Expert Systems With Applications 259 (2025) 125346

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Towards similar alignment and unique uniformity in collaborative filtering

Lei Sang, Yu Zhang, Yi Zhang, Honghao Li, Yiwen Zhang *

School of Computer Science and Technology, Anhui University, Hefei 230601, China

ARTICLE INFO

Keywords:
Collaborative filtering
Recommender systems
Alignment and uniformity
Graph neural network

ABSTRACT

Representation learning, with its desired properties, including Alignment and Uniformity, has recently emerged as an effective approach in collaborative filtering (CF) for recommender systems. Alignment measures the distance between positive pairs, and uniformity describes the distribution of all samples on the unit hypersphere. Despite the effectiveness of existing research, the challenges of insufficient alignment and uniformity bias persist in studies on the desired properties of recommender systems: (1) Sparse interaction information leads to an insufficient alignment representation. (2) Calculating the uniformity loss based on duplicate samples introduces bias, which affects the overall representation uniformity.

Building on aforementioned challenges, we propose a Similar Alignment and Unique Uniformity model called SUAU. SUAU mitigates insufficient alignment by introducing additional item-similar item pairs for model training and employing a uniqueness strategy to prevent uniformity bias. More specifically, we propose a similar items generation method that identifies the most similar items based on the interaction behavior of the original items' related users to form item-similar item pairs. The experimental results obtained on three highly sparse datasets demonstrate that the propose model achieves superior alignment and uniformity compared with state-of-the-art models, notably surpassing the best CF methods in terms of recommendation performance. We have open-sourced the source code on a publicly accessible website: https://github.com/ZzYUuuu/SUAU.

1. Introduction

Recommender systems (RS) are designed to improve user experiences by understanding individual preferences (Wu et al., 2023; Xu et al., 2024a; Yu et al., 2023). As technology advances, RS will play a larger role in various internet applications (e.g., e-commerce, advertising, and user reviews) (Wu et al., 2022a; Xu et al., 2024b). The core of this technology lies in comprehending user historical preferences. By modeling diverse user preferences (He et al., 2017; Koren et al., 2009), we can better understand user interaction behavior, anticipate potential preferences, and offer more personalized recommendations. However, these modeling approaches are often face challenges, such as the sparsity of recommendation datasets, which complicates the task of accurately capturing user potential preferences.

As graph neural networks have evolved (Wu et al., 2022b), the graph structure has become capable of effectively learning multi-hop neighborhood information, which further mitigates data sparsity issues. To more effectively capture potential user preferences and alleviate the sparse problem of recommendation data, an increasing number of graph-based collaborative filtering (CF) methods, such as graph convolutional-based models (He et al., 2020, 2017), graph contrastive-based models (Wei et al., 2024; Xiao et al., 2024; Yu et al., 2022), and graph autoencoder models (Xia et al., 2023), have been proposed. Among these, graph convolutional networks (GCN), notably spearheaded by LightGCN (He et al., 2020), have made significant strides in graph-based recommendation by aggregating information from high-order neighborhoods. Despite their effectiveness, graph-based models generally lack a detailed study of user and item representations. Existing studies only use randomly initialized embedding layers to represent user and item IDs (Sun et al., 2024; Zhang et al., 2024c), without exploring into the inherent properties of these representations. This omission raises questions about whether user preferences are accurately captured, which indicates a critical need for a deeper exploration of recommendation representation learning.

Recent progress in representation learning has led to two desired properties: alignment and uniformity (Wang & Isola, 2020). Alignment reflects the distance between similar sample pairs (*i.e.*, original-augmented pairs), and uniformity reflects the distribution of all samples on the unit hypersphere (*i.e.*, original samples hypersphere). The results demonstrate that by ensuring alignment among positive pairs and uniformity on the unit hypersphere, model performance can be significantly enhanced. In the context of recommender systems, SimGCL (Yu

* Correspondence to: Anhui University, Hefei, China.
E-mail addresses: sanglei@ahu.edu.cn (L. Sang), e23301291@stu.ahu.edu.cn (Y. Zhang), zhangyi@stu.ahu.edu.cn (Y. Zhang), e22201130@stu.ahu.edu.cn (H. Li), zhangyiwen@ahu.edu.cn (Y. Zhang).

https://doi.org/10.1016/j.eswa.2024.125346
Received 24 June 2024; Received in revised form 21 August 2024; Accepted 6 September 2024
Available online 10 September 2024
0957-4174/

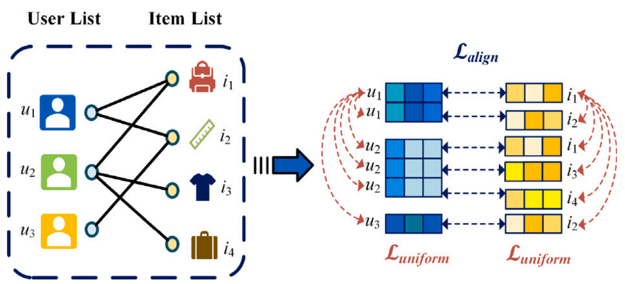


Fig. 1. Left: Example of a user-item interaction. Right: Loss calculation process. .

et al., 2022) points out that uniformity greatly influences the model's final recommendation performance. Although the experimental results align well with this description, the model does not propose methods to optimize uniformity. DirectAU (Wang et al., 2022) directly introduces the alignment loss and uniformity loss functions in Fig. 1. The alignment loss aims to boost the similarity between user-item pairs, bringing them closer, while the uniformity loss ensures a more even distribution of user and item representations. Building on this model, a series of AU-based models are proposed to further optimize the alignment of the representation (Yang et al., 2023) or to incorporate additional self-supervised signals (Ouyang et al., 2024).

Despite the excellence of these AU-based models, we argue that they encounter two significant challenges relative to the alignment and uniformity of user and item representations:

(1) **Insufficient Alignment Representation**. Datasets in recommender systems frequently suffer from interaction sparsity (Jiang et al., 2023; Natarajan et al., 2020), which refers to the lack of user-item interactions. Most existing AU-based models treat user-item interactions as positive pairs (Wang et al., 2022; Yang et al., 2023), and these pairs are used to enhance the alignment of the representations. However, the challenge associated with interaction sparsity is significant because some users and items exhibit only a limited number of sparse positive pairs. This scenario restricts the certain samples' alignment to a small subset of samples, hampering the overall alignment of user and item representations, which could potentially resulting in suboptimal recommendations.

(2) **Uniformity Loss Calculation Introduces Bias**. In practical training scenarios, large amounts of data necessitate batch processing before model training (Li et al., 2014; Masters & Luschi, 2018). Similarly, AU-based models employ this method for training, where the alignment loss and uniformity loss are calculated using batched user-item pairs and batched user and item representations, respectively. However, because of the inherent structure of recommendation data, users/items frequently interact with multiple different items/users, which results in duplicate user and item representations within the same batch. These duplicates can introduce biases in the calculation of uniformity loss, causing a non-uniform distribution of the overall representations that could harm recommendation performance.

In this paper, through a quantitative analysis of the datasets interaction statistics, we observe certain subsets of items with sparse interaction information. This can affect the alignment of items representations and poses challenges in accurately capturing user deep preferences. (1) To address the issue of insufficient alignment, building on the above findings, we introduce a similar items generation method that identifies similar items based on the interaction behavior of the original items' related users and uses cosine similarity to find the most similar items, considering these pairs as item-similar item pairs. (2) Furthermore, we propose a uniqueness strategy to address the uniformity bias problem and ensure that the current user and item representations align more closely with the concept of uniformity. By introducing item-similar item pairs and a uniqueness strategy into the model training process, we ensure a more uniform distribution and more sufficient alignment (as illustrated in Fig. 8). Building on these findings, we propose a Similar Alignment and Unique Uniformity model, named **SUAU**, which aims to mitigate uniformity bias while further enhancing the alignment and uniformity of overall representations. Extensive experiments on three publicly available real datasets demonstrate that the propose model outperforms state-of-the-art models in terms of alignment, uniformity, and recommendation accuracy.

The primary contributions of this study are summarized as follows:

- We analyze the statistics of user/item interactions and find that capturing information from certain items with sparse interaction is particularly challenging. Additionally, our theoretical analysis indicates that duplicate samples for calculating the uniformity loss introduce uniformity bias.
- We propose the SUAU model, which uses the interaction behavior of item-related users to generate item-similar item pairs for training, effectively mitigating insufficient alignment. Furthermore, we introduce a uniqueness strategy to filter duplicate samples, preventing uniformity bias.
- We conduct extensive experiments on three real-world public datasets to evaluate the effectiveness of SUAU. Our analysis of the loss evolution process of AU-based models indicates that the propose model achieves superior alignment and uniformity of the overall representations.

2. Preliminaries

In this section, we present the preparatory knowledge relevant to this study: graph-based collaborative filtering and alignment and uniformity in recommender systems.

2.1. Graph-based collaborative filtering

Given that $\mathcal{U}$ and $\mathcal{I}$ denote users and items, respectively, and given the corresponding rating matrix $\mathcal{R}\{\mathcal{U},\mathcal{I}\}$, where $\mathcal{R}(u,i)=1$ represents an interaction between user u and item i and $\mathcal{R}(u,i)=0$ indicates otherwise. Building upon this foundation, we construct a user-item bipartite graph (He et al., 2020) $\mathcal{G}=(\mathcal{N},\mathcal{E})$. Here, $\mathcal{N}=\{\mathcal{U}\cup\mathcal{I}\}$ denotes the node set of users and items, and $\mathcal{E}$ represents the user-item pairs in the rating matrix $\mathcal{R}\in\mathbb{R}^{M\times N}$, where M and N denote the number of users and items, respectively. Collaborative filtering methods learn from known user interactions to predict the ratings of corresponding items for ranking (Schafer et al., 2007). The top-ranked items are recommended to users based on their scores. This approach attempts to provide users with the most attractive personalized recommendations by deeply analyzing their interaction behavior.

Currently, Graph Convolutional Networks (GCNs) (He et al., 2020) represent one of the most prominent methods in the field of collaborative filtering. The core concept is to capture high-order neighborhood collaborative signals via graph convolutional neural networks. Specifically, the method learns neighborhood information for each layer by leveraging the adjacency matrix derived from the user-item bipartite graph, thereby achieving high-order connectivity as follows:

$$\mathbf{e}_u^{(l+1)}=\sum_{i\in\mathcal{N}_u}\frac{1}{\sqrt{|\mathcal{N}_u||\mathcal{N}_i|}}\mathbf{e}_i^{(l)}, \tag{1}$$

$$\mathbf{e}_i^{(l+1)}=\sum_{u\in\mathcal{N}_i}\frac{1}{\sqrt{|\mathcal{N}_u||\mathcal{N}_i|}}\mathbf{e}_u^{(l)}. \tag{2}$$

where $\mathcal{N}_u$ and $\mathcal{N}_i$ denote user and item one-hop neighbors, respectively, and $\mathbf{e}_u^{(l)}$ and $\mathbf{e}_i^{(l)}$ denote user u and item i embeddings in the lth layer, respectively. To integrate neighbor information from all levels, the model employs weighted aggregation techniques as follows:

$$\mathbf{E} = \frac{1}{L+1} \sum_{l=0}^{L} \mathbf{E}^{(l)}. \tag{3}$$

where $\mathbf{E} = [\mathbf{e}_{u_1}, \mathbf{e}_{u_2}, \ldots, \mathbf{e}_{u_M}, \mathbf{e}_{i_1}, \mathbf{e}_{i_2}, \ldots, \mathbf{e}_{i_N}] \in \mathbb{R}^{(M+N)\times D}$ denotes the final embedding representations of all users and items, and L denotes the total number of layers of the GCN. The parameters are optimized using the point-wise Bayesian Personalization Ranking (BPR) (Rendle et al., 2009) loss as follows:

$$\mathcal{L}_{BPR} = -\log(\text{sigmoid}(\mathbf{e}_u^\top \mathbf{e}_i - \mathbf{e}_u^\top \mathbf{e}_j)). \tag{4}$$

where $\langle u, i, j\rangle$ represents a triplet input, $\mathbf{e}_u$ and $\mathbf{e}_i$ are embeddings of users and interacted items, respectively, $\top$ is the embedding transpose, and $\mathbf{e}_j$ is embedding for randomly selected negative items. After training the model to acquire the final embedding, we can predict the scores $\hat{r}_{ui}$:

$$\hat{r}_{ui} = \mathbf{e}_u^\top \mathbf{e}_i. \tag{5}$$

where $\mathbf{e}_u$ and $\mathbf{e}_i$ represent the final embedding representations of user u and item i, respectively.

2.2. Alignment and uniformity in recommender systems

Recent research in representation learning has demonstrated that alignment and uniformity in desired properties substantially reflect model performance (Wang & Isola, 2020; Wang et al., 2023; Zhang et al., 2023).

To better explain our research, we first introduce concise definitions of alignment and uniformity as follows:

$$l_{align} = \mathop{\mathbb{E}}_{(x,x^+)\sim p_{\text{pos}}} \| f(x) - f(x^+)\|^2, \tag{6}$$

$$l_{uniform} = \log \mathop{\mathbb{E}}_{(y,z)\overset{\text{i.i.d.}}{\sim} p_{\text{data}}} e^{-2\|f(y)-f(z)\|^2}. \tag{7}$$

where $f(\cdot)$ denotes the embedding mapping function, p_{pos} and p_{data} denote the distribution of positive pairs and the distribution of data, respectively, x and x^+ denote $x-x^+$ positive pairs,s y and z denote each y with remaining z on the unit hypersphere to compute uniformity. The alignment and uniformity properties act to reduce the distance between positive pairs and to ensure a uniform distribution across the unit hypersphere, respectively.

In the recommender systems field, DirectAU (Wang et al., 2022) demonstrates a nuanced integration of alignment and uniformity with losses function. The model proposes alignment loss and uniformity loss by treating user-item interactions as positive pairs and envisioning all users/items on the same hypersphere, which directly enhances the collaborative filtering method:

$$\mathcal{L}_{align} = \mathop{\mathbb{E}}_{(u,i)\sim p_{\text{pos}}} \| f(u) - f(i)\|^2, \tag{8}$$

$$\begin{aligned}\mathcal{L}_{uniform} = &\frac{1}{2}\log \mathop{\mathbb{E}}_{(u,u')\overset{\text{i.i.d.}}{\sim} p_{\text{user}}} e^{-2\|f(u)-f(u')\|^2} + \\ &\frac{1}{2}\log \mathop{\mathbb{E}}_{(i,i')\overset{\text{i.i.d.}}{\sim} p_{\text{item}}} e^{-2\|f(i)-f(i')\|^2}.\end{aligned} \tag{9}$$

where u and i denote the corresponding user and item in the user-item pairs p_{pos} for alignment loss, respectively, and $i.i.d.$ denotes samples that are computed pairwise on the hypersphere for uniformity loss, and p_{user} and p_{item} denote the sets of users and items, respectively. The alignment loss is intended to minimize the distances among positive pairs, and the uniformity loss attempts to ensure a uniform and distinguishable distribution of user and item representations.

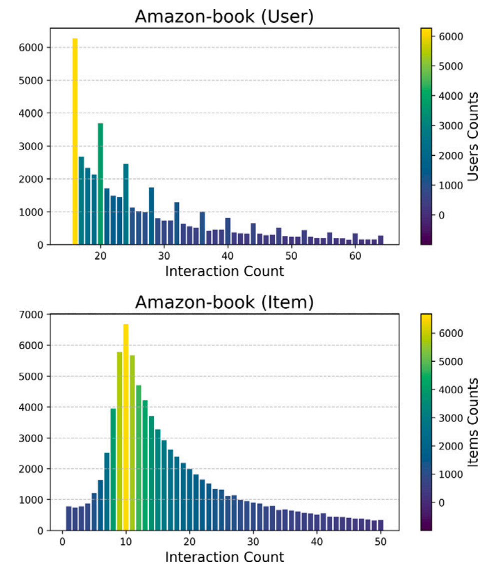


Fig. 2. User/Item interaction statistics results. Top: User interaction statistic. Bottom: Item interaction statistic. The y-axis represents the number of users/items at different interaction scales (Similar results for other datasets).

3. Investigation of alignment and uniformity challenges in the recommendation

In this section, we investigate the frequency of user and item interactions in relevant datasets to uncover potential deficiencies and challenges. In addition, to illustrate uniformity bias problem, we present a comprehensive theoretical analysis of the drawbacks of AU-based models in batch training.

3.1. Data analysis

Despite the large size of datasets commonly employed in recommender systems, data sparsity remains a critical challenge for existing AU-based models (Ouyang et al., 2024; Wang et al., 2022). This is primarily due to these models considering only user-item pairs as positive pairs, ignoring the problem of user-item interaction sparsity, which results in insufficient alignment representation. To gain deeper insights, we analyze the frequency of interactions in relevant datasets, as depicted in Fig. 2. After analyzing User interaction statistic, we uncover that user count decreased consistently as interaction frequency increased. A substantial number of users demonstrate low interaction, which mirrors typical user behavior. In addition, we observe that interactions rarely fall below a specified threshold (*e.g.*, 'Interaction Count'< 15 in the Amazon-book (User)). This trend can be attributed to the fact that in order to better capture user preferences, most datasets (He et al., 2020; Yu et al., 2022) filter out users with few interactions.

In contrast, Item interaction statistics mirror real-world interaction patterns, with item counts initially increasing and then decreasing as interaction frequencies increase. However, it is evident that certain subsets of items have minimal interactions, which presents challenges relative to capturing their latent representations. For example, on the Amazon-book dataset, while the overall dataset has '2984.1k' interaction counts as derived from Table 2, an item may be uniquely aligned with by only a single user, resulting in insufficient alignment representation. Therefore, effectively using the information from this subset of items can alleviate interaction sparsity and better capture user preferences. To counter this issue and tackle the challenge of insufficient alignment, we propose a solution in Section 4.1.

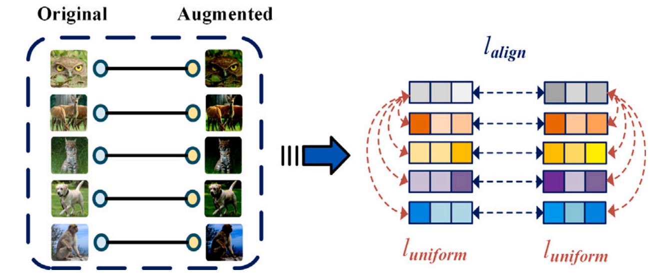


Fig. 3. Left: Image samples: original (left column) and augmented (right column). Right: Calculation process of alignment and uniformity.

3.2. Theoretical analysis

To more effectively demonstrate the presence of uniformity bias, we conduct a theoretical analysis of the uniformity loss in the contexts of representation learning (Wang & Isola, 2020) and recommender systems (Wang et al., 2022), respectively. In representation learning, uniformity loss is typically calculated using different sample representations. As an example, in Fig. 3, to obtain positive pairs, we typically acquire augmented images by augmenting the original images. Subsequently, we treat the original and augmented images are treated as positive pairs to calculate alignment loss. The uniformity loss is calculated using the representations of the original and augmented images, respectively. In contrast, in recommender systems, owing to the different data structures, user-item pairs are used to compute alignment loss, and uniformity loss is computed separately for user and item representations. Assuming that each model iteration can be completed in one step, the calculations of uniformity loss are approximately equivalent across the two domains.

However, due to the limitations of current hardware in processing highly sparse datasets during training—specifically, the inability to compute alignment loss for all positive pairs and uniformity loss for all user and item sets simultaneously with gradient descent—batch processing is commonly used in current research to achieve efficient training (Wang et al., 2022; Yang et al., 2023). To more effectively verify the differences and limitations in calculating the uniformity loss for recommender systems and representation learning, we propose to convert the training conditions to batch training conditions in both domains. We first introduce the transformation process of the target samples in the alignment loss and uniformity loss for representation learning from Eqs. (6) and (7) as follows:

$$(x, x^+) \sim p_{\text{pos}} \Rightarrow (x, x^+) \sim p^b_{\text{pos}} \mid b \in \mathcal{B}, \tag{10}$$

$$(y, z) \overset{\text{i.i.d.}}{\sim} p_{\text{data}} \Rightarrow (y, z) \overset{\text{i.i.d.}}{\sim} p^b_{\text{data}} \mid b \in \mathcal{B}. \tag{11}$$

where $\Rightarrow$ represents the transformation process of the training samples, $\mathcal{B}$ and b denote the total number of batches and samples per batch in each training cycle, respectively. p^b_{pos} and p^b_{data} denote the training samples for alignment loss and uniformity loss in each batch, respectively.

In the recommendation domain, batch training typically involves dividing all user-item pairs into smaller batches and performing sequential training on these batches (Ouyang et al., 2024; Zhang et al., 2023b). DirectAU follows a similar method. It calculates alignment loss by considering batched user-item pairs as positive pairs and separately computes uniformity loss based on user and item representations from these batched pairs. The corresponding conversion process is given by Eqs. (8) and (9) are as follows:

$$(u, i) \sim p_{\text{pos}} \Rightarrow (u, i) \sim p^b_{\text{pos}} \mid b \in \mathcal{B}. \tag{12}$$

$$\begin{aligned} (x, x') &\overset{\text{i.i.d.}}{\sim} p_{\text{user}} \Rightarrow (x, x') \overset{\text{i.i.d.}}{\sim} p^b_{\text{user}} \mid b \in \mathcal{B}, \\ (i, i') &\overset{\text{i.i.d.}}{\sim} p_{\text{item}} \Rightarrow (i, i') \overset{\text{i.i.d.}}{\sim} p^b_{\text{item}} \mid b \in \mathcal{B}. \end{aligned} \tag{13}$$

where p^b_{pos} denotes positive pairs under batch conditions for alignment loss, and p^b_{user} and p^b_{item} denote the target samples of users and items under batch conditions for uniformity loss, respectively.

Although the batch conditions follow a slight modification of the conditions in Eqs. (8) and (9), this sampling strategy may introduce significant challenges to the uniformity of the overall representation. Consider a simple example illustrated in the left panel of Fig. 1. Suppose a specific batch contains three users $\{u_1, u_2, u_3\}$ and four items $\{i_1, i_2, i_3, i_4\}$, generating a total of six user-item interaction pairs ($u_1 - i_1, u_1 - i_2, u_2 - i_1, u_2 - i_3, u_2 - i_4, u_3 - i_2$). Alignment loss is computed based on there six user-item pairs to bring them closer together, while uniformity loss is calculated using users $\mathcal{U} = [u_1, u_1, u_2, u_2, u_2, u_3]$ and items $\mathcal{I} = [i_1, i_2, i_1, i_3, i_4, i_2]$ to ensure a more uniform distribution of $\mathcal{U}$ and $\mathcal{I}$ representations. As depicted in the right panel of Fig. 1, the model employs multiple sets of users and items (with repeated users and items) to calculate the alignment loss and uniformity loss. This raises a crucial question: *Can a calculation of uniformity loss using repeated sets truly ensure the uniform distribution of user and item representations?*

To investigate this issue, we meticulously analyze the loss calculation process in Fig. 1. From the user perspective, the presence of duplicate users u_1 and u_2 in the user set for calculating uniformity loss may cause user representation to increasingly favor users u_1 and u_2 with higher interaction frequencies, potentially leading to uniformity bias. A similar concern arises from the item perspective. In addition, the uniformity definition introduced in Eq. (7) regards p_{data} as unique samples on the unit hypersphere, differing from the hyperspheres p^b_{user} and p^b_{item} (possibly with duplicate samples) during batch conditions. Therefore, we argue that AU-based models' computation of the uniformity loss using user and item representations from batch user-item pairs significantly affects the uniformity of overall representations, which degrades recommendation performance. We propose a uniqueness strategy to mitigate the uniformity bias in Section 4.2.

4. Proposed method

In this section, to address the issues of insufficient alignment and uniformity bias, we propose the SUAU model, which comprises two main parts: a similar items generation method that identifies similar items to create item-similar item pairs, and a uniqueness strategy that filters duplicate samples during batch conditions. Through these parts, SUAU effectively alleviates insufficient alignment and uniformity bias while enhancing the alignment and uniformity of user and item representations. The architecture of the propose model is presented in Fig. 4.

4.1. Similar items generation

In the context of user-item interaction datasets derived from authentic behavioral preferences, data sparsity is a significant challenge. To overcome this challenge, previous studies have proposed various methods, including perturbation-based and graph reconstruction-based data augmentation techniques (Xia et al., 2023; Yu et al., 2022). However, it is difficult to apply these approaches to AU-based models. The main problem with AU-based models is the sparsity of interacting data in the dataset. The analysis of interaction behaviors (Section 3.1) reveals that these datasets often exhibit a subset of items with sparse interactions, which leads to an insufficient alignment representation of AU-based models. To address this, we propose a novel method for generating similar item pairs, which leverages user preferences associated with items and cosine similarity to identify items that are most similar to those items, thereby creating item-similar item pairs.

Fig. 4B illustrates the generation process of item-similar item pairs. Let us consider finding similar items for i_1. To identify the most similar items, we first focus on users with fewer interactions because they often exhibit more pronounced preference behaviors. From the user list that interacted with item i_1, we select the user group with the fewest

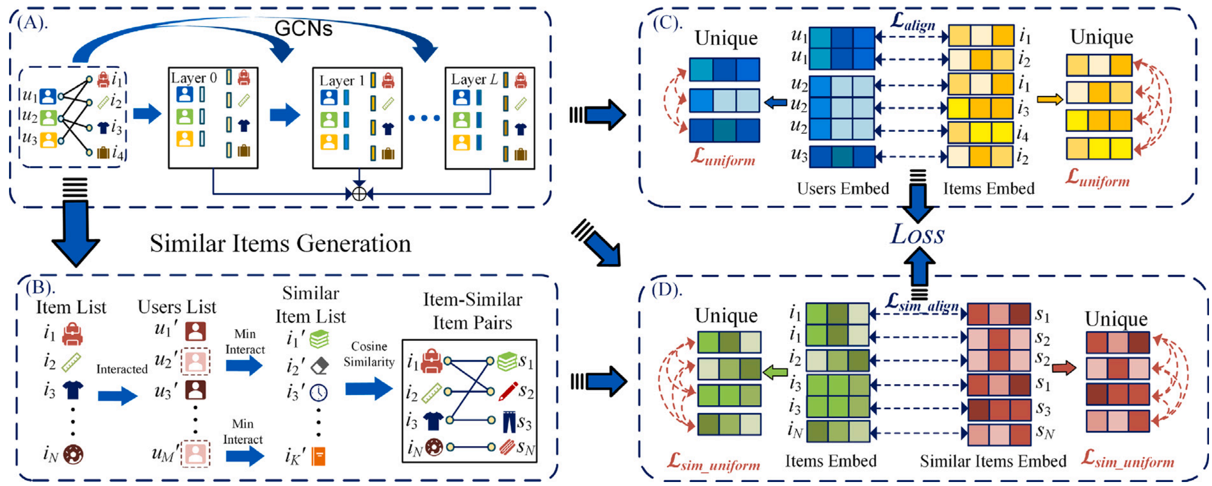


Fig. 4. Four parts of the SUAU model. A: Final embeddings obtained using the graph convolutional network. B: Similar items generation method. We determine the similarity of items by analyzing the interaction number of users associated with the original items and by employing cosine similarity (*i.e.*, Eq. (14)) to select the items that are most similar. (A lighter color for 'Users List' indicates fewer interactions.) C: Calculation of the loss of user-item pairs. D: Calculation of the loss of item-similar item pairs. In C and D, duplicate samples are filtered using a uniqueness strategy (*i.e.*, Eq. (15)) before calculating the uniform loss, respectively.

interactions (*e.g.*, the light-colored users $\{u'_2, \dots, u'_M\}$ with dashed borders). Subsequently, we compile a similar item list (comprising all items interacted with by the user group) and calculate the cosine similarity with item i_1. By ranking the results of the calculations, we can identify the most similar items, which can then be used to create item-similar item pairs $(i_1 - s_1, i_1 - s_2)$. The cosine similarity is computed as follows:

$$\text{cosine\_similarity}(\mathbf{e}_i, \mathbf{e}_{i'}) = \frac{\mathbf{e}_i \cdot \mathbf{e}_{i'}}{\|\mathbf{e}_i\| \cdot \|\mathbf{e}_{i'}\|}. \tag{14}$$

where $\cdot$ denotes the dot product, and $\mathbf{e}_i$ and $\mathbf{e}_{i'}$ denote items embedding and similar items embedding, respectively, $\|\mathbf{e}\|$ denotes the embedding after regularization. Although existing research has typically treated user-item interactions as positive pairs, we argue for a broader interpretation. Consider that positive pairs mean two samples are similar. We propose considering items and their similar items as positive pairs to better capture the close relationships between similar items. Accordingly, we compute the alignment loss and uniformity loss of the item-similar item pairs for model training. To maintain the uniformity of item representations, we extend the search for similar items to encompass all items.

4.2. Uniqueness strategy

Based on the above analysis, we conclude that the original uniformity loss is severely flawed in batch training, which leads to uniformity bias. Inspired by unsupervised contrastive representation learning (Wang & Isola, 2020), we argue that ensuring the uniqueness of samples within batches is essential for achieving overall representation uniformity. To address this issue, we propose a uniqueness strategy designed for recommender systems. By defining the initial sample embedding as $\mathbf{E}$ and the resulting unique sample embedding as $\mathbf{E}'$, we present a uniqueness strategy as follows:

$$\mathbf{E}' = \mathbf{Unique}(\mathbf{E}). \tag{15}$$

After applying the uniqueness strategy $\mathbf{Unique}(\cdot)$ to the initial sample embedding $\mathbf{E}$, the resulting sample embedding $\mathbf{E}'$ effectively filters out all duplicate samples. Referring to the example in Fig. 4C, the user embedding $\mathbf{E}_u = [\mathrm{e}_{u_1}, \mathrm{e}_{u_1}, \mathrm{e}_{u_2}, \mathrm{e}_{u_2}, \mathrm{e}_{u_2}, \mathrm{e}_{u_3}]$ undergoes filtering via the uniqueness strategy, resulting in the unique user embedding $\mathbf{E}'_u = [\mathrm{e}_{u_1}, \mathrm{e}_{u_2}, \mathrm{e}_{u_3}]$, which is used to calculate user uniformity loss. Because of the potentially high computational cost associated with filtering out duplicate embedding representations from the final representations, we use a uniqueness function (*i.e.*, Eq. (15)) to efficiently prefilter duplicate IDs based on user and item IDs before obtaining the final representations. The main loss function for SUAU is described as follows:

$$\mathcal{L}_{align} = \mathop{\mathbb{E}}_{(u,i)\sim p^b_{\text{pos}}|b\in\mathcal{B}} \|f(u) - f(i)\|^2, \tag{16}$$

$$\begin{aligned}\mathcal{L}_{uniform} = {} & \frac{1}{2}\log \mathop{\mathbb{E}}_{(u,u')\overset{\text{i.i.d.}}{\sim}\mathbf{Unique}(p^b_{\text{user}})|b\in\mathcal{B}} e^{-2\|f(u)-f(u')\|^2} + \\ & \frac{1}{2}\log \mathop{\mathbb{E}}_{(i,i')\overset{\text{i.i.d.}}{\sim}\mathbf{Unique}(p^b_{\text{item}})|b\in\mathcal{B}} e^{-2\|f(i)-f(i')\|^2}.\end{aligned} \tag{17}$$

To mitigate uniformity bias in user and item representations, we propose a uniqueness strategy that eliminates duplicate user and item samples during the computation of uniformity loss in batch training. This strategy achieves to achieve a more uniform distribution of users and items on the hypersphere, which enhances the model's recommendation performance.

4.3. Training details of SUAU

In our model training process, we partition the process into two parts. The first part computes the loss of user-item pairs as the main task, and the second part computes the loss of item-similar item pairs as the auxiliary task. To improve alignment and uniformity, we define the joint loss function from Eqs. (16) and (17) to optimize the main task as follows:

$$\mathcal{L}_{main} = \mathcal{L}_{align} + \gamma \mathcal{L}_{uniform}. \tag{18}$$

where γ as a hyperparameter for weighting the alignment loss and uniformity loss. The second part calculates the loss of item-similar item pairs. Considering the need to enhance the overall representation alignment through similar items, we use alignment loss to further improve alignment. However, when optimizing alignment, it is also essential to ensure the uniform distribution of representations. Therefore, we calculate both the alignment loss and uniformity loss for item-similar item pairs to enhance representation alignment as follows:

$$\mathcal{L}_{sim\_align} = \mathop{\mathbb{E}}_{(i,s)\sim p^b_{\text{pos\_sim}}|b\in\mathcal{B}} \|f(i) - f(s)\|^2, \tag{19}$$

$$\begin{aligned}\mathcal{L}_{sim\_uniform} = {} & \frac{1}{2}\log \mathop{\mathbb{E}}_{(i,i')\overset{\text{i.i.d.}}{\sim}\mathbf{Unique}(p^b_{\text{item\_sim}})|b\in\mathcal{B}} e^{-2\|f(i)-f(i')\|^2} + \\ & \frac{1}{2}\log \mathop{\mathbb{E}}_{(s,s')\overset{\text{i.i.d.}}{\sim}\mathbf{Unique}(p^b_{\text{item\_sim}'})|b\in\mathcal{B}} e^{-2\|f(s)-f(s')\|^2}.\end{aligned} \tag{20}$$

Table 1
Time complexity comparisons with AU-based baselines.

Model	Time complexity
LightGCN	$\mathcal{O}(2\|E\|Ld + 2Bd)$
DirectAU	$\mathcal{O}(Bd + B^2d)$
GraphAU	$\mathcal{O}(2\|E\| + 2\|E\|Ld + 4Bd + B^2d)$
AU$^+$	$\mathcal{O}(2\|E\| + 2\|E\|Ld + 3Bd + 3B^2d)$
SUAU	$\mathcal{O}(2\|E\| + 2\|E\|Ld + Bd + B'^2d)$

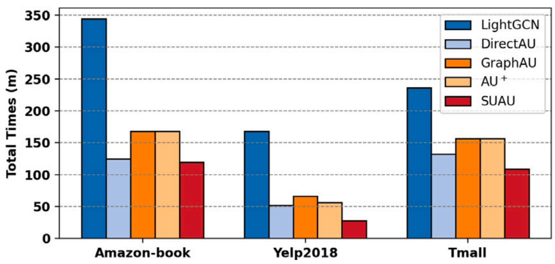


Fig. 5. The overall runtime of SUAU and the state-of-the-art AU-based models.

where $p^b_{\text{pos\_sim}}$ denotes item-similar item pairs for alignment loss, and $p^b_{\text{item\_sim}}$ and $p^b_{\text{item\_sim}'}$ denote the target samples of items and similar items under batch conditions for uniformity loss, respectively. The complete loss function for item-similar item pairs is calculated as follows:

$$\mathcal{L}_{similar} = \mathcal{L}_{sim\_align} + \gamma \mathcal{L}_{sim\_uniform}. \tag{21}$$

where γ here is the same as that defined previously. By combining the above equations, we obtain the total loss function that integrates both the main and auxiliary tasks as follows:

$$Loss = \mathcal{L}_{main} + \beta \mathcal{L}_{similar}. \tag{22}$$

To maintain a balance between the two tasks, we introduce the hyperparameter β. Although not explicitly stated, the final embedding is normalized using the L2-normalization technique before computing all losses.

4.4. Time complexity

To demonstrate the efficiency of our SUAU, we investigate the time complexity and runtime of the AU-based models in Table 1 and Fig. 5. Specifically, the embeddings in SUAU are generated through graph convolution, yielding a time complexity of $\mathcal{O}(2|E| + 2|E|Ld)$, where $|E|$ denotes the number of edges, L denotes the GCN layer, and d denotes the embedding dimensions. In addition, the losses are replaced by alignment loss and uniform loss, the time complexity of loss function become $\mathcal{O}(Bd + B'^2d)$, where $B' < B$ indicates that the batch sample of users or items has been reduced to some extent after applying the uniqueness strategy.

Compared to DirectAU (Wang et al., 2022), we consider using GCN as an encoder to enhance our experimental performance. Although GCN operations are time-consuming, SUAU compensates for this drawback by achieving efficient convergence when incorporating the encoder. In addition, unlike the state-of-the-art baseline AU$^+$ (Ouyang et al., 2024), our model does not require additional contrastive loss functions, thereby making it significantly more efficient in terms of computational complexity.

5. Experiments

5.1. Experimental settings

5.1.1. Datasets

To compare the propose SUAU model with the state-of-the-art baselines, we use three public datasets in real-world for comparative analysis.

Table 2
Statistics of the datasets.

Dataset	#Users	#Items	#Interactions	Density
Amazon-book	52.6k	91.6k	2984.1k	0.06%
Yelp2018	31.7k	38.0k	1561.4k	0.13%
Tmall	47.9k	41.4k	2619.4k	0.13%

- **Amazon-book** (Yu et al., 2022): This dataset comprises users' purchase history and intent regarding books on the Amazon platform, making it a valuable resource for investigating consumer behavior and developing book recommender systems.
- **Yelp2018** (He et al., 2020): This dataset contains user ratings for various business venues on the Yelp platform, making it crucial for studying user preferences and behavior in personalized venue recommendation scenarios.
- **Tmall** (Ren et al., 2023): This dataset contains user product information and purchase records from Alibaba's e-commerce platform Tmall, making it a vital resource for researching consumer behavior and developing product recommender systems.

In preparing the datasets, we follow related works (Zhang et al., 2023b, 2024b), including the removal of duplicates and ensuring each user has a minimum of five interactions. For datasets with ratings, we further preprocess them by labeling data with ratings of 3 or higher as interactions. Details of the preprocessed datasets are summarized in Table 2. During model training, we utilize the training set for training and evaluate the models' performance on the test set using metrics such as Recall@K, NDCG@K, and Precision@K (K=10, 20) (Wang et al., 2022; Yang et al., 2023), averaging the results over five experiments.

5.1.2. Baselines

We compare our model with the state-of-the-art CF models, which mainly consist of MF-based, GCN-based, GCL-based and AU-based models:

- **BPR-MF** (Rendle et al., 2009): This is a basic matrix factorization model that uses a pairwise loss function (BPR) trained on randomly drawn negative samples.
- **NGCF** (Wang et al., 2019): This model captures the high-order neighborhood information of users and items using the graph convolutional neural network.
- **LightGCN** (He et al., 2020): This is one of the most popular GCN model in recommender systems, which works by using simplified graph-convolutional method for collaborative filtering to obtain high-order neighborhoods information.
- **SGL** (Wu et al., 2021): This model pioneers self-supervised learning in recommender systems by generating subgraphs through data augmentation techniques such as dropping interaction edges or nodes. It then employs multiple graph convolutions to obtain different contrastive views for contrastive learning.
- **NCL** (Lin et al., 2022): This model is a novel approach to collaborative filtering. By generating additional contrastive views through structural (i.e., cross-layer) and semantic (i.e., clustering) neighbors, it effectively alleviates data sparsity and enhances the robustness of the model.
- **SimGCL** (Yu et al., 2022): This is a state-of-the-art graph contrastive learning model. It generates effective contrastive views by adding Gaussian noise to embeddings during the graph convolution process and it further captures user preferences through self-supervised signals.
- **DirectAU** (Wang et al., 2022): The model introduces two desired properties from representation learning into recommender systems and treats them as alignment loss and uniformity loss, thereby optimizing the alignment and uniformity of the overall representation.

Table 3
A comparison of overall performance of proposed SUAU model against the state-of-the-art MF-, GCN-, GCL- and AU-based baselines *w.r.t.* Recall@10, NDCG@10, Precision@10. The best value is highlighted in bold, while the second-best value is underlined. 'Improv.%' denotes the relative improvement of SUAU compared to the best baseline, with the enhancement being statistically significant as determined by a two-tailed paired t-test.

Method	Amazon-book			Yelp2018			Tmall		
	Recall@10	NDCG@10	Precision@10	Recall@10	NDCG@10	Precision@10	Recall@10	NDCG@10	Precision@10
BPR-MF	0.0170	0.0182	0.0147	0.0278	0.0317	0.0211	0.0312	0.0287	0.0256
NGCF	0.0199	0.0200	0.0175	0.0331	0.0368	0.0273	0.0374	0.0351	0.0267
LightGCN	0.0228	0.0241	0.0203	0.0362	0.0414	0.0336	0.0435	0.0406	0.0275
SGL	0.0263	0.0281	0.0228	0.0395	0.0448	0.0356	0.0457	0.0434	0.0289
NCL	0.0266	0.0284	0.0230	0.0403	0.0458	0.0361	0.0459	0.0429	0.0290
SimGCL	0.0313	0.0334	0.0252	0.0424	0.0488	<u>0.0383</u>	<u>0.0559</u>	<u>0.0536</u>	<u>0.0355</u>
VGCL	0.0312	0.0332	0.0248	0.0425	0.0485	0.0377	0.0557	0.0533	0.0354
DirectAU	0.0296	0.0297	0.0250	0.0414	0.0477	0.0367	0.0475	0.0443	0.0310
GraphAU	0.0300	0.0310	0.0248	0.0401	0.0463	0.0360	0.0517	0.0488	0.0331
AU^{+}	<u>0.0325</u>	<u>0.0345</u>	<u>0.0263</u>	<u>0.0427</u>	<u>0.0495</u>	0.0374	0.0540	0.0517	0.0343
SUAU(Ours)	**0.0342**	**0.0364**	**0.0277**	**0.0445**	**0.0512**	**0.0395**	**0.0573**	**0.0547**	**0.0365**
Improv.%	5.23%	5.51%	5.32%	4.22%	3.43%	3.13%	2.32%	2.05%	2.82%
p-value	3.08e−8	1.68e−8	5.05e−8	4.93e−8	1.46e−10	3.92e−9	5.86e−7	1.18e−7	7.65e−7

Table 4
A comparison of overall performance of proposed SUAU model against the state-of-the-art MF-, GCN-, GCL- and AU-based baselines *w.r.t.* Recall@20, NDCG@20, Precision@20. The best value is highlighted in bold, while the second-best value is underlined. 'Improv.%' denotes the relative improvement of SUAU compared to the best baseline, with the enhancement being statistically significant as determined by a two-tailed paired t-test.

Method	Amazon-book			Yelp2018			Tmall		
	Recall@20	NDCG@20	Precision@20	Recall@20	NDCG@20	Precision@20	Recall@20	NDCG@20	Precision@20
BPR-MF	0.0308	0.0239	0.0131	0.0486	0.0394	0.0176	0.0547	0.0400	0.0223
NGCF	0.0337	0.0262	0.0152	0.0579	0.0477	0.0232	0.0629	0.0465	0.0225
LightGCN	0.0411	0.0315	0.0169	0.0639	0.0525	0.0288	0.0711	0.0530	0.0226
SGL	0.0478	0.0379	0.0193	0.0675	0.0555	0.0296	0.0738	0.0556	0.0236
NCL	0.0481	0.0373	0.0193	0.0685	0.0577	0.0300	0.0750	0.0553	0.0237
SimGCL	0.0515	0.0414	0.0213	0.0721	0.0601	<u>0.0326</u>	<u>0.0884</u>	<u>0.0674</u>	<u>0.0281</u>
VGCL	0.0515	0.0410	0.0210	0.0715	0.0587	0.0324	0.0880	0.0670	0.0280
DirectAU	0.0506	0.0401	0.0211	0.0703	0.0583	0.0313	0.0752	0.0576	0.0242
GraphAU	0.0502	0.0400	0.0208	0.0691	0.0574	0.0303	0.0840	0.0625	0.0253
AU^{+}	<u>0.0538</u>	<u>0.0434</u>	<u>0.0224</u>	<u>0.0728</u>	<u>0.0604</u>	0.0322	0.0853	0.0651	0.0270
SUAU(Ours)	**0.0572**	**0.0452**	**0.0235**	**0.0750**	**0.0621**	**0.0338**	**0.0902**	**0.0688**	**0.0288**
Improv.%	6.32%	4.15%	4.91%	3.02%	2.81%	3.68%	2.04%	2.08%	2.49%
p-value	6.08e−8	4.74e−11	5.22e−9	2.04e−11	5.34e−11	6.22e−11	2.19e−7	1.38e−8	5.67e−7

- **VGCL** (Yang et al., 2023): This model generates contrastive views through variational graph reconstruction (i.e., adding noise via data augmentation) and designs a cluster-aware dual contrastive learning module (i.e., finding similar samples through clustering) to better exploit self-supervised signals at different scales.
- **GraphAU** (Yang et al., 2023): This model uses a GCN to achieve both alignment and uniformity. For alignment, it aligns higher-order neighborhood embeddings with the original embeddings, further strengthening representation alignment. For uniformity, this model uses the original embeddings.
- **AU^{+}** (Ouyang et al., 2024): This model builds on the original alignment and uniformity losses by using data augmentation to generate contrastive views for contrastive learning, thereby supplementing the missing self-supervised signals in AU-based models.

5.1.3. Hyperparameter settings

Our implementation of the SUAU model is based on PyTorch.[1] To ensure fair comparisons, we replicate the parameter ranges specified in a previous study for grid search and select optimal parameters for performance evaluation. The embedding layer size is set to 64, and the learning rate is fixed at $1e^{-3}$. The batch size for the Yelp2018 dataset is fixed at 2048 while on the Amazon-book and Tmall datasets are fixed at 4096 due to the speed. The number of graph convolution layers from $[1, 2, 3]$ for all GCN models. In SUAU, the number of similar item acquisitions is specified as 1 or 2. Due to significant variation in the weight γ of $\mathcal{L}_{uniform}$ across different datasets, we systematically search within intervals of 0.5, ranging from 1 to 10. The weight β of $\mathcal{L}_{similar}$ is set within the range $[1e^{-3}, 1e^{-2}, 1e^{-1}, 0.2, 0.5, 1]$.

Furthermore, inspire by previous research (Yu et al., 2022), we investigate the effect of excluding the embedding $\mathbf{E}^{(0)}$ in our experiments. Surprisingly, we find that omitting the embedding $\mathbf{E}^{(0)}$ leads to performance improvements on the Yelp2018 and Tmall datasets, but results in a significant performance decline on the Amazon-book dataset. This discrepancy may be attributed to the different datasets requiring alignment and uniformity of user and item representations from different layers. Therefore, we choose to skip the embedding $\mathbf{E}^{(0)}$ during training on the Yelp2018 and Tmall datasets.

5.2. Overall performance comparison

Tables 3 and 4 presents the performance results of the propose model and state-of-the-art models on the three datasets. These findings lead to the following observations:

- The experimental results demonstrate that SUAU outperforms the other models in both top-10 and top-20 scenarios. This success can be attributed to a uniqueness strategy proposed by SUAU, which addresses the uniformity bias by filtering out duplicate samples, thus achieving the optimal uniform distribution of user and item representations. Additionally, SUAU mitigates the sparsity issue in the recommendation dataset by identifying similar items to create item-similar item pairs as additional positive pairs, thereby enhancing the alignment and uniformity of user and item representations. Overall, SUAU effectively tackles the

[1] https://pytorch.org/

Table 5
Ablation study of SUAU, w/o S denotes the removal of item-similar item pairs for loss calculation, w/o U denotes the removal of uniqueness strategies in calculating uniformity loss.

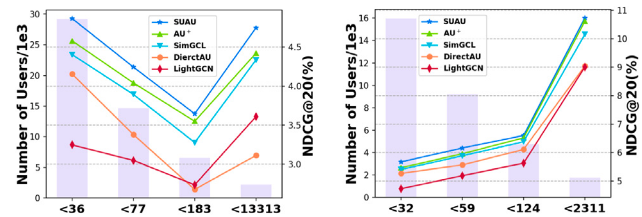
Method	Amazon-book		Yelp2018		Tmall	
	Recall@20	NDCG@20	Recall@20	NDCG@20	Recall@20	NDCG@20
DirectAU	0.0506(-)	0.0401(-)	0.0703(-)	0.0583(-)	0.0752(-)	0.0576(-)
w/o S	0.0559(+10.47%)	0.0442(+10.22%)	0.0741(+5.41%)	0.0614(+5.32%)	0.0889(+18.22%)	0.0679(+17.88%)
w/o U	0.0550(+8.70%)	0.0437(+8.98%)	0.0740(+5.26%)	0.0612(+4.97%)	0.0878(+16.76%)	0.0670(+16.32%)
SUAU	**0.0572(+13.04%)**	**0.0452(+12.72%)**	**0.0750(+6.69%)**	**0.0621(+6.52%)**	**0.0902(+19.95%)**	**0.0688(+19.44%)**

challenges of insufficient alignment and uniformity bias, which led to improved recommendation performance.

- Compare with the most prominent GCN-based model, LightGCN, SUAU exhibits 39.17%, 26.86%, and 17.37% enhancement in performance *w.r.t.* Recall@20 on the datasets Amazon-book, Tmall, and Yelp2018, respectively. This enhancement indicates SUAU's high accuracy in predicting user preferences. Compared with the most popular GCL-based model, SUAU also achieves superior performance, *e.g.*, about 11.07% enhancement in Recall@20 on Amazon-book dataset. Graph contrastive learning models, like SimGCL, primarily generate different contrastive views through data augmentation, employ InfoNCE as an auxiliary loss function for contrastive learning, and extract latent user preferences from these distinct views. However, these models often lack thorough exploration of critical properties, including alignment and uniformity. Due to this limited exploration, the final performance of these models does not surpass that of SUAU.
- The results of our experiments highlight the importance of investigating the alignment and uniformity of overall representations, which also underscores the general lack of research on these properties in existing collaborative filtering methods. By examining these properties from various perspectives, we propose SUAU, which enhances the alignment and uniformity of user and item representations through a similar items generation method and a uniqueness strategy, thereby improving the model's recommendation performance. Compared with AU-based models, which include the state-of-the-art AU^{+}, SUAU maintains a significant advantage, such as approximately 6.32% improvement in Recall@20 on the Amazon-book dataset. Furthermore, as illustrated in Fig. 8, SUAU not only achieves optimal performance but also demonstrates optimal alignment and uniformity. These results provide ample evidence of the effectiveness of SUAU.

5.3. Model ablation study

To demonstrate the effectiveness of our proposed model, we conduct ablation studies on three datasets. We evaluate the performance of SUAU and its variants in a top-20 scenario, and the results are detailed in Table 5. The 'w/o S' variant indicates that the loss from item-similar item pairs is not considered, focusing solely on computing alignment and uniformity losses through user-item pairs. The table clearly shows a significant performance degradation compared to SUAU, supporting the effectiveness of supplementing additional item-similar item pairs to alleviate interaction sparsity, as proposed in our model. Similarly, the 'w/o U' variant, which removes the uniqueness strategy that retains item-similar item pairs for loss calculation, achieves only sub-optimal performance on all datasets, emphasizing the necessity of the uniqueness strategy in uniformity loss calculations to avoid uniformity bias caused by repeated samples. Finally, SUAU shows the most significant performance improvement compared to the two variants, which demonstrates the effectiveness of combining the similar items generation method with the uniqueness strategy. The comparison with the DirectAU model and the analyses conducted, along with the results from other variants, confirm that SUAU enables better optimization of the alignment and uniformity of overall representations, and underscore the importance of studying desired properties to further enhance recommendation performance.



(a) Amazon-book (b) Yelp2018

Fig. 6. Performance comparison *w.r.t.* NDCG@20 of SUAU and other models for different user groups sparsity levels on Amazon-book and Yelp2018 Datasets.

5.4. Model sparsity study

Interaction sparsity has long been a challenge for recommendation datasets. To demonstrate the effectiveness of our propose model in mitigating sparse data issues, we categorize the dataset into four groups based on the number of user interactions, each representing a different sparsity level (Wu et al., 2021; Zhang et al., 2023b). Taking the Amazon-book dataset as an example, the number of users with fewer than 36, 77, 183, and 13,313 interactions accounted for 55.49%, 27.92%, 12.46%, and 4.13% of the total number of users, respectively. To ensure the stability of the experimental results, all training data are used for training, and in the testing phase, we ensure that the number of interactions in each user group is the same before conducting the test. Finally, we present the experimental results *w.r.t.* NDCG@20 for each model in Fig. 6.

As shown in the figure, SUAU consistently outperforms all other models across different sparsity levels. For the sparsest user group, SUAU improves upon state-of-the-art models by 6.23%, 3.76% on the Amazon-book and Yelp2018 datasets, respectively. Furthermore, we observe a decline in recommendation performance for the first three user groups in the Amazon-book dataset, possibly due to the large number of noisy samples in the '< 77' and '< 183' user groups, which could impact the model's learning of user preferences. Nevertheless, our model achieves optimal performance. These experimental results conclusively demonstrate that SUAU effectively addresses interaction sparsity, which significantly enhances the quality of recommendations.

5.5. Alignment and uniformity study

Previous studies have indicated that optimizing the alignment and uniformity can enhance the model's performance to some extent (Wang & Isola, 2020; Yu et al., 2022). To verify whether our propose SUAU can enhance the alignment and uniformity of user and item representations, we present the evolution of the alignment and uniformity for all AU-based models on each of the two datasets in Fig. 7. On the Amazon-book dataset, we observe that SUAU consistently achieves optimal alignment after several iterations. This is primarily due to the similar items generation method employed in SUAU, which involves generating item-similar item pairs for training to mitigate interaction sparsity and ensure a more cohesive connection between positive pairs. Similarly, our observations from other dataset support this finding. In addition, as shown in Fig. 7(b), SUAU achieves almost optimal uniformity across all

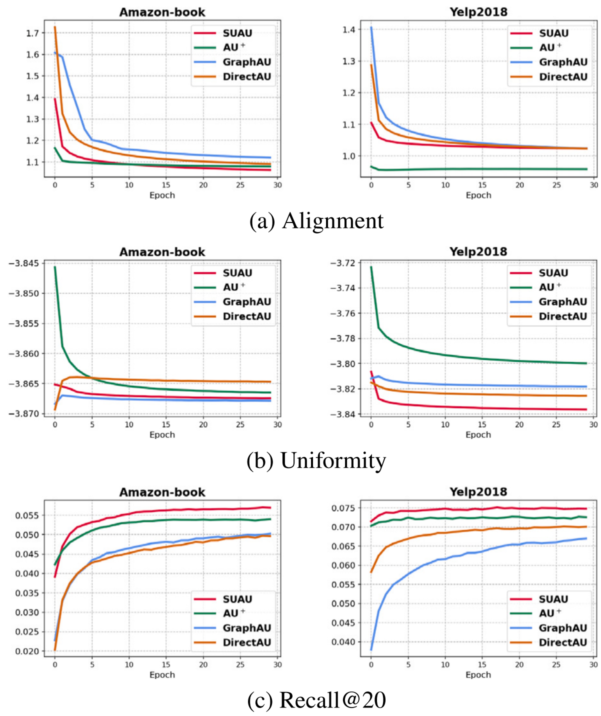


Fig. 7. Evolution of Alignment, Uniformity, and Recall@20 in Our SUAU and AU-based models during training on Amazon-book and Yelp2018 Datasets.

datasets, which indicates the effectiveness of our uniqueness strategy in ensuring a better uniform distribution of user and item representations.

However, an anomaly is observed in the case of the Yelp2018 dataset, where AU^+ exhibits the poorest uniformity among all AU-based models, despite achieving optimal alignment. Based on this finding, we argue that neither superior alignment nor uniformity alone guarantees improved performance. Performance enhancement is only achievable when both desired properties are relatively optimal. Fig. 8 effectively illustrates this point with the AU-based models (DirectAU, AU+, and SUAU), which demostrate significant performance improvements when alignment and uniformity are optimized simultaneously. Therefore, it is essential to consider methods to enhance both alignment and uniformity of user and item representations in the recommender systems. In our propose SUAU, owing to the introduction of a similar items generation method and a uniqueness strategy, both alignment and uniformity are enhanced, which effectively alleviates interaction sparsity, and significantly improves the overall performance of the model.

5.6. Hyperparameter study

In this subsection, we explore the influence of two crucial hyperparameters on the recommendation performance of the SUAU model: γ, which governs the weight of the uniformity loss, and β, which governs the weight of the item-similar item pairs loss.

Fig. 9 depicts the evaluation results of the recommended performance *w.r.t.* Recall@20 and NDCG@20. Regarding γ, as shown in (a) and (b), we observe an overall trend of initially increasing and then decreasing performance. However, the optimal weight for uniformity loss varies across datasets due to their different distributions. To determine the optimal weights, we first perform a brief search for each dataset with γ values ranging from 1.0 to 10.0 in steps of 0.5. After identifying the effective range for each dataset (*e.g.*, 5.0–10.0 for the Amazon-book dataset), we then conduct a detailed experiment to find the precise optimal γ value. In addition, our analysis reveals that

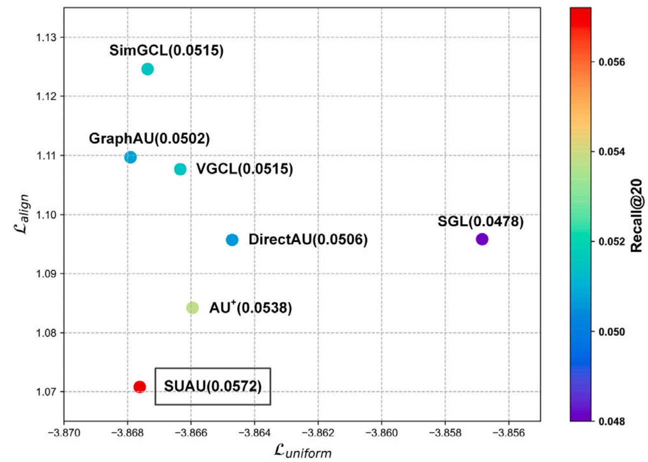


Fig. 8. $\mathcal{L}_{align}$ and $\mathcal{L}_{uniform}$ of the state-of-the-art CF models on Amazon-book dataset (Yu et al., 2022). Scatter color corresponds to the value of the color bar *w.r.t.* Recall@20.

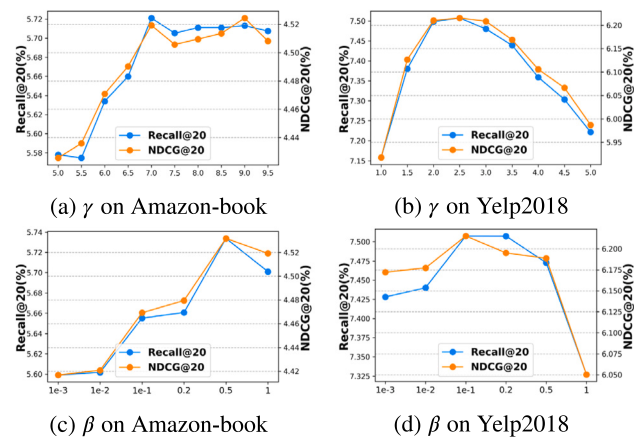


Fig. 9. Results of parameters sensitivity (γ and β) on Amazon-book and Yelp2018 Datasets..

datasets with a higher number of interactions typically require higher uniformity loss weights. This necessity arises because an increase in the number of interactions necessitates a larger uniformity loss to maintain the overall model's uniformity. Considering β, our search within the $[1e^{-3}, 1e^{-2}, 1e^{-1}, 0.2, 0.5, 1]$ showed a similar performance trend, initially rising and then falling. The experimental results suggest that selecting an optimal β (e.g., β=0.5) effectively enhances the SUAU model's alignment and uniformity by capturing the representation of sparse interaction items.

To better implement SUAU, we provide the hyperparameter selections for SUAU on the Amazon-book, Yelp2018, and Tmall datasets. For Amazon-book, γ, β, and the number of GCN layers are set to 7.0, 0.5, and 1, respectively. For Yelp2018, γ, β, and the number of GCN layers are set to 2.5, 0.1, and 2. For Tmall, γ, β, and the number of GCN layers are set to 4.5, 1.0, and 1. It is worth noting that the original embeddings should be removed before running on the Yelp2018 and Tmall datasets.

6. Related work

6.1. Collaborative filtering

Collaborative filtering (CF) approaches are pivotal in recommender systems (Schafer et al., 2007), aiming to personalize recommendations by learning user preferences from known user-item interactions. Various previous approaches have often employed matrix factorization to map user/item IDs to an embedding layer for preference learning (He et al., 2017; Rendle et al., 2009). More recently, the advent of Graph Neural Networks (GNNs) (Thanh et al., 2023; Wu et al., 2023) has led to their adoption in various recommendation scenarios. NGCF (Wang et al., 2019) model introduced graph convolutional networks into CF, leveraging high-order neighborhood information from the adjacency matrix to enhance user preference learning. SGCN (Wu et al., 2019) model further advanced this by eliminating nonlinearities and consolidating multiple weight matrices. Additionally, LightGCN (He et al., 2020) model simplifies the NGCF model by proposing a simplified graph convolution method that focuses on neighborhood aggregation.

Besides, with the wide application of contrastive learning in fields such as language processing (Cheng et al., 2023; Lai et al., 2024), computer vision (Ge et al., 2023; Zhang et al., 2024), etc., researchers introduced contrastive learning into CF to alleviate the issue of sparse supervision. This method enhances the self-supervised signal by contrasting the augmented views. In particular, SGL (Wu et al., 2021) was the most popular one because it introduced contrastive learning into CF, generating diverse contrast views via data augmentation techniques. SimGCL (Yu et al., 2022) incorporates randomly generated Gaussian noise into each embedding layer to generate different contrastive views. AutoCF (Xia et al., 2023) employs a masked graph autoencoder to effectively aggregate global information via masked subgraph structure reconstruction. VGCL (Yang et al., 2023) combines variational graph reconstruction with contrastive learning to produce, producing node-specific Gaussian distributions for view generation and to ensures graph consistency at node and cluster levels. Despite their effectiveness, Graph-based models generally do not investigate the desired properties of the data. In this study, we deeply investigate two desired properties, alignment and uniformity, to enhance user and item representations.

6.2. Alignment and uniformity

With the widespread success of representation learning (Jing et al., 2024; Ju et al., 2024), researchers are increasingly focusing on its essential properties across various domains. Recently, two desired properties, alignment and uniformity, have been proposed (Wang & Isola, 2020). Alignment attempts to reduce the distance between positive pairs, while uniformity ensures an even distribution on the unit hyperspheres. Given the effectiveness of the desired properties, an increasing number of researchers are directing their attention to them (Tan et al., 2023; Wang et al., 2023; Zhang et al., 2023). Wang et al. (2023) propose a method for test time adaptation that includes self-distillation for ensuring feature uniformity and spatial local clustering for achieving feature alignment. Gao et al. (2021) present SimCSE, enhancing sentence embeddings through contrastive learning that improves alignment of positive pairs and promotes uniformity in the embedding space. Tan et al. (2023) used alignment loss and uniformity loss to align target sequence embeddings and drug map embeddings aligned while avoiding collapse.

In recommender systems, DirectAU (Wang et al., 2022) introduces the concept of alignment and uniformity as loss functions in collaborative filtering. uCTRL (Lee et al., 2023) proposes an unbiased alignment function to optimize the alignment and uniformity of CF methods and designs a novel IPW estimation method that eliminates user and item bias. To mitigate the complexity of computing high-order neighborhood alignments, GraphAU (Yang et al., 2023) uses layer-wise alignment to integrate alignment loss. AU^+ (Ouyang et al., 2024) extends the original alignment loss and uniformity loss by introducing self-supervised signals into the model for better capturing user preferences via contrastive learning. Despite the effectiveness of these models, theoretical analysis shows that they still have significant flaws. Therefore, we propose a uniqueness strategy on SUAU and generate additional positive pairs to optimize the alignment and uniformity of model representations for high-quality recommendations.

7. Conclusion

In this study, we investigated the concepts of alignment and uniformity in recommender systems, revealing the serious problems of insufficient alignment and uniformity bias in existing AU-based models. To address the persistent insufficient alignment problem of recommender systems, we analyzed the user/item interaction statistics results and found it is difficult to capture information from sparse interaction items. Furthermore, our theoretical analysis exposed a uniformity bias that arose when computing uniformity loss under batch conditions. To tackle these challenges, a model is proposed called SUAU, which filters duplicate user and item representations via a uniqueness strategy thereby effectively addressing the issue of uniformity bias of overall representations. Additionally, SUAU presented a similar items generation method to address insufficient alignment. The alignment loss and uniformity loss are calculated by identifying similar items based on the interaction behavior of the original items' related users, treating them as additional positive pairs. The experimental results demonstrate that SUAU effectively mitigated interaction sparsity while ensuring alignment and uniformity. Compared with state-of-the-art CF models, SUAU achieved superior recommendation performance.

In the future, we plan to explore high-quality similar items to enhance overall representation alignment. We also plan to delve into the underlying factors of uniformity to improve the efficiency of calculating uniformity loss and boost the recommendation performance of CF methods.

CRediT authorship contribution statement

Lei Sang: Conceptualization, Methodology, Validation, Writing – original draft, Writing – review & editing. **Yu Zhang:** Conceptualization, Methodology, Validation, Writing – original draft, Writing – review & editing. **Yi Zhang:** Methodology, Validation, Writing – review & editing, Visualization. **Honghao Li:** Validation, Software, Writing – Review. **Yiwen Zhang:** Supervision, Writing – review, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

I have shared the link to our code on Github.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. 62272001 and No. 62206002), and the Anhui Provincial Natural Science Foundation (2208085QF195), Xunfei Zhiyuan Digital Transformation Innovation Research Special for Universities (2023ZY001).

References

Cheng, Y., Wei, F., Bao, J., Chen, D., & Zhang, W. (2023). Cico: Domain-aware sign language retrieval via cross-lingual contrastive learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 19016–19026).

Gao, T., Yao, X., & Chen, D. (2021). SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 conference on empirical methods in natural language processing* (pp. 6894–6910). http://dx.doi.org/10.48550/arXiv.2104.08821.

Ge, S., Mishra, S., Kornblith, S., Li, C.-L., & Jacobs, D. (2023). Hyperbolic Contrastive Learning for Visual Representations beyond Objects. In *2023 IEEE/CVF conference on computer vision and pattern recognition (CVPR)* (pp. 6840–6849).

He, X., Deng, K., Wang, X., Li, Y., Zhang, Y., & Wang, M. (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* (pp. 639–648). http://dx.doi.org/10.1145/3397271.3401063.

He, X., Liao, L., Zhang, H., Nie, L., Hu, X., & Chua, T.-S. (2017). Neural Collaborative Filtering. In *Proceedings of the 26th international conference on world wide web* (pp. 173–182). http://dx.doi.org/10.1145/3038912.3052569.

Jiang, Y., Huang, C., & Huang, L. (2023). Adaptive Graph Contrastive Learning for Recommendation. In *Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining* (pp. 4252–4261). http://dx.doi.org/10.1145/3580305.3599768.

Jing, B., Yan, Y., Ding, K., Park, C., Zhu, Y., Liu, H., & Tong, H. (2024). Sterling: Synergistic representation learning on bipartite graphs. *Proceedings of the AAAI Conference on Artificial Intelligence*, *38*(12), 12976–12984. http://dx.doi.org/10.1609/aaai.v38i12.29195.

Ju, W., Fang, Z., Gu, Y., Liu, Z., Long, Q., Qiao, Z., Qin, Y., Shen, J., Sun, F., Xiao, Z., Yang, J., Yuan, J., Zhao, Y., Wang, Y., Luo, X., & Zhang, M. (2024). A comprehensive survey on deep graph representation learning. *Neural Networks*, *173*, Article 106207. http://dx.doi.org/10.1016/j.neunet.2024.106207.

Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, *42*(8), 30–37. http://dx.doi.org/10.1109/MC.2009.263.

Lai, C., Song, S., Meng, S., Li, J., Yan, S., & Hu, G. (2024). Towards More Faithful Natural Language Explanation Using Multi-Level Contrastive Learning in VQA. *Proceedings of the AAAI Conference on Artificial Intelligence*, *38*(3), 2849–2857. http://dx.doi.org/10.1609/aaai.v38i3.28065.

Lee, J.-w., Park, S., Yoon, M., & Lee, J. (2023). uCTRL: Unbiased contrastive representation learning via alignment and uniformity for collaborative filtering. In *Proceedings of the 46th international ACM SIGIR conference on research and development in information retrieval* (pp. 2456–2460). http://dx.doi.org/10.1145/3539618.3592076.

Li, M., Zhang, T., Chen, Y., & Smola, A. J. (2014). Efficient mini-batch training for stochastic optimization. In *Proceedings of the 20th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 661–670). New York, NY, USA: Association for Computing Machinery, http://dx.doi.org/10.1145/2623330.2623612.

Lin, Z., Tian, C., Hou, Y., & Zhao, W. X. (2022). Improving graph collaborative filtering with neighborhood-enriched contrastive learning. In *Proceedings of the ACM web conference 2022* (pp. 2320–2329). New York, NY, USA: Association for Computing Machinery, http://dx.doi.org/10.1145/3485447.3512104.

Masters, D., & Luschi, C. (2018). Revisiting small batch training for deep neural networks. *Computing Research Repository*, URL http://arxiv.org/abs/1804.07612.

Natarajan, S., Vairavasundaram, S., Natarajan, S., & Gandomi, A. H. (2020). Resolving data sparsity and cold start problem in collaborative filtering recommender system using linked open data. *Expert Systems with Applications*, *149*, Article 113248. http://dx.doi.org/10.1016/j.eswa.2020.113248, URL https://www.sciencedirect.com/science/article/pii/S0957417420300737.

Nguyen Thanh, T., Quach, N. D. K., Nguyen, T. T., Huynh, T. T., Vu, V. H., Nguyen, P. L., Jo, J., & Nguyen, Q. V. H. (2023). Poisoning GNN-based Recommender Systems with Generative Surrogate-based Attacks. *ACM Transactions on Information Systems*, *41*(3), 1–24. http://dx.doi.org/10.1145/3567420.

Ouyang, Z., Zhang, C., Hou, S., Zhang, C., & Ye, Y. (2024). How to improve representation alignment and uniformity in graph-based collaborative filtering? *Proc. Int. AAAI Conf. Web and Soc. Media*, *18*(1), 1148–1159. http://dx.doi.org/10.1609/icwsm.v18i1.31379, URL https://ojs.aaai.org/index.php/ICWSM/article/view/31379.

Ren, X., Xia, L., Zhao, J., Yin, D., & Huang, C. (2023). Disentangled contrastive collaborative filtering. In *Proceedings of the 46th international ACM SIGIR conference on research and development in information retrieval* (pp. 1137–1146). New York, NY, USA: Association for Computing Machinery, http://dx.doi.org/10.1145/3539618.3591665.

Rendle, S., Freudenthaler, C., Gantner, Z., & Schmidt-Thieme, L. (2009). BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence* (pp. 452–461). http://dx.doi.org/10.48550/arXiv.1205.2618.

Schafer, J. B., Frankowski, D., Herlocker, J., & Sen, S. (2007). Collaborative filtering recommender systems. In P. Brusilovsky, A. Kobsa, & W. Nejdl (Eds.), *The adaptive web: methods and strategies of web personalization* (pp. 291–324). Springer Berlin Heidelberg, http://dx.doi.org/10.1007/978-3-540-72079-9_9.

Sun, P., Wu, L., Zhang, K., Chen, X., & Wang, M. (2024). Neighborhood-enhanced supervised contrastive learning for collaborative filtering. *IEEE Transactions on Knowledge and Data Engineering*, *36*(5), 2069–2081. http://dx.doi.org/10.1109/TKDE.2023.3317068.

Tan, C., Gao, Z., & Li, S. Z. (2023). Target-Aware Molecular Graph Generation. In *Machine learning and knowledge discovery in databases: applied data science and demo track* (pp. 410–427). http://dx.doi.org/10.1007/978-3-031-43427-3_25.

Wang, X., He, X., Wang, M., Feng, F., & Chua, T.-S. (2019). Neural Graph Collaborative Filtering. In *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval* (pp. 165–174). http://dx.doi.org/10.1145/3331184.3331267.

Wang, T., & Isola, P. (2020). Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In H. D. III, & A. Singh (Eds.), *Proceedings of machine learning research*: *119*, *Proceedings of the 37th international conference on machine learning* (pp. 9929–9939). PMLR, URL https://proceedings.mlr.press/v119/wang20k.html.

Wang, C., Yu, Y., Ma, W., Zhang, M., Chen, C., Liu, Y., & Ma, S. (2022). Towards Representation Alignment and Uniformity in Collaborative Filtering. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining* (pp. 1816–1825). http://dx.doi.org/10.1145/3534678.3539253.

Wang, S., Zhang, D., Yan, Z., Zhang, J., & Li, R. (2023). Feature Alignment and Uniformity for Test Time Adaptation. In *2023 IEEE/CVF conference on computer vision and pattern recognition (CVPR)* (pp. 20050–20060).

Wei, Y., Xu, Y., Zhu, L., Ma, J., & Peng, C. (2024). Multi-level cross-modal contrastive learning for review-aware recommendation. *Expert Systems with Applications*, *247*, Article 123341. http://dx.doi.org/10.1016/j.eswa.2024.123341.

Wu, L., He, X., Wang, X., Zhang, K., & Wang, M. (2022). A Survey on Accuracy-oriented Neural Recommendation: From Collaborative Filtering to Information-rich Recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 1. http://dx.doi.org/10.1109/TKDE.2022.3145690.

Wu, F., Souza, A., Zhang, T., Fifty, C., Yu, T., & Weinberger, K. (2019). Simplifying Graph Convolutional Networks. In K. Chaudhuri, & R. Salakhutdinov (Eds.), *Proceedings of machine learning research*: *97*, *Proceedings of the 36th international conference on machine learning* (pp. 6861–6871).

Wu, S., Sun, F., Zhang, W., Xie, X., & Cui, B. (2022). Graph neural networks in recommender systems: A survey. *ACM Computing Surveys*, *55*(5), http://dx.doi.org/10.1145/3535101.

Wu, S., Sun, F., Zhang, W., Xie, X., & Cui, B. (2023). Graph Neural Networks in Recommender Systems: A Survey. *ACM Computing Surveys*, *55*(5), 1–37. http://dx.doi.org/10.1145/3535101.

Wu, J., Wang, X., Feng, F., He, X., Chen, L., Lian, J., & Xie, X. (2021). Self-supervised Graph Learning for Recommendation. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval* (pp. 726–735). http://dx.doi.org/10.1145/3404835.3462862.

Xia, L., Huang, C., Huang, C., Lin, K., Yu, T., & Kao, B. (2023). Automated Self-Supervised Learning for Recommendation. In *Proceedings of the ACM web conference 2023* (pp. 992–1002). http://dx.doi.org/10.1145/3543507.3583336.

Xiao, Y., Huang, J., & Yang, J. (2024). Tfcsrec: Time–frequency consistency based contrastive learning for sequential recommendation. *Expert Systems with Applications*, *245*, Article 123118. http://dx.doi.org/10.1016/j.eswa.2023.123118.

Xu, B., Li, X., & Fan, Z.-P. (2024). Selection and visiting sequence of daily attractions: Multi-day travel itinerary recommendation based on multi-source online data. *Expert Systems with Applications*, *250*, Article 123895. http://dx.doi.org/10.1016/j.eswa.2024.123895.

Xu, H., Yang, B., & Liu, X. (2024). Reverse-graph enhanced graph neural networks for session-based recommendation. *Expert Systems with Applications*, *245*, Article 122995. http://dx.doi.org/10.1016/j.eswa.2023.122995.

Yang, L., Liu, Z., Wang, C., Yang, M., Liu, X., Ma, J., & Yu, P. S. (2023). Graph-based Alignment and Uniformity for Recommendation. In *Proceedings of the 32nd ACM international conference on information and knowledge management* (pp. 4395–4399). http://dx.doi.org/10.1145/3583780.3615185.

Yang, Y., Wu, Z., Wu, L., Zhang, K., Hong, R., Zhang, Z., Zhou, J., & Wang, M. (2023). Generative-Contrastive Graph Learning for Recommendation. In *Proceedings of the 46th international ACM SIGIR conference on research and development in information retrieval* (pp. 1117–1126). http://dx.doi.org/10.1145/3539618.3591691.

Yu, J., Yin, H., Xia, X., Chen, T., Cui, L., & Nguyen, Q. V. H. (2022). Are Graph Augmentations Necessary?: Simple Graph Contrastive Learning for Recommendation. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval* (pp. 1294–1303). http://dx.doi.org/10.1145/3477495.3531937.

Yu, J., Yin, H., Xia, X., Chen, T., Li, J., & Huang, Z. (2023). Self-Supervised Learning for Recommender Systems: A Survey. *IEEE Transactions on Knowledge and Data Engineering*, 1–20. http://dx.doi.org/10.1109/TKDE.2023.3282907.

Zhang, T., He, S., Dai, T., Wang, Z., Chen, B., & Xia, S.-T. (2024). Vision-language pre-training with object contrastive learning for 3D scene understanding. *Proceedings of the AAAI Conference on Artificial Intelligence*, *38*(7), 7296–7304. http://dx.doi.org/10.1609/aaai.v38i7.28559.

Zhang, D., Li, C., Li, H., Huang, W., Huang, L., & Zhang, J. (2023). Rethinking alignment and uniformity in unsupervised image semantic segmentation. *Proceedings of the AAAI Conference on Artificial Intelligence*, *37*, 11192–11200. http://dx.doi.org/10.1609/aaai.v37i9.26325.

Zhang, Y., Zhang, Y., Yan, D., Deng, S., & Yang, Y. (2023). Revisiting Graph-based Recommender Systems from the Perspective of Variational Auto-Encoder. *ACM Transactions on Information Systems*, *41*(3), 1–28. http://dx.doi.org/10.1145/3573385.

Zhang, Y., Zhang, Y., Yan, D., He, Q., & Yang, Y. (2024). NIE-gcn: Neighbor item embedding-aware graph convolutional network for recommendation. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *54*(5), 2810–2821. http://dx.doi.org/10.1109/TSMC.2024.3350658.

Zhang, Y., Zhang, Y., Zhao, Y., Deng, S., & Yang, Y. (2024). Dual variational graph reconstruction learning for social recommendation. *IEEE Transactions on Knowledge and Data Engineering*, http://dx.doi.org/10.1109/TKDE.2024.3386895.