the loss $\mathcal{L}_{\text{pri}}^{\text{train}}$ of the primary recommendation task to obtain the final joint loss. To alleviate the difficulty of the bi-level optimization [63], the parameters of recommendation encoder $\mathbf{W}^{(k)}$ can be approximated with the updated parameters $\hat{\mathbf{W}}^{(k)}$ in meta learning paradigm as follows:

$$\begin{aligned} \hat{\mathbf{W}}^{(k)}(\Theta) = \mathbf{W}^{(k)} - \alpha \nabla_{\mathbf{W}^{(k)}} & \left(\sum_{i=1}^{|D^{\text{pri-train}}|} \frac{1}{|D^{\text{pri-train}}|} \mathcal{L}_{\text{pri},i}^{\text{train}}(\mathbf{W}^{(k)}) \right. \\ & \left. + \sum_{s=1}^{|\text{SSL}|} \sum_{i=1}^{N_{\text{ssl}_s}} \frac{1}{N_{\text{ssl}_s}} V([\mathcal{L}_{\text{ssl}_s,i}(\mathbf{W}^{(k)}); e_{\text{ssl}_s}]; \Theta^{(k)}) \mathcal{L}_{\text{ssl}_s,i}(\mathbf{W}^{(k)}) \right), \end{aligned} \quad (12)$$

where α is the learning rate for \mathbf{W} . $\mathcal{L}_{\text{pri},i}^{\text{train}}$ is the i -th instance loss function of data on $D^{\text{pri-train}}$.

3.5.2. Stage II: Updating Parameters of Automatic Weighting Network

Once the parameters $\mathbf{W}^{(k)}$ of the recommendation encoder are updated, the parameters $\Theta^{(k)}$ of the automatic weighting network can then be updated accordingly with the updated parameters $\hat{\mathbf{W}}^{(k)}$ on $D^{\text{pri-meta}}$, which is defined

Algorithm 1: The proposed AusRec Framework

Input: Training data for the primary recommendation task D^{pri} and SS-A tasks D^{SSL} , max iteration K , learning rate α, β .

Output: Social recommendation parameters \mathbf{W}^K , and Automatic Weighting Network parameters Θ^K .

```
1 Initialization parameters  $\mathbf{W}^1, \Theta^1$ ;  
2 for  $k = 1 \dots K$  do  
3   Sample minibatch  $D_n^{\text{pri}}$  from primary task training data  $D^{\text{pri}}$ ;  
4   Sample minibatch  $D_n^{\text{SSL}}$  from SS-A task training data  $D^{\text{SSL}}$ ;  
5   Split  $D_n^{\text{pri}}$  into  $\{D_n^{\text{pri-train}}, D_n^{\text{pri-meta}}\}$ ;  
6   Update  $\hat{\mathbf{W}}^{(k)}$  by Eq. (12) with  $D_n^{\text{pri-train}}$  and  $D_n^{\text{SSL}}$ ;  
7   Update  $\Theta^{(k)}$  by Eq. (13) with  $D_n^{\text{pri-meta}}$ ;  
8   Update  $\mathbf{W}^{(k)}$  by Eq. (14) with  $D_n^{\text{pri}}$  and  $D_n^{\text{SSL}}$ ;  
9 end
```

as follows:

$$\Theta^{(k+1)} = \Theta^{(k)} - \beta \nabla_{\Theta^{(k)}} \mathcal{L}_{\text{pri}}^{\text{meta}}(\hat{\mathbf{W}}^{(k)}(\Theta^{(k)})), \quad (13)$$

where β is the meta learning rate for updating Θ . We update parameters of the automatic weighting network on the meta data (i.e., $D^{\text{pri-meta}}$) from the primary recommendation task. This update allows softly select SS-A tasks and balances them for the primary recommendation task, so as to enhance the representation learning of users and items in social recommendations. Conversely, without such an automatic weighting network to balance various SS-A tasks on the joint loss (i.e., Eq. 7), multiple SS-A tasks may result in negative knowledge transfer on the primary recommendation task via domi-

430 nating the main training objective.

3.5.3. Stage III: Meta Updating Parameters of Recommendation Encoder

At last, with the updated parameters $\Theta^{(k+1)}$ of the automatic weighting network, the parameters $\mathbf{W}^{(k)}$ of recommendation encoder can be further updated through $D^{\text{pri}} = \{D^{\text{pri-train}}, D^{\text{pri-meta}}\}$ and D^{SSL} as follows:

$$\begin{aligned} \mathbf{W}^{(k+1)} = \mathbf{W}^{(k)} - \alpha \nabla_{\mathbf{W}^{(k)}} \left(\sum_{i=1}^{|D^{\text{pri}}|} \frac{1}{|D^{\text{pri}}|} \mathcal{L}_{\text{pri},i}^{\text{train+meta}}(\mathbf{W}^{(k)}) + \right. \\ \left. \sum_{s=1}^{|D^{\text{SSL}}|} \sum_{i=1}^{N_{\text{SSL}_s}} \frac{1}{N_{\text{SSL}_s}} V([\mathcal{L}_{\text{SSL}_s,i}(\mathbf{W}^{(k)}); e_{\text{SSL}_s}]; \Theta^{(k+1)}) \mathcal{L}_{\text{SSL}_s,i}(\mathbf{W}^{(k)}) \right). \end{aligned} \quad (14)$$

435 4. Experiment

In this section, we conduct comprehensive experiments to validate the effectiveness of our proposed method. We begin by presenting the experimental settings, followed by a comparison of our proposed method with various baselines on representative datasets. Finally, we perform detailed
440 ablation studies on the proposed method.

4.1. Experimental Settings

4.1.1. Datasets

To evaluate how effective the proposed method is, we carry out comprehensive tests using three representative datasets, including LastFM⁶, Epinions⁷, and DBook⁸. A statistical overview of these three datasets appears in
445 Table 1.

⁶<https://www.last.fm>

⁷<http://www.epinions.com>

⁸<https://book.douban.com>

Table 1: Statistics of the experimental data.

Dataset	users	items	Interactions	Density(R)	Connections	Density(S)
LastFM	1,880	4,489	52,668	0.00624	25,434	0.00720
Epinions	5,957	11,027	132,819	0.00202	98,959	0.00277
DBook	13,024	22,347	792,062	0.01137	169,150	0.00100

- 450 • **LastFM.** This dataset is provided by last.fm, a UK-based internet radio and music community. It contains data from social networks, making it a popular choice for testing social recommendation models. The dataset includes information on each user’s favorite artists along with their play counts, as well as user-generated application tags that can be used to create content vectors. Specifically, the dataset comprises 1,880 users and 4,489 items.
- 455 • **Epinions.** This dataset is sourced from Epinions, a product review website where users can freely register and write subjective reviews on various items, including software, music, TV shows, hardware, and office appliances. The version of the dataset used in our proposed method consists of 5,957 users and 11,027 items.
- 460 • **DBook.** This dataset is collected from the Douban.Book website and includes users’ book ratings (ranging from 1 to 5) as well as information about social networks between users. This is the largest of the three datasets used in this paper, containing 13,024 users and 22,347 items.

4.1.2. Parameter Setting

During training our proposed method, the parameters are set as follows:
465 batch size is 2048, consistent with LightGCN. User and item embedding sizes d are searched within the range [128, 160, 192, 224, 256] with a step size of 32. There are two learning rates in our proposed method, including the learning rate lr for parameters updating of the target recommendation encoder, and the meta learning rate mlr for parameters updating of the Automatic
470 Weighting Network. We select lr from [0.001, 0.005, 0.01, 0.05] and mlr from [0.0001, 0.0005, 0.001, 0.005], using a grid search to identify the optimal parameters. The optimal configuration is $lr = 0.001$ and $mlr = 0.0001$. We divide the interactions into training and testing sets at a 4 : 1 ratio. Specific experiment is described below. Besides, we set the ratio of meta data in
475 dataset as 5%. Adam optimizer is employed for updating the model’s parameters. To ensure a fair comparison, we adhere to the default settings of the baselines. All experiments are performed on a system with an Intel(R) Xeon(R) Gold 5120 CPU and an NVIDIA Titan RTX 2080Ti 24G GPU.

4.1.3. Baselines

480 To assess effectiveness, we compare the proposed method against three categories of representative baselines, covering recommendation models without social network information (**MF** [5], **NeuMF** [65], **NGCF** [66], **LightGCN** [26]), social recommendation systems (**SocialMF** [13], **LightGCN-S**, **DiffNet** [10]), and self-supervised recommendation systems (**SGL** [17],
485 **MHCN** [19], **HGCL** [67]). These baseline methods are summarized as follows:

- MF [5]: This method stands as a classic collaborative filtering approach, utilizing only user-item direct interactions as the primary input.
- NeuMF [65]: This approach represents a cutting-edge matrix factorization model built on a neural network architecture.
- NGCF [66]: This is the first GNN-based model to integrate high-order connectivity between users and items.
- LightGCN [26]: This is a collaborative filtering GNNs-based recommendation method by simplifying NGCF model [66], achieving promising recommendation performance.
- LightGCN-S: This model improves upon LightGCN by integrating social network information.
- SocialMF [13]: This method integrates trust information propagation into the MF model for recommendation systems.
- DiffNet [10]: This GCN-driven social recommendation approach captures recursive and dynamic social influence across both user and item domains.
- SGL [17]: This advanced self-supervised recommendation method employs diverse data augmentation techniques, such as node drop, edge drop, and random walk, to enhance recommendation performance.
- MHCN [19]: This recent GCN-based social recommendation method incorporates self-supervised learning into hypergraph convolutional networks.

Table 2: Performance comparison of different recommendation methods.

Datasets	Metrics	MF	NeuMF	NGCF	LightGCN	LightGCN-S	SocialMF	DiffNet	SGL	MHCN	HGCL	AusRec	Improv.
LastFM	Recall@5	0.0786	0.0835	0.0950	0.1225	0.1254	0.1003	0.0908	0.1261	0.1237	<u>0.1268</u>	0.1313	3.55%
	Recall@10	0.1290	0.1281	0.1483	0.1925	0.1925	0.1538	0.1425	0.1920	0.1944	<u>0.1964</u>	0.2029	3.31%
	Recall@20	0.1943	0.1873	0.2215	0.2748	0.2774	0.2360	0.2144	<u>0.2818</u>	0.2789	0.2816	0.2909	3.23%
	NDCG@5	0.1034	0.1183	0.1223	0.1640	0.1678	0.1313	0.1297	0.1687	0.1653	<u>0.1690</u>	0.1734	2.60%
	NDCG@10	0.1156	0.1316	0.1371	0.1791	0.1814	0.1423	0.1442	0.1808	0.1818	<u>0.1833</u>	0.1889	3.06%
	NDCG@20	0.1426	0.1601	0.1670	0.2148	0.2152	0.1771	0.1781	0.2189	0.2191	<u>0.2200</u>	0.2244	2.00%
Epinions	Recall@5	0.0191	0.0192	0.0309	0.0372	0.0364	0.0278	0.0259	0.0341	0.0313	<u>0.0376</u>	0.0421	11.99%
	Recall@10	0.0306	0.0339	0.0501	0.0585	0.0574	0.0450	0.0416	0.0531	0.0523	<u>0.0586</u>	0.0645	10.07%
	Recall@20	0.0470	0.0567	0.0777	0.0908	0.0904	0.0679	0.0669	0.0796	0.0823	<u>0.0913</u>	0.1011	10.73%
	NDCG@5	0.0164	0.0185	0.0251	0.0319	0.0321	0.0237	0.0261	0.0304	0.0338	<u>0.0346</u>	0.0365	5.49%
	NDCG@10	0.0208	0.0250	0.0324	0.0398	0.0399	0.0302	0.0330	0.0373	0.0430	<u>0.0441</u>	0.0447	1.36%
	NDCG@20	0.0259	0.0336	0.0421	0.0505	0.0508	0.0378	0.0413	0.0461	0.0541	<u>0.0552</u>	0.0564	2.17%
DBook	Recall@5	0.0621	0.0428	0.0649	0.0734	<u>0.0762</u>	0.0570	0.0556	0.0729	0.0749	0.0758	0.0826	8.40%
	Recall@10	0.0922	0.0723	0.0976	0.1105	0.1172	0.0864	0.0870	0.1043	0.1122	<u>0.1174</u>	0.1218	3.75%
	Recall@20	0.1355	0.1154	0.1427	0.1637	<u>0.1716</u>	0.1336	0.1290	0.1431	0.1670	0.1712	0.1721	0.29%
	NDCG@5	0.0596	0.0421	0.0625	0.0752	0.0779	0.0578	0.0602	0.0513	0.0771	<u>0.0813</u>	0.0865	6.40%
	NDCG@10	0.0677	0.0532	0.0727	0.0854	0.0900	0.0666	0.0728	0.0618	0.0872	<u>0.0904</u>	0.0968	7.08%
	NDCG@20	0.0813	0.0690	0.0870	0.1024	0.1069	0.0815	0.0876	0.0723	0.1058	<u>0.1100</u>	0.1135	3.18%

- HGCL [67]: This is a contrastive learning model dynamically applies knowledge from side information to model user-item interactions.

4.2. Performance Comparison of Recommender Systems

We begin by comparing the performance of our proposed method, AusRec, with all baseline methods. Table 2 presents the overall performance comparison based on Recall@5/10/20 and NDCG@5/10/20 metrics across three datasets. From these results, we observe the following:

- Our proposed AusRec demonstrates strong performance across three datasets, outperforming baseline methods on both metrics. For example, our proposed method improves one of the most effective baselines

LightGCN at least 10% under Recall metric on Epinions dataset, and
520 also outperforms social recommendation methods (i.e., LightGCN-S
and SocialMF) and SSL methods (i.e., SGL and MHCN) on three
datasets. This improvement highlights the effectiveness of our proposed
method and the adaptability of the automatic weighting mechanism to
various SS-A tasks.

- 525 • NGCF and LightGCN as representative GNNs-based recommendations
can show much better performance than traditional recommendation
methods (i.e., MF). On the other hand, compared with social recom-
mendations, their performance is still competitive, indicating the power
of GNNs in representation learning for recommendations.
- 530 • SGL as a general self-supervised recommendation method can achieve
promising performance in some cases, though no social information is
enhanced. This can be attributed to the integration of self-supervision
signals. This observation demonstrates the effectiveness of self-supervised
learning techniques in supporting the primary recommendation task.
- 535 • Although MHCH and HGCL are state-of-the-art self-supervised social
recommendation methods, they do not achieve the best performance
across all cases. Our proposed method AusRec can outperform all the
baseline, which is attributed to considering the automatic weights on
various SS-A tasks.
- 540 • AusRec achieves varying degrees of improvement over baseline meth-
ods across the three datasets. Specifically, the performance gain is
the smallest on the LastFM dataset, with a 3.55% improvement in

Recall@5, while the largest improvement is observed on the Epinions dataset, reaching 11.99% in Recall@5. This variation may be attributed to the differing densities of the underlying social networks. Excessively high network density (LastFM) may introduce considerable noise in social relations, thereby degrading the quality of auxiliary tasks and consequently affecting the performance of the primary recommendation task. Conversely, overly sparse social networks (DBook) may fail to adequately capture users’ preferences, which can likewise hinder the overall effectiveness of the model.

In summary, our proposed method AusRec achieves significant performance improvements over various baselines, demonstrating the effectiveness of the proposed framework. The superior performance of AusRec can be attributed to its ability to incorporate multiple types of social relations as additional knowledge, which enriches the representation learning process. Moreover, the automatic weighting network effectively balances different self-supervised auxiliary tasks with the primary recommendation objective, enabling the model to capture more comprehensive user–item interactions and further enhance overall recommendation accuracy.

4.3. Ablation Study

In this part, to gain a deeper understanding of our framework, AusRec, we examine the impact of its components and hyper-parameters through comprehensive experiments.

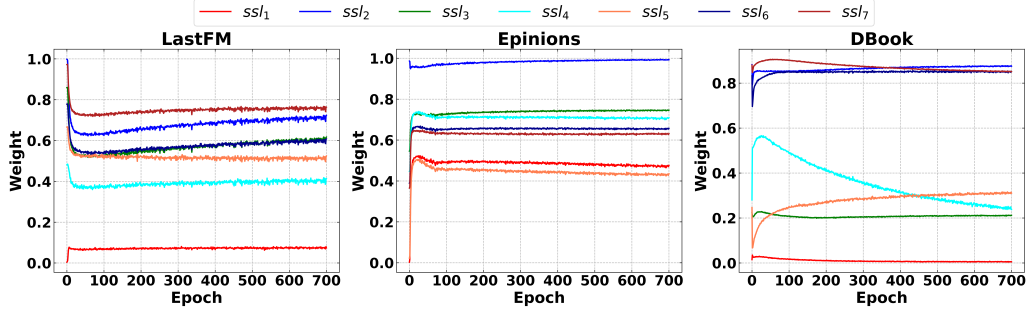


Figure 4: The change of automatic weights on various SS-A tasks during training (Best viewed in color).

4.3.1. The Change of Automatic Weights on Various SS-A Tasks during Training

To grasp the workings of our proposed AusRec, we now analyze the change of automatic weights λ_{ssl_s} on various SS-A tasks during the model training process. Note that weights λ_{ssl_s} (learned by the automatic weighting network $V(\cdot; \Theta)$) aim to control the contribution of SS-A tasks ssl_s on primary recommendation task, which can affect model’s performance significantly.

Figure 4 presents the experimental results, which reveal that (1) the automatic weights on most SS-A tasks are unstable in the early training stage, while they can converge to a specific value at the end; and (2) the weights on different SS-A tasks are different over three datasets. More specifically, the weight on SS-A task ssl_2 (i.e., joint triangle relation) is much largest than that of other SS-A tasks on Epinions dataset, while the weight on SS-A task ssl_7 (i.e., meta-path $uvvu$) contributes more on primary recommendation task than that of other SS-A tasks on LastFM dataset. These observations suggest that the proposed automatic weighting mechanism can automatically control the contribution of various SS-A tasks across different datasets, so

Table 3: Results of AusRec with the same weight towards SS-A tasks.

Dataset	LastFM			Epinions			DBook		
Metrics	Recall@5	Recall@10	NDCG@10	Recall@5	Recall@10	NDCG@10	Recall@5	Recall@10	NDCG@10
AusRec	0.1313	0.2029	0.1889	0.0421	0.0645	0.0447	0.0826	0.1218	0.0968
w/o AW	0.1213	0.1873	0.1758	0.0364	0.0589	0.0409	0.0705	0.1050	0.0841
Improv.	8.24%	8.33%	7.45%	15.66%	9.51%	9.29%	17.16%	16.00%	15.10%

as to transfer self-supervision signals in various SS-A tasks to enhance social recommendation.

4.3.2. Effect of Automatic Weighting Mechanism

585 To further study the effect of the automatic weighting mechanism, the weights of seven kinds of SS-A tasks are equally set to 1 without differentiating the importance of SS-A tasks (w/o AW for short). Table 3 provides the experimental results across three datasets, showing that the model without the automatic weighting network (i.e., w/o AW) experiences a significant drop in recommendation performance compared to the proposed Aus-
590 Rec. Therefore, we can see that removing the automatic weighting network can degrade performance significantly on three datasets, indicating that the automatic weighting mechanism can contribute to automatically leveraging various auxiliary tasks for enhancing social recommendation.

595 4.3.3. Effect of automatic weighting mechanism for only One SS-A task

To further study the effect of the automatic weighting mechanism, we use the automatic weighting mechanism with only one SS-A task, while the weight on variant (w/o AW) towards each SS-A task is set to 1 without differentiating the important weights of the samples in the SS-A task. Here,

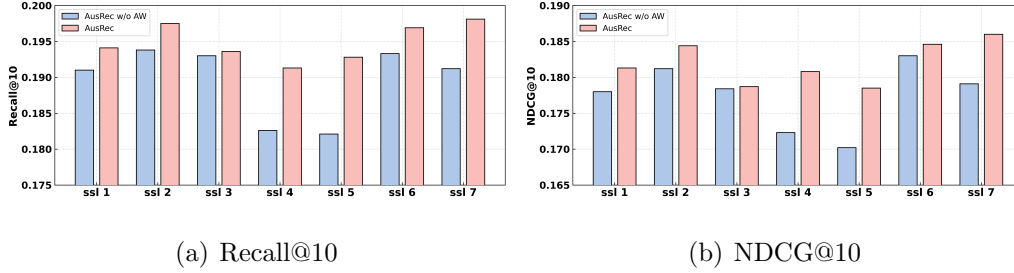


Figure 5: Results of ablation experiments (single auxiliary task).

we run the experiment on the LastFM dataset, as similar patterns are observed in other datasets. Figure 5 shows that the automatic weighting network positively impacts model performance with only one SS-A task under the evaluation metrics Recall@10 and NDCG@10. For instance, among the seven SS-A tasks, the model with the automatic weighting network on SS-A task ssl₅ can achieve the largest performance improvement compared to the model without the automatic weighting network. And the model with SS-A task ssl₃ has only a slight performance improvement compared to that of the model without the automatic weighting network. Note that ssl₃ and ssl₅ denote the SS-A tasks based on 1-hop and 3-hop social relations, respectively. The reason for these observations is that SS-A task ssl₅ can explore the high-order signals (3-hop) in social networks, hence providing more performance improvement by learning better representations, while SS-A task ssl₃ only explores the shallow signals (1-hop) in social networks, which provides limited signals to learn social information for enhancing the performance of social recommendation. Besides, the observations mentioned above also suggest the model is not just learning a single weight per task, but is learning to weight per-instance losses dynamically, which is a more sophisticated form

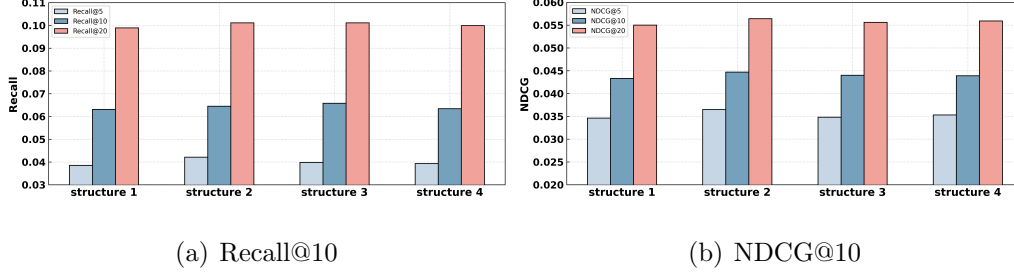


Figure 6: Effect of Automatic Weighting Network Architecture.

of optimization. In summary, the phenomenon indicates that the automatic weighting network not only automatically selects SS-A tasks to improve the primary recommendation task, but also improves the impact of a single SS-A task on the primary recommendation task.

4.3.4. Effect of Automatic Weighting Network Architecture

To examine the impact of the MLP architecture in the automatic weighting network, we conduct experiments on Epinions dataset with various MLP architecture configurations. Structures 1, 2, 3, and 4 correspond to the four structure configurations: 8-500-1, 8-1000-1, 8-1000-1000-1, and 8-100-100-1, respectively. Figure 6 presents the experimental results, showing that the network structure configuration 8-1000-1 achieves the best performance among the four configurations Epinions datasets. For example, the network structure setting 8-1000-1 achieves the best performance under NDCG@10, while the rest of the three network structure settings only achieve suboptimal performance. Therefore, we select the network structure setting 8-1000-1 as the network structure setting for the automatic weighting network in our proposed method, achieving higher recommendation performance.

Table 4: Average training time per epoch (sec/epoch) on the LastFM dataset.

Model	LightGCN	DiffNet	SGL	MHCN	HGCL	AusRec	AusRec(one)
sec/epoch	0.73	1.02	1.27	2.43	3.87	4.1	2.1

635 4.4. Training Efficiency

To evaluate the training efficiency of our model, we compared the time required to train our model for one epoch with that of other baseline models. The experimental results are presented in Table 4. As shown in the table, LightGCN and DiffNet require the shortest training time per epoch, since
640 they are trained solely on graph-structured recommendation data without incorporating any SS-A tasks. SGL and MHCN incorporate different numbers of SS-A tasks, which consequently lead to an increase in their training time. HGCL introduces multiple auxiliary tasks and employs a more complex network architecture, thereby further increasing the training time. Our
645 proposed model AusRec incorporates the largest number (seven) of auxiliary tasks among all compared approaches and adopts a meta-learning training paradigm. Consequently, its training efficiency is roughly comparable to that of HGCL. It is noteworthy that the training efficiency of AusRec can be further improved by reducing the number of SS-A tasks. As illustrated
650 in Table 4, AusRec (one) represents the variant trained with only a single auxiliary task, requires approximately half the training time per epoch compared with the full version of the model. Improving the training efficiency of the model is also an important direction for our future research.

5. Limitation and Future Work

655 Although AusRec achieves strong performance across various datasets, it still has several limitations that deserve further investigation. First, the auxiliary tasks introduced in this work are relatively limited in diversity, as they are primarily constructed from one-hop or multi-hop user-user relations. Such relationships may not be sufficient to capture the full spectrum of user
660 intentions and social influences in real-world scenarios. Second, the framework involves multiple auxiliary tasks and relies on meta-optimization for automatic weighting, which inevitably increases computational complexity compared with single-task recommendation models.

In future work, we plan to explore more diverse and semantically rich social relations (e.g., group-level, contextual, or temporal social interactions) to
665 construct auxiliary tasks that better reflect users’ multifaceted preferences. We also aim to design more efficient optimization strategies to mitigate computational costs while preserving or enhancing recommendation accuracy.

6. Conclusion

670 In this work, we proposed an automatic self-supervised learning framework for social recommendations (**AusRec**), which can automatically balance the importance of various Self-Supervised Auxiliary (SS-A) tasks to enhance representation learning in social recommendations. What’s more, our model provides advanced components with two following advantages:
675 (1) Our proposed method can incorporate many SS-A tasks for social recommendations, which can take advantage of different kinds of social relation information across different datasets and downstream tasks; (2) The proposed

automatic weighting mechanism facilitates the soft selection and balancing of SS-A tasks with the primary recommendation task, thereby enhancing user and item representation learning in social recommendations. Extensive experiments on three real-world datasets demonstrate the effectiveness of our proposed method. In addition, the ablation studies also demonstrate that the automatic weighting mechanism can adapt multiple SS-A tasks for improving the primary recommendation task.

References

- [1] Y. Liu, X. Zhou, H. Kou, Y. Zhao, X. Xu, X. Zhang, L. Qi, Privacy-preserving point-of-interest recommendation based on simplified graph convolutional network for geological traveling, *ACM Transactions on Intelligent Systems and Technology* 15 (4) (2024) 1–17.
- [2] W. Fan, Y. Ma, Q. Li, J. Wang, G. Cai, J. Tang, D. Yin, A graph neural network framework for social recommendations, *IEEE Transactions on Knowledge and Data Engineering* (2020).
- [3] L. Meng, Z. Liu, D. Chu, Q. Z. Sheng, J. Yu, X. Song, Poi recommendation for occasional groups based on hybrid graph neural networks, *Expert Systems with Applications* 237 (2024) 121583.
- [4] M. McPherson, L. Smith-Lovin, J. M. Cook, Birds of a feather: Homophily in social networks, *Annual review of sociology* (2001) 415–444.
- [5] Y. Koren, R. Bell, C. Volinsky, Matrix factorization techniques for recommender systems, *Computer* 42 (8) (2009) 30–37.

- 700 [6] W. Fan, Y. Ma, D. Yin, J. Wang, J. Tang, Q. Li, Deep social collaborative filtering, in: Proceedings of the 13th ACM Conference on Recommender Systems, 2019, pp. 305–313.
- [7] S. Ni, W. Zhou, J. Wen, L. Hu, S. Qiao, Enhancing sequential recommendation with contrastive generative adversarial network, Information Processing & Management 60 (3) (2023) 103331.
- 705 [8] P. R. de Souza, F. A. Durão, Exploiting social capital for improving personalized recommendations in online social networks, Expert Systems with Applications 246 (2024) 123098.
- [9] W. Fan, Y. Ma, Q. Li, Y. He, E. Zhao, J. Tang, D. Yin, Graph neural networks for social recommendation, in: The world wide web conference, 710 2019, pp. 417–426.
- [10] L. Wu, P. Sun, Y. Fu, R. Hong, X. Wang, M. Wang, A neural influence diffusion model for social recommendation, in: Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval, 715 2019.
- [11] J. Yu, H. Yin, J. Li, M. Gao, Z. Huang, L. Cui, Enhance social recommendation with adversarial graph convolutional networks, IEEE Transactions on Knowledge and Data Engineering (2020).
- [12] Y. A. Kim, J. Srivastava, Impact of social influence in e-commerce decision making, in: Proceedings of the ninth international conference on Electronic commerce, 720 2007, pp. 293–302.

- [13] M. Jamali, M. Ester, A matrix factorization technique with trust propagation for recommendation in social networks, in: Proceedings of the fourth ACM conference on Recommender systems, 2010.
- 725 [14] J. Tang, X. Hu, H. Liu, Social recommendation: a review, *Social Network Analysis and Mining* 3 (4) (2013) 1113–1133.
- [15] Z. Wu, X. Yang, X. Wei, P. Yuan, Y. Zhang, J. Bai, A self-supervised anomaly detection algorithm with interpretability, *Expert Systems with Applications* 237 (2024) 121539.
- 730 [16] E. Hashmi, S. Y. Yayilgan, M. M. Yamin, M. Abomhara, M. Ullah, Self-supervised hate speech detection in norwegian texts with lexical and semantic augmentations, *Expert Systems with Applications* 264 (2025) 125843.
- [17] J. Wu, X. Wang, F. Feng, X. He, L. Chen, J. Lian, X. Xie, Self-supervised graph learning for recommendation, in: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2021.
- 735 [18] X. Xia, H. Yin, J. Yu, Q. Wang, L. Cui, X. Zhang, Self-supervised hypergraph convolutional networks for session-based recommendation, in: Proceedings of the AAAI conference on artificial intelligence, Vol. 35, 2021, pp. 4503–4511.
- 740 [19] J. Yu, H. Yin, J. Li, Q. Wang, N. Q. V. Hung, X. Zhang, Self-supervised multi-channel hypergraph convolutional network for social recommendation, in: Proceedings of the Web Conference 2021, 2021.

- 745 [20] R. Tian, J. Cai, C. Li, J. Wang, Self-supervised pre-training model based on multi-view for mooc recommendation, *Expert Systems with Applications* 252 (2024) 124143.
- [21] S. Rendle, C. Freudenthaler, Z. Gantner, L. Schmidt-Thieme, Bpr: Bayesian personalized ranking from implicit feedback, *arXiv preprint arXiv:1205.2618* (2012).
- 750 [22] X. Juan, M. Peng, X. Wang, Dynamic self-training with less uncertainty for graph imbalance learning, *Expert Systems with Applications* (2025) 126643.
- [23] X. Juan, X. Liang, H. Xue, X. Wang, Multi-strategy adaptive data aug-
755 mentation for graph neural networks, *Expert Systems with Applications* 258 (2024) 125076.
- [24] Y. Wang, H. Xue, K. Zhou, T. Chen, X. Wang, A two-stage co-adversarial perturbation to mitigate out-of-distribution generalization of large-scale graph, *Expert Systems with Applications* (2024) 124472.
- 760 [25] W. Fan, X. Liu, W. Jin, X. Zhao, J. Tang, Q. Li, Graph trend networks for recommendations, *arXiv preprint arXiv:2108.05552* (2021).
- [26] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, M. Wang, Lightgcn: Simplifying and powering graph convolution network for recommendation, in: *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, 2020.
- 765

- [27] W. Fan, Q. Li, M. Cheng, Deep modeling of social relations for recommendation, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 32, 2018.
- [28] W. Zhou, A. U. Haq, L. Qiu, J. Akbar, Multi-view social recommendation via matrix factorization with sub-linear convergence rate, Expert Systems with Applications 237 (2024) 121687.
- [29] X. Chen, Z. Liu, S. Xie, K. He, Deconstructing denoising diffusion models for self-supervised learning, arXiv preprint arXiv:2401.14404 (2024).
- [30] M. Assran, Q. Duval, I. Misra, P. Bojanowski, P. Vincent, M. Rabbat, Y. LeCun, N. Ballas, Self-supervised learning from images with a joint-embedding predictive architecture, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 15619–15629.
- [31] A. Baevski, W.-N. Hsu, Q. Xu, A. Babu, J. Gu, M. Auli, Data2vec: A general framework for self-supervised learning in speech, vision and language, in: International conference on machine learning, PMLR, 2022, pp. 1298–1312.
- [32] Y.-A. Chung, Y. Zhang, W. Han, C.-C. Chiu, J. Qin, R. Pang, Y. Wu, W2v-bert: Combining contrastive learning and masked language modeling for self-supervised speech pre-training, in: 2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), IEEE, 2021, pp. 244–250.

- [33] Z. Peng, W. Huang, M. Luo, Q. Zheng, Y. Rong, T. Xu, J. Huang, Graph representation learning via graphical mutual information maximization, in: Proceedings of The Web Conference 2020, 2020, pp. 259–270.
- [34] J. Qiu, Q. Chen, Y. Dong, J. Zhang, H. Yang, M. Ding, K. Wang, J. Tang, Gcc: Graph contrastive coding for graph neural network pre-training, in: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2020, pp. 1150–1160.
- [35] K. Sun, Z. Lin, Z. Zhu, Multi-stage self-supervised learning for graph convolutional networks on graphs with few labeled nodes, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34, 2020, pp. 5892–5899.
- [36] P. Velickovic, W. Fedus, W. L. Hamilton, P. Liò, Y. Bengio, R. D. Hjelm, Deep graph infomax., ICLR (Poster) 2 (3) (2019) 4.
- [37] A. Sankar, J. Wang, A. Krishnan, H. Sundaram, Beyond localized graph neural networks: An attributed motif regularization framework, in: 2020 IEEE International Conference on Data Mining (ICDM), IEEE, 2020, pp. 472–481.
- [38] D. Hwang, J. Park, S. Kwon, K. Kim, J.-W. Ha, H. J. Kim, Self-supervised auxiliary learning with meta-paths for heterogeneous graphs, Advances in Neural Information Processing Systems 33 (2020) 10294–10305.
- [39] L. Wu, H. Lin, C. Tan, Z. Gao, S. Z. Li, Self-supervised learning on

- 810 graphs: Contrastive, generative, or predictive, *IEEE Transactions on Knowledge and Data Engineering* (2021).
- [40] W. Jin, T. Derr, H. Liu, Y. Wang, S. Wang, Z. Liu, J. Tang, Self-supervised learning on graphs: Deep insights and new direction, *arXiv preprint arXiv:2006.10141* (2020).
- 815 [41] J. Yu, H. Yin, X. Xia, T. Chen, J. Li, Z. Huang, Self-supervised learning for recommender systems: A survey, *arXiv preprint arXiv:2203.15876* (2022).
- [42] H. Liu, H. Lin, W. Fan, Y. Ren, B. Xu, X. Zhang, D. Wen, N. Zhao, Y. Lin, L. Yang, Self-supervised learning for fair recommender systems, *Applied Soft Computing* (2022) 109126.
- 820 [43] A. Jaiswal, A. R. Babu, M. Z. Zadeh, D. Banerjee, F. Makedon, A survey on contrastive self-supervised learning, *Technologies* 9 (1) (2020) 2.
- [44] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, *arXiv preprint arXiv:1810.04805* (2018).
- 825 [45] M. Joshi, D. Chen, Y. Liu, D. S. Weld, L. Zettlemoyer, O. Levy, Spanbert: Improving pre-training by representing and predicting spans, *Transactions of the Association for Computational Linguistics* 8 (2020) 64–77.
- 830 [46] A. Conneau, G. Lample, Cross-lingual language model pretraining, *Advances in neural information processing systems* 32 (2019).

- [47] K. Song, X. Tan, T. Qin, J. Lu, T.-Y. Liu, Mass: Masked sequence to sequence pre-training for language generation, arXiv preprint arXiv:1905.02450 (2019).
- 835 [48] J. Yu, H. Yin, M. Gao, X. Xia, X. Zhang, N. Q. Viet Hung, Socially-aware self-supervised tri-training for recommendation, in: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, 2021, pp. 2084–2092.
- [49] M. Li, S. Hu, F. Zhu, Q. Zhu, Few-shot learning for cold-start recommendation, in: Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), 2024, pp. 7185–7195.
- 840 [50] N. Li, J. Wang, X. Liu, L. Wang, L. Zhang, Contrastive learning-based meta-learning sequential recommendation, in: 2024 International Conference on Distributed Computing and Optimization Techniques (ICD-COT), IEEE, 2024, pp. 1–5.
- [51] X. Pan, Y. Chen, C. Tian, Z. Lin, J. Wang, H. Hu, W. X. Zhao, Multi-modal meta-learning for cold-start sequential recommendation, in: Proceedings of the 31st ACM international conference on information & knowledge management, 2022, pp. 3421–3430.
- 850 [52] J. Chen, H. Dong, Y. Qiu, X. He, X. Xin, L. Chen, G. Lin, K. Yang, Autodebias: Learning to debias for recommendation, in: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2021, pp. 21–30.

- 855 [53] C. Finn, P. Abbeel, S. Levine, Model-agnostic meta-learning for fast adaptation of deep networks, in: International conference on machine learning, PMLR, 2017, pp. 1126–1135.
- [54] H. Lee, J. Im, S. Jang, H. Cho, S. Chung, Melu: Meta-learned user preference estimator for cold-start recommendation, in: Proceedings of the 860 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019, pp. 1073–1082.
- [55] H. Bharadhwaj, Meta-learning for user cold-start recommendation, in: 2019 International Joint Conference on Neural Networks (IJCNN), IEEE, 2019, pp. 1–8.
- 865 [56] Y. Chen, B. Chen, X. He, C. Gao, Y. Li, J.-G. Lou, Y. Wang, λ opt: Learn to regularize recommender models in finer levels, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019, pp. 978–986.
- [57] Y. Zhang, F. Feng, C. Wang, X. He, M. Wang, Y. Li, Y. Zhang, How 870 to retrain recommender system? a sequential meta-learning method, in: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2020, pp. 1479–1488.
- [58] R. Miao, K. Zhou, Y. Wang, N. Liu, Y. Wang, X. Wang, Rethinking 875 independent cross-entropy loss for graph-structured data, arXiv preprint arXiv:2405.15564 (2024).

- [59] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, U. Alon, Network motifs: simple building blocks of complex networks, *Science* (2002).
- 880 [60] M. Newman, *Networks*, Oxford university press, 2018.
- [61] C. Shi, Y. Li, J. Zhang, Y. Sun, S. Y. Philip, A survey of heterogeneous information network analysis, *IEEE Transactions on Knowledge and Data Engineering* (2016).
- [62] A. R. Benson, D. F. Gleich, J. Leskovec, Higher-order organization of
885 complex networks, *Science* 353 (6295) (2016) 163–166.
- [63] H. Wang, H. Zhao, B. Li, Bridging multi-task learning and meta-learning: Towards efficient training and effective adaptation, in: *International Conference on Machine Learning*, PMLR, 2021, pp. 10991–11002.
- 890 [64] J. Shu, Q. Xie, L. Yi, Q. Zhao, S. Zhou, Z. Xu, D. Meng, Meta-weight-net: Learning an explicit mapping for sample weighting, *Advances in neural information processing systems* 32 (2019).
- [65] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, T.-S. Chua, Neural collaborative filtering, in: *Proceedings of the 26th international conference on world wide web*, 2017.
895
- [66] X. Wang, X. He, M. Wang, F. Feng, T.-S. Chua, Neural graph collaborative filtering, in: *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*, 2019.

- [67] M. Chen, C. Huang, L. Xia, W. Wei, Y. Xu, R. Luo, Heterogeneous
900 graph contrastive learning for recommendation, in: Proceedings of the
sixteenth ACM international conference on web search and data mining,
2023, pp. 544–552.