

# M2GNN: Metapath and Multi-interest Aggregated Graph Neural Network for Tag-based Cross-domain Recommendation

Anonymous Author(s)

## ABSTRACT

Cross-domain recommendation (CDR) is an effective way to alleviate the data sparsity problem. Content-based CDR is one of the most promising branches since most kinds of products can be described by a piece of text, especially when cold-start users or items have few interactions. However, two vital issues are still under-explored: (1) From the content modeling perspective, sufficient long-text descriptions are usually scarce in a real recommender system, more often the light-weight textual features, such as a few keywords or tags, are more accessible, which is improperly modeled by existing methods. (2) From the CDR perspective, not all inter-domain interests are helpful to infer intra-domain interests. Caused by domain-specific features, there are part of signals benefiting for recommendation in the source domain but harmful for that in the target domain. Therefore, how to distill useful interests is crucial. To tackle the above two problems, we propose a **metapath and multi-interest aggregated graph neural network (M2GNN)**. Specifically, to model the tag-based contents, we construct a heterogeneous information network to hold the semantic relatedness between users, items, and tags in all domains. The metapath schema is predefined according to domain-specific knowledge, with one metapath for one domain. User representations are learned by GNN with a hierarchical aggregation framework, where the intra-metapath aggregation firstly filters out trivial tags and the inter-metapath aggregation further filters out useless interests. Offline experiments and online A/B tests demonstrate that M2GNN achieves significant improvements over the state-of-the-art methods and current industrial recommender system in Dianping, respectively. Further analysis shows that M2GNN offers an interpretable recommendation.

## CCS CONCEPTS

- Information systems → Recommender systems.

## KEYWORDS

Cross-domain Recommendation, Graph Neural Network, Tag-based Recommendation

### ACM Reference Format:

Anonymous Author(s). 2022. M2GNN: Metapath and Multi-interest Aggregated Graph Neural Network for Tag-based Cross-domain Recommendation. In *Proceedings of 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2023)*. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/1122445.1122456>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

SIGIR 2023, 23-27 July, 2023, Taipei

© 2018 Association for Computing Machinery.  
ACM ISBN 978-1-4503-XXXX-X/18/06...\$15.00  
<https://doi.org/10.1145/1122445.1122456>

## 1 INTRODUCTION



Figure 1: Dianping review feed recommendaiton and some examples to explain how tags are associated with the review, search, and consumption domain. Best viewed in color.

Dianping is one of the most popular online platforms in China<sup>1</sup> for discovering local life services, including dining, shopping, entertainment, hotel, etc.. With the purpose of seeking trusted local business information, users are eager to search and consume in Dianping and write their reviews to share experiences. To increase user retention rates, Dianping establishes a review feed recommender system on the homepage as shown on the left side of Figure 1, which faces a severe data sparsity problem in the collection log<sup>2</sup>. Take one-month data (Nov. 2022) as an example, Figure 2(a) shows the distribution of collection numbers across the users. Cold-start users account for about 96% of all users, while active users (with more than ten collect histories) less than 0.2%.

To solve this dilemma, we introduce the more accessible light-weight textual side information called tags. Tag is a word or phrase, which refers to an object or activity, like hot pot and skiing. Take Figure 1 as an example, we input *beijing barbecue* in the search bar (blue box) and the associated tags are *beijing* and *barbecue*. Obviously, user tag-based interests extracted from other domains are useful for suggesting reviews on the homepage. For example, a user might get interested in a bar review because he searches for wines. We count tags associated with three domains (review, search, and consumption) via the same one-month data as shown in Figure 1. The results are in Figure 2(b) and show that users have rich behaviors in the latter two domains, especially for cold-start and inactive users. Therefore, **(Q1) how to leverage tag-based**

<sup>1</sup>Monthly, hundreds of million UV and ten million user-generated reviews are accumulated.

<sup>2</sup>Here, our goal is to improve the collection rate rather than the click-through rate, because the former is the most important indicator for profit growth in our business data analysis.

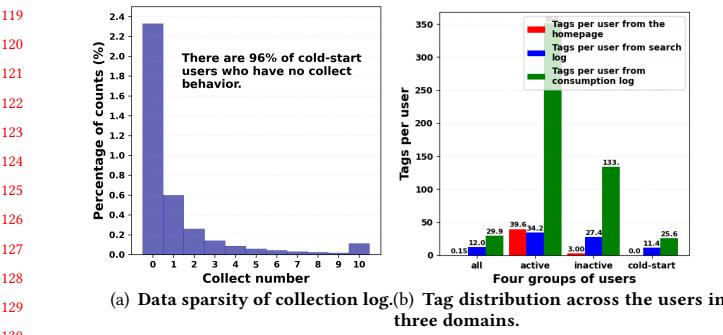


Figure 2: User cross-domain behavior analysis.

### contents to enhance recommendations in the target domain is a key issue.

Furthermore, we analyze whether all associated tags of search and consumption are advantageous for review feed recommendation, that is, the proportion of useful tags. We gather tags associated with the above three domains in Nov. 2022, called training tags, and then collect tags just associated with reviews in the following week (from Dec. 1, 2022 to Dec. 7, 2022), called testing tags. All tags are embedded by SBERT [19], which is pretrained on Dianping corpus specifically. For testing tags, we mine the relevant training tags by setting a threshold of cosine similarity between two embeddings. Figure 3 visualizes tag embeddings of a randomly selected user and shows the percentage of useful tags by averaging all users at the bottom right, which reveals a reasonable phenomenon that only a small part of tags (less than 10% when the threshold is 0.7) can represent user interests towards reviews. Therefore, **(Q2) another pivotal problem is to distill a fraction of useful tags representing intra-domain interests from a large quantity, most of which are noises.**

Cross-domain recommendation (CDR) is an intuitive branch to solve the above problems. Early research [23, 36] focus on different mapping strategies for user cross-domain preferences. However, all of them don't introduce content information. To achieve content enhanced recommendation, some single-domain workings realize great success [22, 32, 33, 40]. Based on the previous efforts, content enhanced CDR methods firstly extract content-aware interests and then transfer them across domains. CATN [39] adopts text convolution to learn aspect-level features, which are matched between the source and target domain via a correlation matrix. MTNet [10] attentively distills useful contents via a memory network and selectively feeds them into a transfer network. However, these approaches neglect how to model tag-based contents. In recent years, graph neural network (GNN) based CDR has attracted considerable research attention for its superiority in modeling high-order connections, like PPGN [38] and HecRec [37]. However, the aggregation via the GNN layer in the above methods overlooks noisy neighbor nodes, which are indeed harmful when the useless nodes account for the most.

In this paper, we propose a metapath and multi-interest aggregated graph neural network for tag-based cross-domain recommendation, called M2GNN. To solve **Q1**, we leverage the ability of graph to recognize the pattern of non-euclidean data [41] including tag-based contents. Technically, a tag-associated heterogeneous graph

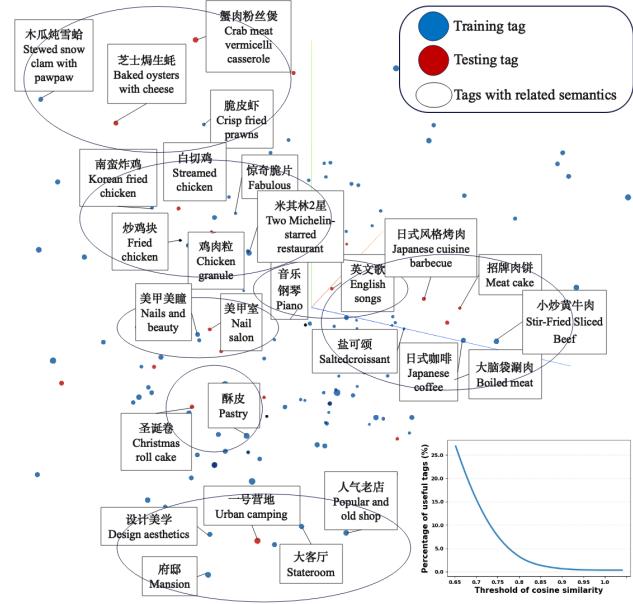


Figure 3: The proportion of useful tags and an example.

is established to model the multi-homed behaviors, where each domain is represented by a metapath. Then a GNN-based method is designed to explore high-order tag-based cross-domain interests for enhancing recommendation in the target domain. To address **Q2**, we introduce a novel two-step aggregation mechanism employing the dynamic routing network and self-attention network. The former filters out the trivial tags and learns multiple high-level user interests in each domain. The latter furthermore transfers more significant interests into the target domain. Moreover, we introduce the skip-gram regularization to model the semantic correlation between tags. To summarize, the main contributions of this work are as follows:

- We propose a novel CDR problem whose side information is tags and design a GNN-based method to solve it.
- We present a hierarchical aggregation schema to learn user preferences when noisy neighbor nodes are in the majority.
- We conduct offline experiments on a newly proposed industrial dataset and a public dataset, which proves the effectiveness and interpretability of M2GNN. Online A/B tests demonstrate the improvement to the current industrial recommender system.

## 2 PRELIMINARIES

In this section, we first formulate the problem and then describe the heterogeneous graph extracted from Dianping with some notations.

### 2.1 Problem Setup

**Tag-based cross-domain recommendation (TCDR).** We have a target domain (e.g. review collection) and some source domains (e.g. search and consumption), denoted as  $D^t$  and  $D^s$ , where  $S = \{s_1, s_2, \dots\}$  represents there are  $|S|$  source domains. Each domain includes its users  $u$ , items  $i$ , and interactions  $\hat{y}_{ui}$ . Note that users are full overlap, also known as multi-homed users, while items are non-overlap. We identify the item in different domains by its superscript, such as  $i^t$ ,  $i^{s_1}$ , and  $i^{s_2}$ . According to interaction

numbers in the target domain, users are divided into three groups, which are cold-start, inactive and active users. Let  $U^{cs}$ ,  $U^{ia}$  and  $U^a$  represent them, respectively. In this paper, the bound to distinguish the latter two groups is set to be 10 times. Cold-start users have no interactions with items in  $D^t$ , but with items in  $D^s$ . Each item in all domains has the corresponding content information denoted by  $D_i$ . Different from the document with a common structure of the linguistic sequence,  $D_i$  is a set of tags. **Note that all tags come from the same vocabulary<sup>3</sup>, which is shared within all domains.**

The goal of TCDR is to improve the recommendation performance in the target domain by exploring latent tag-based interests behind cross-domain behaviors, which is especially needed for  $U^{cs}$  and  $U^{ia}$ . In this work, we target at the top-K recommendation task<sup>4</sup>.

## 2.2 Graph Construction

**Definition 2.1. Tag-associated Heterogenous Graph.** Figure 4 illustrates three kinds of interactions from the review feed, search, and consumption domain in Dianping APP. There are six kinds of entities in this TCDR task, which are user, review feed, search query, POI, review of POI, and tag. We denote them as  $\mathcal{U} = \{u\}, \mathcal{R} = \{r\}, \mathcal{Q} = \{q\}, \mathcal{P} = \{p\}, \mathcal{RP} = \{rp\}$  and  $\mathcal{T} = \{t\}$ , respectively. Obviously,  $i^t$  is identical to  $r$ ,  $i^s = \{i^{s_1}, i^{s_2}\} = \{q, p\}$  and  $D_i = \{t\}$ . Eight relations are used to connect the above nodes, which are  $\{u - r, u - q, u - p, p - rp, r - t, q - t, rp - t, t - t\}$ . The first seven edges are easy to understand and we explain the last relation.  $t - t$  represents two tags that have related semantics, like *Japanese barbecue* and *Japanese coffee*. We use Elasticsearch (ES) to store all tags and utilize its search engine to construct  $t - t$  by regarding each tag as a query. Then a heterogeneous graph  $\mathcal{G}$  is defined as  $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$  with nodes  $\mathcal{V}$  and edges  $\mathcal{E}$ . In Dianping dataset,  $\mathcal{V} = \{\mathcal{U}, \mathcal{R}, \mathcal{Q}, \mathcal{P}, \mathcal{RP}, \mathcal{T}\}$  and  $\mathcal{E} = \{u - r, u - q, u - p, p - rp, r - t, q - t, rp - t, t - t\}$ . Note that only  $t - t$  is a symmetric relation and the others are directed.

**Definition 2.2. Domain-specific Metapath.** A metapath  $\rho$  is a path defined on the graph  $\mathcal{G}$ , and is denoted in the form of ordered nodes  $A_1 \rightarrow A_2 \rightarrow \dots \rightarrow A_L$ .  $d_\rho$  is the number of metapaths. In Dianping dataset, there are three metapaths ( $d_\rho = 3$ ) to represent domain-specific knowledge.  $u - r - t$  represents user interested tags from the collected review feed.  $u - q - t$  contains user potential preference inferred from his search log.  $u - p - rp - t$  describes what content of a local business can develop users' willingness to consume.

**Definition 2.3. Metapath-based neighbor.** Given a metapath  $\rho$ , the metapath-based neighbors  $\mathcal{N}_v^\rho$  of a node  $v$  is defined as the set of nodes that connect with  $v$  using  $\rho$ .

## 3 THE M2GNN FRAMEWORK

In this section, we start from two motivations: (1) why we adopt the GNN framework to solve this TCDR task; (2) why we propose such an aggregation mechanism to distill intra-domain interests. Then the technical details of the proposed M2GNN are described. Figure 5 presents the framework of M2GNN.

<sup>3</sup>Dianping constructs a large-scale tag database extracted from user-generated contents via a customized tokenizer and provides a unified API to associate tags with contents.

<sup>4</sup>Dianping review feed recommender system is a complex industrial system with a tandem structure. After the top-K rating, also known as the matching or recall process, a customized CTR module will rate all candidate items by taking into account a cluster of additional features, like user age, occupation, and whether paid for advertising. Therefore, this work just aims to improve the recall performance.

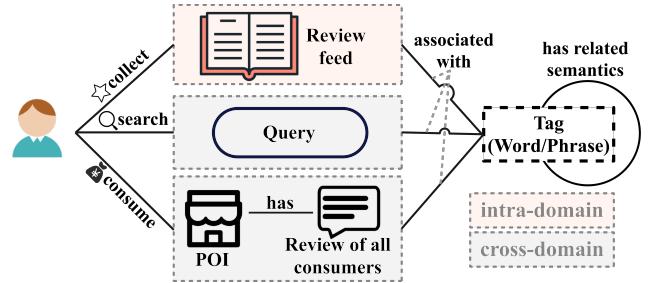


Figure 4: The tag-associated heterogeneous graph constructed by Dianping data. Best viewed in color.

### 3.1 Motivation

As aforementioned in Q1, the content side information in this TCDR task is tags. Different from contents with the linguistic sequence, tags have no natural sequential distribution, leading to the recent works focusing on sequential feature extraction like CATN, which will be compared in Section 4, is less applicable. What's more, tags have an important feature, that is global shareability, which means any items in all domains are described by the same tag dict. This inspires us to recall the review feed including the same or similar tags appeared in other domains. For example, user  $\xrightarrow{\text{search}}$  *korean barbecue*  $\xrightarrow{\text{associated with}}$  *barbecue*  $\xleftarrow{\text{has related semantics}}$  *janpanese barbecue*  $\xleftarrow{\text{associated with}}$  *review feed* can help to provide a review about janpanese barbecue for a user who searches korean barbecue. GNN, an intuitive tool to explore such a high-order reasoning chain, motivates us to build a tag-based graph and corresponding neural networks.

With determining to employ the GNN framework, we face severe noisy information in aggregation as shown in Q2. This is an inevitable problem when the number of attention members is large, which also appears in GraphAir [8]. Hard attention [35], also known as nondeterministic attention, is a close solution to focus on exactly one or a few members, which will be compared in Section 4. However, the tag is a fine-grained object, which means a cluster of related tags indicates one entire high-level user preference. Therefore, discarding most of tags directly adopted in hard attention will lose content side information and degrade performance. Inspired by the success of multi-interest recommender systems (RSs), we find extracting multiple interests from each domain can distill intra-domain interests in a corpus where noisy contents account for the most with the following two reasons: (1) learning different aspects of user preference can preserve main but filter out trivial tags; (2) attention mechanism works better in transferring high-level interests into target domain than transferring tags directly because the number of interests is much less than that of tags.

### 3.2 Overview

A general framework that can generalize most GNN-based RSs (e.g. NGCF [29], LightGCN [6], KGAT [27], KGIN [30], GC-SAN [34] and SURGE [2]) adopts the following scheme: aggregate feature (neighbor) nodes to user or item (center) nodes and update  $\rightarrow$  pool node representations after multiple layers  $\rightarrow$  train the model

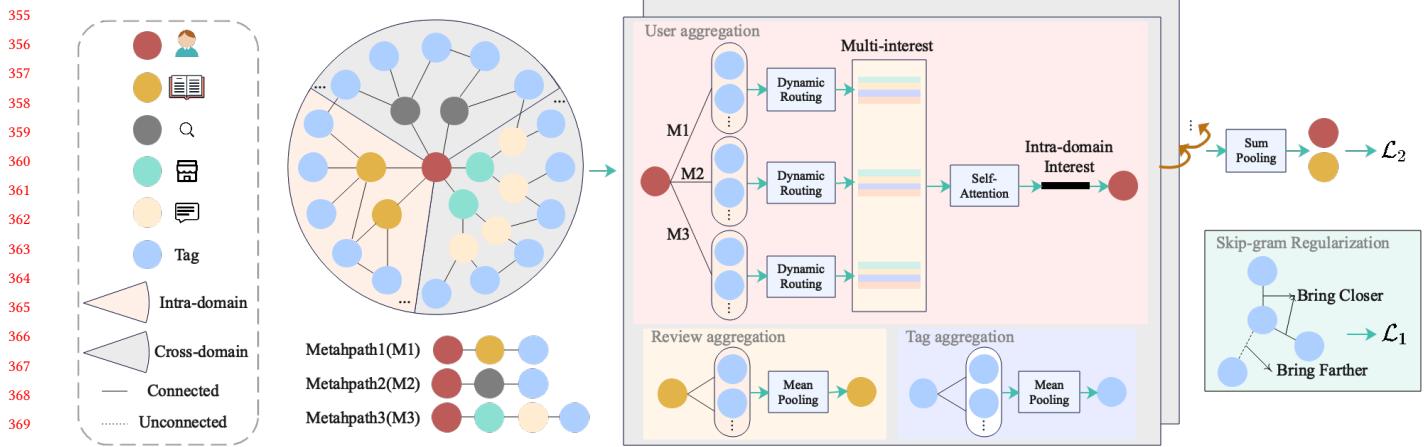


Figure 5: The framework of M2GNN. Best viewed in color.

according to downstream tasks like top-K recommendation or CTR prediction.

Inspired by the above straightforward but effective idea, M2GNN aggregates tag nodes into user nodes by domain-specific metapaths without self-connection, learns item embeddings by their corresponding tags, and enhances representations of tag nodes by their homogeneous neighbors with related semantics. Since the number of tags associated with an item is small, adopting complex aggregation methods is unnecessary and here we use average pooling. The inherence of the tag-to-tag update is to bring the vectors of two tags with a similar meaning closer in latent semantic space, we also utilize average pooling and leave other aggregation functions like attention as a future work. Then sum operation is conducted to pool multi-layer embeddings, which is widely used in GNN for the recommendation task to fuse multi-hop collaborative filtering signals. Finally, the inner product is adopted to predict top-K items. Formally, the above message passing and recommendation mechanism are defined as follows,

$$\mathbf{e}_t^{l+1} \leftarrow \text{Mean}_{t' \in N_t^{l-t}} (\mathbf{e}_{t'}^l) \quad (1)$$

$$\mathbf{e}_r^{l+1} \leftarrow \text{Mean}_{t \in N_r^{l-t}} (\mathbf{e}_t^l) \quad (2)$$

$$\mathbf{e}_u^{l+1} \leftarrow \text{Aggregate}(\mathbf{e}_t^l)_{t \in N_u^\rho} \quad (3)$$

$$\mathbf{e}_v^* = \sum_{l=0}^L \mathbf{e}_v^l \quad (4)$$

$$y(u, r) = \mathbf{e}_u^* \top \mathbf{e}_r^* \quad (5)$$

where  $v$  can be either a user or an item node in target domain  $r$ ,  $\mathbf{e}_v^l \in \mathbb{R}^{d \times 1}$  denotes the representation of node  $v$  in the  $l$ -th layer, superscript \* indicates the final representation for predicting,  $y(u, r)$  is the matching score of a pair of user and item.

Obviously, the key component is equation 3. As mentioned in Section 3.1, we first learn multiple interests in each domain (cf. Section 3.3) and transfer them into the target domain (cf. Section 3.4).

### 3.3 Intra-domain Aggregation

We utilize the dynamic routing network from [21] as multi-interest extractor. For each domain (metapath), tag embeddings and user preferences are regarded as low and high-level capsules, respectively. Given a set of metapath-based neighbor tag nodes  $t \in \mathcal{N}_u^\rho$ , we learn interest capsules as

$$\mathbf{c}_{u,j}^{\rho,l} = \sum_{t \in \mathcal{N}_u^\rho} w_{tj} \mathbf{S} \mathbf{e}_t^l \quad (6)$$

where  $\mathbf{c}_{u,j}^{\rho,l}$  denotes  $j$ -th interest capsule of user  $u$  using metapath  $\rho$  in  $l$ -th layer,  $w_{tj}$  represents the contribution degree of low-level capsule  $t$  to high-level capsule  $j$ ,  $\mathbf{S} \in \mathbb{R}^{d \times d}$  is transformation matrix and is shared across all interest capsules and metapaths.

The superiority of dynamic routing is to update  $w_{tj}$  via a new parameter called routing logit  $b_{tj}$  in an iterative way. We initialize  $b_{tj}$  randomly in the first iteration.  $w_{tj}$  is calculated by performing softmax on  $b_{tj}$  as

$$w_{tj} = \frac{\exp b_{tj}}{\sum_{z=1}^K \exp b_{tz}} \quad (7)$$

where  $K$  is the number of interest capsules and is adaptively calculated by the number of interacted tags as

$$K = \max(1, \min(K_{\max}, \lceil \log_2(|\mathcal{N}_u^\rho|) \rceil)) \quad (8)$$

where  $K_{\max}$  is the maximum number of interest capsules and  $\lceil x \rceil$  refers to the smallest integer that is more than or equal to  $x$ .

To normalize the length of the output vector of a capsule between 0 and 1, which benefits updating  $b_{tj}$  in next iteration, a non-linear squashing function is utilized as

$$\mathbf{z}_{u,j}^{\rho,l} = \text{squash}(\mathbf{c}_{u,j}^{\rho,l}) = \frac{\|\mathbf{c}_{u,j}^{\rho,l}\|^2}{1 + \|\mathbf{c}_{u,j}^{\rho,l}\|^2} \frac{\mathbf{c}_{u,j}^{\rho,l}}{\|\mathbf{c}_{u,j}^{\rho,l}\|} \quad (9)$$

Then  $b_{tj}$  is updated by

$$b_{tj} = b_{tj} + \mathbf{z}_{u,j}^{\rho,l} \mathbf{S} \mathbf{e}_t^l \quad (10)$$

Finally, the above routing process is repeated multiple times, which is set to be three, to obtain the converged output of interest capsules.

### 473 3.4 Inter-domain Aggregation

474 We collect all interest embeddings as a matrix  $Z_u^l = [z_{u,1}^{\rho_1,l}, \dots, z_{u,K_{\max}}^{\rho_1,l}, z_{u,1}^{\rho_2,l}, \dots, z_{u,K_{\max}}^{\rho_2,l}] \in \mathbb{R}^{d \times (d_{\rho} \cdot K_{\max})}$ . For faster and unified tensor calculation, we pad zero vectors when  $K < K_{\max}$ . Now each user has an interest matrix ( $Z_u^l$ ) consisting of many interest embeddings ( $z_{u,j}^{\rho,l}$ ). To transfer useful preferences to the target domain, we assign a weight  $\alpha_{u,j}^{\rho,l}$  to each interest embedding and output an intra-domain interest embedding by pooling  $Z_u^l$  attentively.

483 Since sparse data in the target domain is undesirable to train  
484 accurate weights, a reasonable alternative solution is to utilize  
485 user similarity. For example, if two users have the same search  
486 and consumption log, it is reasonable to recommend a review feed  
487 collected by one user to the other. Technologically, we aim to assign  
488 the same weight to a pair of close interest embeddings from two  
489 users with similar portraits (a.k.a.  $Z_u^l$ ). Therefore, a global shared  
490 mechanism is required. To achieve this, a self-attention module is  
491 designed as

$$492 H_u^l = S_2^\top \tanh(S_1 Z_u^l) \quad (11)$$

493 where  $H_u^l \in \mathbb{R}^{1 \times (d_{\rho} \cdot K_{\max})}$  is the attention vector,  $S_1 \in \mathbb{R}^{d \times d}$  and  
494  $S_2 \in \mathbb{R}^{d \times 1}$  are trainable matrixes, which are shared across all users.

495 Q2 reveals that only a fraction of interests contributes to the  
496 target domain, therefore we introduce the element-wise exponentiation  
497 operator before a softmax function to calculate weights  
498 as:

$$500 \alpha_{u,j}^{\rho,l} = \frac{\exp(\text{pow}(h_{u,j}^l, \gamma))}{\sum_{z=1}^{d_{\rho} \cdot K_{\max}} \exp(\text{pow}(h_{u,z}^l, \gamma))} \quad (12)$$

501 where  $h_{u,j}^l$  is the  $j$ -th element in  $H_u^l$  and  $\gamma$  is a hyperparameter to  
502 control the attention distribution. As  $\gamma$  increases, more significant  
503 interest embeddings will receive more attention and bigger weights,  
504 and vice versa. Consider the limit case,  $\gamma = 0$  is identical to mean  
505 pooling and  $\gamma = \infty$  means the hard attention which chooses only  
506 one candidate as output. Therefore,  $\gamma$  is set to be bigger than 1,  
507 which can alleviate the problem in Q2.

508 Finally, the user representation is inferred to represent his intra-  
509 domain interests as

$$510 e_u^{l+1} = Z_u^l A_u^l {}^\top \quad (13)$$

511 where  $A_u^l = [\alpha_{u,1}^{\rho,l}, \dots, \alpha_{u,d_{\rho} \cdot K_{\max}}^{\rho,l}] \in \mathbb{R}^{1 \times (d_{\rho} \cdot K_{\max})}$  is the weight ma-  
512 trix of user  $u$  in  $l$  layer.

### 517 3.5 Skip-gram Regularization

518 It is clear that the quality of tag embeddings deeply affects the  
519 clustering performance in Section 3.3 and weight computing in  
520 Section 3.4. To further draw related tags closer and unconcerned  
521 tags removed in latent semantic space, we adopt skip-gram idea to  
522 regularize the training process. Specifically, a tag node is regarded  
523 as the center word, its homogeneous neighbor nodes are set to  
524 be nearby words. The principle is to maximize the co-occurrence  
525 probability of related tags, and vice versa. We randomly sample a  
526 negative node for each positive pair of tags and define skip-gram  
527 objective loss as

$$528 \mathcal{L}_1 = -\frac{1}{|\mathcal{E}^{t-t}|} \sum_{t \in \mathcal{T}} \sum_{t' \in N_t^{t-t}, t' \notin N_t^{t-t}} (p(t' | t) - p(t'' | t)) \quad (14)$$

532 where  $|\mathcal{E}^{t-t}|$  denotes the number of all edges belonging to type  
533  $t - t$  in graph  $\mathcal{G}$ .

534 The basic formulation to model probability  $p$  is the global soft-  
535 max performed on all tags, which is impractical for the huge cost of  
536 time and calculation. Inspired by hierarchical softmax, an auxiliary  
537 set of trainable embeddings  $\mathcal{V} \in \mathbb{R}^{d \times |\mathcal{T}|}$  is introduced to define  $p$   
538 as

$$539 p(t' | t) = \log \sigma(v_{t'}^\top e_t^0) \quad (15)$$

540 where  $v_{t'}$  is the embedding of tag  $t'$  from  $\mathcal{V}$ ,  $e_t^0$  is the initial em-  
541 bedding of tag  $t$ .

### 543 3.6 Model Optimization

544 With the user representation  $e_u^*$  and the review representation  $e_r^*$   
545 ready, equation 5 is adopted to predict top-K items. We employ the  
546 BPR loss [20] as

$$547 \mathcal{L}_2 = -\frac{1}{|\mathcal{O}|} \sum_{(u, r^+, r^-) \in \mathcal{O}} \ln \sigma(y(u, r^+) - y(u, r^-)) + \lambda \|\Theta\|_2^2 \quad (16)$$

551 where  $\mathcal{O} = \{(u, r^+, r^-) | (u, r^+) \in \hat{y}_{ui}, (u, r^-) \notin \hat{y}_{ui}\}$  denotes the  
552 training set, where  $r^+$  is the observed interaction and  $r^-$  is a nega-  
553 tive counterpart which is sampled from the unobserved interactions  
554 randomly,  $\sigma(\cdot)$  is the sigmoid function.  $\Theta = \{e_x^0, \mathcal{S}, S_1, S_2, v_t | x \in$   
555  $\{\mathcal{U}, \mathcal{R}, \mathcal{T}\}, t \in \mathcal{T}\}$  is the model parameter set.  $L_2$  regularization  
556 parameterized by  $\lambda$  on  $\Theta$  is conducted to prevent overfitting. Finally,  
557 the joint loss is

$$558 \mathcal{L} = \mathcal{L}_1 + \mathcal{L}_2 \quad (17)$$

## 559 4 EXPERIMENTS

560 We evaluate our proposed M2GNN method on two benchmark  
561 datasets to answer the following research questions:

- 562 • **RQ1:** Does our proposed M2GNN outperform the state-of-the-art  
563 recommendation methods?
- 564 • **RQ2:** How do different components (i.e., two-step aggregation  
565 module and skip-gram regularization) and hyperparameters (i.e.  
566 capsule numbers, GNN layer numbers, and exponent before soft-  
567 max) affect M2GNN?
- 568 • **RQ3:** Does M2GNN achieve great improvement towards cold-  
569 start and inactive users?
- 570 • **RQ4:** Is there an online improvement of review feed recom-  
571 mendation in Dianping homepage by M2GNN?
- 572 • **RQ5:** Can M2GNN provide potential explanations about tag-  
573 based preferences of users?

### 576 4.1 Experimental Settings

577 **Dataset Description.** We propose a novel industrial dataset, called  
578 DPB (Dianping Beijing). The dataset is encrypted and desensitized,  
579 and does not contain any personally identifiable information. To  
580 avoid data leakage, we collect the multi-site data in one month  
581 (from Nov. 1, 2022, to Nov. 30, 2022) as the training and validation  
582 set. Then the latter one-week data (from Dec. 1, 2022, to Dec. 7, 2022)  
583 is the testing set. We also use a common dataset, Amazon review  
584 dataset<sup>5</sup>, where the book and movie are chosen as the source and  
585 target domain. To ensure data quality, we remove the interaction  
586 records without review text and some low-quality items following  
587 earlier studies [7, 11]. We split the review text into individual words

588 <sup>5</sup><http://jmcauley.ucsd.edu/data/amazon/>

**Table 1: Statistics of two datasets.  $m$  and  $b$  denote movie and book, respectively. ‘int’ indicates the interaction data of each domain.**

|           | DPBJ      |               | Amazon      |           |
|-----------|-----------|---------------|-------------|-----------|
| # $u$     | 6,156,837 |               | 41,465      |           |
| # $t$     | 94,029    |               | 20,000      |           |
| # $t-t$   | 1,679,832 |               | 184,776     |           |
| $D^t$     | item      | # $r$         | 2,036,665   | # $m$     |
|           | int.      | # $u-r$       | 297,658     | # $u-m$   |
|           |           | # $r-t$       | 2,917,242   | # $m-t$   |
| $D^{S_1}$ | item      | # $q$         | 3,436,125   | # $b$     |
|           | int.      | # $u-q$       | 54,527,624  | # $u-b$   |
|           |           | # $q-t$       | 5,936,250   | # $b-t$   |
| $D^{S_2}$ | item      | # $p$         | 207,766     |           |
|           |           | # $r_p$       | 2,103,670   |           |
|           |           | # $u-p$       | 15,473,732  |           |
|           | int.      | # $p-r_p$     | 160,263     |           |
|           |           | # $r_p-t$     | 3,646,553   |           |
| metapath  |           | # $u-r-t$     | 497,532     | # $u-m-t$ |
|           |           | # $u-q-t$     | 73,904,447  | # $u-b-t$ |
|           |           | # $u-p-r_p-t$ | 184,203,329 |           |

and filter out stopwords by word tokenization tool like nltk<sup>6</sup>. The above all words comprise the tag dict. The detailed statistics of these two datasets are reported in Table 1.

**Baselines.** To demonstrate the effectiveness, we compare M2GNN with **single-domain** (MF, lightGCN), **content-free cross-domain** (EMCDR, PTUPCDR), **content-based cross-domain** (CATN, MTNet), and **GNN-based cross-domain** (HeroGRAPH, HCDIR, GA) methods.

- **MF** [20] is a matrix factorization method optimized by the Bayesian personalized ranking (BPR) loss.
- **lightGCN** [6] simplifies NGCF [29] and provides simple but effective model for user-item bipartite graph-based RS.
- **EMCDR** [17] first pretrains embeddings in each domain and then learns a mapping function to transfer interests across domains.
- **PTUPCDR** [43] designs a meta network fed with users’ characteristic embeddings to generate personalized bridge functions to achieve personalized transfer of user preferences.
- **CATN** [27] extracts multiple aspects from the documents and learns aspect correlations across domains attentively.
- **MTNet** [9] designs a memory component to find the words highly relevant to the user preferences, which are then fused with source domain-specific knowledge to improve final prediction.
- **HeroGRAPH** [3] constructs a heterogeneous graph to fuse all interactions as a whole and learn additional global embeddings to enhance recommendation in all domains.
- **HCDIR** [1] separately learns user embeddings in each domain via an insurance information network and then employs MLP to perform latent space matching across domains.
- **GA** [42] proposes a unified framework, which constructs separate heterogeneous graphs to generate more representative

<sup>6</sup><https://github.com/nltk/nltk>

embeddings and uses an element-wise attention mechanism to effectively combine them of common entities.

**Parameter Settings.** We implement our M2GNN model in Pytorch and Deep Graph Library (DGL)<sup>7</sup>, which is a Python package for deep learning on graphs. Since DPBJ has million-scale nodes and billion-scale edges, the API of distributed data parallel (DDP) in Pytorch is used to train M2GNN and the data loading package in DGL is adopted to generate subgraph samples for each batch. We released all implementations (code, datasets, parameter settings, and training logs) to facilitate reproducibility<sup>8</sup>. The embedding size of all models is set to be 64 for a fair comparison. We adopt Adam [12] as the optimizer and the batch size is fixed at 1024 for all methods. For EMCDR, two domains in DBLP are combined into a whole source domain and we use the version, which employs BPR as its latent factor model and an MLP as the mapping function. For CATN, SBERT [19] generates pretrained embeddings of tags for DPBJ while the 300-dimension word embeddings pretrained in Google News<sup>9</sup> [18] is utilized for Amazon dataset. For MTNet, given an interaction  $(u, i)$ , the corresponding document consists of tags associated with this user and item. For HCDIR, we predefine the same metapaths used in M2GNN for a fair comparison. For GA, we use GA-MTCDR-P version because it performs best.

**Evaluation Metrics.** As mentioned in Section 2.1, Recall@ $K$  and Hit@ $K^{10}$  are used in the task of top-K recommendation.

## 4.2 Performance Comparison (RQ1)

We report the empirical results in Table 2. The observations are as follows:

- M2GNN consistently achieves the best performance on two datasets in terms of all measures. Specifically, it achieves significant improvements over the strongest baselines w.r.t. Recall@50 by 18.46%, 6.80% in DPBJ and Amazon, respectively. The results prove the effectiveness of M2GNN.
- Despite some particular cases, the average performances of content-aware (content-based&GNN-based) cross-domain, content-free cross-domain, and single-domain models are in decreasing order. A clear reason is whether to introduce the auxiliary information, like contents or cross-domain interactions.
- MTNet outperforms CATN in DPBJ while underperforms CATN in Amazon. The reason is the difference between content extractors in these two models. MTNet adopts an attention network to model unstructured text without regarding the sequential feature, thus performing better in DPBJ dataset. CATN utilizes text convolution and has superiority in modeling sequential signals, which is more suitable for the Amazon dataset.
- HeroGRAPH outperforms GA-MTCDR-P (HCDIR) in DPBJ while underperforms GA-MTCDR-P (HCDIR) in Amazon. DPBJ has a large number of noisy tags, leading to a big gap between intra and cross-domain interests. Therefore, using multiple embeddings to represent domain-specific interests separately (HeroGRAPH) performs better than using one representation (GA-MTCDR-P/HCDIR). However, Amazon dataset has higher consistency

<sup>7</sup><https://github.com/dmlc/dgl>

<sup>8</sup><https://anonymous.4open.science/r/M2GNNonony/>

<sup>9</sup><https://code.google.com/archive/p/word2vec/>

<sup>10</sup> $K$  is set to be 50 rather than a commonly used value like 20 or 100. The reason is that the subsequent CTR module only ranks the top 50 items from the recall phase.

**Table 2: Performance comparison. ▲% denotes the relative improvement of M2GNN over the best SOTA algorithm.**

|                            |            | DPBJ          |               |               |               | Amazon        |               |               |               |
|----------------------------|------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|
|                            |            | Recall@50     | Recall@100    | Hit@50        | Hit@100       | Recall@50     | Recall@100    | Hit@50        | Hit@100       |
| single-domain              | MF         | 0.0265        | 0.0421        | 0.0529        | 0.0796        | 0.0612        | 0.0989        | 0.3874        | 0.5218        |
|                            | LightGCN   | 0.0388        | 0.0595        | 0.0697        | 0.1045        | <u>0.1044</u> | 0.1599        | 0.5232        | 0.6475        |
| content-free cross-domain  | EMCDR      | 0.0425        | 0.0699        | 0.0752        | 0.1137        | 0.0670        | 0.1056        | 0.4271        | 0.5790        |
|                            | PTUPCDR    | 0.0499        | 0.0731        | 0.0825        | 0.1331        | 0.0727        | 0.1341        | 0.4901        | 0.6515        |
| content-based cross-domain | CATN       | 0.0304        | 0.0495        | 0.0560        | 0.0829        | 0.0815        | 0.1354        | 0.5323        | 0.6811        |
|                            | MTNet      | <u>0.0531</u> | <u>0.0822</u> | 0.0877        | 0.1362        | 0.0707        | 0.1199        | 0.4761        | 0.6377        |
| GNN-based cross-domain     | HeroGRAPH  | 0.0437        | 0.0713        | <u>0.0884</u> | <u>0.1381</u> | 0.0884        | 0.1523        | 0.5514        | 0.7065        |
|                            | HCDIR      | 0.0320        | 0.0562        | 0.0653        | 0.1099        | 0.1005        | 0.1680        | 0.5785        | 0.7194        |
|                            | GA-MTCDR-P | 0.0341        | 0.0577        | 0.0690        | 0.1153        | 0.1028        | <u>0.1703</u> | <u>0.5977</u> | <u>0.7318</u> |
| M2GNN                      |            | <b>0.0629</b> | <b>0.0953</b> | <b>0.0967</b> | <b>0.1464</b> | <b>0.1115</b> | <b>0.1783</b> | <b>0.6471</b> | <b>0.7768</b> |
| ▲%                         |            | 18.46         | 15.94         | 9.39          | 6.01          | 6.80          | 4.70          | 8.27          | 6.15          |

**Table 3: Impact of two-step aggregation and skip-gram regularization.**

|                           | DPBJ      |        | Amazon    |        |
|---------------------------|-----------|--------|-----------|--------|
|                           | Recall@50 | Hit@50 | Recall@50 | Hit@50 |
| M2GNN-mean                | 0.0511    | 0.0762 | 0.0864    | 0.5264 |
| M2GNN-softmax             | 0.0524    | 0.0791 | 0.0837    | 0.4879 |
| M2GNN-HAN                 | 0.0544    | 0.0809 | 0.0898    | 0.5631 |
| M2GNN-hard                | 0.0586    | 0.0873 | 0.0951    | 0.6062 |
| M2GNN-w/o $\mathcal{L}_1$ | 0.0526    | 0.0822 | 0.0903    | 0.5400 |

in user preferences toward books and movies, thus aggregating cross-domain interests is more applicable.

### 4.3 Study of M2GNN (RQ2)

**Impact of two-step aggregation module.** To demonstrate the superiority of the two-step aggregation module, we compare the performance of M2GNN with the following four variants which adopt different methods of aggregation.

- **M2GNN-mean** uses average pooling in both two steps.
- **M2GNN-softmax** adopts the standard schema of attention, proposed in Transformer [25], in both two steps.  $K$  is user embedding, while  $Q$  and  $V$  are tag embeddings.
- **M2GNN-HAN** regards intra-domain and inter-domain aggregation as node-level and semantic-level attention in HAN [31].
- **M2GNN-hard** utilizes local alignment attention in [16], which extends original hard attention [35] to select a subset of input members via implementing a softmax distribution. For a fair comparison, the number of chosen nodes is equal to that of interests in each domain and inter-domain aggregation adopts the same self-attention module in M2GNN.

The results are shown in the upper part of Table 3 and the major findings are as follows:

- M2GNN-mean and M2GNN-softmax achieve the worst performances on the two datasets. The reason is both of them don't track the noisy content problem. The former regards all tags as equally important, while the latter receives a large number of attention members, leading to the weights of significant tags are not trained as expected.

- M2GNN-HAN slightly outperforms the first two variants. A possible reason is that the different mechanism to calculate attention scores using the type-specific transformation matrix and the weight coefficient helps to increase the weights of useful tags.
- M2GNN-hard performs the best among the four variants. A clear reason is that only a subset of significant tags are aggregated into user representations, which avoids using all tags to infer final embeddings. However, as aforementioned in Section 3.1, it is inevitable to discard some useful tags in hard attention, which causes that M2GNN-hard underperforms M2GNN.

**Impact of skip-gram regularization.** To demonstrate the superiority of the skip-gram regularization, we discard  $\mathcal{L}_1$  and only leverage collaborative filtering signals to train the model, termed M2GNN-w/o  $\mathcal{L}_1$ . The lower part of Table 3 shows that M2GNN-w/o  $\mathcal{L}_1$  sharply underperforms M2GNN without regularizing tag embeddings via  $\mathcal{L}_1$ , which proves the necessity of the skip-gram regularization.

**Impact of hyperparameters.** We investigate the influence of three hyperparameters: layer numbers, capsule numbers, and exponent before softmax, all of which greatly affect multi-interest extraction and transfer.  $L$ ,  $K_{\max}$  and  $\gamma$  are searched in  $\{1, 2, 3\}$ ,  $\{2, 4, 6, 8\}$  and  $\{2, 4, 6, 8\}$ , respectively. Figure 6 shows the results in the DPBJ dataset and it has a similar conclusion in the Amazon dataset, which is omitted. From the observations above, we make several conclusions. Firstly,  $L = 2$  results in the best performance, which means 2-order connections integrate more similar tags into users and stacking more layers continually brings in noisy tags, which is harmful to the recommendation. Secondly, the increase of interest capsules indeed improves the recall list since local businesses contain multiple aspects, while there is a saturation limit which has a similar conclusion in [13]. Thirdly, the optimal number of  $\gamma$  is 6~8. The reason is that a proper exponent helps the softmax function to identify more important interests and assign bigger scores.

### 4.4 Cold-start and Inactive Users Analysis (RQ3)

One motivation for exploiting CDR is to alleviate the severe sparsity problem in Dianping, where an overwhelming majority of users are cold-start in Figure 2. Therefore, we investigate whether introducing CDR helps alleviate this issue. Towards this end, we divide the testing set into four groups based on the interaction number

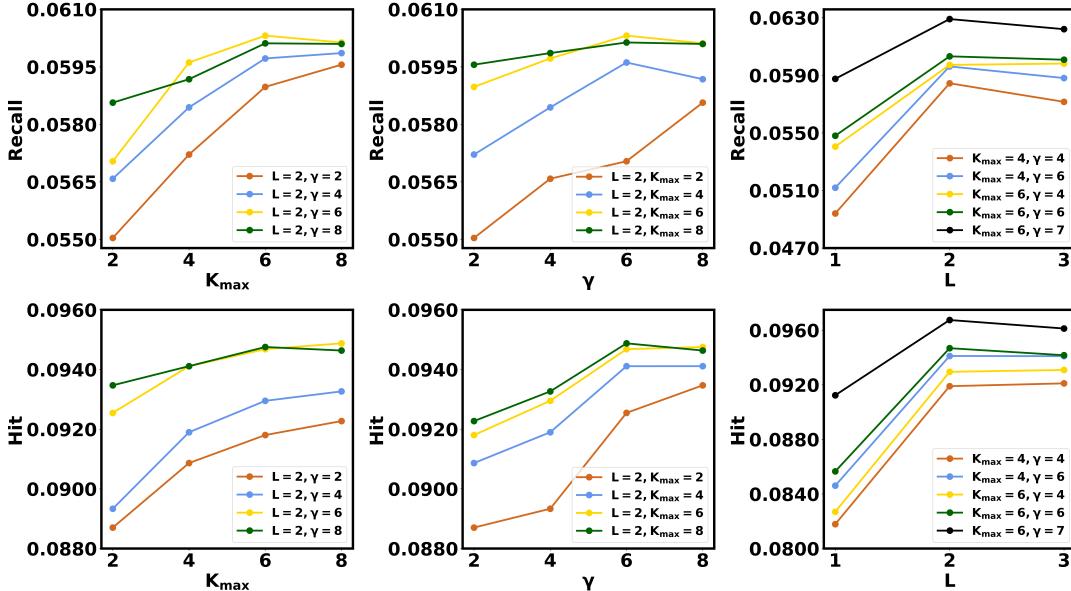


Figure 6: Effect of hyperparameters. Best viewed in color.

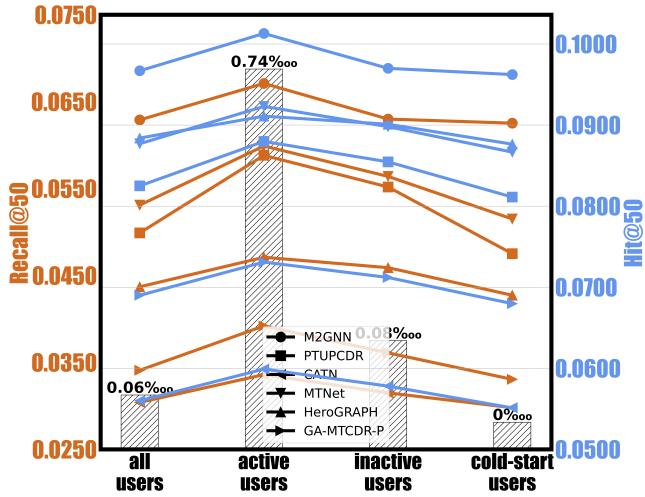


Figure 7: Performance comparison over the sparsity distribution of user groups. The background histograms indicate the density of each user group and %oo means permyriad.

per user, which has the same criterion in Section 1. Metrics are calculated for each group individually. EMCDR and HCDIR have similar performances with PTUPCDR and GA-MTCDR-P respectively, therefore they are omitted. Figure 7 illustrates the results and we have two conclusions: (1) All models perform better for active users since they have richer interactions in the target domain. (2) There is a marked decrease caused by the data sparsity problem for inactive and cold-start users in baselines, while M2GNN has the least impact. This proves M2GNN has stronger robustness to serve the inactive and cold-start users.

Table 4: Online metrics in a week.

| Metrics                       | CN       | UV-CN    | CR       |
|-------------------------------|----------|----------|----------|
| active users                  | +0.4715% | +0.3749% | +0.4663% |
| inactive and cold-start users | +1.0957% | +1.3447% | +1.1986% |

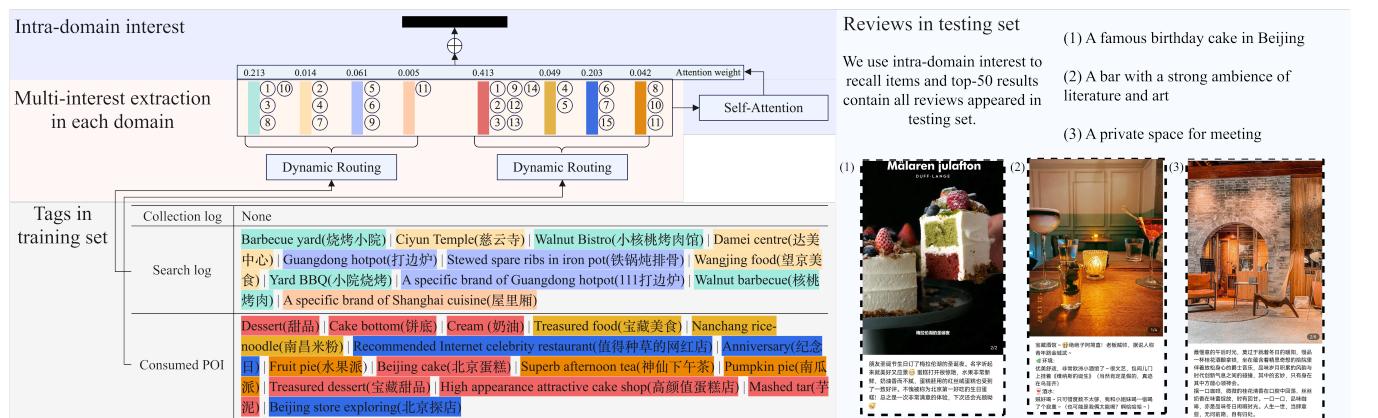
#### 4.5 Online A/B Test (RQ4)

We report a 2-week online A/B test experiment of M2GNN in Beijing, from Dec. 1, 2022, to Dec. 15, 2022. Specifically, the current recommender system on the Dianping homepage<sup>11</sup> is the base model. M2GNN is integrated into it as an additional recall method and directly offers 200 reviews for the later rank stage, which is regarded as the tested model. 3.4 million users participated in this A/B test. Three kinds of metrics are chosen. ‘Collection number’ (CN) indicates the number of all collected reviews, which is an absolute amount. ‘Collection number per user’ (UV-CN) means the number of collected reviews per user, which eliminates users with no collection behaviors. ‘Collection rate’ (CR) is the ratio of collected reviews to all exposed reviews. We just give the increase rate since the true value of each metric is sensitive. As in Table 4, M2GNN achieves great improvements relative to the base model. What’s more, the improvement of inactive and cold-start users (1.1986% CR) is more significant than that of active users (0.4663% CR), which demonstrates the superiority of M2GNN in alleviating data sparsity problems by transferring cross-domain knowledge.

#### 4.6 Case Study (RQ5)

To prove the interpretability of M2GNN, we visualize the two-step aggregation process to show how to distill intra-domain interests from tags, most of which are noises. Figure 8 takes a specific user as an example. The bottom lists all tags from three domains, and obviously he is a cold-start user because of no interactions in the

<sup>11</sup>This is not disclosed for confidentiality reasons.



**Figure 8: Case study. Each tag is represented by a color block. The original Chinese text is in brackets and we translate it into English. Best viewed in color.**

target domain. In the pink block, there are four interest capsules in each domain according to equation 8. The number  $\otimes$  near to a capsule  $z$  represents that  $x$ -th tag has the largest routing logit towards  $z$  among four capsules, which is also reflected by highlighting tags with the same color. The results indicate that tags are clustered via their semantics and the interest capsule represents the high-level preference of each domain. For example, in the consumption domain, the first capsule consists of tags about cakes (e.g. *Dessert*, *Cream* and *Beijing cake*). What's more, the influence of trivial tags (e.g. *Mashed tar*) is weakened since a capsule can only represent one aspect of semantics. Then eight weights are calculated by the self-attention module as shown in the blue block. Three capsules, which are concerned with barbecue (0.213), cake (0.413), and popular shop (0.203) respectively, account for the major portion of final intra-domain interest. Finally, the top-100 candidate reviews contain all ground-truth items at the right of the figure, which are highly relevant to the above three interest capsules and confirm the reasonability and interpretability of M2GNN.

## 5 RELATED WORK

Our work is highly related to two subareas of recommendation, which are content-based and GNN-based CDR.

**Content-based CDR.** The underlying assumption of this method is that the interests extracted from descriptive information in the content can be shared in different domains because of the consistency of user preferences. According to the method for modeling text features, there are two kinds of content-based CDR. The first way is directly projecting documents into the common low-dimensional space [4, 5, 9]. RC-DFM [5] learns latent vectors of reviews by applying a max pooling operation over pretrained word embeddings and utilizes the output of hidden layer under SDAE [26] framework to recommend items. The second category of solutions fulfills high-level semantics extraction, also known as aspects [24, 39]. CATN [39] leverages the aspect gate as a soft on-off switch to control which latent feature derived from the text convolution layer is relevant to the aspect, and then performs pair-wise aspect matchings as the final rating prediction. Despite their success in processing text content, solutions in both categories don't specially take into

consideration the corpus where the majority of tokens are noise. To our best knowledge, M2GNN is the first attempt to filter out noisy content and distill intra-domain interests simultaneously.

**GNN-based CDR.** This category of algorithms aims to leverage the high-order connectivity of GNN by transforming domain-specific knowledge into graphs. There are two common graph data topologies: (1) user-item bipartite graph [3, 14, 15, 28]. BiTGCf [15] incorporates graph neural network to a bi-direction dual transfer learning approach, where each domain has a unique user-item bipartite graph and the output of each GNN layer is integrated across domains in the propagation process. (2) heterogeneous graph [1, 37]. HCDIR [1] transforms four types of insurance products with their properties and relations into an insurance graph and utilizes relational neighbor aggregation to learn embeddings in each domain, which is fed into an MLP to transfer knowledge. However, all of the above models fail to incorporate review information into the graph explicitly and just use documents to generate initial representations. To our best knowledge, M2GNN is the first attempt to regard the words (tags) as one type of node in GNN-based CDR.

## 6 CONCLUSION

In this paper, we aim to improve the review feed recommender system in the Dianping app. The proposed M2GNN adopts the GNN-based cross-domain framework to extract content-based interests via tags from other sites like search and consumption logs to enhance target recommendation. A key contribution is a novel two-step aggregation mechanism, which can distill intra-domain interests from a large number of tags, most of which are noises. Both offline and online experiments demonstrate the effectiveness and explainability of M2GNN. For future work, we plan to investigate how to filter out useless tags via node sampling technology in GNN.

## REFERENCES

- [1] Ye Bi, Liqiang Song, Mengqiu Yao, Zhenyu Wu, Jianming Wang, and Jing Xiao. 2020. A heterogeneous information network based cross domain insurance recommendation system for cold start users. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2211–2220.

1004  
1005  
1006  
1007  
1008  
1009  
1010  
1011  
1012  
1013  
1014  
1015  
1016  
1017  
1018  
1019  
1020  
1021  
1022  
1023  
1024  
1025  
1026  
1027  
1028  
1029  
1030  
1031  
1032  
1033  
1034  
1035  
1036  
1037  
1038  
1039  
1040  
1041  
1042  
1043  
1044  
1045  
1046  
1047  
1048  
1049  
1050  
1051  
1052  
1053  
1054  
1055  
1056  
1057  
1058  
1059  
1060  
1061  
1062

- 1063 [2] Jianxin Chang, Chen Gao, Yu Zheng, Yiqun Hui, Yanan Niu, Yang Song, Depeng  
1064 Jin, and Yong Li. 2021. Sequential Recommendation with Graph Neural Networks.  
1065 In *SIGIR*. 378–387.
- 1066 [3] Qiang Cui, Tao Wei, Yafeng Zhang, and Qing Zhang. 2020. HeroGRAPH: A Het-  
1067 erogeneous Graph Framework for Multi-Target Cross-Domain Recommendation.  
In *ORSUM@ RecSys*.
- 1068 [4] Ali Mamdouh Elkahky, Yang Song, and Xiaodong He. 2015. A multi-view deep  
1069 learning approach for cross domain user modeling in recommendation systems.  
In *Proceedings of the 24th international conference on world wide web*. 278–288.
- 1070 [5] Wenjing Fu, Zhaohui Peng, Senzhang Wang, Yang Xu, and Jin Li. 2019. Deeply  
1071 fusing reviews and contents for cold start users in cross-domain recommendation  
systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33.  
1072 94–101.
- 1073 [6] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng  
1074 Wang. 2020. 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*. 639–648.
- 1075 [7] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng  
1076 Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th international  
conference on world wide web*. 173–182.
- 1077 [8] Fenyu Hu, Yanqiao Zhu, Shu Wu, Weiran Huang, Liang Wang, and Tieniu Tan.  
2021. Graphair: Graph representation learning with neighborhood aggregation  
and interaction. *Pattern Recognition* 112 (2021), 107745.
- 1078 [9] Guangneng Hu, Yu Zhang, and Qiang Yang. 2018. MTNet: a neural approach for  
cross-domain recommendation with unstructured text. *KDD Deep Learning Day*  
(2018), 1–10.
- 1079 [10] Wenyi Huang and Jack W Stokes. 2016. MtNet: a multi-task neural network  
for dynamic malware classification. In *International conference on detection of  
intrusions and malware, and vulnerability assessment*. Springer, 399–418.
- 1080 [11] SeongKu Kang, Junyoung Hwang, Dongha Lee, and Hwanjo Yu. 2019. Semi-  
1081 supervised learning for cross-domain recommendation to cold-start users. In  
*Proceedings of the 28th ACM International Conference on Information and Knowl-  
edge Management*. 1563–1572.
- 1082 [12] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic opti-  
1083 mization. *arXiv preprint arXiv:1412.6980* (2014).
- 1084 [13] Chao Li, Zhiyuan Liu, Mengmeng Wu, Yuchi Xu, Huan Zhao, Pipei Huang,  
Guoliang Kang, Qiwei Chen, Wei Li, and Dik Lun Lee. 2019. Multi-interest  
1085 network with dynamic routing for recommendation at Tmall. In *KDD*. 2615–  
2623.
- 1086 [14] Jin Li, Zhaohui Peng, Senzhang Wang, Xiaokang Xu, Philip S Yu, and Zhenyun  
1087 Hao. 2020. Heterogeneous Graph Embedding for Cross-Domain Recom-  
1088 mendation Through Adversarial Learning. In *International Conference on Database  
Systems for Advanced Applications*. Springer, 507–522.
- 1089 [15] Meng Liu, Jianjun Li, Guohui Li, and Peng Pan. 2020. Cross domain recom-  
1090 mendation via bi-directional transfer graph collaborative filtering networks. In  
*Proceedings of the 29th ACM International Conference on Information & Knowledge  
Management*. 885–894.
- 1091 [16] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective  
1092 approaches to attention-based neural machine translation. *EMNLP* (2015).
- 1093 [17] Tong Man, Huawei Shen, Xiaolong Jin, and Xueqi Cheng. 2017. Cross-domain  
1094 recommendation: An embedding and mapping approach. In *IJCAI*, Vol. 17. 2464–  
2470.
- 1095 [18] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013.  
Distributed representations of words and phrases and their compositionality.  
1096 *Advances in neural information processing systems* 26 (2013).
- 1097 [19] Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings  
1098 using siamese bert-networks. *arXiv preprint arXiv:1908.10084* (2019).
- 1099 [20] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-  
1100 Thieme. 2012. BPR: Bayesian personalized ranking from implicit feedback. *arXiv  
preprint arXiv:1205.2618* (2012).
- 1101 [21] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. 2017. Dynamic routing  
1102 between capsules. *Advances in neural information processing systems* 30 (2017).
- 1103 [22] Sungyong Seo, Jing Huang, Hao Yang, and Yan Liu. 2017. Interpretable con-  
1104 volutional neural networks with dual local and global attention for review rating  
1105 prediction. In *Proceedings of the eleventh ACM conference on recommender systems*.  
297–305.
- 1106 [23] Ajit P Singh and Geoffrey J Gordon. 2008. Relational learning via collective matrix  
1107 factorization. In *Proceedings of the 14th ACM SIGKDD international conference on  
Knowledge discovery and data mining*. 650–658.
- 1108 [24] Tianhang Song, Zhaohui Peng, Senzhang Wang, Wenjing Fu, Xiaoguang Hong,  
1109 and Philip S Yu. 2017. Based cross-domain recommendation through joint  
1110 tensor factorization. In *International conference on database systems for advanced  
applications*. Springer, 525–540.
- 1111 [25] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones,  
Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all  
1112 you need. *Advances in neural information processing systems* 30 (2017).
- 1113 [26] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, Pierre-Antoine  
1114 Manzagol, and Léon Bottou. 2010. Stacked denoising autoencoders: Learning  
1115 useful representations in a deep network with a local denoising criterion. *Journal  
of machine learning research* 11, 12 (2010).
- 1116 [27] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019. Kgat:  
1117 Knowledge graph attention network for recommendation. In *Proceedings of the  
25th ACM SIGKDD International Conference on Knowledge Discovery & Data  
Mining*. 950–958.
- 1118 [28] Xiang Wang, Xiangnan He, Liqiang Nie, and Tat-Seng Chua. 2017. Item silk road:  
1119 Recommending items from information domains to social users. In *Proceedings  
of the 40th International ACM SIGIR conference on Research and Development in  
Information Retrieval*. 185–194.
- 1120 [29] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019.  
1121 Neural graph collaborative filtering. In *Proceedings of the 42nd international ACM  
SIGIR conference on Research and development in Information Retrieval*. 165–174.
- 1122 [30] Xiang Wang, Tinglin Huang, Dingxian Wang, Yancheng Yuan, Zhenguang Liu,  
1123 Xiangnan He, and Tat-Seng Chua. 2021. Learning Intents behind Interactions with  
1124 Knowledge Graph for Recommendation. In *Proceedings of the Web Conference  
2021*. 878–887.
- 1125 [31] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S  
1126 Yu. 2019. Heterogeneous graph attention network. In *WWW*. 2022–2032.
- 1127 [32] Libing Wu, Cong Quan, Chenliang Li, Qian Wang, Bolong Zheng, and Xiangyang  
1128 Luo. 2019. A context-aware user-item representation learning for item recom-  
1129 mendation. *ACM Transactions on Information Systems (TOIS)* 37, 2 (2019), 1–29.
- 1130 [33] Haifeng Xia, Zengmao Wang, Bo Du, Lefei Zhang, Shuai Chen, and Gang Chun.  
2019. Leveraging ratings and reviews with gating mechanism for recommenda-  
1131 tion. In *Proceedings of the 28th ACM International Conference on Information and  
Knowledge Management*. 1573–1582.
- 1132 [34] Chengfeng Xu, Pengpeng Zhao, Yanchi Liu, Victor S Sheng, Jiajie Xu, Fuzhen  
1133 Zhuang, Junhua Fang, and Xiaofang Zhou. 2019. Graph Contextualized Self-  
Attention Network for Session-based Recommendation.. In *IJCAI*, Vol. 19. 3940–  
3946.
- 1134 [35] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan  
1135 Salakhutdinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural  
image caption generation with visual attention. In *International conference on  
machine learning*. PMLR, 2048–2057.
- 1136 [36] Huan Yan, Xiangning Chen, Chen Gao, Yong Li, and Depeng Jin. 2019. Deepapf:  
1137 Deep attentive probabilistic factorization for multi-site video recommendation.  
*IJCAI 2019* (2019).
- 1138 [37] Jiangyi Yin, Yuchun Guo, and Yishuai Chen. 2019. Heterogenous Information  
1139 Network Embedding Based Cross-Domain Recommendation System. In *2019  
International Conference on Data Mining Workshops (ICDMW)*. IEEE, 362–369.
- 1140 [38] Cheng Zhao, Chenliang Li, and Cong Fu. 2019. Cross-domain recommendation  
via preference propagation graphnet. In *Proceedings of the 28th ACM International  
Conference on Information and Knowledge Management*. 2165–2168.
- 1141 [39] Cheng Zhao, Chenliang Li, Rong Xiao, Hongbo Deng, and Aixin Sun. 2020. CATN:  
1142 Cross-domain recommendation for cold-start users via aspect transfer network.  
In *Proceedings of the 43rd International ACM SIGIR Conference on Research and  
Development in Information Retrieval*. 229–238.
- 1143 [40] Lei Zheng, Vahid Noroozi, and Philip S Yu. 2017. Joint deep modeling of users  
1144 and items using reviews for recommendation. In *Proceedings of the tenth ACM  
international conference on web search and data mining*. 425–434.
- 1145 [41] Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu,  
Lifeng Wang, Changcheng Li, and Maosong Sun. 2020. Graph neural networks:  
1146 A review of methods and applications. *AI Open* 1 (2020), 57–81.
- 1147 [42] Feng Zhu, Yan Wang, Jun Zhou, Chaochao Chen, Longfei Li, and Guanfeng Liu.  
2021. A unified framework for cross-domain and cross-system recommendations.  
*IEEE Transactions on Knowledge and Data Engineering* (2021).
- 1148 [43] Yongchun Zhu, Zhenwei Tang, Yudan Liu, Fuzhen Zhuang, Ruobing Xie, Xu  
Zhang, Leyu Lin, and Qing He. 2022. Personalized transfer of user preferences for  
1149 cross-domain recommendation. In *Proceedings of the Fifteenth ACM International  
Conference on Web Search and Data Mining*. 1507–1515.
- 1150 [44] [44] [44]
- 1151 [45] [45] [45]
- 1152 [46] [46] [46]
- 1153 [47] [47] [47]
- 1154 [48] [48] [48]
- 1155 [49] [49] [49]
- 1156 [50] [50] [50]
- 1157 [51] [51] [51]
- 1158 [52] [52] [52]
- 1159 [53] [53] [53]
- 1160 [54] [54] [54]
- 1161 [55] [55] [55]
- 1162 [56] [56] [56]
- 1163 [57] [57] [57]
- 1164 [58] [58] [58]
- 1165 [59] [59] [59]
- 1166 [60] [60] [60]
- 1167 [61] [61] [61]
- 1168 [62] [62] [62]
- 1169 [63] [63] [63]
- 1170 [64] [64] [64]
- 1171 [65] [65] [65]
- 1172 [66] [66] [66]
- 1173 [67] [67] [67]
- 1174 [68] [68] [68]
- 1175 [69] [69] [69]
- 1176 [70] [70] [70]
- 1177 [71] [71] [71]
- 1178 [72] [72] [72]
- 1179 [73] [73] [73]