

# Dimension Independent Mixup for Hard Negative Sample in Collaborative Filtering

Xi Wu  
YanShan University  
Qinhuangdao, China  
wuxi@stumail.yzu.edu.cn

Chao Zhou  
YanShan University  
Qinhuangdao, China  
chaozhou1999@gmail.com

Liangwei Yang  
University of Illinois Chicago  
Chicago, USA  
lyang84@uic.edu

Tianyu Lin  
YanShan University  
Qinhuangdao, China  
1163265504@qq.com

Jibing Gong  
YanShan University  
Qinhuangdao, China  
gongjibing@ysu.edu.cn

Xiaolong Liu  
University of Illinois Chicago  
Chicago, USA  
xliu262@uic.edu

Philip S. Yu  
University of Illinois Chicago  
Chicago, USA  
psyu@uic.edu

## ABSTRACT

Collaborative filtering (CF) is a widely employed technique that predicts user preferences based on past interactions. Negative sampling plays a vital role in training CF-based models with implicit feedback. In this paper, we propose a novel perspective based on the sampling area to revisit existing sampling methods. We point out that current sampling methods mainly focus on Point-wise or Line-wise sampling, lacking flexibility and leaving a significant portion of the hard sampling area un-explored. To address this limitation, we propose Dimension Independent Mixup for Hard Negative Sampling (DINS), which is the first Area-wise sampling method for training CF-based models. DINS comprises three modules: Hard Boundary Definition, Dimension Independent Mixup, and Multi-hop Pooling. Experiments with real-world datasets on both matrix factorization and graph-based models demonstrate that DINS outperforms other negative sampling methods, establishing its effectiveness and superiority. Our work contributes a new perspective, introduces Area-wise sampling, and presents DINS as a novel approach that achieves state-of-the-art performance for negative sampling. Our implementations are available in PyTorch<sup>1</sup>.

## CCS CONCEPTS

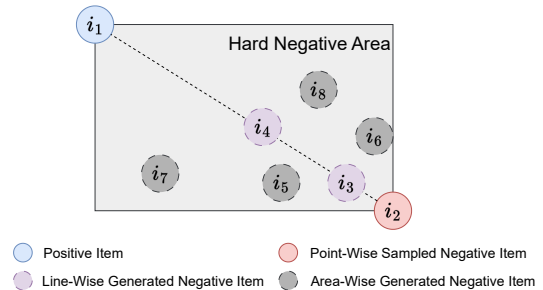
• **Information systems** → **Recommender systems; Collaborative filtering.**

## KEYWORDS

Recommendation System, Negative Sample, Mixup

## 1 INTRODUCTION

In the contemporary era of voluminous data [15], individuals are inundated with an incessant influx of content generated by the internet. To address the issue of information overload, Recommender Systems (RecSys) are employed to assist users in locating the most relevant information and are increasingly pivotal in online services



**Figure 1: Illustration of items sampled by Point/Line/Area-wise sampling methods.**

such as news feed [27], music suggestion [5], and online shopping [9]. Collaborative filtering (CF) [13], a highly effective method that predicts a user’s preference based on their past interactions, is widely employed. The latest CF-based models [10, 25] incorporate historical interactions into condensed user/item vectors and predict a user’s preference for each item based on the dot product of the corresponding vectors. These models have garnered significant research attention and have been demonstrated to be effective in a range of application contexts [30, 33].

Negative sampling [2, 35] is a key technique when training CF-based models with implicit feedback [18], which is inferred from user behavior, such as clicks, views, and purchases, rather than being explicitly provided by the user. Since implicit feedback is prevalent in most online platforms, it is frequently utilized in training RecSys [10, 11, 33]. These behaviors only signify a user’s positive feedback, necessitating the integration of a negative sampling module to provide negative feedback. The training process of CF-based models involves the differentiation between positive and negative examples, enhancing its ability to suggest items of interest to the user. The negative sampling approach has a significant impact on the ultimate performance of CF-based models [6, 11, 20, 23, 35].

<sup>1</sup><https://anonymous.4open.science/r/DINS-1EE3>

For each observed user-item interaction, the negative sampling module samples one or multiple negative items [3]. By introducing the concept of sampling area analysis, we offer a fresh perspective for understanding and categorizing these methods. In accordance with the proposed framework, this paper explores the sampling of negative items in relation to observed user-item interactions. As shown in Figure 1,  $i_1$  is the observed positive item. Notably, the sampled negative items, denoted as  $i_2$  to  $i_8$ , are obtained through diverse methodologies, which can be further categorized into Point/Line/Area-wise negative sampling methods.

- (1) The **Point-wise** sampling method ( $i_2$ ) selects a particular item from the candidate item set using a predetermined sample distribution. In terms of embedding space interpretation, these methods involve selecting a single point from a predefined set of candidate points, each representing a potential candidate item. This category encompasses a majority of existing methods [7, 16, 19, 23, 32, 35], as they employ sampling techniques within the discrete space.
- (2) The **Line-wise** sampling method ( $i_3, i_4$ ) involves selecting a pseudo-negative item positioned along a line within the embedding space. A notable representative work in this category is MixGCF [11], which incorporates the *mixup* technique [34]. MixGCF employs a mixing coefficient sampled from the beta distribution to blend the positive item and the sampled negative item. This blending process creates a linear interpolation between the positive and negative items, generating a pseudo-negative item located precisely on the line connecting the positive and negative instances. By acquiring a challenging negative item, the model gains improved discriminative capabilities between positive and negative items.
- (3) This paper introduces the novel **Area-wise** sampling approach ( $i_5-i_8$ ), which involves selecting a pseudo-negative item that resides within a specific area within the embedding space. Figure 1 serves as an illustrative example. The hard negative area, depicted as a grey square, represents the region bounded by the dimensions of a positive item and a point-wise sampled negative item. When compared to Point-wise and Line-wise sampling methods, the Area-wise sampling technique offers a more extensive exploration space and greater flexibility. It allows for sampling from the hard negative area, which provides increased capacity for differentiating between positive and negative instances in multiple dimensions, thereby offering varying degrees of hardness.

The proposed Area-wise sampling method faces several significant challenges that must be addressed. Firstly, the definition of boundaries for the hard negative area is crucial to the effectiveness of this approach. The sampling area plays a pivotal role, as an excessively large area may fail to provide informative negative items, while an overly small area could result in false negative item generation [7]. The second challenge involves enabling Area-wise sampling within the determined hard negative area. Previous methods have primarily focused on sampling from discrete spaces or utilizing the *mixup* technique to generate Line-wise pseudo-negative items. The ability to sample within a continuous area represents a novel research question in this context. Lastly, given the flexibility introduced by sampling within an area, an additional

challenge is effectively regularizing the sampling method to generate informative pseudo-negative items. Developing appropriate regularization techniques is necessary to ensure the quality and relevance of the generated negative samples.

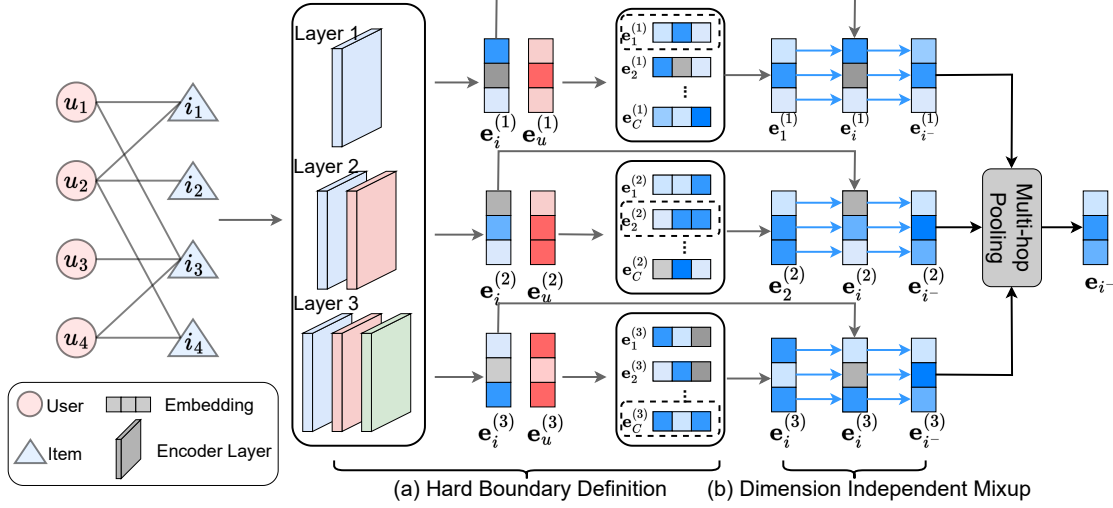
This paper presents Dimension Independent Negative Sampling (DINS) as a solution to facilitate Area-wise sampling in the context of collaborative filtering. DINS comprises three distinct modules, each designed to address the aforementioned challenges in a targeted manner. Firstly, the proposed **Hard Boundary Definition** module within DINS determines the appropriate boundaries for the hard negative sampling area. It accomplishes this by selecting the item with the highest dot-product similarity to the positive item, thus establishing a boundary that closely aligns with the positive instance. Following the establishment of boundaries, DINS introduces a novel **Dimension Independent Mixup** module, enabling the sampling of items from the corresponding area. This module employs distinct linear interpolation weights for different dimensions, thereby extending the Line-wise sampling approach to support Area-wise sampling. Lastly, DINS proposes the **Multi-hop Pooling** module to regularize the sampling process and generate informative pseudo-negative items based on multiple hops of neighborhood information. By leveraging this module, DINS achieves effective regularization, thereby enhancing the quality and relevance of the sampled negative instances. By integrating these three modules, DINS samples from the hard negative area successfully, and yield superior performance compared to other negative sampling methods when applied to various backbone models. The contributions of this work can be summarized as follows.

- **Novel Perspective:** We offer a fresh perspective from the embedding space to comprehend and analyze existing negative sampling methods, providing insights into their mechanisms.
- **Area-wise Sampling:** We are the first to introduce Area-wise negative sampling, recognizing the challenges, and presenting corresponding solutions.
- **DINS:** We propose DINS as the first Area-wise negative sampling method, enabling highly flexible sampling over an area. This novel approach surpasses existing methods and achieves state-of-the-art performance in collaborative filtering tasks.
- **Experimental Validation:** We conduct extensive experiments on three real-world datasets, employing different backbone models. The results demonstrate the effectiveness and superiority of DINS, confirming its performance enhancement compared to other negative sampling methods.

The remaining paper is organized as follows. Section 2 represents the preliminaries of this paper. Section 3 illustrates different parts of DINS in detail. Section 4 conducts experiments comparing other methods and further experiments to show the effectiveness of DINS. Section 5 represents the most related works for reference, and we conclude DINS and discuss future research directions in Section 6.

## 2 PRELIMINARIES

In this section, we first formulate the collaborative filtering (CF) problem, and then illustrate the negative sampling problem in CF-based model training.



**Figure 2: Main framework of DINS.** For each observed interaction  $(u, i)$ , DINS first encodes  $u$ 's and  $i$ 's information within the graph with different numbers of GNN layers. For the output of each layer, (a) DINS defines the sampling area by the Hard Boundary Definition module and then (b) mixes the sampled item with the corresponding positive item via the proposed Dimension Independent Mixup module. Finally, (c) DINS generates the synthetic negative item by integrating the negative signal from different hops of neighborhoods with a Multi-hop Pooling module.

## 2.1 Collaborative Filtering

Collaborative filtering aims to predict user's preferences based on users' historical interactions. It has been shown as an effective and powerful tool [1, 14] for RecSys. With implicit feedback, we have a set of users  $\mathcal{U} = \{u_1, u_2, \dots, u_{|\mathcal{U}|}\}$ , a set of items  $\mathcal{I} = \{i_1, i_2, \dots, i_{|\mathcal{I}|}\}$ , and the observed user-item interactions  $\mathbf{R} \in \mathbb{R}^{|\mathcal{U}| \times |\mathcal{I}|}$ , where  $R_{u,i} = 1$  if user  $u$  has interacted with item  $i$ , or  $R_{u,i} = 0$  otherwise. Learning from historical interactions, current advanced CF-based methods [10, 25] learn an encoder function  $f(\cdot)$  to map each user/item into a dense vector embedding, i.e.,  $f(u), f(i) \in \mathbb{R}^d$ , where  $d$  is the dimension of vector embedding. The predicted score from  $u$  to  $i$  is then calculated as the similarity of two vectors (e.g., dot product similarity,  $R_{u,i} = f(u)^\top f(i)$ ). Then these methods rank all items based on the prediction score and select the top  $k$  items  $\{i_1, i_2, \dots, i_k\}$  as recommendation results for each user.

## 2.2 Graph Neural Network for Recommendation

In real-life applications, users can only interact with a limited number of items, which leads to the data sparsity problem [36]. To alleviate the problem, the most advanced CF-based models [10, 28] explicitly utilize high-order connections by representing the historical interactions as a user-item bipartite graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ , where  $\mathcal{V} = \mathcal{U} \cup \mathcal{I}$  and there is an edge  $(u, i) \in \mathcal{E}$  between  $u$  and  $i$  if  $R_{u,i} = 1$ . By utilizing graph neural network (GNN) as the encoder function  $f(\cdot)$ , these methods learn the representations of user/item embeddings by aggregating information from their neighbors, so that connected nodes in the graph structure tend to have similar embeddings. The operation of a general GNN computation can be

expressed as follows:

$$\mathbf{e}_u^{(l+1)} = \mathbf{e}_u^{(l)} \oplus \text{AGG}^{(l+1)}(\{\mathbf{e}_i^{(l)} \mid i \in \mathcal{N}_u\}), \quad (1)$$

where  $\mathbf{e}_u^{(l)} \in \mathbb{R}^d$  is  $u$ 's embedding on the  $l$ -th layer,  $\mathcal{N}_u$  is the neighborhoods of  $u$ ,  $\text{AGG}^{(l)}(\cdot)$  is a function that aggregates neighbors' embeddings into a single vector for layer  $l$ , and  $\oplus$  combines  $u$ 's embedding with its neighbor's information.  $\text{AGG}(\cdot)$  and  $\oplus$  can be simple or complicated functions. Item is calculated in the same way.

After stacking  $L$  layer GNN convolution, we can obtain  $L$  user/item embedding from different GNN layers. Take the user as an example; we will obtain  $(\mathbf{e}_u^{(1)}, \mathbf{e}_u^{(2)}, \dots, \mathbf{e}_u^{(L)})$  for each user after the aggregation.  $\mathbf{e}_u^{(l)}$  encodes the information within  $l$ -hop neighborhoods, which provides a unique user representation with local influence scope. A pooling function (e.g., attention, mean, sum) is used to combine them together  $\mathbf{e}_u = \text{Pool}(\mathbf{e}_u^{(1)}, \mathbf{e}_u^{(2)}, \dots, \mathbf{e}_u^{(L)})$ .

## 2.3 The Negative Sampling Problem

The observed implicit feedback in RecSys only indicates the user's positive interest. Training with only positive labels would cause the model degradation without the ability to distinguish different items. Thus, the training of CF-based models involves the negative sampling procedure to provide the negative signal. The corresponding training method trains the model to give higher scores to observed interactions while lower scores to negative sampled interactions. The most renowned is pair-wise BPR loss [19]:

$$\mathcal{L}_{BPR} = \frac{1}{|\mathcal{E}|} \sum_{(u,i) \in \mathcal{E}} -\log(\text{sigmoid}(s(u, i) - s(u, i^-))), \quad (2)$$

where  $s(u, i)$  is the predicted rating score from  $u$  to  $i$ , and  $i^-$  is the negative sampled item. The quality of sampled  $i^-$  critically impacts the model performance. Hard negative samples can effectively

assist the model in learning better boundaries between positive and negative items [33]. In terms of sampling methods, discrete sampling [19, 23, 33, 35] can be considered as point-wise sampling, which selects negative items from a discrete sample space. Negative sampling based on mixup [11, 20], on the other hand, can be classified as line-wise sampling because this method performs a linear interpolation between two items to obtain a new negative sample along the diagonal line. Both point-wise sampling and line-wise sampling, in fact, do not adequately approximate the positives in the sample space and, therefore, cannot train a more powerful encoder. In this paper, we propose the first area-wise sampling method that greatly extends the sampling area.

### 3 PROPOSED METHOD

This section illustrates the proposed DINS that enables area-wise negative sampling. As Figure 2 demonstrates, DINS mainly consists of three parts, *i.e.*, Hard Boundary Definition, Dimension Independent Mixup, and Multi-hop Pooling.

#### 3.1 Hard Boundary Definition

The first question to answer in the proposed area-wise negative sampling is how to define the continuous sampling area. The hard boundary definition module is shown in Figure 2(a). It samples a boundary item to define the sampling area. For each interaction  $(u, i)$ , DINS defines the area by sampling a boundary item  $i_*$ . In each sampling, DINS pre-samples a candidate set  $C = (i_1, i_2, \dots, i_{|C|})$  to reduce the computation workload as previous researches [7, 35]. Then DINS selects the item with the highest dot-product with  $u$  to define the boundary:

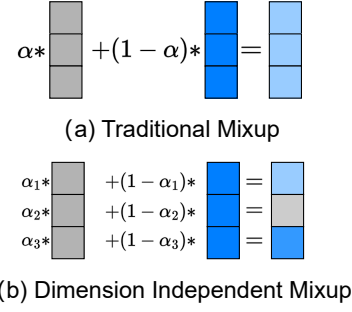
$$i_* = \arg \max_i (\mathbf{e}_u^\top \mathbf{e}_{i_1}, \mathbf{e}_u^\top \mathbf{e}_{i_2}, \dots, \mathbf{e}_u^\top \mathbf{e}_{i_{|C|}}). \quad (3)$$

The boundary item  $i_*$  defines the hard negative area together with the corresponding positive item  $i$  in the embedding space. As shown in Figure 1, the hard negative area is defined as the space between  $i$  and  $i_*$ . The majority of previous research [7, 19, 35] samples discrete existing items. While MixGCF [11] allows the sampling on continuous space along the diagonal line from  $i$  to  $i_*$ , it still leaves the large-volume hard negative area un-explored.

#### 3.2 Dimension Independent Mixup

The second question to answer is how to explore the non-diagonal space within the hard negative area and extend the line-wise to area-wise sampling. To achieve this, we propose the novel Dimension Independent Mixup method in Figure 2(b). It enables a dimension independent mix between the boundary item  $i_*$  and positive item  $i$ .

A further comparison to the traditional Mixup is shown in Figure 3. Line-wise sampling methods [11, 20] utilize the traditional Mixup to generate a negative item. As shown in Figure 3(a), traditional Mixup synthesizes the negative item via linear interpolation of the positive and boundary items with a unified weight  $\alpha$  on all dimensions. With all dimensions having the same linear interpolation, it can only generate a synthetic item falling on the line between the two mixed items in the continuous embedding space. Figure 3(b) shows the proposed Dimension Independent Mixup, which is the core idea to support area-wise negative samples. It mixes the two items dimension independently by calculating specific interpolation



**Figure 3: (a) Traditional Mixup assigns the same interpolation weights on all dimensions. (b) The proposed Dimension Independent Mixup assigns different interpolation weights on different dimensions.**

weights for different dimensions. It is more flexible and increases the exploration space from a line to the whole hard negative area.

For each interaction  $(u, i)$ , Section 3.1 samples a boundary item  $i_*$  to define the sampling area. The mixup for the  $d$ -th dimension is calculated as:

$$e_{i_-}^d = \alpha_d * e_{i_*}^d + (1 - \alpha_d) * e_i^d, \quad (4)$$

where  $e_i^d$  is the  $d$ -th dimension value of  $\mathbf{e}_i$ .  $\alpha_d$  is the interpolation weight for the  $d$ -th dimension, which is calculated as :

$$\alpha_d = \frac{\exp(e_u^d * e_{i_*}^d)}{\beta * \exp(e_u^d * e_{i_*}^d) + \exp(e_u^d * e_i^d)}, \quad (5)$$

where  $e_u^d$  is the  $d$ -th dimension value of  $\mathbf{e}_u$ , and  $\beta$  is a hyper-parameter to tune the relative weight for the mixup. A larger  $\beta$  leads to a smaller  $\alpha_d$ , and  $e_{i_*}^d$  is more similar with positive item. We concatenate the dimensions together to obtain the mixed item:

$$\mathbf{e}_{i_-} = \text{Concat}(e_{i_-}^1, e_{i_-}^2, \dots, e_{i_-}^D) \quad (6)$$

$\mathbf{e}_{i_-}$  is the generated negative item embedding for the interaction  $(u, i)$ . It can be directly used to calculate the ranking loss in Equation 2. Calculating the mixup weight independently for each dimension extends the exploration area from a line to the whole hard negative area. It greatly improves the final recommendation performance by providing more varied items, as illustrated in Section 4.

#### 3.3 Multi-hop Pooling

We further extend DINS to better support graph neural network (GNN) based collaborative filtering methods [10, 29]. These methods are shown to be effective by explicitly considering the high-order connections and encode user/item high-order neighborhood information into dense vectors. The output of different GNN layers encodes neighborhoods within different hops. Regarding the negative sample, MixGCF [11] also proves to be effective when considering different hops of neighborhoods. To utilize the high-order information, DINS samples negative items based on each dense vector extracted by different GNN layers and utilizes a pooling module to obtain the final negative item embedding.

When we utilize GNN as the encoder model  $f(\cdot)$ , we can obtain one dense embedding from each GNN layer to encode the corresponding neighborhoods' information. After  $L$  layers GNN

encoding on the user-item bipartite graph  $\mathcal{G}$ , we obtain  $L$  dense representations for each user  $u$  and item  $i$ :

$$\begin{aligned} f(u) &= (\mathbf{e}_u^{(1)}, \mathbf{e}_u^{(2)}, \dots, \mathbf{e}_u^{(L)}) \\ f(i) &= (\mathbf{e}_i^{(1)}, \mathbf{e}_i^{(2)}, \dots, \mathbf{e}_i^{(L)}). \end{aligned} \quad (7)$$

Following Section 3.1 and Section 3.2, we synthesize a negative item embedding  $\mathbf{e}_{i^-}^{(l)}$  based on  $\mathbf{e}_u^{(l)}$  and  $\mathbf{e}_i^{(l)}$ . Then we obtain the final negative sample via a Pooling function:

$$\mathbf{e}_{i^-} = \text{Pool}(\mathbf{e}_{i^-}^{(1)}, \mathbf{e}_{i^-}^{(2)}, \dots, \mathbf{e}_{i^-}^{(L)}), \quad (8)$$

where the Pool can be any function pools multiple tensors into a single tensor. For simplicity, we test mean pooling and concatenation function for all datasets.

### 3.4 Complexity Analysis

The training process of DINS is illustrated step by step in Algorithm 1. The time complexity for each sampling process comes from the three proposed modules. Hard Boundary Definition module takes  $O(MD)$ , where  $M$  is the candidate set budget and  $D$  is the dimension size. Dimension Independent Mixup module takes  $O(D)$ . Thus, the time complexity without considering multi-hop neighbors is  $O(MD)$ . When considering the multi-hop neighbors, DINS needs an extra sampling procedure for each GNN layer. The total time complexity is then  $O(LMD)$ , where  $L$  is the number of GNN layers. To be noted that, the time complexity is the same as MixGCF, but DINS extends to a much larger exploration space.

---

#### Algorithm 1 The training process with DINS

---

**Require:** Training set  $\mathcal{G}$ , Recommendation user/item encoder  $f(\cdot)$ , a candidate set budget  $M$ , a parameter  $\beta$  to control positive mixing

- 1: **for**  $t = 1, 2, 3, \dots$ , to  $T$  **do**
  - 2:   Sample a mini-batch of user-item positive pairs  $\{(u, i)\}$ .
  - 3:   Initialize loss  $\mathcal{L}_{BPR} = 0$ .
  - 4:   **for** each  $(u, i)$  pair **do**
  - 5:     Get the user and positive item embeddings by encoder  $f(\cdot)$ .
  - 6:     Get ID embedding of uniformly sampled  $M$  negatives as the candidate set.
  - 7:     Get the boundary negative item  $i_*$  by (3).
  - 8:     Calculate the independent dimensional weight matrix of boundary item  $i_*$  by (5).
  - 9:     Synthesize a hard negative item  $e_{i^-}$  by (4) and (6).
  - 10:    **if**  $f(\cdot)$  is GNN-based model **then**
  - 11:     Further synthesize hard negative item for output of different GNN layers
  - 12:     Pooling the synthesized negative items by (8)
  - 13:    **end if**
  - 14:    Calculate  $\mathcal{L}_{BPR}$  by (8).
  - 15:    **end for**
  - 16:    Update  $\theta$  by descending the gradients  $\nabla_{\theta} \mathcal{L}_{BPR}$ .
  - 17: **end for**
- 

**Table 1: Statistics of the datasets.**

Dataset	#User	#Items	#Interactions	Density
Alibaba	106,042	53,591	907,407	0.016%
Yelp2018	31,668	38,048	1,561,406	0.13%
Amazon	192,403	63,001	1,689,188	0.014%

## 4 EXPERIMENTS

This section empirically evaluates the proposed DINS on three real-world datasets with three different backbones. The goal is to answer the four following research questions (RQs).

- **RQ1:** Can DINS provide informative negative samples to improve the performance of recommendation?
- **RQ2:** Does every module contributes to the effectiveness?
- **RQ3:** What is the impact of different hyper-parameters on DINS?
- **RQ4:** Is DINS really supporting area-wise negative sampling?

### 4.1 Experimental Setup

**4.1.1 Datasets.** For a fair comparison, we also evaluate DINS on three benchmark datasets: Alibaba [11], Yelp2018 [25], and Amazon [11] following previous research [11]. The detailed Statistics of three public datasets are shown in Table 1, which exhibits the variation in scale and sparsity. Yelp2018 demonstrates the highest level of density among these three datasets under consideration, while the other two datasets exhibit relatively lower levels of density.

**4.1.2 Baselines.** To validate the effectiveness of DINS, we chose three commonly used backbone networks, LightGCN [10] and NGCF [25] as GNN-based encoders and MF [19] as a non-GNN-based encoder. Additionally, we selected five baseline negative sampling methods for comparison.

- **Popularity:** It samples negative items by assigning a higher sampling probability of more popular items.
- **RNS [19]:** Random negative sampling (RNS) strategy is the most common and widely used approach, which applies a uniform distribution to sample an item that the user has never interacted with as a negative item.
- **DNS [35]:** Dynamic negative sampling (DNS) strategy adaptively selects the highest-scoring negative item by the current recommendation model among randomly selected items. Such a negative item is considered a hard negative item for training.
- **IRGAN [23]:** It is a GAN-based strategy for generating negative sampling distribution.
- **MixGCF [11]:** MixGCF is the state-of-the-art sampling method based on Mixup, which applies positive mixing and hop mixing to synthesize new negative items.

**4.1.3 Evaluation Method.** We use Recall@K (R@K) and NDCG@K (N@K) to evaluate the performance of the top-K recommendation of our model, both of which are widely used in the recommendation system. By default, the value of K is set as 20. We present the average metrics for all users in the test set, calculating these metrics based on the rankings of non-interacted items. In accordance with prior research [10, 25], we employ the complete full-ranking technique, which involves ranking all items that have not yet been interacted with by the user.

**4.1.4 Experimental Settings.** We implement our DINS and other baseline models based on Pytorch 2.0 and Python 3.8. Both the user embedding size and the item embedding size are fixed to 64 for backbone recommendation models (NGCF, LightGCN and MF). To achieve better performance, we use Xavier[8] to initialize embedding parameters for all encoders, and the number of aggregation layers in NGCF or LightGCN is set as 3 by default. Adam [12] is used to optimize all encoders. We set the batch size as 2048 for LightGCN and MF, while 1024 for NGCF. For the remaining hyper-parameters, we used the grid search technique to find the optimal settings for each recommender: the learning rate is searched in  $\{0.0001, 0.001, 0.01\}$ , coefficient of weight decay is tuned in  $\{10^4, 10^5, 10^6\}$ . Moreover, the size of the candidate pool for DNS, MixGCF, and our DINS is searched in  $\{8, 16, 32, 64\}$ , and the hyper-parameter  $\beta$ , which is used in our method is searched from 0 to 10, steps of 0.5 each.

## 4.2 RQ1: Performance Evaluation

We report the overall performance of the five baselines on the three backbones in Table 2. We can have the following observations:

- DINS consistently outperforms all baselines by a large margin on three datasets across three classical encoders. Specifically, DINS accomplishes remarkable improvements over the second-best baseline, especially on the Amazon and Alibaba datasets which improved Recall@20 by 23%, and 36.1%, respectively. This further demonstrates the effectiveness of area-wise sampling, as it enhances the performance of not only GNN-based encoders but also non-GNN-based encoders.
- In most cases, the line-wise sampling method (MixGCF) is better than the rest point-wise sampling methods. It shows the advantage of extending exploration space from points to lines.
- DINS exhibits greater improvement on the two datasets of lower density (Alibaba and Amazon), ranging from 8.0% to 36.1%. This highlights DINS's ability to explore the continuous embedding space and effectively enhance performance on sparse datasets.
- MixGCF employs "hop mixing" and "positive mixing" to improve the quality of negative samples. However, MF is not a GNN-based encoder, so "hop mixing" cannot be used; we only use "positive mixing" in MixGCF (marked with \*) under MF. Compared with other baselines, MixGCF without "hop mixing" achieves state-of-the-art results solely on the Amazon dataset when utilizing the MF-based encoder. However, it falls short of surpassing DNS on both the Alibaba and Yelp2018 datasets. On the other hand, our DINS not only achieves state-of-the-art results in the GNN-based model but also excels in the Non-GNN-based model, establishing its superiority across the board.

## 4.3 RQ2: Ablation Study

In this section, we conduct an ablation study of the three modules of DINS on LightGCN by building 3 variants. A is built by removing the Hard Boundary Definition module, where we randomly sample a boundary item from the candidate set. B is built by replacing the dimension independent mixup module with the traditional mixup. C is built by removing the multi-hop pooling module by only considering the negative item for the first layer aggregation. Corresponding experimental results are presented in Figure 4. We can have the following observations:

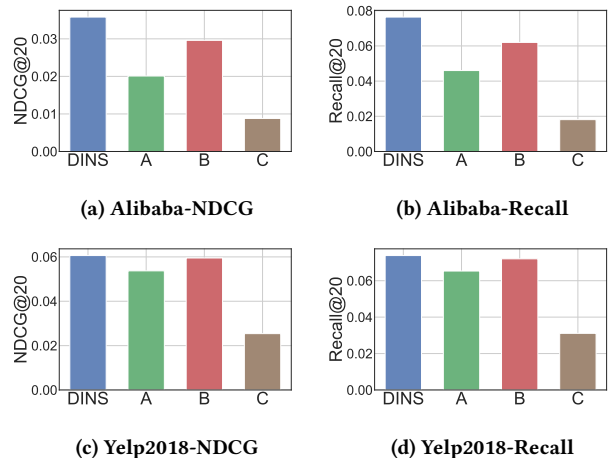


Figure 4: Ablation experiments on LightGCN.

- DINS always performs the best. Removing any module would have a negative effect on the performance. It reveals each part of DINS contributes to the performance.
- Removing the Hard Boundary Definition module (Variant A) results in a substantial drop in performance across the datasets, indicating the importance of maintaining an optimal size for the negative sampling region. A too-small region may lead to the loss of valuable feature information from the negative samples. At the same time, a too-large region hampers the effective fusion of information from the positive samples.
- Compared with the other two modules, changing the dimension independent mixup module to the traditional mixup (Variant B) has a relatively small performance drop. It shows the mixup technique is already powerful by generating negative items in the continuous space. The proposed dimension independent mixup further enhances the improvement, especially on the Alibaba.
- Removing the multi-hop pooling module has a great impact on the performance. It reveals the importance of considering high-order information during the negative sampling procedure. This observation aligns with the finding of MixGCF [11].

By observing the data, it can be seen that the area of negative sampling can neither be too large nor too small. According to Equation 4, if the area is too small, the hard negative item is synthesized too close to the positive items and thus the features of the negatives will be lost, while if the area is too far, the features of the positive items cannot be incorporated into the hard negative items at all.

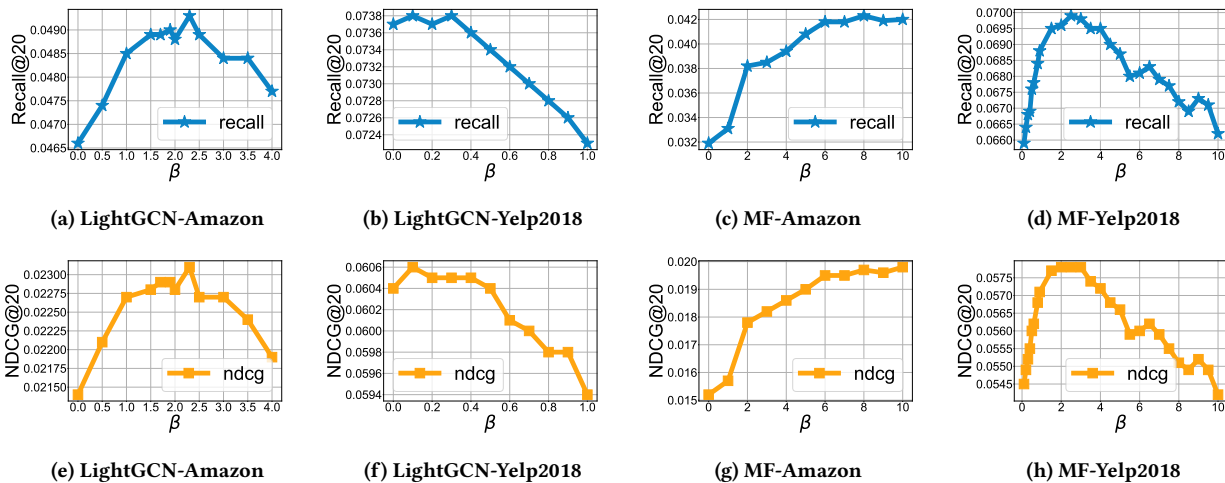
## 4.4 RQ3: Parameter Sensitivity

In this section, we focus on how the hyper-parameters ( $\beta$ ,  $M$ ) and the boundary item selection method actually affect DINS.

**4.4.1 Impact of the  $\beta$  value.**  $\beta$  is a distinctive hyper-parameter within our DINS framework, serving to regulate the incorporation of information from the positive item as a collective entity into our synthetic hard negative item.  $\beta$  is a propensity coefficient that controls whether our synthetic **Hard Negative Samples** (HNS) as a whole converge toward positive or negative items. Larger  $\beta$  means the synthetic HNS is closer to the positive item. We conduct

**Table 2: Performance Comparison. The best and runner-ups are marked in bold and underlined separately.**

Backbone Model	Sampling Method	Amazon		Alibaba		Yelp208	
		Recall@20	NDCG@20	Recall@20	NDCG@20	Recall@20	NDCG@20
LightGCN	Popularity	0.0323	0.0153	0.0481	0.0231	0.0469	0.0369
	RNS	0.0399	0.0178	0.0550	0.0251	0.0605	0.0493
	DNS	0.0453	0.0211	0.0576	0.0258	<u>0.0706</u>	<u>0.0581</u>
	IRGAN	0.0338	0.0150	0.0551	0.0255	0.0535	0.0251
	MixGCF	<u>0.0456</u>	<u>0.0214</u>	<u>0.0689</u>	<u>0.0332</u>	0.0691	0.0565
	DINS	<b>0.050</b>	<b>0.0236</b>	<b>0.0764</b>	<b>0.0358</b>	<b>0.0738</b>	<b>0.0604</b>
	Improvement	9.6%	10.3%	10.9%	7.9%	4.5%	4.0%
NGCF	Popularity	0.0115	0.0047	0.0180	0.0080	0.0253	0.0196
	RNS	0.0288	0.0119	0.0337	0.0144	0.0561	0.0457
	DNS	0.0304	0.0131	0.0475	0.0228	0.0634	0.0520
	IRGAN	0.0194	0.0078	0.0280	0.0116	0.0438	0.0353
	MixGCF	0.0350	0.0150	0.0562	0.0268	0.0686	0.0567
	DINS	<b>0.0379</b>	<b>0.0163</b>	<b>0.0607</b>	<b>0.0277</b>	<b>0.0709</b>	<b>0.0586</b>
	Improvement	8.3%	8.7%	8.0%	3.4%	3.4%	3.4%
MF	Popularity	0.0148	0.0122	0.0215	0.0103	0.0382	0.0317
	RNS	0.0245	0.0104	0.0301	0.0144	0.0558	0.0449
	DNS	0.0320	0.0154	<u>0.0487</u>	<u>0.0240</u>	<u>0.0663</u>	<u>0.0547</u>
	IRGAN	0.0281	0.0119	0.0307	0.0139	0.0412	0.0338
	MixGCF*	<u>0.0342</u>	<u>0.0156</u>	0.0480	0.0232	0.0642	0.0525
	DINS	<b>0.0423</b>	<b>0.0197</b>	<b>0.0663</b>	<b>0.0320</b>	<b>0.0699</b>	<b>0.0579</b>
	Improvement	23.7%	26.3%	36.1%	33.3%	8.9%	10.3%

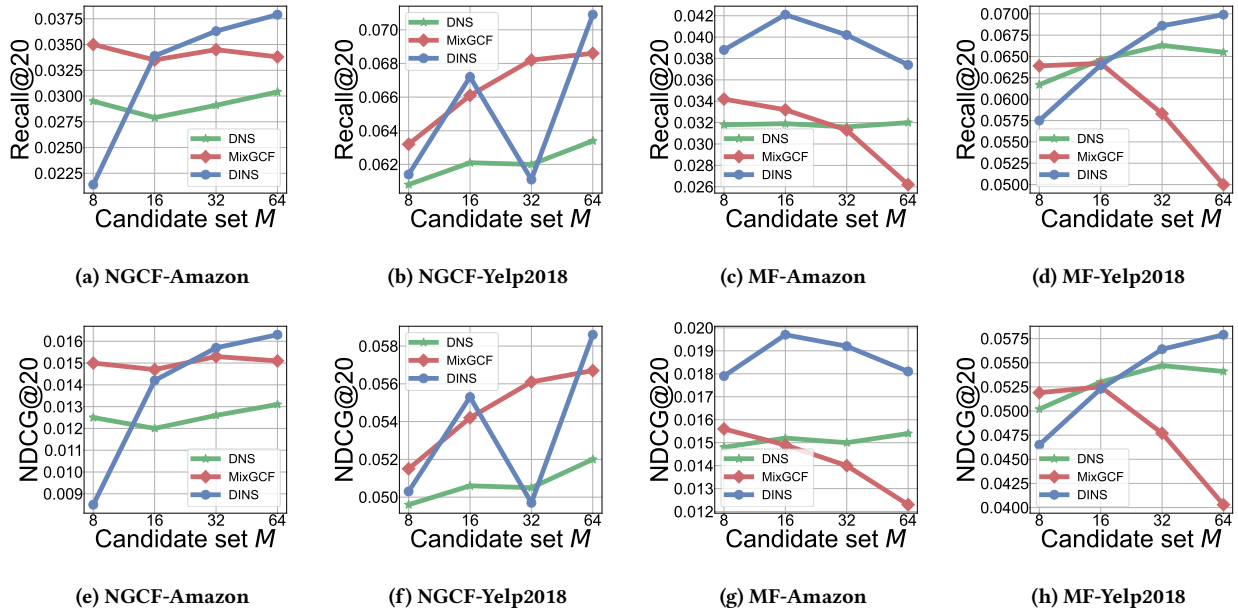
**Figure 5: The impact of the  $\beta$  value on model performance.**

experiments with all 3 datasets on 3 recommenders. Due to the space limit, we illustrate parts of the results in Figure 5. Similar trends are also observed on other experiments. The observations are as follows:

- As the dataset becomes sparser, the encoder necessitates the fusion of a greater amount of information about the positive items. According to the statistics in Table 1, it is clear that Yelp208 is much denser than Amazon and Alibaba. On the Yelp208 dataset, LightGCN achieved the best results at around  $\beta = 0.1$  and MF

at around  $\beta = 2$ . In contrast, on the Amazon dataset, LightGCN needs to be around 2.3 and MF needs to be around 8 to achieve the best results. This may be due to the fact that the sparser the dataset the larger the embedding space of users and items is, resulting in a larger sampling boundary for our determination. Even after dimension-wise mixup, the synthesized negative items can still be far away from the positive one, so we need to move closer to the positive item as a whole with a larger  $\beta$ .

- The stronger the encoding capability of the encoder, the less information from the positive item is needed for DINS. LightGCN

Figure 6: The impact of the candidate set size  $M$ .

is a state-of-the-art CF method that employs a message-passing mechanism on the bipartite user-item graph to learn a better embedding and MF is a straightforward model which recommends by a factorization-based approach. The  $\beta$ -values required for LightGCN to achieve the best performance on all three datasets are much smaller than those of MF and NGCF. This may be because LightGCN uses a message-passing mechanism on the graph to automatically aggregate a portion of the information from the positive items, while MF does not have a message-passing mechanism to naturally aggregate the information from the positive items. For example, on the Amazon dataset, LightGCN achieves a significantly higher performance compared to MF. While LightGCN only requires an  $\beta$ -value of 2.3, MF necessitates an  $\beta$ -value of 8. Remarkably, LightGCN outperforms MF by 18.2% on Recall@20 and 19.8% on NDCG@20 metrics. This observation highlights the value of gathering information from both neighboring nodes and positive examples when training the encoder. The effectiveness of LightGCN, with its message-passing mechanism for automatic information aggregation from positive examples, demonstrates the benefit of incorporating such information to enhance the encoder’s performance.

**4.4.2 Impact of the candidate set size  $M$ .** We also conducted an experimental analysis on the candidate set size for negative items in DNS and MixGCF, which are included as baselines. We systematically tested these baselines using candidate set sizes of 8, 16, 32, and 64. Detail experiments are illustrated in Figure 6. We can observe that the  $M$  impacts all the sampling methods. Generally, increasing the candidate set size  $M$  tends to improve the performance of the experiment. For example, the best results are mostly achieved with  $M = 64$ . Interestingly, MixGCF exhibits lower stability than DNS and DINS, even yielding contrasting outcomes on the Amazon

dataset. Notably, as the candidate set increases, MixGCF demonstrates a significant decline. We attribute this disparity in results to the attributes of the dataset.

**4.4.3 Impact of the boundary item selection method.** To delve deeper into the influence of the boundary item selection method in Section 3.1, we further conduct experiments on the impact of the selection method. In the experiment, we opt for LightGCN as the encoder due to its superior performance. To be noted that the sampled boundary item decides the sampling area together with the positive item. The area is the continuous space between the boundary and the positive item. By multiplying all the absolute differences in each dimension, we can obtain the volume of the sampling area. Based on this principle, we further design 3 sampling methods to find the boundary item as:

- **Random.** Randomly select an item from the candidate set as the boundary item.
- **Min.** Find the item that constitutes the minimum sampling area volume as the boundary item.
- **Max.** Find the item that constitutes the maximum sampling area volume as the boundary item.
- **Dot Product (DP).** The method used in DINS as Equation 3.

After obtaining the boundary item, we also conduct the dimension independent mixup and multi-hop pooling. Subsequently, we conducted experiments on LightGCN to examine the impact of the boundary item selection method across the three datasets. The detailed experimental results are shown in Table 3. We can observe that Min and Max perform badly in most cases, which reveals the importance of selecting a suitable area for generating negative items. Even random selection outperforms Min and Max methods. The method used by DINS (DP) always performs the best across



**Table 3: Performance Comparison of different boundary item selection methods.**

	Amazon		Alibaba		Yelp2018	
	R@20	N@20	R@20	N@20	R@20	N@20
Random	0.0300	0.0123	0.0460	0.0201	0.0653	0.0537
Min	0.0145	0.0055	0.0279	0.0129	0.0441	0.0360
Max	0.0246	0.0084	0.0182	0.0081	0.0357	0.0294
DP	<b>0.0493</b>	<b>0.0231</b>	<b>0.0764</b>	<b>0.0358</b>	<b>0.0738</b>	<b>0.0606</b>

the 3 datasets. It shows we have selected a suitable sampling area to generate the negative item.

#### 4.5 RQ4: Case Study

To answer RQ4: *Does DINS really support area-wise hard negative sampling?* we conduct a case study for readers to understand the sampling principle behind DINS. Experiments are conducted on the Yelp2018 dataset with LightGCN as the backbone recommender. We train the model with RNS, MixGCF, and DINS for 60 epochs. For a fixed user-item interaction, we store the embedding of the positive item and the sampled negative item by different sampling methods in each iteration. Then we obtain the averaged positive item representation by mean pooling over all the collected positive item embedding. Then we concatenate the averaged positive item and all collected negative items embedding together and visualize the distribution via t-SNE [22]. For better visualization, we move the positive item to the center of the visualization by subtracting all the reduced 2-d embedding from the reduced positive item embedding.

The visualization is shown in Figure 7. We draw a circle (with a radius represented by  $R$ ) to show the farthest sampled negative item. We can have three observations:

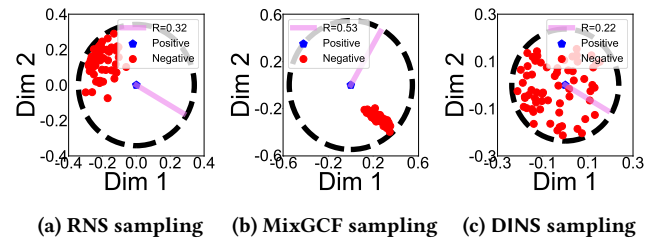
- The RNS sampling method samples varied negative items that are far from the positive item. It exhibits the characteristic of the point-wise negative sampling method. It samples other existing items that are restricted in the upper-left corner area based on currently learned embedding.
- The MixGCF method, as expected for the line-wise sampling method, exhibits line-style negative sampling results. It is due to the traditional Mixup method that generates a linear interpolation of two embeddings with the same weight on all dimensions.
- DINS shows the characteristics of the area-wise sampling method. We can observe the sampled negative items spans across the whole circle, which gives a sufficient exploration of the embedding space. At the same time, the sample radius of DINS is only 0.22, which is the smallest among the three methods. It shows DINS samples the hard negatives for assist model training.

## 5 RELATED WORK

In this section, we introduce the related work for DINS, which includes the graph-based recommendation and negative sampling method in recommendation.

### 5.1 Graph-based recommendation

Recent years have witnessed the rapid development of the emerging direction of GNN-based recommender systems [10, 17, 21, 24, 25, 29–31, 33] where the user-item interactions are presented as a bipartite

**Figure 7: The hard negatives sampled by RNS, MixGCF, DINS**

graph, and graph neural network methods are employed to learn the representation of each node via exploring structure information. For example, Pinsage [33] samples neighborhoods according to the visit counts of a node through a random-walk sampling approach. GC-MC [21] proposes a graph auto-encoder framework to construct the node representation by directly aggregating the information of its neighbors. NGCF [25] captures high-order connectivities by stacking multiple embedding propagation layers and utilizes the combination of different layers' output for the rating prediction. Compared with NGCF, LightGCN [10] achieves better training efficiency and generation capability by removing the feature transformation and nonlinear activation function. Finally, SVD-GCN [17] further simplifies LightGCN by replacing neighborhood aggregation with exploiting K-largest singular vectors for the close relation between GCN-based and low-rank methods.

### 5.2 Negative sampling method

Negative sampling in RecSys has gained significant attention due to its ability to accelerate training and greatly enhance model performance. According to a widely accepted classification, these methods can be categorized into the following groups.

**Static Sampler** methods sample from an item space that users have not interacted with using predefined distributions, such as RNS[10, 19, 25] (random negative sampling) which selects negative items uniformly or PNS[4] (popularity-biased sampling) which considers item popularity. **Hard Negative Sampler** techniques like DNS[35] (dynamic negative sampling) choose items with the highest scores from the current recommender, while **GAN-based methods** like IRGAN[23] and AdvIR[16] use adversarial learning to generate negative items and improve robustness. Mixup-based methods like MixGCF[11] generate new negative items by performing "positive mixing" and "hop mixing" operations. **Auxiliary-based Samplers** leverage additional information, such as the knowledge graph in KGPolicy[26] or Personalized PageRank scores in PinSage[33], to sample hard negative instances. These approaches serve the purpose of selecting unexplored items to improve RecSys training.

We categorized the above sampling methods into Point/Line/Area-wise negative sampling methods. Methods like DNS[35], RNS[19], etc. directly select items that users have not yet interacted with as the negatives are categorized as point sampling. On the other hand, methods like MixGCF[11] and SDNS[20], which employ linear interpolation between negative and positive items, are classified as line sampling. But DINS introduces a novel approach to perform area sampling, setting it apart from previous methods. It determines an optimal negative sampling boundary point, ensuring an appropriate negative sampling area that strikes a balance between being

too large and too small. Subsequently, DINS employs Dimension Independent Fusion, a process that enables the newly synthesized negatives to locally approximate the positive items in each embedding dimension and then globally approach the positives. Through this combination of local approximation and overall approach, the synthesized negative items effectively retain their negative characteristics while exhibiting an unprecedented level of approximation to the positive items. This enhancement significantly amplifies the model's ability to discern between positive and negative items.

## 6 CONCLUSION

In conclusion, this paper provides a novel perspective to revisit the current negative sampling methods based on continuous sampling area and classifies them into point-wise and line-wise sampling methods. In a further step, we design the first area-wise sampling method, named DINS, by proposing the Dimension Independent Mixup method. DINS can easily support both matrix factorization and graph-based backbone recommenders. Extensive experiments demonstrate superior performance compared with other methods, making it a state-of-the-art solution for negative sampling when training collaborative filtering with implicit feedback. The contributions of this work include a fresh perspective on negative sampling methods, the introduction of Area-wise sampling, and the development of the innovative DINS method. These findings have the potential to enhance RecSys capabilities and improve user experiences in various online services.

## REFERENCES

- [1] Dong-Kyu Chae, Jin-Soo Kang, Sang-Wook Kim, and Jung-Tae Lee. 2018. Cfgan: A generic collaborative filtering framework based on generative adversarial networks. In *Proceedings of the 27th ACM international conference on information and knowledge management*. 137–146.
- [2] Chong Chen, Weizhi Ma, Min Zhang, Chenyang Wang, Yiqun Liu, and Shaoping Ma. 2023. Revisiting negative sampling vs. non-sampling in implicit recommendation. *ACM Transactions on Information Systems* 41, 1 (2023), 1–25.
- [3] Ting Chen, Yizhou Sun, Yue Shi, and Liangjie Hong. 2017. On sampling strategies for neural network-based collaborative filtering. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 767–776.
- [4] Ting Chen, Yizhou Sun, Yue Shi, and Liangjie Hong. 2017. On Sampling Strategies for Neural Network-Based Collaborative Filtering. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (Halifax, NS, Canada) (KDD '17)*. Association for Computing Machinery, New York, NY, USA, 767–776. <https://doi.org/10.1145/3097983.3098202>
- [5] Yian Chen, Xing Xie, Shou-De Lin, and Arden Chiu. 2018. WSDM cup 2018: Music recommendation and churn prediction. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. 8–9.
- [6] Jingtao Ding, Yuhan Quan, Quanming Yao, Yong Li, and Depeng Jin. 2020. Simplify and Robustify Negative Sampling for Implicit Collaborative Filtering. In *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (Eds.), Vol. 33. Curran Associates, Inc., 1094–1105. [https://proceedings.neurips.cc/paper\\_files/paper/2020/file/0c7119e3a6a2209da6a5b90e5b5b75bd-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2020/file/0c7119e3a6a2209da6a5b90e5b5b75bd-Paper.pdf)
- [7] Jingtao Ding, Yuhan Quan, Quanming Yao, Yong Li, and Depeng Jin. 2020. Simplify and robustify negative sampling for implicit collaborative filtering. *Advances in Neural Information Processing Systems* 33 (2020), 1094–1105.
- [8] Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *International Conference on Artificial Intelligence and Statistics*. 249–256.
- [9] Yulong Gu, Zhuoye Ding, Shuaiqiang Wang, and Dawei Yin. 2020. Hierarchical user profiling for e-commerce recommender systems. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. 223–231.
- [10] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yong Dong Zhang, and Meng Wang. 2020. LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation. *SIGIR 2020 - Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. <https://doi.org/10.1145/3397271.3401063>
- [11] Tinglin Huang, Yuxiao Dong, Ming Ding, Zhen Yang, Wenzheng Feng, Xinyu Wang, and Jie Tang. 2021. MixGCF: An Improved Training Method for Graph Neural Network-based Recommender Systems. *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining* (2021).
- [12] Diederik P. Kingma and Jimmy Lei Ba. 2014. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations*.
- [13] Yehuda Koren, Steffen Rendle, and Robert Bell. 2021. Advances in collaborative filtering. *Recommender systems handbook* (2021), 91–142.
- [14] Kelong Mao, Jieming Zhu, Jinpeng Wang, Quanyu Dai, Zhenhua Dong, Xi Xiao, and Xiuqiang He. 2021. SimpleX: A simple and strong baseline for collaborative filtering. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 1243–1252.
- [15] Viktor Mayer-Schönberger and Kenneth Cukier. 2013. *Big data: A revolution that will transform how we live, work, and think*. Houghton Mifflin Harcourt.
- [16] Dae Hoon Park and Yi Chang. 2019. Adversarial sampling and training for semi-supervised information retrieval. In *The World Wide Web Conference*. 1443–1453.
- [17] Shaowen Peng, Kazunari Sugiyama, and Tsunenori Mine. 2022. SVD-GCN: A Simplified Graph Convolution Paradigm for Recommendation. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management, Atlanta, GA, USA, October 17-21, 2022*, Mohammad Al Hasan and Li Xiong (Eds.). ACM, 1625–1634.
- [18] Steffen Rendle and Christoph Freudenthaler. 2014. Improving pairwise learning for item recommendation from implicit feedback. In *Proceedings of the 7th ACM international conference on Web search and data mining*. 273–282.
- [19] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2012. BPR: Bayesian personalized ranking from implicit feedback. In *Conference on Uncertainty in Artificial Intelligence*.
- [20] Kexin Shi, Yun Zhang, Bingyi Jing, and Wenjia Wang. 2022. Soft BPR Loss for Dynamic Hard Negative Sampling in Recommender Systems. *arXiv preprint arXiv:2211.13912* (2022).
- [21] Rianne van den Berg, Thomas N. Kipf, and Max Welling. 2017. Graph Convolutional Matrix Completion. *CoRR* abs/1706.02263 (2017).
- [22] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, 11 (2008).
- [23] Jun Wang, Lantao Yu, Weinan Zhang, Yu Gong, Yinghui Xu, Benyou Wang, Peng Zhang, and Dell Zhang. 2017. IRGAN: A Minimax Game for Unifying Generative and Discriminative Information Retrieval Models. In *Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. 515–524.
- [24] Shen Wang, Liangwei Yang, Jibing Gong, Shaojie Zheng, Shuying Du, Zhiwei Liu, and S Yu Philip. 2022. MetaKRec: Collaborative Meta-Knowledge Enhanced Recommender System. In *2022 IEEE International Conference on Big Data (Big Data)*. IEEE, 665–674.
- [25] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural Graph Collaborative Filtering. *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval* (2019).
- [26] Xiang Wang, Yaokun Xu, Xiangnan He, Yixin Cao, Meng Wang, and Tat-Seng Chua. 2020. Reinforced Negative Sampling over Knowledge Graph for Recommendation. In *WWW*.
- [27] Chuhan Wu, Fangzhao Wu, Mingxiao An, Jianqiang Huang, Yongfeng Huang, and Xing Xie. 2019. NPA: neural news recommendation with personalized attention. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 2576–2584.
- [28] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. 2021. Self-supervised graph learning for recommendation. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*. 726–735.
- [29] Liangwei Yang, Zhiwei Liu, Yingdong Dou, Jing Ma, and Philip S Yu. 2021. Consisrec: Enhancing gnn for social recommendation via consistent neighbor aggregation. In *Proceedings of the 44th international ACM SIGIR conference on Research and development in information retrieval*. 2141–2145.
- [30] Liangwei Yang, Zhiwei Liu, Yu Wang, Chen Wang, Ziwei Fan, and Philip S Yu. 2022. Large-scale personalized video game recommendation via social-aware contextualized graph neural network. In *Proceedings of the ACM Web Conference 2022*. 3376–3386.
- [31] Liangwei Yang, Shengjie Wang, Yunzhe Tao, Jiankai Sun, Xiaolong Liu, Philip S Yu, and Taiping Wang. 2022. DGRec: Graph Neural Network for Recommendation with Diversified Embedding Generation. *arXiv preprint arXiv:2211.10486* (2022).
- [32] Zhen Yang, Ming Ding, Xu Zou, Jie Tang, Bin Xu, Chang Zhou, and Hongxia Yang. 2023. Region or Global? A Principle for Negative Sampling in Graph-Based Recommendation. *IEEE Transactions on Knowledge and Data Engineering* 35, 6 (2023), 6264–6277. <https://doi.org/10.1109/TKDE.2022.3155155>
- [33] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec. 2018. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 974–983.
- [34] Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. 2018. mixup: Beyond Empirical Risk Minimization. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3,*

- 2018, *Conference Track Proceedings*. OpenReview.net. <https://openreview.net/forum?id=r1Ddp1-Rb>
- [35] Weinan Zhang, Tianqi Chen, Jun Wang, and Yong Yu. 2013. Optimizing top-n collaborative filtering via dynamic negative item sampling. In *Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*.
- [36] Lei Zheng, Chaozhuo Li, Chun-Ta Lu, Jiawei Zhang, and Philip S Yu. 2019. Deep Distribution Network: Addressing the Data Sparsity Issue for Top-N Recommendation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1081–1084.