Revisiting Alignment and Uniformity for Recommendation via Discrimination and Reliable Assessment

Xinzhe Jiang, Lei Sang, Shun Lian, Yi Zhang, and Yiwen Zhang

***Abstract*—Utilizing alignment and uniformity for recommendation has shown success in considering similarities between users and items. Despite this effectiveness, we argue that they suffer from two limitations: (1) alignment loss as a measure of model quality fluctuates significantly during adjustment, leading to inaccurate assessments. (2) Current methods ignore potential connections for user-user and item-item, resulting in incomplete understanding of user preferences and item characteristics.To address these issues, we propose using the trace of user and item correlation matrices as a new assessment metric to replace traditional alignment for the first time. This design reduces the impact of hyperparameters on model assessment, ensuring that trace and model quality are optimized simultaneously, thereby improving recommendation accuracy. Based on this, we introduce a new model *Alignment and Uniformity with Discrimination*, which additionally considers the similarities for user-user and item-item. Specifically, DiscrimAU calculates the Euclidean distance between the user (item) relevance matrix and its fully aligned matrix, distinguishing the relevance levels among different users (items). This process ensures that highly relevant users and items are more closely aligned, capturing more information. Extensive experiments on three datasets show that the proposed model achieves a maximum improvement of 6.29%, clearly demonstrating its effectiveness.**

***Index Terms*—Recommendation system, self-supervised learning, alignment and uniformity, graph neural network.**

I. Introduction

With the Internet of Everything continues to grow, electronic shopping has become a dominant form of consumer behavior. This shift allows consumers to browse and purchase a wide range of products from the comfort of their homes, saving time and effort. However, the vast choices on e-commerce platforms can lead to decision fatigue. Shoppers often struggle to find products that meet their needs. As a result, potential purchases may be missed [1]. To improve the online shopping experience, effective product recommendations are essential [2]. These recommendations guide consumers toward suitable options. They enhance customer satisfaction and help conserve resources by reducing unnecessary shipping.

In this context, collaborative filtering (CF) [3] captures and analyzes historical user-item interactions to build personalized recommendation [4]. So far, various types of CF-based methods have been proposed to project users and items into a latent embeddings [5], such as matrix factorization [6], [7], autoencoder [8] and GCN-based methods [9]. However, traditional CF models face the challenge of users clicking on irrelevant products due to over-recommendation of popular items [10], leading to inaccurate information being learned directly from the user-item interaction [11]. To address this issue, self-supervised learning (SSL) extracts the general features from unlabeled data [12]. SSL-based methods incorporate additional loss functions alongside the BPR loss [13], such as InfoNCE loss [14], cosine contrastive loss [15], which have been shown to bring more robust improvements [16]. Meanwhile, the new SSL paradigm DirectAU [17] employs alignment and uniformity on the hypersphere for CF. The alignment loss directly highlights the similarity between users and their interacted items, and the uniformity loss ensures that users and items are evenly distributed across a hypersphere, thus maintaining their distinguishability [18]. These losses ensure that even non-popular items gain can be noticed and recommended to users, enhancing the model robustness.

Despite the ability of alignment and uniformity to optimize user-item distributions, these methods still pose several drawbacks.

On the one hand: Traditional AU-based methods such as DirectAU [17] and GraphAU [19] capture model performance by adjusting the degree of alignment and uniformity. During this process, enhancing alignment increases the difficulty of maintaining uniformity, and vice versa. We need to manage the trade-off between alignment and uniformity. Additionally, alignment loss, which is calculated as the Euclidean distance between users and items, shows significant fluctuation during adjustment. Directly using it as an assessment metric leads to a biased trade-off between alignment and uniformity, resulting in inaccurate model performance capture.

On the other hand: Most research on alignment and uniformity overlooks the direct relationships for similar user-user and item-item, focusing only on the connections between users and items. This will result in biased distributions of users and items, which limits the modeling of learnable information. As shown in Fig. 1, by constraining the alignment and uniformity,

Received 1 November 2024; revised 12 December 2024; accepted 3 January 2025. Date of publication 8 January 2025; date of current version 12 June 2025. This work was supported in part by the National Natural Science Foundation of China under Grant 62272001 and Grant 62206002, and in part by the Xunfei Zhiyuan Digital Transformation Innovation Research Special for Universities under Grant 2023ZY001. *(Corresponding author: Yiwen Zhang.)*

The authors are with the School of Computer Science and Technology, Anhui University, Hefei 230601, Anhui, China (e-mail: e23301307@stu.ahu.edu.cn; sanglei@ahu.edu.cn; lianshun@ustc.edu; zhangyi.ahu@gmail.com; zhangyiwen@ahu.edu.cn).

Digital Object Identifier 10.1109/TCE.2025.3527007

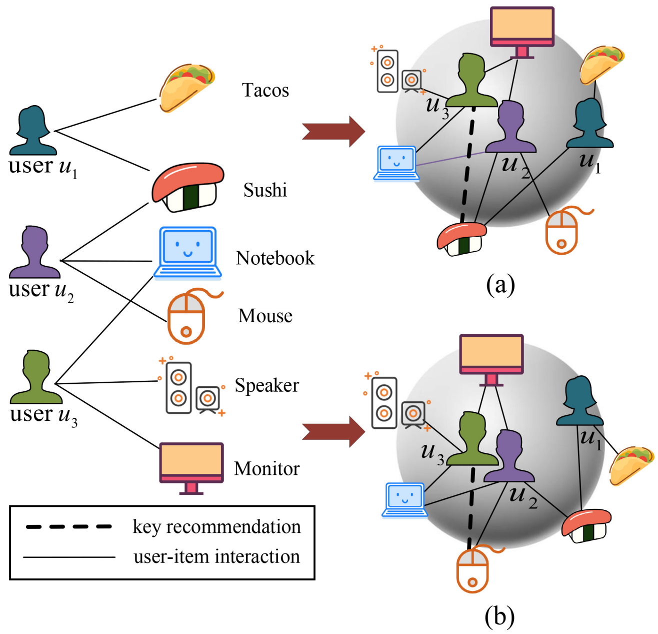


Fig. 1. Illustration of user-item interaction on the hypersphere in recommendation scenarios. The closer the entities are, the greater their similarity. (a) denotes the defective distribution; (b) denotes the excellent distribution.

highly relevant users and items are brought closer together and distributed uniformly on the hypersphere. According to CF principles, recommender systems suggest new items that are most relevant or closest to the user [20], [21]. In Fig. 1(a), the monitor is the closest non-interacted item for user u_2, suggesting it should be recommended for user u_2. However, as shown by the bold dashed lines, the system with recommend *Sushi* for user u_3 instead of the mouse, as *Sushi* is closer for user u_3 than the mouse. In reality, based on historical interactions of user u_3 with speaker and monitor, user u_3 might likely prefer a recommendation for the mouse over *Sushi*. To address this issue, we hope that the distribution of users and items on the hypersphere is as shown in Fig. 1 (b), where the mouse is closer to the more similar notebook and speaker, and highly relevant user u_2 and user u_3. In this case, as indicated by the bold dashed lines, we can correctly recommend the mouse for user u_3 according to CF principles.

Based on these, we encounter two primary challenges:

- How to achieve accurate and reliable model quality assessment?
- How to use direct user-user and item-item relationships to optimize distributions on the hypersphere?

To address the above issues, we propose the **Alignment and Uniformity with Discrimination (DiscrimAU)** model. Facing the first issue, we propose assessing alignment using the **trace** (Tr) of the user and item correlation matrices. The trace optimizes traditional model assessment methods, helping the model accurately and efficiently capture its optimal state. Based on this, we argue that the trace is an excellent method for assessing model quality. Facing the second issue, DiscrimAU normalizes user and item embeddings and further introduces the discriminative loss among users (items), building on direct alignment and uniformity. The discriminative loss helps to distinguish users and items by their relevance, capturing implicit relationships between similar user-user and item-item. Extensive experiments are conducted on three real-world datasets, showing that DiscrimAU produces significant improvements compared to state-of-the-art CF methods.

The contributions of our paper are summarized as follows:

- To the best of our knowledge, this is the first work to propose a new reliable assessment method based on the trace of the user and item correlation matrices. It can be an ideal alternative to alignment loss, improving the accuracy of model quality assessment.
- We propose the *Alignment and Uniformity with Discrimination (DiscrimAU)* model. By incorporating discriminative loss to capture user-user and item-item relationships, our model enhances alignment in recommender system.
- Extensive experiments on three public datasets show that the proposed DiscrimAU outperforms current leading CF methods, achieving superior alignment and significant performance gains.

II. Preliminaries

A. Graph-Based Collaborative Filtering

Given a matrix $\mathbf{R}$, which size is $|\mathcal{U}| \times |\mathcal{I}|$. We let $U = \{u_1, u_2, \ldots, u_{|U|}\}$, $I = \{i_1, i_2, \ldots, i_{|I|}\}$. We use the matrix $\mathbf{R} \in \mathbb{R}_{m \times n}$ to describe user-item interactions, where each element $r_{ij} = 1$ if the user i interacts with the item j, otherwise $r_{ij} = 0$. Collaborative filtering assumes that similar users are likely to have similar interests. Based on this principle, it ranks all items to predict unobserved user-item interactions. To model historical interactions, a bipartite graph of user-item interactions is constructed: $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, where $\mathcal{N} = \mathcal{U} \cup \mathcal{I}$, $\mathcal{E}$ denotes the edge $(u, i) \in E$ in the case of $r_{ij} = 1$. The final score is calculated by the inner product between user u and item i: $\hat{y}_{ui} = \mathbf{e}_u^\top \mathbf{e}_i$. The BPR loss is adopted to optimize model parameters:

$$\mathcal{L}_{\text{BPR}} = \sum_{(u,i,j) \in O} -\log \sigma\left(\hat{y}_{ui} - \hat{y}_{uj}\right), \tag{1}$$

where $O = \{(u, i, j) | (u, i) \in O^+, (u, j) \in O^-\}$ is the training data, O^+ means positive pairs and O^- means negative pairs. This loss function is designed to optimize prediction scores so that positive items have higher scores than randomly selected negative items.

B. Alignment and Uniformity

Recently, Wang and Isola [18] have clarified two key properties of contrastive loss: alignment of positive pairs and uniformity of the induced feature distribution. DirectAU [17] applies these to the field of recommender system, optimizing user and item embeddings. The alignment loss is calculated as follows:

$$l_{\text{align}} \triangleq \underset{(x,x^+) \sim pos}{\mathbb{E}} \left\| f(x) - f\left(x^+\right) \right\|^2, \tag{2}$$

where $(x, x^+) \sim pos$ represents the distribution of a given positive pair, $f(\cdot)$ is the l_2 normalized representations.

The alignment loss is defined as the expected distance between positive pairs of embeddings. Its purpose is to emphasize the similarity between users and items, bringing similar user-item pairs as close as possible. The uniformity loss is calculated as follows:

$$l_{\text{uniform}} \triangleq \log \underset{x,y\sim p_{data}}{\mathbb{E}} e^{-2\|f(x)-f(y)\|^2}, \tag{3}$$

where $x, y \sim p_{data}$ represents the distribution of data. This training objective ensures that users and items should be evenly distributed on the hypersphere.

In summary, these two metrics ensure that positive instances remain close to one another, while random instances are dispersed across the hypersphere. Finally, The loss function of DirectAU is set as:

$$\mathcal{L}_{\text{DirectAU}} = l_{\text{align}} + \gamma l_{\text{uniform}}. \tag{4}$$

The weight γ controls the desired degree of uniformity. DirectAU achieves superior recommendation performance by directly optimizing for alignment and uniformity, replacing traditional BPR loss. Although DirectAU analyzes the roles of alignment and uniformity in recommender systems and directly optimizes for these two metrics, solely relying on these optimizations can lead to insufficient alignment between users and items. This can result in incomplete information learning, thereby impacting the overall recommendation performance.

III. Methodology

A. Alignment Analysis

In this section, based on the issues identified in DirectAU, we investigate how hyperparameter variations impact the outcomes of the alignment loss. We find that directly using alignment loss might pose difficulties for analysis. DirectAU uses BPR loss as an illustrative example, demonstrating that with a perfectly aligned and uniformly distributed encoder, BPR loss can be minimized. However, using the value of alignment loss directly as a metric can be affected by multiple factors, making it an unreliable measurement criterion. Determining a direct measure for evaluating whether a model achieves superior alignment is challenging. To tackle the above issue, we introduce a new criterion that aims to serve as a more effective measure of the model's alignment state, potentially replacing the traditional alignment loss.

Theorem 1: The alignment of an encoder can be accurately gauged by calculating the trace of its embeddings.

Proof: Assuming the $f(u)$ and $f(i)$ represent the embeddings for user u and item i, and we have:

$$\begin{aligned}
&\underset{(u,i)\sim pos}{\mathbb{E}} \|f(u) - f(i)\|^2 \\
&= \underset{(u,i)\sim pos}{\mathbb{E}} \left[\left(f(u)^\top - f(i)^\top (f(u) - f(i)) \right) \right] \\
&= \underset{(u,i)\sim pos}{\mathbb{E}} \left[2 - 2f(u)^\top f(i) \right] \\
&= 2 - \frac{2}{|n|} tr\left((\mathbf{H}_\text{user})^\top \mathbf{H}_\text{item} \right),
\end{aligned} \tag{5}$$

where, $\mathbf{H}_{\text{user}}$ and $\mathbf{H}\_\text{item}$ means user and item matrices, respectively. And tr means the sum of the diagonal elements of the obtained matrix $(\mathbf{H}_{\text{user}})^\top \mathbf{H}\_\text{item}$, $|n|$ means the size of the obtained matrix. We observe that alignment loss essentially calculates the Euclidean distance between users and items. We can transform it into $f(u)^\top f(i)$, showing that alignment is equivalent to Eq. (5). Thus, the trace can be used to measure alignment. ■

TABLE I
DirectAU With Different γ

γ	Align Loss	Uniform Loss	Tr	R@20	N@20
1.0	0.8583	-3.719	37.2737	0.0697	0.0574
1.5	0.9434	-3.789	**38.4137**	**0.0709**	**0.0586**
2.0	0.9931	-3.819	38.1809	0.0708	0.0584
2.5	1.0269	-3.832	37.8988	0.0700	0.0579

The above proof shows that a larger trace of the optimization target corresponds to a smaller overall alignment loss, indicating better alignment. To validate the effectiveness of our approach, we perform experiments on the Yelp dataset [22]. The findings demonstrate that our proposed metric, trace, can effectively replace the traditional alignment loss in evaluating the model's alignment quality. Taking DirectAU as an example, we calculate the correlation between model loss and trace values with changes in hyperparameters. As shown in Table I, the change rates of the alignment loss are 9.915%, 5.268%, and 3.403%, while the change rates of the uniformity loss are 1.882%, 0.792%, and 0.340%. Meanwhile, the change rates of the trace are 3.058%, 0.610%, and 0.744%. The uniformity and trace metrics are less sensitive to changes in hyperparameters and better reflect the actual state of the model. Additionally, the trace metric more accurately reflects performance changes: when the model's performance peaks, the trace metric also reaches its highest point. Based on this analysis, we decide to use the trace as a replacement for the alignment loss in our assessments.

B. Alignment and Uniformity With Discrimination

DirectAU often falls short in terms of alignment, leading us to propose a new learning objective. On top of the existing alignment and uniformity, we add a discriminative term between users (items) to achieve better recommendation results, named DiscrimAU. As shown in Fig. 2, the positive user-item pairs are first encoded into embeddings using LightGCN as the encoder. Next, we multiply the obtained user and item embeddings by their respective transposes to derive the relevance matrices between users (items). DiscrimAU consists of two modules: alignment and uniformity learning, and discriminative learning. Recommendation are made based on the distribution results obtained after training. Below, we introduce each component in detail.

1) Alignment and Uniformity Learning: For user and item embeddings, we utilize the normalized alignment and uniformity losses as defined in Eqs. (6) and (7):

$$l_{\text{norm\_align}} = \underset{(u,i)\sim pos}{\mathbb{E}} \left\| f\left(\frac{1}{F}u\right) - f\left(\frac{1}{F}i\right) \right\|^2; \tag{6}$$

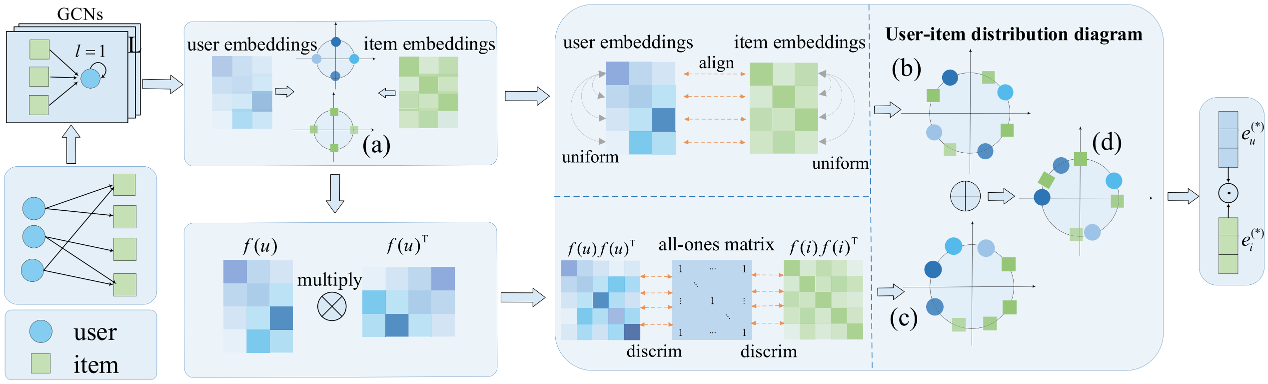


Fig. 2. The overall model framework of DiscrimAU. In this framework: (a) denotes the initial user and item embeddings; (b) denotes the distribution after alignment and uniformity learning; (c) denotes the distribution post discriminative term learning; (d) denotes the final distribution obtained through the DiscrimAU.

$$l_{\text{norm\_uniform}} = \log \mathop{\mathbb{E}}_{u,u' \sim p_{user}} e^{-2\left\| f\left(\frac{1}{F}u\right) - f\left(\frac{1}{F}u'\right)\right\|^2} /2 + \log \mathop{\mathbb{E}}_{i,i' \sim p_{item}} e^{-2\left\| f\left(\frac{1}{F}i\right) - f\left(\frac{1}{F}i'\right)\right\|^2} /2, \quad (7)$$

where $(u, i) \sim pos$ means the positive user-item pair, $u, u' \sim p_{user}$ and $i, i' \sim p_{item}$ means the user and item distribution, respectively. And $1/F$ means the normalization of user and item embeddings. Contrary to traditional approaches that directly apply alignment and uniformity losses, we first normalize the user and item embeddings. Normalization ensures that no single dimension or feature becomes overly influential, thereby providing a balanced contribution of all features. This leads to more accurate learning and unbiased recommendation results [23].

Similar samples inherently possess similar attributes, and the goal of alignment and uniformity learning is to ensure that these analogous entities are brought closer together within the embedding space. The alignment facilitates the accurate capture of positive interaction signals, thereby improving the quality of recommendation. Concurrently, we assess the uniformity loss for the embeddings of users and items independently, striving for distribution that is as uniform as possible. A uniform distribution is beneficial as it preserves the highest amount of information, enabling the model to capture and learn from a more extensive array of content [18].

By employing this model, we expect the distribution of users and items to resemble that shown in Fig. 2(b). In this configuration, user-item pairs with higher relevance are closely clustered on the hypersphere, while users and items are uniformly distributed across the hypersphere.

2) Discriminative Learning: The purpose of discriminative loss is to distinguish users and items based on their degree of relevance, respectively. For user (item) pairs with high relevance, the discriminative loss encourages them to be close in the embedding space. This ensures that the distribution of users and items accurately reflects their relevance, helping the model to capture the complex relationships between users and items more accurately. The discriminative loss as defined in Eqs. (8) and (9):

TABLE II
THE IMPACT OF DISCRIMINATIVE LOSS ON TRACE

Models	Amazon-Book	Yelp2018	Tmall
w/o discrim	53.8164	40.7485	34.4909
DiscrimAU	**54.5637**	**41.6001**	**35.2358**

$$l_{\text{discrim\_i}} = \mathop{\mathbb{E}}_{i \in p_{item}} \left\| f\left(\frac{1}{F}i\right) f\left(\frac{1}{F}i\right)^{\top} - \mathbf{1}_{m\times m} \right\|^2; \quad (8)$$

$$l_{\text{discrim\_u}} = \mathop{\mathbb{E}}_{u \in p_{user}} \left\| f\left(\frac{1}{F}u\right) f\left(\frac{1}{F}u\right)^{\top} - \mathbf{1}_{m\times m} \right\|^2, \quad (9)$$

where $\mathbf{1}$ means the all-ones matrix and $m \times m$ means the number of row and column of the matrix is m. And in the matrix $\mathbf{1}_{m\times m}$, the element $r_{ij} = 1$ at the i row and j column indicates that the users (items) i and j are fully aligned. In this loss function, we utilize the obtained relevance matrices between users (items). Through the discriminative term, we calculate the Euclidean distance between the relevance matrices and the perfectly aligned matrix to ensure that similar users (items) maintain alignment as closely as possible. The discriminative loss ensures the distinguishability of information, allowing for more accurate learning of positive samples and preventing the issue of over-uniformity.

Meanwhile, our discriminative term increases the trace values for users and items, thereby enhancing the alignment. In Table II, the 'w/o discrim' denotes the version without the discriminative loss. Across the three datasets, the inclusion of the discriminative loss leads to an increase in the trace values to varying degrees, with increases of 1.39%, 2.09%, and 2.16%. As shown in Fig. 2(c), similar users (items) cluster together to some extent, exhibiting a tightly packed arrangement. Under the influence of the discriminative loss, the model is able to discern subtle differences among similar users or items, leading to more precise recommendation.

Finally, the distribution of users and items is shown in Fig. 2(d), which is uniformly distributed overall. Similar users and items are relatively closer, ensuring the quality of learnable information. Our total loss is as follows:

$$\mathcal{L} = l_{\text{norm\_align}} + \alpha l_{\text{discrim\_u}} + \beta l_{\text{discrim\_i}} + \gamma l_{\text{norm\_uniform}}. \quad (10)$$

TABLE III
THE COMPARISON OF TIME COMPLEXITY

Component	SimGCL	DirectAU	DiscrimAU
Adjacency Matrix	$O(2\|E\|)$	$O(2\|E\|)$	$O(2\|E\|)$
Graph Convolution	$O(6\|E\|Ld)$	$O(2\|E\|Ld)$	$O(2\|E\|Ld)$
Loss	$O(3Bd{+}BMd)$	$O(Bd{+}Md{+}M^2d)$	$O(3(Bd{+}Md){+}M^2d)$

TABLE IV
EFFICIENCY COMPARISON ON YELP2018, INCLUDING THE AVERAGE TRAINING TIME PER EPOCH, THE NUMBER OF EPOCHS TO CONVERGE, AND THE TOTAL TRAINING TIME (S: SECOND, M: MINUTE, H: HOUR)

Method	time/epoch	#epoch	total time
LightGCN	32.4s	312	2h48m
DirectAU	67.6s	72	1h21m
DiscrimAU	70.3s	51	59m

The weights α and β control the strength of the discriminative perturbation and the weight γ controls the desired degree of uniformity.

C. Model Analysis

1) Space Complexity: Compared to traditional embedding-based collaborative filtering, the model parameters only take up an additional space of $d*d$ complete correlation matrix, which d denote the embedding size. The additional storage space is very small and can be neglected.

2) Time Complexity: We compare the time complexity of DirectAU and SimGCL, where DirectAU is the earliest method in the recommendation field that incorporates alignment and uniformity, and SimGCL is a classic method based on the InfoNCE loss. Let $|E|$ denote the edge number of the graph, d denote the embedding size, L denote the number of layers, B denote the batch size and M represent the node number in a batch. We can derive:

- For all three methods, no graph augmentations are required, so they must normalize the original adjacency matrix, which contains $2|E|$ non-zero elements. During the graph convolution stage, these methods do not require data augmentation and thus only need a single encoder.
- As for recommendation loss, LightGCN uses the BPR loss, the time complexity is $O(2Bd)$. DirectAU uses the align loss and uniformity loss, with time complexities of $O(Bd+Md)$ and $O(M^2d)$, respectively.The uniformity loss requires calculating the distances between M^2 node pairs, with each distance calculation involving d dimensions. Thus, the complexity of computing the pairwise distance matrix is $O(M^2d)$. Building upon DirectAU, DiscrimAU adds two discriminative losses, resulting in a computational cost of $O(3(Bd+Md)+M^2d)$.
- We trained our models using a GeForce RTX 3090 GPU. On the Yelp2018 dataset, LightGCN takes 32 seconds per epoch, DirectAU takes 67 seconds, and DiscrimAU takes 70 seconds. Additionally, the loss functions used by the three algorithms differ, leading to variations in convergence speed. The total training time required for each method is 168 minutes, 81 minutes, and 59 minutes, respectively. DiscrimAU's training time is shorter than DirectAU, possibly because the discriminative loss helps accelerate convergence.

TABLE V
STATISTICS OF THE EXPERIMENTAL DATASETS

Datasets	Users	Items	Interactions	Density
Yelp2018	31,668	38,084	1,561,406	0.00130
Amazon-book	52,643	91,599	2,984,108	0.00062
Tmall	47,939	41,390	2,619,389	0.00132

IV. EXPERIMENTS

A. Experimental Settings

1) Datasets: To evaluate the recommendation performance of DiscrimAU, we perform extensive experiments on three public datasets:

- *Yelp2018* [22]: This widely-used dataset contains user ratings on business venues collected from the Yelp platform. Following previous work, we use transaction records from 2018 onwards.
- *Amazon-Book* [16]: This dataset includes users' rating behavior on products with book categories on the Amazon platform.
- *Tmall* [24]: It contains customer purchase behaviors from the online retailer Tmall.

Table V provides statistical information for three datasets. To ensure fairness and consistency, we adopt the same processing method with existing efforts [9], [16]. Specifically, the 10-core setting is taken to ensure the quality of the datasets (i.e., users and items with fewer than 10 interactions are filtered out). Subsequently, we transfer all explicit feedback (0,1,…, 5) to implicit feedback (0,1). We consider all ratings '>3' as presence of interactions (i.e., presence of interaction is 1 otherwise 0). Consistent with previous work, we divide all interactions into training (70%), validation (10%), and testing sets (20%). We measure the performance of all recommendation models via Recall@20 and NDCG@20 [25].

2) Baselines: We compare the performance of DiscrimAU with various state-of-the-art methods:

- *BPRMF* [13]: This is a typical negative sampling method that uses BPR loss to optimize MF, which randomly takes negative samples from the item set.
- *NGCF* [25]: This method is a cutting-edge model based on Graph Convolutional Networks (GCNs). In addition to following the standard GCN framework, it incorporates an additional Hadamard product between user and item embeddings, enhancing the interaction modeling beyond the conventional GCN approach.
- *LightGCN* [9]: This is a simplified graph convolution network for CF that obtains multi-hop neighborhood information in user-item bipartite graph.
- *SGL-ED* [26]: This uses graph contrastive learning. It uses the random deletion method to generate two views for graph contrast learning. This paper uses SGL-ED with the best performance as the comparison method.
- *NCL* [27]: This is a graph contrastive learning method that considers the neighbors of users (or items) from

TABLE VI
RECOMMENDATION PERFORMANCE ON THREE DATASETS. THE BEST RESULTS ARE IN BOLD FACE, AND THE BEST BASELINES ARE UNDERLINED

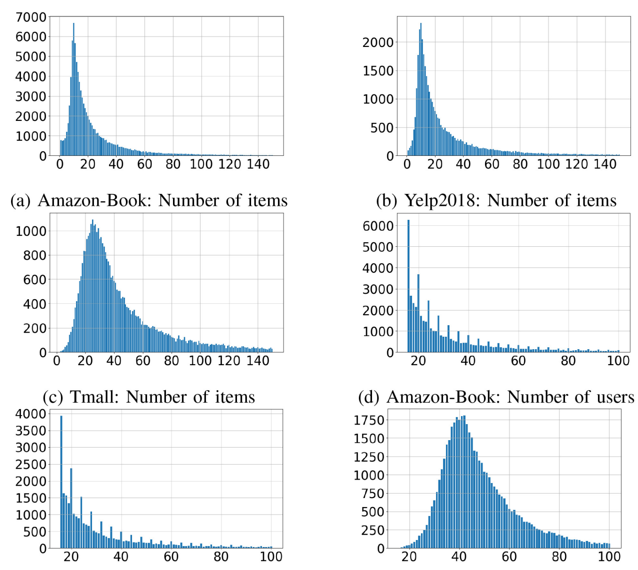
	Amazon-Book		Yelp2018		Tmall	
	Recall@20	NDCG@20	Recall@20	NDCG@20	Recall@20	NDCG@20
BPRMF	0.0320	0.0248	0.0545	0.0446	0.0615	0.0426
NGCF	0.0337	0.0262	0.0579	0.0477	0.0629	0.0465
LightGCN	0.0403	0.0314	0.0635	0.0521	0.0718	0.0499
SGL-ED	0.0478	0.0379	0.0675	0.0547	0.0721	0.0506
NCL	0.0481	0.0373	0.0685	0.0577	0.0750	0.0553
SimGCL	0.0517	0.0412	0.0721	0.0596	0.0885	0.0632
DirectAU-LGCN	0.0508	0.0406	0.0709	0.0586	0.0835	0.0593
CGCL	0.0483	0.0380	0.0690	0.0560	0.0880	0.0614
GraphAU	0.0510	0.0409	0.0663	0.0544	0.0842	0.0589
VGCL	0.0511	0.0406	0.0714	0.0587	0.0885	0.0635
RecDCL	0.0525	0.0407	0.0690	0.0567	0.0853	0.0632
BIGCF	0.0500	0.0398	0.0730	0.0603	0.0876	0.0623
DiscrimAU	**0.0558**(±0.004)*	**0.0446**(±0.003)*	**0.0743**(±0.004)*	**0.0617**(±0.002)*	**0.0899**(±0.006)*	**0.0641**(±0.002)*
p-value	1.13e-11	1.63e-11	2.89e-9	5.67e-10	9.89e-6	4.65e-7

Symbol * denotes that the improvement is significant with a p value < 0.001 based on a two-tailed paired $t-test$. The data in parentheses in parentheses indicate the margin of error.

both the graph structure and semantic space. It explicitly incorporates potential neighbors into the contrastive pairs.

- *SimGCL* [22]: This is a graph contrastive learning method that uses random noise for data augmentation to compare the views generated under two random noise conditions.
- *DirectAU-LGCN* [17]: This method provides a new learning objective for CF, which measures representation quality based on alignment and uniformity. This paper uses LightGCN with the best performance as the comparison method.
- *CGCL* [28]: This method mitigates the data sparsity problem and obtains high-quality node embeddings by constructing structural neighbor contrastive learning targets, candidate contrastive learning targets, and candidate structural neighbor contrastive learning targets.
- *GraphAU* [19]: Based on DirectAU, this uses a layer-wise alignment pooling module to integrate alignment losses layer-wise.
- *VGCL* [29]: This method leverages variational graph reconstruction to estimate a Gaussian distribution of each node, using twofold contrastive learning of node-level and cluster-level.
- *RecDCL* [30]: This is a dual contrastive approach that optimises using both batch-wise contrastive learning and feature-wise contrastive learning.
- *BIGCF* [16]: From a causal perspective, this method divides interaction motivations into collective intentions, and individual intentions and leverages these intentions to guide the reconstruction of the interaction graph.

3) Parameter Settings: We use PyTorch to implement our DiscrimAU model. Adam [31] is used as the default optimizer. The batch size is 2048 and embedding size is fixed to 64. Early stop is adopted if NDCG@20 on the validation dataset continues to drop for 10 epochs. The default encoder is LightGCN. For our DiscrimAU model, we turn the weight γ in [0.5, 1.5, 3, 4.5, 6.5, 8.5, 10], and turn the discriminative weights α and β in [0, 0.0015, 0.003, 0.005, 0.01]. For the baseline specific hyperparameters, we tune them within the ranges recommended by the original paper. All parameters are initialized using Xavier initialization [32].



(a) Amazon-Book: Number of items (b) Yelp2018: Number of items (c) Tmall: Number of items (d) Amazon-Book: Number of users (e) Yelp2018: Number of users (f) Tmall: Number of users

Fig. 3. User and item interactions of three datasets.The x-axis represents the interaction counts.

B. Overall Performance

Table VI presents the performance of various baseline methods compared to DiscrimAU. From the experimental results, we can make the following observations:

- DiscrimAU consistently outperforms all baselines on all three datasets. Specifically, DiscrimAU improves Recall@20 compared to DirectAU by 7.93% on Amazon-Book, 4.79% on Yelp2018, and 7.54% on Tmall. Additionally, compared to the strongest baseline, DiscrimAU shows superior performance, with improvements in Recall@20 by 6.29%, 1.78%, and 1.46%, respectively. Additionally, the results of the two-tailed paired $t-test$ indicate that the improvement of our model is statistically significant.
- On the Tmall dataset, methods that directly use alignment and uniformity generally perform weaker than algorithms utilizing InfoNCE. As shown in Fig. 3, this may be due to the higher number of interactions for users and items in

TABLE VII
ABLATION STUDY OF DISCRIMAU ON DIFFERENT DATASETS (MEASURED BY RECALL@20 AND NDCG@20)

	Amazon-Book		Yelp2018		Tmall	
	R@20	N@20	R@20	N@20	R@20	N@20
LightGCN	0.0403	0.0314	0.0635	0.0521	0.0718	0.0499
align&uniform to BPR	0.0414	0.0321	0.0660	0.0544	0.0756	0.0524
w/o discrim	0.0550	0.0439	0.0732	0.0607	0.0883	0.0634
DiscrimAU	**0.0558**	**0.0446**	**0.0743**	**0.0617**	**0.0899**	**0.0641**

TABLE VIII
DISCRIMAU WITH DIFFERENT γ VARIATION

γ	Align Loss	Uniform Loss	Tr	R@20	N@20
1	0.8475	-3.7090	38.2462	0.0709	0.0586
2	0.9853	-3.8139	40.4558	0.0738	0.0614
3	1.0451	-3.8451	**41.6001**	**0.0743**	**0.0617**
4	1.0838	-3.8542	40.5168	0.0730	0.0606
5	1.1010	-3.8610	40.6298	0.0718	0.0596

the Tmall dataset, typically ranging between twenty and sixty. In contrast, the peak interactions for Amazon-Book and Yelp2018 are around twenty. This higher interaction range in Tmall could lead to uniform training diluting the connections between similar users and items. DiscrimAU addresses these issues by normalizing the alignment and uniformity losses and incorporating discriminative terms beyond alignment and uniformity, thus surpassing the current state-of-the-art InfoNCE method.

- By observing Figs. 3(a), 3(b) and 3(c), it becomes clear that the long-tail effect is especially pronounced in the Yelp2018 and Amazon-Book datasets, where the majority of items have relatively few interactions and are often overlooked in the recommendation process. In contrast, the Tmall dataset exhibits a more balanced interaction distribution, with most items receiving sufficient attention. As a result, the long-tail effect has a less pronounced impact on recommendations derived from this dataset. This explains why, despite Tmall and Yelp2018 having similar densities, the Recall and NDCG performance on Tmall are notably superior. Furthermore, the more significant improvements observed on the Amazon dataset suggest that our method may be particularly effective at addressing the challenges posed by sparse datasets.
- All graph contrastive learning methods achieve impressive performance, validating the effectiveness of incorporating SSL. SimGCL, VGCL, BIGCF and RecDCL achieve the best baseline performance on different datasets, indicating that GCL with data augmentation is more effective than structural augmentation. Notably, although DiscrimAU does not involve data augmentation, the discriminative loss we introduce yields a similar effect: enhancing the learnable information while preserving the invariance of the original graph to improve recommendation performance. Consequently, DiscrimAU exhibits superior performance compared to DirectAU.

C. Ablation Study

We perform an ablation study to demonstrate the necessity of combining the designed components. As shown in Table VII, 'w/o discrim' indicates the removal of the discriminative loss. 'align&uniform to BPR' refers to replacing the normalized alignment and uniformity losses with BPR loss. (i) By comparing the 'w/o discrim' and 'DiscrimAU' settings, we can conclude that the discriminative loss demonstrates its effectiveness across all three datasets. When the normalized alignment and uniformity losses are replaced with BPR loss, the performance remains superior to LightGCN. This indicates that the discriminative loss could enhance the alignment effect in other methods as well, thereby improving recommendation performance. (ii) Replacing alignment loss and uniformity loss with BPR loss leads to a sharp decline in performance, undoubtedly due to the crucial role of SSL learning in recommendations, as previously validated by prior work. However, using alignment and uniformity for representation learning causes the model to overly depend on extracted attributes, compromising the generality of the representations and further resulting in poor performance on test data. Using discrim loss can alleviate this problem to some extent through the discriminant term. Based on the above analysis, all components contribute positively to the recommendation performance of DiscrimAU.

D. Alignment and Uniformity Analysis

1) Role of Alignment and Uniformity: Through the study of the baseline training process, we further analyze the roles of alignment and consistency in recommendation. As shown in Fig. 4, the two solid lines represent changes in trace and uniformity during the training of VGCL. It is evident that when performance is at its worst, uniformity reaches its peak. However, due to poor alignment at this stage, the model struggles to effectively capture positive information, making recommendations difficult. As training continues, excessive uniformity is alleviated. Both trace and uniformity reach an optimal state when performance peaks. Subsequently, the alignment weakens significantly. As a result, the model struggles to capture positive samples. This makes it difficult to generate recommendations, leading to a decline in performance. Overall, monitoring the status of alignment and uniformity can help guide the model in completing recommendation tasks.

2) Impact of Hyperparameter on Trace Values: Based on the analysis in the **Alignment Analysis** section, we statistically analyze the changes in the model loss and trace values of the DiscrimAU as the hyperparameter γ changes, as shown in Table VIII. As the hyperparameter γ increases, the change rates of alignment loss are 16.259%, 6.069%, 3.703%, and 1.587%, while the change rates of trace are 5.777%, 2.828%, 2.673%, and 0.279%. Similarly, the change rates of uniformity loss are 2.828%, 0.818%, 0.237%, and 0.176%. It is evident that the trace is less affected by hyperparameters compared to alignment loss. Notably, the value of alignment loss increases with γ, but the trace first rises and then falls as γ increases, which generally matches the model's performance changes. The trace reaches its peak when the model captures the optimal

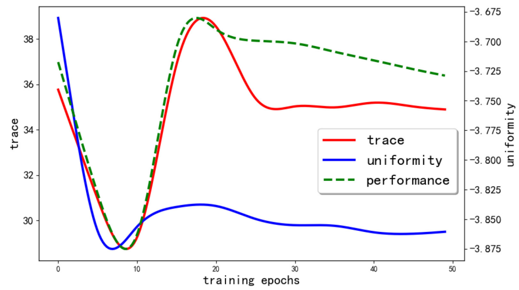
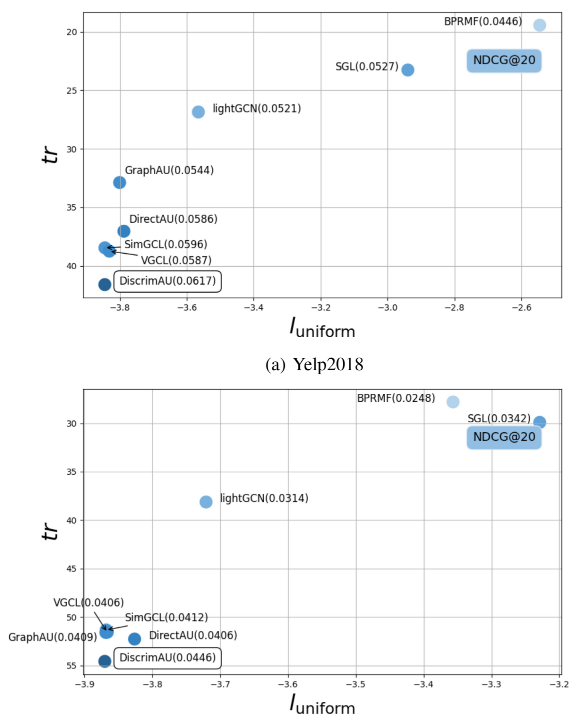


Fig. 4. The trends of tr and l_uniform during training (solid line) and the learning curve (dashed line) of VGCL method on the Yelp2018 dataset.



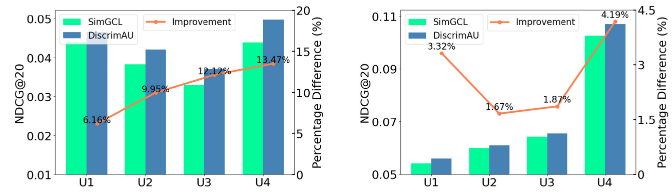
(a) Yelp2018

(b) Amazon-Book

Fig. 5. $\text{tr} - l_{uniform}$ plot of different methods. For $l_{uniform}$, lower numbers is better. For tr, higher numbers is better. Numbers in parentheses indicate NDCG@20.

hyperparameters. This indicates that the trace can indeed be used as a more reliable measure of alignment.

3) Performance Analysis of Trace and Uniformity: Based on this analysis, we present the trace and uniformity performance of different methods, as shown in Fig. 5. Overall, methods with larger trace values and greater absolute uniformity tend to achieve better performance. Although GraphAU demonstrates good uniformity, its alignment effectiveness is insufficient, leading to relatively suboptimal performance. This shortcoming is less noticeable on the Amazon-Book dataset, possibly because the large number of nodes in this dataset partially compensates for the lack of alignment. LightGCN's poor alignment causes relatively related users and items to be overly dispersed. This dispersion makes it difficult to capture sufficient information, leading to suboptimal learning outcomes. In contrast, VGCL and SimGCL not only maintain excellent uniformity but also achieve outstanding alignment.



(a) Amazon-Book Dataset

(b) Yelp2018 Dataset

Fig. 6. Performance comparisons under different user groups.

DirectAU, which directly trains on alignment and uniformity, also achieved excellent recommendation performance. Building on this foundation, our proposed DiscrimAU model further enhances learning capabilities, achieving superior alignment and uniformity, and delivering the best recommendation performance.

E. Sparse Data Experiment Analysis

To investigate the impact of data sparsity on recommendation performance, we conduct a comparative analysis across different user groups. Specifically, we classify all users into four distinct groups based on their interaction frequency and evaluate the recommendation performance for each group. As shown in Fig. 6, all models achieve the best performance in the user group with the highest interaction density, which is expected since denser data allows models to better capture user preferences and make more accurate predictions.

However, our proposed method, DiscrimAU, consistently outperforms the other models across all user groups, demonstrating superior recommendation quality irrespective of interaction density. Notably, DiscrimAU exhibits more substantial improvements in groups with higher interaction density, particularly in the U4 group of both the Amazon-book and Yelp2018 datasets, where performance increases by 13.47% and 4.19%, respectively. These results underscore the model's robustness and its capacity to effectively handle more complex data patterns.

A particularly striking finding emerges in the Yelp2018 dataset, where the sparsest user group (U1) experiences a notable performance boost. DiscrimAU mitigates the issue of sparse interactions from inactive users by strengthening the connections between users and items. This approach enhances the implicit information that can be extracted from these users locally, thereby reducing the negative impact of sparse data on recommendation performance.

F. Parameter Sensitivity Analysis

1) Effect of Discriminative Loss Weights α and β: Since the discriminative loss directly impacts the alignment strength, it should be kept within a small range to prevent excessive influence on user-user and item-item pairs. An overly strong discriminative loss could cause similar users and items to collapse into a single point, making them difficult to distinguish through uniformity. Based on previous experience, we performed a grid search for α and β in the range of 0 to 0.01, with the grid divided as follows: [0, 0.0015, 0.003, 0.005, 0.01].

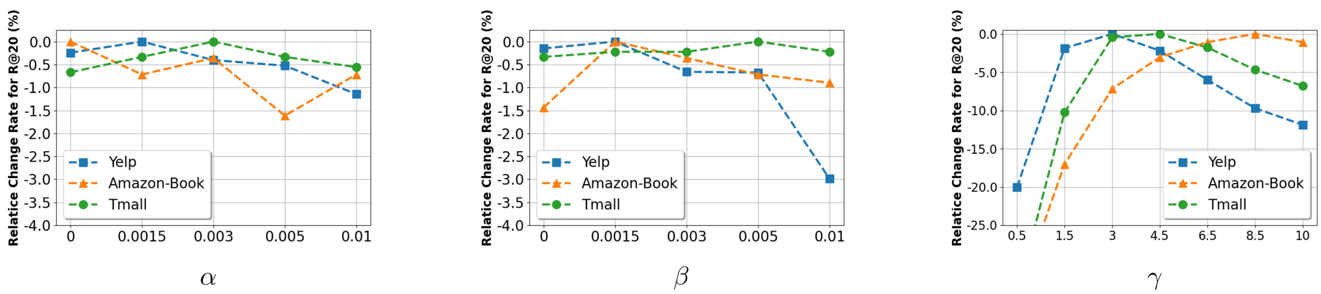


Fig. 7. Hyperparameter Sensitivities for (a) discrimination for user weight α; (b) discrimination for item weight β; and (c) uniformity weight γ w.r.t Recall@20 on three datasets.

As shown in Figs. 7(a) and 7(b), we find that for varying α and β, the resulting performance is different. On the Yelp2018 dataset, the best performance is achieved when both α and β are set to 0.0015. And on the Tmall dataset, the optimal values of α and β are 0.005 and 0.003, respectively. On the Amazon-Book dataset, the optimal performance is achieved with α set to 0.0015 and set β to 0. This could be due to Amazon-Book having a sufficient number of items to produce a satisfactory alignment, making additional discriminative loss unnecessary. Meanwhile, when α and β are kept within a relatively small range (below 0.01), the overall performance difference is minimal. Even compared to peak performance, the performance gap does not exceed four percent. This indicates that the discriminative loss is insensitive to hyperparameters. And a moderate level of discriminative perturbation positively contributes to enhancing alignment and improving recommendation performance.

2) Effect of Uniformity Loss Weight γ: For the hyperparameter γ, we performed a grid search from 0 to 10. Based on prior experiments with DirectAU, we found that after γ reaches a larger value, a broader range is needed. Therefore, we further refined the grid into [0.5, 1.5, 3, 4.5, 6.5, 8.5, 10].

As shown in Fig. 7(c), we compare experimental results of different γ on three datasets. Low levels of uniformity may hard to lead that users and items may over-align, making it hard to learn enough information. Conversely, high levels of uniformity might cause overly similar users and items to become too separated, leading to overly uniform information distribution and ineffective learning.

The optimal value of γ varies with the number of users, items, and interactions. By observing the experimental results on the Yelp and Tmall datasets, we found that as the number of users and items increases and the number of interactions grows, the optimal γ value also tends to be higher. This is likely because larger datasets make maintaining consistency more challenging. Since the Amazon-Book dataset has a larger number of users and items, it produces sparser interactions, necessitating a higher level of uniformity. Consequently, the optimal γ value was found to be 8.5.

V. Related Work

A. Graph-Based Recommender Systems

To address the complex user-item interaction relationships, Graph Neural Networks (GNNs) have been widely applied in recommender systems in recent years [33], [34]. These architectures can effectively capture multi-hop links between users and items, allowing the model to capture more complex and nuanced relationships. Some architectures, like NGCF [25], which propagates embeddings across the user-item graph, leverages its structure to model high-order connectivity effectively. This approach explicitly incorporates collaborative signals into the embedding process. Furthermore, LightGCN [9] simplifies non-linear transformations and removes unnecessary deep learning operations, as detailed below:

$$e_u^{(k+1)} = \sum_{i \in \mathcal{N}_u} \frac{1}{\sqrt{|\mathcal{N}_u|}\sqrt{|\mathcal{N}_i|}} e_i^{(k)},$$
$$e_i^{(k+1)} = \sum_{u \in \mathcal{N}_i} \frac{1}{\sqrt{|\mathcal{N}_u|}\sqrt{|\mathcal{N}_i|}} e_u^{(k)}. \tag{11}$$

where $\mathcal{N}_u$ and $\mathcal{N}_i$ respectively denotes the set of items that are interacted by user u, and the set of users that are interacted by item i, $e_u^{(k)}$ and $e_i^{(k)}$ denotes the refined embedding of user u and item i after k layers propagation, respectively. The symmetric normalization term $\frac{1}{\sqrt{|\mathcal{N}_u|}\sqrt{|\mathcal{N}_i|}}$ prevents the embedding size from growing with each graph convolution operation, thus enhancing the efficiency of embedding extraction. These methods preserve multi-hop connections within the embeddings.

B. Self-Supervised Learning in Recommendation

In the context of Graph Convolutional Neural Networks (GCNs), self-supervised learning (SSL) has proven to be an effective approach for graph learning. One contribution of SSL in graphs is the use of contrastive learning to analyze graph data from different perspectives, such as heterogeneous graphs [35] and knowledge graphs [36].

Self-supervised learning has also been widely applied in recommender systems, particularly in graph contrastive learning [37] and generative self-supervised learning [38]. For example, LightGT [39] uses a layer-wise positional encoder and lightweight self-attention blocks to improve efficiency, optimizing user and item representations effectively and efficiently through generated content. LightGCL [37] employs singular value decomposition for contrastive enhancement, enabling unconstrained structural refinement in global collaborative relationship modeling, generating two views for contrastive learning. Additionally, self-supervised learning has also achieved success in other recommendation domains. SimKGCL [40] applies graph contrastive learning to knowledge graph recommendation, achieving excellent performance by contrasting knowledge graphs and interaction graphs. HGCL [41] uses meta-network enhanced heterogeneous graphs for contrastive learning, achieving adaptive augmentation by integrating heterogeneous side information into the recommender systems. The effect of SSL is very significant, and we adopt this principle as the learning objectives.

VI. Conclusion

In this work, we investigated the limitations of using alignment as a discriminative measure in recommender systems. We proposed a novel assessment based on the trace of the users and items interaction matrices, which effectively captured the distribution of users and items on hypersphere after model training. Furthermore, we proposed the DiscrimAU model, which incorporated an additional discriminative loss. This enhanced the relationships between similar users (items) and optimized distribution characteristics, thereby improving the overall performance of the recommender systems. We conducted experiments on three real-world datasets, which provided evidence for its superiority in contrastive recommendation.

References

[1] P. Lv et al., "DSMN: An improved recommendation model for capturing the multiplicity and dynamics of consumer interests," *IEEE Trans. Consum. Electron.*, vol. 70, no. 1, pp. 1236–1245, Feb. 2024.

[2] H. Kim, G. Y. Kim, and J. Y. Kim, "Music recommendation system using human activity recognition from accelerometer data," *IEEE Trans. Consum. Electron.*, vol. 65, no. 3, pp. 349–358, Aug. 2019.

[3] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Adv. Artif. Intell.*, vol. 2009, Oct. 2009, Art. no. 421425.

[4] Z. Qu, J. Ding, R. H. Jhaveri, Y. Djenouri, X. Ning, and P. Tiwari, "FedSarah: A novel low-latency federated learning algorithm for consumer-centric Personalized recommendation systems," *IEEE Trans. Consum. Electron.*, vol. 70, no. 1, pp. 2675–2686, Feb. 2024. [Online]. Available: https://doi.org/10.1109/TCE.2023.3342100

[5] L. Xia, C. Huang, C. Huang, K. Lin, T. Yu, and B. Kao, "Automated self-supervised learning for recommendation," in *Proc. ACM Web Conf. WWW*, 2023, pp. 992–1002.

[6] Y. Koren, R. M. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, Aug. 2009.

[7] R. Zhang et al., "Cognition-driven structural prior for instance-dependent label transition matrix estimation," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jan. 8, 2024, doi: 10.1109/TNNLS.2023.3347633.

[8] Y. Zhang, Y. Zhang, D. Yan, S. Deng, and Y. Yang, "Revisiting graph-based recommender systems from the perspective of variational auto-encoder," *ACM Trans. Inf. Syst.*, vol. 41, no. 3, p. 28, 2023.

[9] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, "LightGCN: Simplifying and powering graph convolution network for recommendation," in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2020, pp. 639–648.

[10] Y. Jiang, C. Huang, and L. Huang, "Adaptive graph contrastive learning for recommendation," in *Proc. 29th ACM SIGKDD Conf. Knowl. Disc. Data Min.*, 2023, pp. 4252–4261.

[11] X. Wang, H. Jin, A. Zhang, X. He, T. Xu, and T. Chua, "Disentangled graph collaborative filtering," in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2020, pp. 1001–1010.

[12] Y. Zhang, L. Sang, Y. Zhang, and Y. Zhang, "High-order fusion graph contrastive learning for recommendation," 2024, *arXiv:2407.19692*.

[13] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "BPR: Bayesian Personalized ranking from implicit feedback," in *Proc. 25th Conf. Uncertainty Artif. Intell.*, 2009, pp. 452–461.

[14] C. Zhou, J. Ma, J. Zhang, J. Zhou, and H. Yang, "Contrastive learning for debiased candidate generation in large-scale recommender systems," in *Proc. 27th ACM SIGKDD Conf. Knowl. Disc. Data Min.*, 2021, pp. 3985–3995.

[15] K. Mao et al., "SimpleX: A simple and strong baseline for collaborative filtering," in *Proc. 30th ACM Int. Conf. Inf. Knowl. Manage.*, 2021, pp. 1243–1252.

[16] Y. Zhang, L. Sang, and Y. Zhang, "Exploring the individuality and collectivity of intents behind interactions for graph collaborative filtering," in *Proc. 47th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2024, pp. 1253–1262.

[17] C. Wang et al., "Towards representation alignment and uniformity in collaborative filtering," in *Proc. 28th ACM SIGKDD Conf. Knowl. Disc. Data Min.*, 2022, pp. 1816–1825.

[18] T. Wang and P. Isola, "Understanding contrastive representation learning through alignment and uniformity on the hypersphere," in *Proc. 37th Int. Conf. Mach. Learn.*, vol. 119, 2020, pp. 9929–9939.

[19] L. Yang et al., "Graph-based alignment and uniformity for recommendation," in *Proc. 32nd ACM Int. Conf. Inf. Knowl. Manage.*, 2023, pp. 4395–4399.

[20] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T. Chua, "Neural collaborative filtering," in *Proc. 26th Int. Conf. World Wide Web*, 2017, pp. 173–182.

[21] L. Xia, C. Huang, J. Shi, and Y. Xu, "Graph-less collaborative filtering," in *Proc. ACM Web Conf.*, 2023, pp. 17–27.

[22] J. Yu, H. Yin, X. Xia, T. Chen, L. Cui, and Q. V. H. Nguyen, "Are graph augmentations necessary?: Simple graph contrastive learning for recommendation," in *Proc. 45th Int. ACM SIGIR Conf. Res. Develop. Inf.*, 2022, pp. 1294–1303.

[23] C. Garbin, X. Zhu, and O. Marques, "Dropout vs. batch normalization: An empirical study of their impact to deep learning," *Multimedia Tools Appl.*, vol. 79, nos. 19–20, pp. 12777–12815, 2020.

[24] X. Ren, L. Xia, J. Zhao, D. Yin, and C. Huang, "Disentangled contrastive collaborative filtering," in *Proc. 46th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2023, pp. 1137–1146.

[25] X. Wang, X. He, M. Wang, F. Feng, and T. Chua, "Neural graph collaborative filtering," in *Proc. 42nd Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2019, pp. 165–174.

[26] J. Wu et al., "Self-supervised graph learning for recommendation," in *Proc. 44th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2021, pp. 726–735.

[27] Z. Lin, C. Tian, Y. Hou, and W. X. Zhao, "Improving graph collaborative filtering with Neighborhood-enriched contrastive learning," in *Proc. ACM Web Conf. Virtual Event*, 2022, pp. 2320–2329.

[28] W. He, G. Sun, J. Lu, and X. S. Fang, "Candidate-aware graph contrastive learning for recommendation," in *Proc. 46th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2023, pp. 1670–1679.

[29] Y. Yang et al., "Generative-contrastive graph learning for recommendation," in *Proc. 46th Int. ACM SIGIR Conf. Res. Develop. Inf.*, 2023, pp. 1117–1126.

[30] D. Zhang et al., "RecDCL: Dual contrastive learning for recommendation," in *Proc. ACM Web Conf.*, 2024, pp. 3655–3666.

[31] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017, *arXiv:1412.6980*.

[32] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. 13th Int. Conf. Artif. Intell. Stat.*, vol. 9, 2010, pp. 249–256.

[33] S. Jin, Y. Zhang, X. Li, and M. Lu, "Heterogeneous graph convolutional network for E-commerce product recommendation with adaptive denoising training," *IEEE Trans. Consum. Electron.*, vol. 70, no. 1, pp. 3259–3268, Feb. 2024.

[34] R. Zhang et al., "Detail-aware network for infrared image enhancement," *IEEE Trans. Geosci. Remote Sens.*, vol. 63, pp. 1–14, 2025.

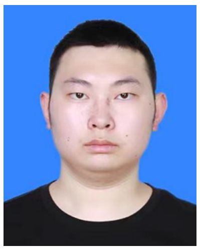
[35] D. Hwang, J. Park, S. Kwon, K. Kim, J. Ha, and H. J. Kim, "Self-supervised auxiliary learning with meta-paths for heterogeneous graphs," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 1–14.

[36] P. Zhu, D. Cheng, S. Luo, R. Xu, Y. Liang, and Y. Luo, "Leveraging enterprise knowledge graph to infer Web events' influences via self-supervised learning," *J. Web Semant.*, vol. 74, Oct. 2022, Art. no. 100722.

[37] X. Cai, C. Huang, L. Xia, and X. Ren, "LightGCL: Simple yet effective graph contrastive learning for recommendation," in *Proc. 11th Int. Conf. Learn. Represent.*, 2023, pp. 1–14.

[38] C. Li, L. Xia, X. Ren, Y. Ye, Y. Xu, and C. Huang, "Graph transformer for recommendation," in *Proc. 46th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2023, pp. 1680–1689.

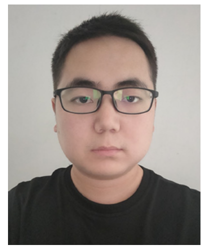
[39] Y. Wei, W. Liu, F. Liu, X. Wang, L. Nie, and T. Chua, "LightGT: A light graph transformer for multimedia recommendation," in *Proc. 46th Int. ACM SIGIR Conf. Res. Develop. Inf.*, 2023, pp. 1508–1517.
[40] H. Ye, X. Li, Y. Yao, and H. Tong, "On the sweet spot of contrastive views for knowledge-enhanced recommendation," 2023, *arXiv:2309.13384*.
[41] M. Chen, C. Huang, L. Xia, W. Wei, Y. Xu, and R. Luo, "Heterogeneous graph contrastive learning for recommendation," in *Proc. 16th ACM Int. Conf. Web Search Data Min.*, 2023, pp. 544–552.



Xinzhe Jiang received the bachelor's degree in software engineering from Anhui University, Hefei, China, in 2023, where he is currently pursuing the master's degree with the School of Computer Science and Technology. His current research interests include graph neural network, personalized recommender systems, and data mining.

Lei Sang received the Ph.D. degree from the Faculty of Engineering and Information Technology, University of Technology Sydney, Sydney, Australia, in 2021. He is currently a Lecturer with the School of Computer Science and Technology, Anhui University, Hefei, Anhui, China. His current research interests include natural language processing, data mining, and recommender systems.

Shun Lian received the master's degree in software engineering from the University of Science and Technology of China in 2016. He is currently pursuing the Ph.D. degree with the School of Computer Science and Technology, Anhui University, Hefei, China. His current research directions include natural language processing and intelligent computing services.



Yi Zhang received the bachelor's and master's degrees in computer science and technology from Anhui University, Hefei, China, in 2020 and 2023, respectively, where he is currently pursuing the Ph.D. degree with the School of Computer Science and Technology. He has more than ten publications in several flagship journals and conferences, including IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS, IEEE TRANSACTIONS ON BIG DATA, *ACM Transactions on Information Systems*, and ACM SIGIR. His current research interests include graph neural network, personalized recommender systems, and service computing.

Yiwen Zhang received the Ph.D. degree in management science and engineering from the Hefei University of Technology in 2013. He is currently a Full Professor with the School of Computer Science and Technology, Anhui University. He has published more than 100 papers in highly regarded conferences and journals, including IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, IEEE TRANSACTIONS ON MOBILE COMPUTING, IEEE TRANSACTIONS ON SERVICES COMPUTING, *ACM Transactions on Information Systems*, IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, *ACM Transactions on Knowledge Discovery from Data*, SIGIR, and ACL. His research interests include service computing, recommender systems, and big data analytics.