How to Improve Representation *Alignment* and *Uniformity* in Graph-based Collaborative Filtering?

Zhongyu Ouyang[1], Chunhui Zhang[2], Shifu Hou[1], Chuxu Zhang[3], Yanfang Ye[1*]

[1]University of Notre Dame, Notre Dame, IN 46556
[2]Dartmouth College, Hanover, NH 03755
[3]Brandeis University, Waltham, MA 02453
{zouyang2,shou,yye7}@nd.edu, chunhui.zhang.gr@dartmouth.edu, chuxuzhang@brandeis.edu,

Abstract

Collaborative filtering (CF) is a prevalent technique utilized in recommender systems (RSs), and has been extensively deployed in various real-world applications. A recent study in CF focuses on improving the quality of representations from the perspective of alignment and uniformity on the hyperspheres for enhanced recommendation performance. It promotes alignment to increase the similarity between representations of interacting users and items, and enhances uniformity to have more uniformly distributed user and item representations within their respective hyperspheres. However, although alignment and uniformity are enforced by two different optimized objectives, respectively, they jointly constitute the supervised signals for model training. Models trained with only supervised signals in labeled data can inevitably overfit the noise introduced by label sampling variance, even with *i.i.d.* datasets. This overfitting to noise further compromises the model's generalizability and performance on unseen testing data. To address this issue, in this study, we aim to mitigate the effect caused by the sampling variance in labeled training data to improve representation generalizability from the perspective of alignment and uniformity. Representations with more generalized alignment and uniformity further lead to improved model performance on testing data. Specifically, we model the data as a user-item interaction bipartite graph, and apply a graph neural network (GNN) to learn the user and item representations. This graph modeling approach allows us to integrate self-supervised signals into the RS, by performing self-supervised contrastive learning on the user and item representations from the perspective of label-irrelevant alignment and uniformity. Since the representations are less dependent on label supervision, they can capture more label-irrelevant data structures and patterns, leading to more generalized alignment and uniformity. We conduct extensive experiments on three benchmark datasets to demonstrate the superiority of our framework (i.e., improved performance and faster convergence speed). Our codes: https://github.com/zyouyang/AUPlus

Introduction

The development of recommender systems (RSs) has been widely explored to assist in information filtering that alleviates the data overload problem among multiple fields (McAuley et al. 2015; Covington, Adams, and Sargin 2016). The goal of a recommender system is to predict future interactions given the historical interactions currently observed between users and items. RSs are mainly categorized into content-based models (Lops, De Gemmis, and Semeraro 2011; Tay, Luu, and Hui 2018), collaborative filtering (CF)-based models (Schafer et al. 2007; Chen et al. 2020a; Yang et al. 2022b; Wang et al. 2019; He et al. 2020b; Wu et al. 2021; Lee et al. 2021; Lin et al. 2022), and hybrid models (Burke 2002; Zhang et al. 2016; Dong et al. 2017; Wang, Shi, and Yeung 2016). In the realm of CF-based methods, the Bayesian Personalized Ranking (BPR) loss (Rendle et al. 2009), a pairwise supervised loss function, has been widely adopted as an optimized objective. It encourages the posterior probabilities of the observed user-item interactions to be higher than their unobserved counterparts. Models that adopt the BPR loss as the main loss function include matrix factorization (Koren, Bell, and Volinsky 2009) and other graph-based CF methods (Ying et al. 2018; Wang et al. 2019; Yu et al. 2019; Sun et al. 2020; He et al. 2020b). In addition to the default BPR loss, some works adopt other objectives (Weston, Bengio, and Usunier 2011; Weston, Yee, and Weiss 2013; Hsieh et al. 2017) to provide training supervisions in CF-based models.

The resurgence of contrastive learning (CL) in deep representation learning (Chen et al. 2020b) has inspired multiple studies related to the essence of CL (Wang and Isola 2020; Gao, Yao, and Chen 2021; Yu et al. 2022; Yue et al. 2022; Qian et al. 2022; Yu et al. 2023a; Zhang et al. 2023c). Some (Yu et al. 2022, 2023a) provide empirical evidence to suggest that the contrastive loss in CL is the predominant factor contributing to enhanced model generalizability. Following this line, from the self-supervised perspective, some recent CF methods (Wu et al. 2021; Yu et al. 2021b; Xia et al. 2021; Lin et al. 2022) strategically design self-supervised auxiliary contrastive tasks to jointly optimize for improved model generalization ability. From the supervised perspective, there also exist a recent work (Zhang et al. 2023a) that directly apply supervised contrastive loss to train the CF model. Apart from the angle of generalizability, some other studies in CL (Wang and Isola 2020; Gao, Yao, and Chen 2021) focus on the quality of learned representations, and identify two key properties, alignment and uniformity, related to the contrastive loss. This discovery inspires DirectAU (Wang et al. 2022), a recent CF method that disassembles supervised

*Corresponding author

contrastive loss as the alignment and uniformity loss, and directly optimizes the jointed loss to learn representations with improved alignment and uniformity. Specifically, DirectAU aligns the representations of observed user-item pairs via the alignment loss, and encourages user and item presentations to distribute uniformly in the respective hyperspheres via the uniformity loss.

However, although the alignment and uniformity loss are two distinct optimized objectives, they jointly constitute the supervised signals in model training – the alignment loss provides the collaborative filtering signals in the interactions, and the uniformity loss prevents the model from collapsing to a trivial solution, where all users and items have identical representations. Since the model is trained with only supervised signals from labeled training data, it can inevitably overfit the noise introduced by label sampling variance, even with *i.i.d.* datasets. This overfitting to noise compromises the learned representations' generalizability in alignment and uniformity, and further the model performance on unseen testing data.

To address the above issue, in this study, we aim to mitigate the model's overfitting to the noise caused by sampling variance in labeled training data, and to learn representations with more generalized alignment and uniformity. Representations with more generalized qualities further lead to improved model performance on testing data. In particular, we propose AU$^+$, a framework that enhances label-irrelevant representation alignment and uniformity by performing self-supervised CL on user and item representations. Specifically, AU$^+$first models the data as a user-item interaction bipartite graph, and applies a graph neural network (GNN) to learn the user and item representations. Then, AU$^+$augments the user and item presentations with the devised 0-layer embedding perturbation mechanism to obtain the positive and negative view pairs. This mechanism minimally yet effectively augments the data without the need of tuning among the classical graph augmentation operators, such as edge dropout and node dropout. Finally, in addition to the alignment and uniformity loss calculated from labeled data, AU$^+$performs self-supervised CL on the augmented user and item representations views to promote label-irrelevant alignment and uniformity. With the enhancement from self-supervised CL, the learned representations are less dependent on label supervision, and can capture label-irrelevant data structures and patterns, leading to more generalized qualities of alignment and uniformity. We conduct extensive experiments on three benchmark datasets to demonstrate that our AU$^+$outperforms existing CF methods with improved performance and faster convergence speed. In summary, the main contributions of this work are as follows:

- We introduce a hypothesis, which states that the sampling variance in labeled training data can compromise the generalizability of learned representations. Based on the hypothesis, we identify the generalizability limitation in the existing CF method from the perspective of representation alignment and uniformity.
- Drawing from our insights regarding the generalizability of alignment and uniformity, we propose AU$^+$, a framework that mitigates the model's overfitting to noise by performing self-supervised CL on the user and item representations from the perspective of alignment and uniformity. Within this framework, we devise a 0-layer embedding perturbation mechanism to perform minimal yet sufficient data augmentation, obviating the need of tuning among the classical graph augmentation operators.
- We conduct comprehensive experiments on three benchmark datasets to show that our AU$^+$not only outperforms the other baseline models, but also converges faster in training. We also present AU$^+$'s training trajectory w.r.t. the alignment and uniformity loss to demonstrate the improved generalizability in alignment and uniformity.

Preliminaries

In this section, we present concepts that are closely related to this work. Specifically, in section *Graph-based Collaborative Filtering* we introduce the common paradigm of graph-based CF methods, where the backbone of our framework belongs to. This approach allows for the integration of self-supervised CL, which enhances label-irrelevant alignment and uniformity. Then in section *Loss Function*, we present the common supervised losses (i.e., the BPR loss and the supervised CL loss) used in training RSs, as well as the auxiliary self-supervised CL loss commonly utilized to enhance model generalizability. Finally in section *Alignment and Uniformity in Representations*, we outline the two important properties, alignment and uniformity, originally identified in CL, and their corresponding losses. For further demonstration purposes, we briefly present how a recent CF method, namely DirectAU (Wang et al. 2022), utilizes the two losses in their model training.

Graph-based Collaborative Filtering

Collaborative filtering in recommendation relies on the collaborative relations among users who interact with the same items to implicitly learn the representations. Specifically, let $\mathcal{U}$ and $\mathcal{I}$ denote the set of users and items respectively. The interaction matrix is denoted as $R \in \{0,1\}^{|\mathcal{U}|\times|\mathcal{I}|}$, where $r_{uv} = 1$ represents an observed interaction between user u and item v, and 0 otherwise. To extract collaborative signals, the interaction matrix R is abstracted to a bipartite graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V} = \mathcal{U} \cup \mathcal{I}$ is the set of nodes and $\mathcal{E} = \{(u,v)|u \in \mathcal{U}, v \in \mathcal{I}, r_{uv} = 1\}$ is the set of edges.

GNN is widely adopted for representation learning on graphs to capture high-order connectivity and has been utilized in multiple domains for general tasks such as node classification (Kipf and Welling 2017; Wen et al. 2022b; Zhang et al. 2023b; Wu et al. 2022), edge prediction (Ouyang et al. 2023), and graph classification (Wen et al. 2022a; Guo et al. 2023; Wen et al. 2024). In addition to utility, some other works improve GNN to improve their properties in robustness (Zhang et al. 2023d; Yuan et al. 2024), fairness (Liu et al. 2023; Jia, Zhang, and Vosoughi 2024), privacy (Liu et al. 2024), and heterogeneity (Li, Zhang, and Zhang 2023; Tian et al. 2023).

In general, at each layer of a GNN and for each node, the information from the neighborhood is aggregated and then combined with the information from the node itself. The combined information is then passed to the next layer for

further aggregation and combination. Information received at each layer is summarized via a readout function at last to obtain the node embeddings. Formally, for any node $i \in \mathcal{V}$, the processes of obtaining the corresponding embedding $\mathbf{z}_i \in \mathbb{R}^m$ where m is the embedding dimension, are formulated as:

$$\begin{aligned}\mathbf{z}_i^{(l)} &= \mathrm{COM}^{(l)}\left(\mathbf{z}_i^{(l-1)}, \mathrm{AGG}^{(l)}\left(\left\{\mathbf{z}_j^{(l-1)}, \forall j \in N_i\right\}\right)\right), \\ \mathbf{z}_i &= \mathrm{READOUT}([\mathbf{z}_i^{(0)}, \mathbf{z}_i^{(1)}, ..., \mathbf{z}_i^{(L)}]),\end{aligned} \tag{1}$$

where $\mathrm{COM}(\cdot)$, $\mathrm{AGG}(\cdot)$, $\mathrm{READOUT}(\cdot)$ are neighbor combination, neighbor aggregation, and readout function respectively, N_i is the neighbor set of node i, $\mathbf{z}_i^{(l)}$ is the embedding of node i at layer l, and L is the number of layers.

To illustrate the learning scheme, we here demonstrate the modeling process of LightGCN (He et al. 2020b), one of the state-of-the-art graph-based RS. Formally, at each layer of LightGCN, the information is aggregated and read out via a simple weighted sum defined as follows:

$$\mathbf{z}_i^{(l+1)} = \sum_{j \in N_i} \frac{1}{\sqrt{|N_i|}\sqrt{|N_j|}} \mathbf{z}_j^{(l)}, \quad \mathbf{z}_i = \sum_{l=0}^{L} a_l \mathbf{z}_i^{(l)}, \tag{2}$$

where a_l is the readout coefficient for each layer-l's output, and is conventionally set to $1/(L+1)$ in this work. After learning the node embeddings, the preference score $\hat{y}_{u,v}$ of item v to user u is calculated as $\hat{y}_{u,v} = \mathbf{z}_u^\intercal \mathbf{z}_v$. Intuitively, the more similar the two representations are, the higher the output score is. Note that the learnable parameters Θ in LightGCN are the initialized embeddings only, i.e., $\Theta = \{\mathbf{z}_u^{(0)}, \mathbf{z}_v^{(0)} | \forall u \in \mathcal{U}, \forall i \in \mathcal{I}\}$.

Loss Function

The Bayesian Personalized Ranking (BPR) loss (Rendle et al. 2009) is a pairwise supervised loss function widely adopted in various CF-based methods (Koren, Bell, and Volinsky 2009; Wang et al. 2019; He et al. 2020b). It encourages the predictions of the observed user-item pairs to be higher than their unobserved counterparts. Formally, it is defined as:

$$\mathcal{L}_{\text{BPR}} = -\sum_{u=0}^{|\mathcal{U}|} \sum_{v \in N_u} \sum_{k \notin N_u} \log \sigma(\hat{y}_{u,v} - \hat{y}_{u,k}), \tag{3}$$

where $\sigma(\cdot)$ is the sigmoid function, $(u, v) \in \mathcal{E}$ are observed pairs, and $(u, k) \notin \mathcal{E}$ are unobserved.

In addition to the BPR loss, the supervised CL loss (Khosla et al. 2020) is also adopted in the realm of personalized recommendation (Yang et al. 2022a; Zhang et al. 2023a) to provide training supervisions. Specifically, the loss function of personalized recommendation with InfoNCE (Oord, Li, and Vinyals 2018) loss can be formulated as follows:

$$\mathcal{L}^s_{\text{InfoNCE}} = -\sum_{v \in N_u} \log \frac{\exp(\hat{y}_{u,v})}{\exp(\hat{y}_{u,v})) + \sum_{k \notin N_u} \exp(\hat{y}_{u,k})}. \tag{4}$$

The contrastive loss in self-supervised CL is widely employed in multiple graph-based CF methods in combination with the BPR loss to improve the generalization ability of the models (Wu et al. 2021; Yu et al. 2022; Lin et al. 2022). It does not require labeled data, and contrasts between views augmented from the original pairs. Formally, let $\mathbf{z}_{i'}, \mathbf{z}_{i''}$ be the two augmented views from the node representation $\mathbf{z}_i$. The self-supervised InfoNCE loss is defined as:

$$\mathcal{L}^u_{\text{InfoNCE}} = -\sum_{i \in \mathcal{V}} \log \frac{\exp(s(\mathbf{z}_{i'}, \mathbf{z}_{i''})/\tau)}{\sum_{j \neq i} \exp(s(\mathbf{z}_{i'}, \mathbf{z}_{j''})/\tau))}, \tag{5}$$

where $s(\cdot)$ is the similarity function, and is set as the cosine similarity function in this work; τ is the the temperature hyper-parameter in softmax function.

Alignment and Uniformity in Representations

A recent study (Wang and Isola 2020) identifies two critical properties of representations - alignment and uniformity - that are closely related to self-supervised contrastive loss described in Eq. 5. The alignment measures the degree of similarity between two node representations, and the uniformity evaluates how uniformly the user and item representations distribute in their respective hyperspheres. A following work in natural language processing named SimCSE (Gao, Yao, and Chen 2021) confirms that representations with better alignment and uniformity lead to better model performance. Formally, given a distribution p_{pos} of positive nodes pairs $(i, j) \sim p_{\text{pos}}$, we align them with the alignment loss defined as the expected distance between the positive pairs over p_{pos}:

$$\ell_{\text{align}} = \mathop{\mathbb{E}}_{(\mathbf{z}_i, \mathbf{z}_j) \sim p_{\text{pos}}} \|f(\mathbf{z}_i) - f(\mathbf{z}_j)\|^2, \tag{6}$$

where $f(\cdot)$ is the $L2$ normalization. Given a data distribution p_{data} for node pairs $(i, j) \sim p_{\text{data}}$, based on the Gaussian potential kernel (Cohn and Kumar 2007), the uniformity loss is defined as the logarithm of the expected pairwise Gaussian potential that measures how well the embeddings distribute uniformly on the hypersphere:

$$\ell_{\text{uniform}} = \log \mathop{\mathbb{E}}_{(\mathbf{z}_i, \mathbf{z}_j) \sim p_{\text{data}}} \exp(-2\|f(\mathbf{z}_i) - f(\mathbf{z}_j)\|^2). \tag{7}$$

To improve alignment and uniformity of representations in personalized recommendation, a recent work in CF named DirectAU (Wang et al. 2022) extends the supervised CL loss described in Eq. 4 to the alignment and uniformity loss defined in Eq. 6 and Eq. 7, respectively. They define the positive node pair distribution p_{pos} as the observed interacted user-item pairs: $(\mathbf{z}_u, \mathbf{z}_v) \sim p_{\text{pos}}$, where $(u, v) \in \mathcal{E}$. They encourage the user and item representations to distribute uniformly in their respective hyperspheres under two data distributions: the user distribution $(\mathbf{z}_{u_1}, \mathbf{z}_{u_2}) \sim p_{\text{user}}$, where $u_1, u_2 \in \mathcal{U}, u_1 \neq u_2$, and the item distribution $(\mathbf{z}_{v_1}, \mathbf{z}_{v_2}) \sim p_{\text{item}}$, where $v_1, v_2 \in \mathcal{I}, v_1 \neq v_2$. The alignment and uniformity loss in DirectAU is formulated as:

$$\begin{aligned}\mathcal{L}_{\text{align}} &= \mathop{\mathbb{E}}_{(u,v) \in \mathcal{E}} \|f(\mathbf{z}_u) - f(\mathbf{z}_v)\|^2, \\ \mathcal{L}_{\text{uniform}} &= \log \mathop{\mathbb{E}}_{\substack{(u_1, u_2) \in \mathcal{U} \\ u_1 \neq u_2}} \exp(-2\|f(\mathbf{z}_{u_1}) - f(\mathbf{z}_{u_2})\|^2)/2 + \\ &\qquad \log \mathop{\mathbb{E}}_{\substack{(v_1, v_2) \in \mathcal{I} \\ v_1 \neq v_2}} \exp(-2\|f(\mathbf{z}_{v_1}) - f(\mathbf{z}_{v_2})\|^2)/2.\end{aligned} \tag{8}$$

The two losses are then linearly combined as the supervision loss for DirectAU to jointly optimize:

$$\mathcal{L}_{\text{DirectAU}} = \mathcal{L}_{\text{alignment}} + \gamma \mathcal{L}_{\text{uniform}}, \tag{9}$$

where γ is the weight coefficient of $\mathcal{L}_{\text{uniform}}$.

Methodology

In this section, we aim to present our proposed framework, namely AU$^+$, that enhances the generalizability from the perspective of alignment and uniformity.

Enhanced Alignment and Uniformity: AU$^+$

While the previous CF method DirectAU is intentionally designed to improve representations from the perspective of alignment and uniformity, the optimized objectives jointly constitutes the supervised signals for model training. This is because the model relies on the alignment loss to learn the collaborative filtering signals among observed interactions, which must be optimized in combination with the uniformity loss, which prevents the model from collapsing into a trivial solution where all users and items have identical representations. This normalization from uniformity also exists in the BPR loss – while it encourages representations of interacted users and items to align to each other, it utilizes the negative pairs to restrain the model from collapsing. Therefore, DirectAU can inevitably overfit the sample variance in labeled training data, impairing the performance on unseen testing data. To mitigate the effect brought by the sample variance, we propose a label-irrelevant self-supervised CL, that enhances the generalizability from the perspective of alignment and uniformity.

The structure of our framework is depicted in Figure 1. Our framework is quite simple – it consists of the supervised and self-supervised CL part. The total losses is then defined as the linear combination of three terms:

$$\mathcal{L}^{\text{AU}^+} = \mathcal{L}^s_{cl} + \lambda_1 \mathcal{L}^u_{cl} + \lambda_2 \|\Theta\|^2, \tag{10}$$

where $\mathcal{L}^s_{cl}$ is the supervised contrastive loss that provides supervised signals from the perspective of alignment and uniformity, and $\mathcal{L}^u_{cl}$ is the self-supervised contrastive loss that promotes label-irrelevant alignment and uniformity, Θ are the learnable parameters, λ_1, λ_2 are the coefficients for the self-supervised loss and the norm of the learnable parameters, respectively. To symbolise $\mathcal{L}^s_{cl}$, in AU$^+$we let it be the alignment and uniformity loss defined in Eq. 8. That is, we let $\mathcal{L}^s_{cl} = \mathcal{L}_{\text{alignment}} + \gamma \mathcal{L}_{\text{uniform}}$.

Then, to utilizes self-supervised CL to promote label-irrelevant alignment and uniformity, we model the user-item interaction histories as a user-item bipartite graph, and adopts LightGCN (He et al. 2020b) as the encoder to obtain the node representations. In order to perform self-supervised CL, two augmented node representation views are first generated, as shown in the left side in Figure 1. The augmentation process does not require labels, and should not impair the representations severely such that it is completely corrupted. This augmentation process is conducted via our devised 0-layer perturbation mechanism, which minimally yet sufficiently augment the node representations. We detail this mechanism in the following section *Maximum Efficacy with Minimal Data Augmentation*. Then, we require views augmented from the same node to align each other, and encourage all node views to distribute uniformly in the hypersphere. There are two ways to fulfill the above requirements: (i) Applying the alignment and uniformity loss defined in Eq. 6-7 to promote label-irrelevant alignment and uniformity. Specifically, the positive node pair distribution p_{pos} is defined as $(\mathbf{z}_{i'}, \mathbf{z}_{i''}) \sim p_{\text{pos}}$, where $\mathbf{z}_{i'}, \mathbf{z}_{i''}$ are the two augmented views from the node representation $\mathbf{z}_i$. The data distribution p_{data} in the uniformity loss is defined as $(\mathbf{z}_{i'^{(\prime)}}, \mathbf{z}_{j'^{(\prime)}}) \sim p_{\text{data}}$, where $\mathbf{z}_{i'^{(\prime)}}, \mathbf{z}_{i'^{(\prime)}}$ are the augmented views from the different node representation $\mathbf{z}_i$ and $\mathbf{z}_j$, respectively. Formally, the auxiliary loss $\mathcal{L}^u_{\text{cl}}$ is formulated as follows:

$$\begin{aligned}
\mathcal{L}^u_{\text{cl}} = & \mathop{\mathbb{E}}_{i \in \mathcal{V}} \|f(\mathbf{z}_{i'}) - f(\mathbf{z}_{i''})\|^2 + \\
& \gamma_p \log \mathop{\mathbb{E}}_{\substack{i,j \in \mathcal{V} \\ i \neq j}} \exp(-2\|f(\mathbf{z}_{i'}) - f(\mathbf{z}_{j'})\|^2)/2 + \\
& \gamma_p \log \mathop{\mathbb{E}}_{\substack{i,j \in \mathcal{V} \\ i \neq j}} \exp(-2\|f(\mathbf{z}_{i''}) - f(\mathbf{z}_{j''})\|^2)/2.
\end{aligned} \tag{11}$$

We denote this variant as **AU$^+$-AU**. (ii) Directly applying the self-supervised contrastive loss defined in Eq. 5 to the augmented views. That is, let $\mathcal{L}^u_{\text{cl}} = \mathcal{L}^u_{\text{InfoNCE}}$. This is our designed **AU$^+$** framework.

Additionally, the classical graph augmentation operators are also suffice the needs for node view generation. To demonstrate the effectiveness of self-supervised CL, we define another model variant named **AU$^+$-SGL**, where the node representations are augmented through edge dropout, and the corresponding self-contrastive loss is calculated via Eq. 5. In the following experiment section, we present the results of the three model variants to show that AU$^+$yields the best performance among all.

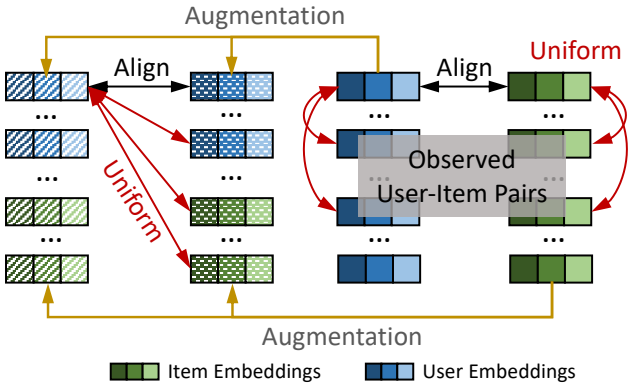

Figure 1: The overall framework of our proposed AU$^+$.

Maximum Efficacy with Minimal Data Augmentation

The process of data augmentation is crucial in representation learning, as the augmented views should preserve the data structure and patterns in the original data. In the realm of

graph augmentation, classical graph augmentation operators are adopted (Wu et al. 2021) such as edge drop, node drop, and random walk. Earlier work (Yu et al. 2022) finds that such operators are not necessary, and SimCSE (Gao, Yao, and Chen 2021) also demonstrates the feasibility of perturbing the initialized embeddings as data augmentation. Therefore, instead of utilizing classical graph augmentation operators to generate the augmented node views, we devise the 0-layer embedding perturbation mechanism that adds randomly generated noise to perturb the initialized embeddings as data augmentation. Specifically, we perturb only the initialized learnable embeddings with a d-dimensional random noise Δ. Formally, the augmented view is created as follows:

$$\mathbf{z}_{u'}^{(0)} = \mathbf{z}_u^{(0)} + \Delta', \mathbf{z}_{u''}^{(0)} = \mathbf{z}_u^{(0)} + \Delta'', \tag{12}$$

where Δ', Δ'' both subject to $\|\Delta\|_2 = \epsilon, \Delta = \bar{\Delta} \odot \text{sign}(\mathbf{z}_u^{(0)})$, and $\bar{\Delta} \in \mathcal{R}^d \sim U(0,1)$. The two perturbed representations are then feed to the encoder to obtain the final perturbed learned embeddings. With this devised mechanism, our AU$^+$is able to learn representation alignment and uniformity through two distinct yet semantically-justified augmented perspectives without tuning among the classical graph operators.

We here additionally show how the original node representation is modified through our augmentation mechanism. Based on the message passing operation defined in Eq. 2, the perturbed node representations are modified as:

$$\begin{aligned}\mathbf{Z}'_{\text{AU}^+} =& \frac{1}{L}\left(\hat{A}(\mathbf{Z}^{(0)} + \Delta) + \ldots + \hat{A}^L(\mathbf{Z}^{(0)} + \Delta)\right) \\ =& \frac{1}{L}\sum_{i=1}^{L}\hat{A}^i\mathbf{Z}^{(0)} + \frac{1}{L}\sum_{i=1}^{L}\hat{A}^i\Delta,\end{aligned} \tag{13}$$

where $\hat{\mathbf{A}}$ is the normalized adjacency of the user-item bipartite graph, and Δ is the generated uniform noise added to the initial representations. The left term $\frac{1}{L}\sum_{i=1}^{L}\hat{A}^i\mathbf{Z}^{(0)}$ are the original node representations. The right term $\frac{1}{L}\sum_{i=1}^{L}\hat{A}^i\Delta$ are the mean of the propagated noises through L layers. Since the noise has a zero mean, the propagated noise (i.e., the right term) only contains structural knowledge in the graph, excluding from any else information. Therefore, the augmented node representations abstract the same collaborative filtering signals as the orignal node representaitons.

Experiments

In this section, we aim to compare our AU$^+$and its variants with other baselines to demonstrate its superiority in model performance and convergence speed. We first outline the experimental settings in this study for reproducibility, and briefly introduce the baseline models. To demonstrate our AU$^+$'s superiority in performance and convergence speed, we then compare AU$^+$and its variants with other CL-based methods, where LightGCN (He et al. 2020b) is utilized as the backbone model. Later on, we additional compare AU$^+$with other CF methods w.r.t. from the perspective of performance. Lastly, we perform an ablation study to show the necessity of combining the designed components. The standard deviation of all reported results is omitted due to their small magnitudes.

Dataset	# User	# Item	# Iteraction	Density
Douban-book	13,024	22,347	792,062	0.00272
Yelp2018	31,668	38,048	1,561,406	0.0013
Amazon-book	52,643	91,599	2,984,108	0.00062

Table 1: Statistics of the benchmark datasets.

Experimental Settings

We select three public benchmark datasets – *Yelp2018* (Wang et al. 2019), *Amazon-book* (Wu et al. 2021), and *Douban-book* (Yu et al. 2021a) – under the public splittings to train and evaluate our model. The statistics of each dataset are outlined in Table 1. We split the public training set with the ratio 8:2 for training and validation, and the model is tested on the public test set. Each model's performance is evaluated by the metrics Recall@K and NDCG@K, and $K = 20$ for all the reported results in this paper. For the baseline models, we first choose methods that adopt LightGCN (He et al. 2020b) as the backbone, and perform self-supervised CL tasks to enhance model generalizability. These models include SGL (Wu et al. 2021), NCL (Lin et al. 2022), and SimGCL (Yu et al. 2022). Furthermore, we compare our framework with models that adopt either different model structures or objectives. The models include BPRMF (Rendle et al. 2009), Mult-VAE (Liang et al. 2018), BUIR (Lee et al. 2021), and DirectAU (Wang et al. 2022). We reproduce the results of DirectAU under the public split, and the results of other methods are copied from the paper for SimGCL. We adopt SELFRec (Yu et al. 2023b) as the code structure for model implementation. All experiments are conducted on an NVIDIA RTX 3090 GPU with 24 GB of memory.

- **BPRMF** (Rendle et al. 2009) learns embeddings by randomly sampling negative items coupled with positive items to optimize the BPR loss.
- **Mult-VAE** (Liang et al. 2018) uses a variational autoencoder and aims to reconstruct the user-item click matrix.
- **LightGCN** (He et al. 2020b) linearly propagates and aggregates the neighborhood information on the user-item bipartite graph.
- **SGL** (Wu et al. 2021) promotes performance through the auxiliary contrast task which maximizes the agreement of each node under different graph-augmented views.
- **SimGCL** (Yu et al. 2022) adjusts the uniformity of the representations by contrasting between the node views, where different uniform noises are added to each layer of the aggregated embeddings.
- **BUIR** (Lee et al. 2021) uses bootstrapping to maintain two encoders that learn from each other and have one approximate the higher-level features learned by the other.
- **NCL** (Lin et al. 2022) optimizes the structure- and semantic-contrastive objectives to capture the layer- and semantic-wise relations among the identified neighbors.
- **DirectAU** (Wang et al. 2022) replaces the BPR loss with the combination of the alignment and uniformity loss, which leads to higher-quality representations.

Method		*Yelp2018*		*Amazon-book*		*Douban-book*	
		Recall	NDCG	Recall	NDCG	Recall	NDCG
1-Layer	LightGCN	0.0631	0.0515	0.0384	0.0298	0.1288	0.1081
	NCL	-	-	-	-	-	-
	SGL	0.0643(1.9%)	0.0529(2.7%)	0.0451(17.4%)	0.0353(18.5%)	0.1658(28.7%)	0.1491(37.9%)
	SimGCL	0.0689(9.2%)	0.0572(11.1%)	0.0453(18.0%)	0.0358(20.1%)	0.1720(33.5%)	0.1519(40.5%)
	AU$^+$-SGL	0.0711(12.7%)	0.0594(15.3%)	0.0504(31.3%)	0.0405(35.9%)	0.1706(32.5%)	0.152(40.6%)
	AU$^+$-AU	**0.0726(15.1%)**	**0.0608(18.1%)**	0.0540(40.6%)	0.0436(46.3%)	0.1746(35.6%))	0.1574(45.6%)
	AU$^+$	0.0725(14.9%)	0.0610(18.4%)	**0.0535(39.3%)**	**0.0432(45.0%)**	**0.1767(37.2%)**	**0.1586(46.7%)**
2-Layer	LightGCN	0.0622	0.0504	0.0411	0.0315	0.1485	0.1272
	NCL	0.0655(5.3%)	0.0545(8.1%)	0.0424(3.2%)	0.0331(5.1%)	0.1628(9.6%)	0.1426(12.1%)
	SGL	0.0668(7.4%)	0.0549(8.9%)	0.0468(13.9%)	0.0371(17.8%)	0.1721(15.9%)	0.1525(19.9%)
	SimGCL	0.0719(15.6%)	0.0601(19.2%)	0.0507(23.4%)	0.0405(28.6%)	0.1770(19.2%)	0.1582(24.4%)
	AU$^+$-SGL	0.0717(15.3%)	0.0601(19.2%)	0.0507(23.4%)	0.0408(29.5%)	0.1756(18.2%)	0.1576(23.9%)
	AU$^+$-AU	0.0729(17.2%)	0.0611(21.2%)	0.0531(29.2%)	0.043(36.5%)	0.1779(19.8%)	0.1602(25.9%)
	AU$^+$	**0.0730(17.4%)**	**0.0613(21.6%)**	**0.0538(30.9%)**	**0.0434(37.8%)**	**0.1804(21.5%)**	**0.1628(28.0%)**
3-Layer	LightGCN	0.0639	0.0525	0.0410	0.0318	0.1392	0.1188
	NCL	0.0666(4.2%)	0.0555(5.7%)	0.0440(7.3%)	0.0341(7.2%)	0.1625(16.7%)	0.1401(17.9%)
	SGL	0.0675(5.6%)	0.0555(5.7%)	0.0478(16.6%)	0.0379(19.2)	0.1732(24.4%)	0.1551(30.6%)
	SimGCL	0.0721(12.8%)	0.0601(14.5%)	0.0515(25.6%)	0.0414(30.2)	0.1772(27.2%)	0.1583(33.2%)
	AU$^+$-SGL	0.0718(12.4%)	0.0600(14.3%)	0.0502(22.4%)	0.0403(26.7%)	0.1737(24.8%)	0.1539(29.6%)
	AU$^+$-AU	0.0726(13.6%)	0.0611(16.4%)	0.0528(28.8%)	0.0427(34.3%)	0.1745(25.4%)	0.1557(31.1%)
	AU$^+$	**0.0730(14.2%)**	**0.0614(17%)**	**0.0536(30.7%)**	**0.0432(35.8%)**	**0.1776(27.6%)**	**0.1597(34.4%)**

Table 2: Performance comparison between the CL-based methods with our model and its variants on the three datasets. The best results are in bold and the runner-ups are underlined. Relative improvements are calculated based on LightGCN. We omit the standard deviation of all reported results due to their small magnitudes.

Comparison with CL-based Methods

We here compare our AU$^+$with the CL-based methods, including SGL, SimGCL, and NCL. The overall performance of CL-based methods with three layer settings are shown in Table 2. We do not further increase the number of layers since models with more than three layers suffer from the over-smoothing problem. For a fair comparison, we reproduce the results of NCL on each dataset with the public splits under either the best hyper-parameter settings reported in the original paper or the one we find via grid search.

Performance Comparison The overall performance comparison with CL-based methods is shown in Table 2:

- Adding CL as the auxiliary task empirically improves the performance of LightGCN, regardless of the augmentation types. The superiority of SimGCL over SGL can be attributed to the layer-wise perturbation mechanism, which preserves some essential collaborative signals that might be corrupted by graph augmentation such as edge drop.
- The performance of NCL is slightly worse than SGL, possibly because the contrasting views between the node and its identified structure and semantic neighbors introduce inductive bias inconsistent with the downstream task.
- In comparison, our model consistently outperforms other CL-based methods. The fact that both our model and its variants yield better performance suggests that the self-supervised CL task does promote the representation generalizability in alignment and uniformity, which is the main reason for better model performance.
- Our AU$^+$and AU$^+$-AU, both of which rely on our 0-layer embedding perturbation for the CL task, generally perform better than AU$^+$-SGL. This is because the perturbation-based augmentation strategy minimally hurts the essential collaborative signals while providing necessary self-supervised signals to promote the generalizability in alignment and uniformity.

Convergence Speed Comparison We compare our model with other CL-based models in terms of convergence speed, and plot each model's learning curve with respect to recall under their best performance settings shown in Figure 2. From the figure, we see that our model achieves nearly state-of-the-art performance after only 5 epochs of training. A slight performance increment can be further obtained after a few more epochs, but 50 epochs are generally sufficient for convergence. In contrast, the performance of LightGCN and NCL slowly increases as the training process proceeds, and evidently needs more epochs for final convergence. While SGL and SimGCL require relatively fewer epochs to converge, their performance fluctuates and is not stabilized after 15 to 20 epochs of training. We credit our model's fast convergence to the self-supervised CL, which helps the model to quickly identify the most general perspective to optimize at the early training stages. The initial high-quality embeddings (evidenced by fast convergence speed with superior

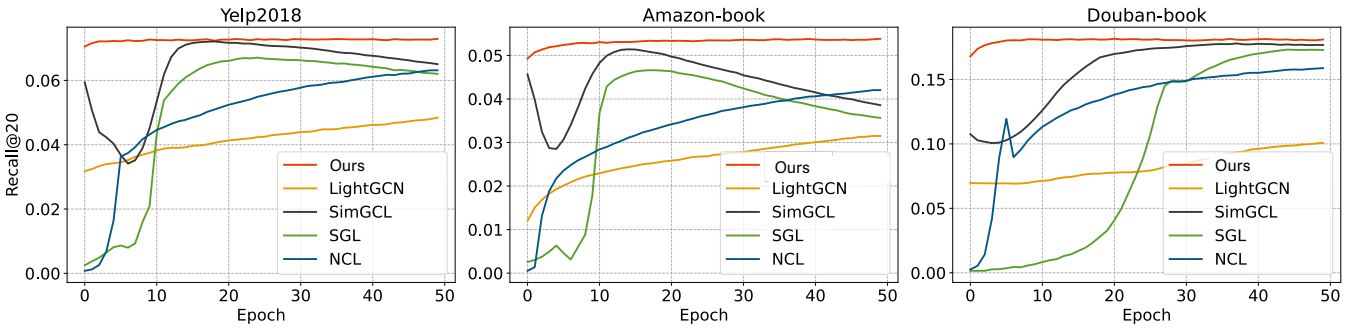

Figure 2: The learning curve w.r.t. recall@20 for the dataset of *Yelp2018*, *Amazon-book*, and *Douban-book*. All curves are plotted based on the corresponding model's best performance setting, and only the previous 50 epochs are shown.

Method	*Yelp2018*		*Amazon-book*		*Douban-book*	
	Recall	NDCG	Recall	NDCG	Recall	NDCG
BPRMF (Rendle et al. 2009)	0.0488	0.0398	0.0298	0.0233	0.1286	0.1051
Mult-VAE (Liang et al. 2018)	0.0584	0.0450	0.0407	0.0315	0.1310	0.1103
LightGCN (He et al. 2020b)	0.0639	0.0525	0.0411	0.0315	0.1485	0.1272
BUIR (Lee et al. 2021)	0.0578	0.0461	0.0423	0.0326	0.1533	0.1317
DirectAU (Wang et al. 2022)	0.0699	0.0593	0.0435	0.03501	0.1623	0.1463
AU^+-SGL	0.0718	0.0600	0.0507	0.0408	0.1746	0.1574
AU^+-AU	0.0729	0.0611	**0.0540**	**0.0436**	0.1779	0.1602
AU^+	**0.0730**	**0.0614**	0.0538	0.0434	**0.1804**	**0.1628**

Table 3: Performance comparison between other methods and our model as well as its variants. The best performance is in bold and the runner-ups are underlined. We omit the standard deviation of all reported results due to their small magnitudes

performance) makes our model a head start, laying out the foundations of the future optimization directions for alignment and uniformity.

Comparison with Other Methods

We compare our model with methods that improve performance from other perspectives such as structure modification and objective function substitution, and the results are shown in Table 3. The results suggest that our model consistently outperforms other methods. We attribute the disadvantaged performance of BPRMF and Mult-VAE to their incapability in capturing high-order connectivity information, which is essential in collaborative filtering. DirectAU outperforms other baselines in that it directly aligns the representations with the supervised signals while restraining the representation uniformity. Representations with better alignment and uniformity have been proven to be effective in previous works (Gao, Yao, and Chen 2021). However, the optimization process is affected by the label variance in the training data, and the model can inevitably overfit the noise. In contrast, our model and its two variants exhibit better performance compared to the baselines, due to the improved generalizability in alignment and uniformity, which is achieved by the self-supervised CL task. Therefore, instead of rigidly optimizing the supervised loss, our framework is able to find a more general optimization path, leading to representations with more generalized properties. Additionally, we note that the performance of AU^+-SGL is slightly inferior to that of AU^+-AU and our AU^+. This discrepancy can be attributed to the fact that graph augmentation operators such as edge drop may hurt the structural knowledge, which is correspondent to the collaborative filtering signals in the bipartite graph in the recommendation case. Therefore, performing self-supervised CL among impaired node representation views can lead to inferior performance. Our proposed 0-level embedding perturbation mechanism circumvents the problem by augmenting the data while retaining the graph structure, thus leading to better performance in $AU^+$and AU^+-AU.

Ablation Study

In this context, we perform an ablation study to demonstrate the necessity of combining the designed components. Specifically, we replace the alignment and uniformity losses from the original representations with the BPR loss, and denote this variant as **CL-BPR**. We remove the alignment and uniformity constraints from the augmented views, and this degenerates our model to **DirectAU**. Each of the ablated variants is tuned to their best performance on each of the datasets. The performance comparison is shown in Table 4. From the table, we see that removing and replacing either of the modules causes performance decrement: (i) While the self-supervised CL task in CL-BPR promotes the alignment and uniformity among the augmented views, the two properties are label-irrelevant. The BPR loss utilized in CL-BPR does not enforce uniformity among the original node representations. Therefore, the representations learned in CL-BPR demonstrate in-

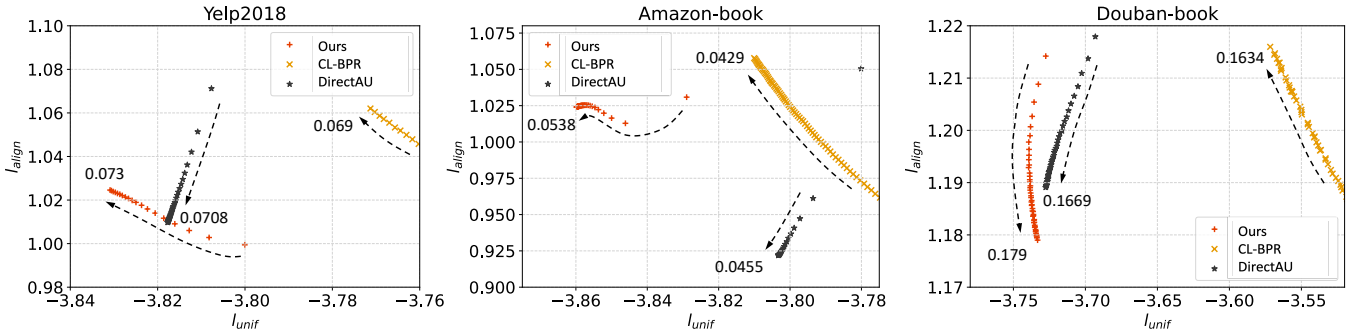

Figure 3: The learning trajectories of CL-BPR, DirectAU, and our model on the three datasets w.r.t. alignment and uniformity losses. The denoted numbers represent the final converged recall@20 and the arrows point to the converging directions.

Method	*Yelp2018*		*Amazon-book*		*Douban-book*	
	Recall	NDCG	Recall	NDCG	Recall	NDCG
CL-BPR	0.0690	0.0573	0.0429	0.0335	0.1634	0.1430
DirectAU	0.0708	0.0592	0.00455	0.0364	0.1669	0.1497
AU^+	**0.0730**	**0.0614**	**0.0538**	**0.0434**	**0.1804**	**0.1628**

Table 4: Performance comparison between our model and its ablated variants: CL-BPR replaces the supervised loss as the BPR loss; DirectAU does not conduct the self-supervised CL task.

sufficient uniformity, making the model yield relatively less ideal performance. (ii) The alignment and uniformity properties demonstrated in the representations learned in DirectAU are extracted from labeled data. This label dependency impairs the representation generalizability, and further leads to inferior model performance on the testing data.

Furthermore, we depict the learning trajectories of these variants concerning the alignment and uniformity losses in Figure 3. From the figure, we see that during the training process: (i) CL-BPR always favor uniformity over alignment – it continuously sacrifices alignment for improved uniformity. (ii) DirectAU always rigidly minimizes the joint of the two losses. (iii) In contrast, our $AU^+$is able to dynamically adjust the optimization object – it neither overly fits the noise in the labeled data by heading towards the directions where the joint loss decreases, nor constantly favors one property over the other. Our AU^+looks for a point where the alignment and uniformity are more generalized, by dynamically adjust the optimization object and directions.

Related Work

GNN-based Recommender Systems

Recently, the advances of graph neural networks (Kipf and Welling 2017; Hamilton, Ying, and Leskovec 2017; Veličković et al. 2018) offer new opportunities for recommender systems to capture high-order structure information in the observed interactions (Gao et al. 2022), making GNN-based recommender systems the new state-of-the-art approaches. For example, GCMC (Berg, Kipf, and Welling 2018) transforms the interaction matrix completion problem into a link prediction problem on the bipartite interaction graph. NGCF (Wang et al. 2019) encodes the collaborative signals into the embedding process for explicitly modeling high-order connectivity. LightGCN (He et al. 2020b) simplifies the design of GCN by removing the linear aggregation weights and the non-linear activation functions in each layer, making the model more concise and appropriate for the recommendation task. In addition, domain knowledge has been utilized as side information to enhance the quality of recommendation (Chen et al. 2019; Wu et al. 2019b,a; Huang et al. 2021). Despite the differences in details, the above methods follow the general idea, which is to gather and propagate neighborhood information for high-order connectivity abstraction. Our work also follows this paradigm – it not only allows the model to capture the structural knowledge in graph, but also fits to our devised augmentation strategy, which minimally yet sufficiently generate augmented node views for self-supervised learning.

Contrastive Learning for Recommendation

Self-supervised contrastive learning was first brought up in the domain of computer vision (Ye et al. 2019; He et al. 2020a; Chen et al. 2020b; Caron et al. 2020), and was quickly adapted to multiple application areas including natural language processing (Gao, Yao, and Chen 2021; Giorgi et al. 2021), graph mining (Liu et al. 2022), as well as recommendation (Wu et al. 2021; Yu et al. 2022; Lee et al. 2021; Lin et al. 2022; Chen et al. 2022), due to its alleviation of the data sparsity issue. Many works have adopted this technique in the realm of the personalized recommendation (Chen et al. 2022; Wu et al. 2021; Lee et al. 2021; Yu et al. 2022; Lin et al. 2022). For example, ICL (Chen et al. 2022) leverages the EM algorithm to learn latent intent variables and maximizes the agreement of a view with its intent variable. SGL (Wu et al.

2021) relies on graph augmentation such as node drop, edge drop, and random walk to create contrastive views. They also theoretically analyze that self-supervised contrastive learning with InfoNCE loss mines hard negative samples by properly tuning the temperature hyperparameter. BUIR (Lee et al. 2021) relieves the burden of negative sampling to create contrastive views by maintaining two distinct encoders that learn from each other. SimGCL (Yu et al. 2022) creates contrastive views by adding uniform distributed noises to every layer of LightGCN. They also find that this auxiliary task improves the user/item embedding uniformity, which not only mitigates the popularity bias but also improves the training performance and efficiency. NCL (Lin et al. 2022) leverages the EM algorithm to learn the neighbors of a node in the structure space, and its semantic prototype in the semantic space. Positive contrastive views are created between the node and its structure neighbors and semantic prototype. The general paradigm of the contrastive learning that the above models follow is to first identify the label-invariant views that filter irrelevant noises with respect to the downstream task, and then improve model robustness by pulling positive views together and pushing negative views away. Apart from the fact that our model also follows this paradigm, we further focus on the coherent effects between the main and auxiliary tasks from the perspective of representation properties, i.e., alignment and uniformity. In addition, our devised augmentation strategy is free from the requirement of traditional graph augmentation as most of the previous works.

Conclusion

In this paper, we revisit the representations learned in the current CF-based method, and identify their label dependency from the perspective of alignment and uniformity. The identified label dependency can inevitably results in the model's overfitting to the noise in labeled data, and further compromises the model's generalizability to unseen testing data. To mitigate such label dependency, we propose AU$^+$, a framework that utilizes self-supervised CL to improve the generalizability in representation alignment and uniformity. Within this framework, we devise the 0-layer perturbation mechanism, that minimally yet sufficiently augments the data for self-supervised CL, circumventing the requirement of classical graph augmentation operators. We conduct extensive experiments over three benchmark datasets to demonstrate the superiority of our AU$^+$. Results show that the integration of self-supervised CL enhances the generalizability of the learned representations from the perspective of alignment and uniformity, leading to improved performance and faster convergence speed. Finally, we provide further discussions regarding our work's limitations and impacts in Appendix.

Acknowledgments

This work was supported by the NSF under grant CMMI-2146076. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the sponsors.

References

Berg, R. v. d.; Kipf, T. N.; and Welling, M. 2018. Graph convolutional matrix completion. In *SIGKDD*.

Burke, R. 2002. Hybrid recommender systems: Survey and experiments. *User modeling and user-adapted interaction*.

Caron, M.; Misra, I.; Mairal, J.; Goyal, P.; Bojanowski, P.; and Joulin, A. 2020. Unsupervised learning of visual features by contrasting cluster assignments. In *NeurIPS*.

Chen, C.; Zhang, M.; Wang, C.; Ma, W.; Li, M.; Liu, Y.; and Ma, S. 2019. An efficient adaptive transfer neural network for social-aware recommendation. In *SIGIR*.

Chen, C.; Zhang, M.; Zhang, Y.; Ma, W.; Liu, Y.; and Ma, S. 2020a. Efficient heterogeneous collaborative filtering without negative sampling for recommendation. In *AAAI*.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020b. A simple framework for contrastive learning of visual representations. In *ICML*.

Chen, Y.; Liu, Z.; Li, J.; McAuley, J.; and Xiong, C. 2022. Intent contrastive learning for sequential recommendation. In *The Web Conference*.

Cohn, H.; and Kumar, A. 2007. Universally optimal distribution of points on spheres. *Journal of the American Mathematical Society*.

Covington, P.; Adams, J.; and Sargin, E. 2016. Deep neural networks for youtube recommendations. In *RecSys*.

Dong, X.; Yu, L.; Wu, Z.; Sun, Y.; Yuan, L.; and Zhang, F. 2017. A hybrid collaborative filtering model with deep structure for recommender systems. In *AAAI*.

Gao, C.; Zheng, Y.; Li, N.; Li, Y.; Qin, Y.; Piao, J.; Quan, Y.; Chang, J.; Jin, D.; He, X.; et al. 2022. A Survey of Graph Neural Networks for Recommender Systems: Challenges, Methods, and Directions. *ACM Transactions on Recommender Systems*.

Gao, T.; Yao, X.; and Chen, D. 2021. SimCSE: Simple Contrastive Learning of Sentence Embeddings. In *EMNLP*.

Giorgi, J.; Nitski, O.; Wang, B.; and Bader, G. 2021. DeCLUTR: Deep Contrastive Learning for Unsupervised Textual Representations. In *ACL*.

Guo, Z.; Zhang, C.; Fan, Y.; Tian, Y.; Zhang, C.; and Chawla, N. V. 2023. Boosting Graph Neural Networks via Adaptive Knowledge Distillation. In *AAAI Conference on Artificial Intelligence*.

Hamilton, W.; Ying, Z.; and Leskovec, J. 2017. Inductive representation learning on large graphs. In *NIPS*.

He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. 2020a. Momentum contrast for unsupervised visual representation learning. In *CVPR*.

He, X.; Deng, K.; Wang, X.; Li, Y.; Zhang, Y.; and Wang, M. 2020b. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *SIGIR*.

Hsieh, C.-K.; Yang, L.; Cui, Y.; Lin, T.-Y.; Belongie, S.; and Estrin, D. 2017. Collaborative metric learning. In *WWW*.

Huang, C.; Xu, H.; Xu, Y.; Dai, P.; Xia, L.; Lu, M.; Bo, L.; Xing, H.; Lai, X.; and Ye, Y. 2021. Knowledge-aware coupled graph neural network for social recommendation. In *AAAI*.

Jia, Y.; Zhang, C.; and Vosoughi, S. 2024. Aligning Relational Learning with Lipschitz Fairness. In *ICLR*.

Khosla, P.; Teterwak, P.; Wang, C.; Sarna, A.; Tian, Y.; Isola, P.; Maschinot, A.; Liu, C.; and Krishnan, D. 2020. Supervised contrastive learning. In *NeurIPS*.

Kipf, T. N.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR*.

Koren, Y.; Bell, R.; and Volinsky, C. 2009. Matrix factorization techniques for recommender systems. *IEEE Computer*.

Lee, D.; Kang, S.; Ju, H.; Park, C.; and Yu, H. 2021. Bootstrapping user and item representations for one-class collaborative filtering. In *SIGIR*.

Li, J.; Zhang, C.; and Zhang, C. 2023. Heterogeneous Temporal Graph Neural Network Explainer. In *ACM International Conference on Information and Knowledge Management*.

Liang, D.; Krishnan, R. G.; Hoffman, M. D.; and Jebara, T. 2018. Variational autoencoders for collaborative filtering. In *The Web Conference*.

Lin, Z.; Tian, C.; Hou, Y.; and Zhao, W. X. 2022. Improving Graph Collaborative Filtering with Neighborhood-enriched Contrastive Learning. In *The Web Conference*.

Liu, Y.; Jin, M.; Pan, S.; Zhou, C.; Zheng, Y.; Xia, F.; and Yu, P. 2022. Graph self-supervised learning: A survey. *IEEE Transactions on Knowledge and Data Engineering*.

Liu, Z.; Dou, G.; Tian, Y.; Zhang, C.; Chien, E.; and Zhu, Z. 2024. Breaking the Trilemma of Privacy, Utility, and Efficiency via Controllable Machine Unlearning. In *WWW*.

Liu, Z.; Zhang, C.; Tian, Y.; Zhang, E.; Huang, C.; Ye, Y.; and Zhang, C. 2023. Fair graph representation learning via diverse mixture-of-experts. In *WWW*.

Lops, P.; De Gemmis, M.; and Semeraro, G. 2011. Content-based recommender systems: State of the art and trends. *Recommender systems handbook*.

McAuley, J.; Targett, C.; Shi, Q.; and Van Den Hengel, A. 2015. Image-based recommendations on styles and substitutes. In *SIGIR*.

Oord, A. v. d.; Li, Y.; and Vinyals, O. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Ouyang, Z.; Hou, S.; Ma, S.; Chen, C.; Zhang, C.; Li, T.; Xiao, X.; Zhang, C.; and Ye, Y. 2023. Prompt Learning Unlocked for App Promotion in the Wild. In *NeurIPS 2023 Workshop: New Frontiers in Graph Learning*.

Qian, Y.; Zhang, C.; Zhang, Y.; Wen, Q.; Ye, Y.; and Zhang, C. 2022. Co-Modality Graph Contrastive Learning for Imbalanced Node Classification. In *Advances in Neural Information Processing Systems*.

Rendle, S.; Freudenthaler, C.; Gantner, Z.; and Schmidt-Thieme, L. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *The Conference on Uncertainty in Artificial Intelligence*.

Schafer, J. B.; Frankowski, D.; Herlocker, J.; and Sen, S. 2007. Collaborative filtering recommender systems.

Sun, J.; Zhang, Y.; Guo, W.; Guo, H.; Tang, R.; He, X.; Ma, C.; and Coates, M. 2020. Neighbor interaction aware graph convolution networks for recommendation. In *SIGIR*.

Tay, Y.; Luu, A. T.; and Hui, S. C. 2018. Multi-pointer co-attention networks for recommendation. In *SIGKDD*.

Tian, Y.; Dong, K.; Zhang, C.; Zhang, C.; and Chawla, N. V. 2023. Heterogeneous graph masked autoencoders. In *AAAI Conference on Artificial Intelligence*.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph Attention Networks. In *ICLR*.

Wang, C.; Yu, Y.; Ma, W.; Zhang, M.; Chen, C.; Liu, Y.; and Ma, S. 2022. Towards Representation Alignment and Uniformity in Collaborative Filtering. In *SIGKDD*.

Wang, H.; Shi, X.; and Yeung, D.-Y. 2016. Collaborative recurrent autoencoder: Recommend while learning to fill in the blanks.

Wang, T.; and Isola, P. 2020. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *ICML*.

Wang, X.; He, X.; Wang, M.; Feng, F.; and Chua, T.-S. 2019. Neural graph collaborative filtering. In *SIGIR*.

Wen, Q.; Ju, M.; Ouyang, Z.; Zhang, C.; and Ye, Y. 2024. Graph Representation Learning with Multi-granular Semantic Ensemble.

Wen, Q.; Ouyang, Z.; Zhang, C.; Qian, Y.; Ye, Y.; and Zhang, C. 2022a. Graph Contrastive Learning with cross-view Reconstruction. In *NeurIPS 2022 Workshop: New Frontiers in Graph Learning*.

Wen, Q.; Ouyang, Z.; Zhang, J.; Qian, Y.; Ye, Y.; and Zhang, C. 2022b. Disentangled dynamic heterogeneous graph learning for opioid overdose prediction. In *SIGKDD*.

Weston, J.; Bengio, S.; and Usunier, N. 2011. Wsabie: Scaling up to large vocabulary image annotation.

Weston, J.; Yee, H.; and Weiss, R. J. 2013. Learning to rank recommendations with the k-order statistic loss. In *Proceedings of the 7th ACM Conference on Recommender Systems*.

Wu, F.; Souza, A.; Zhang, T.; Fifty, C.; Yu, T.; and Weinberger, K. 2019a. Simplifying graph convolutional networks. In *ICML*.

Wu, J.; Wang, X.; Feng, F.; He, X.; Chen, L.; Lian, J.; and Xie, X. 2021. Self-supervised graph learning for recommendation. In *SIGIR*.

Wu, J.; Zhang, C.; Liu, Z.; Zhang, E.; Wilson, S.; and Zhang, C. 2022. Graphbert: Bridging graph and text for malicious behavior detection on social media. In *ICDM*.

Wu, L.; Sun, P.; Fu, Y.; Hong, R.; Wang, X.; and Wang, M. 2019b. A neural influence diffusion model for social recommendation. In *SIGIR*.

Xia, X.; Yin, H.; Yu, J.; Wang, Q.; Cui, L.; and Zhang, X. 2021. Self-supervised hypergraph convolutional networks for session-based recommendation. In *AAAI*.

Yang, C.; Zou, J.; Wu, J.; Xu, H.; and Fan, S. 2022a. Supervised contrastive learning for recommendation. *Knowledge-Based Systems*.

Yang, M.; Li, Z.; Zhou, M.; Liu, J.; and King, I. 2022b. Hicf: Hyperbolic informative collaborative filtering. In *SIGKDD*.

Ye, M.; Zhang, X.; Yuen, P. C.; and Chang, S.-F. 2019. Unsupervised embedding learning via invariant and spreading instance feature. In *CVPR*.

Ying, R.; He, R.; Chen, K.; Eksombatchai, P.; Hamilton, W. L.; and Leskovec, J. 2018. Graph convolutional neural networks for web-scale recommender systems. In *SIGKDD*.

Yu, J.; Xia, X.; Chen, T.; Cui, L.; Hung, N. Q. V.; and Yin, H. 2023a. XSimGCL: Towards extremely simple graph contrastive learning for recommendation. *IEEE Transactions on Knowledge and Data Engineering*.

Yu, J.; Yin, H.; Gao, M.; Xia, X.; Zhang, X.; and Viet Hung, N. Q. 2021a. Socially-aware self-supervised tri-training for recommendation. In *SIGKDD*.

Yu, J.; Yin, H.; Li, J.; Wang, Q.; Hung, N. Q. V.; and Zhang, X. 2021b. Self-supervised multi-channel hypergraph convolutional network for social recommendation. In *The Web Conference*.

Yu, J.; Yin, H.; Xia, X.; Chen, T.; Cui, L.; and Nguyen, Q. V. H. 2022. Are graph augmentations necessary? simple graph contrastive learning for recommendation. In *SIGIR*.

Yu, J.; Yin, H.; Xia, X.; Chen, T.; Li, J.; and Huang, Z. 2023b. Self-supervised learning for recommender systems: A survey. *IEEE Transactions on Knowledge and Data Engineering*.

Yu, L.; Zhang, C.; Liang, S.; and Zhang, X. 2019. Multi-order attentive ranking model for sequential recommendation. In *AAAI*.

Yuan, X.; Zhang, C.; Tian, Y.; Ye, Y.; and Zhang, C. 2024. Mitigating Severe Robustness Degradation on Graphs. In *ICLR*.

Yue, H.; Zhang, C.; Zhang, C.; and Liu, H. 2022. Label-invariant Augmentation for Semi-Supervised Graph Classification. In *Advances in Neural Information Processing Systems*.

Zhang, A.; Sheng, L.; Cai, Z.; Wang, X.; and Chua, T.-S. 2023a. Empowering Collaborative Filtering with Principled Adversarial Contrastive Loss.

Zhang, C.; Huang, C.; Tian, Y.; Wen, Q.; Ouyang, Z.; Li, Y.; Ye, Y.; and Zhang, C. 2023b. When sparsity meets contrastive models: less graph data can bring better class-balanced representations. In *ICML*.

Zhang, C.; Liu, H.; Li, J.; Ye, Y.; and Zhang, C. 2023c. Mind the Gap: Mitigating the Distribution Gap in Graph Few-shot Learning. *Transactions on Machine Learning Research*.

Zhang, C.; Tian, Y.; Ju, M.; Liu, Z.; Ye, Y.; Chawla, N.; and Zhang, C. 2023d. Chasing all-round graph representation robustness: Model, training, and optimization. In *ICLR*.

Zhang, F.; Yuan, N. J.; Lian, D.; Xie, X.; and Ma, W.-Y. 2016. Collaborative knowledge base embedding for recommender systems. In *SIGKDD*.

A Hyper-parameter Setting

For all the baselines, we either refer to the best hyper-parameter settings in the original papers or tune the parameters through grid search. Overall, we add a $L2$ regularization to each of the models and set the regularization coefficient

Table 5: Additional hyper-parameter settings for the reproduced baselines and our model.

Model	Hyper-parameter	*Yelp2018*	*Amazon-book*	*Douban-book*
SGL	r	0.1	0.1	0.2
	τ	0.2	0.2	0.2
	λ_1	0.1	0.5	0.1
	L	3	3	3
SimGCL	ϵ	0.1	0.1	0.2
	λ_1	0.5	2	0.2
	L	3	3	3
NCL	τ	0.05	0.05	0.05
	λ_1	1e-6	1e-6	1e-6
	α	1.5	0.8	1.5
	λ_{proto}	1e-7	1e-7	1e-7
	n_c	2000	2000	2000
	L	3	3	3
	λ_2	1e-4	1e-6	1e-4
DirectAU	γ	2	1.5	0.5
	L	3	3	3
AU$^+$	λ_1	0.5	2	0.2
	ϵ	0.2	0.2	0.05
	τ	0.2	0.2	0.2
	γ	1	1	0.3
	L	3	2	2
AU$^+$-SGL	λ_1	1	1	0.2
	ϵ	0.2	0.2	0.2
	τ	0.2	0.2	0.2
	γ	2	2	0.5
	r	0.2	0.2	0.2
	L	3	2	2
AU$^+$-AU	λ_1	1	2	0.5
	ϵ	0.2	0.2	0.1
	τ	0.2	0.2	0.2
	γ	2	2	0.5
	γ_p	1	2	1
	L	3	2	2

λ_2 as $1e{-}4$. The batch size is set to 2048 and we use Adam optimizer with a learning rate $1e{-}3$. For DirectAU, AU$^+$and its variants, we tune γ from $\{0.2, 0.5, 1.0, 2.0, 3.0\}$, and λ_1 from $\{0.2, 0.5, 1.0, 2.0\}$. We also tune ϵ from $\{0.01, 0.05, 0.1, 0.2, 0.5\}$ for our model. Following the original setting for SGL and SimGCL, we set the temperature τ as 0.2 and keep it the same for AU$^+$and its variants for a fair comparison. Table 5 shows the detailed hyper-parameter settings.

B Hyper-parameter Sensitivity

In this context, we analyze our model's performance sensitivity with respect to the supervised uniformity coefficient γ and the unsupervised contrastive loss coefficient λ_1.

B.1 Supervised Uniformity Coefficient γ.

We test the performance sensitivity of our model with respect to the hyperparameter γ, and show the results in Table 4. We note that γ for *Yelp2018* and *Amazon-book* ranges from $\{0.1, 0.2, 0.5, 1, 2\}$, and for *Douban-book* it ranges from $\{0.2, 0.3, 0.5, 1, 2\}$. From the figure, we see that our model is more sensitive on the dataset *Douban-book* with respect to the hyperparameter γ, and is less sensitive on the other two datasets. We attribute the difference to the sizes of the datasets - the larger the dataset is, the less sensitive it is to γ. In addition, we observe that although the performance differs as γ changes, within a certain range our model is still able to achieve decent performance. For example, on *Amazon-book*, $\gamma = 1/\gamma = 2$ yields similar state-of-the-art performance. We credit this to the InfoNCE loss for the self-supervised contrastive task, which implicitly promotes representation uniformity. Furthermore, we notice that the larger the dataset is, the larger γ it is for the model in terms of the final performance. This might be because the representations of a smaller dataset is a smaller community, and therefore more sensitive to uniformity loss.

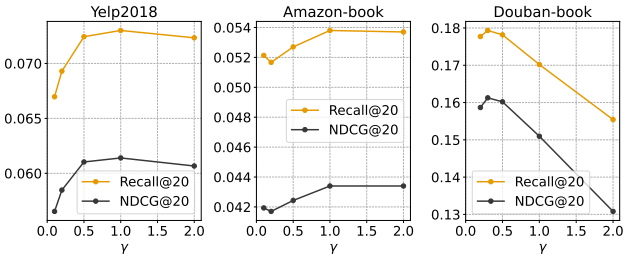

Figure 4: Our model's sensitivity with respect to γ.

Contrastive Loss Coefficient λ_1. We test our model's performance sensitivity concerning the hyperparameter λ_1 and show the results in Figure 5. Note that the test range for each dataset differs, and we select the ranges among which the best possible performance might lie based on the estimation via the first few epochs' results. From the figure, we see that our model shows up different levels of sensitivity to different datasets. For *Yelp2018*, the performance does not fluctuate violently with λ_1 ranging from 0.1 to 1. In contrast, our model displays relatively higher sensitivity towards *Amazon-book*

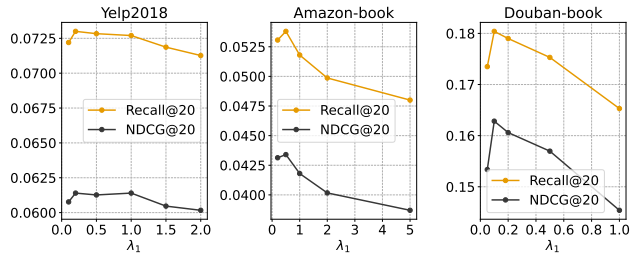

Figure 5: Our model's sensitivity with respect to λ_2.

and *Douban-book* in terms of λ_1. We observe that a value around 0.2 generally yields good performance for all the datasets. For models with λ_1 that is too small, they cannot exploit the benefits of the unsupervised RAU loss, therefore gradually degenerate to DirectAU, which yields worse performance than ours according to Table 3. Models with λ_1 that are too large bring about performance decrement, in that they pay too much effort in aligning the noises while ignoring the essential supervised signals.

C Limitations

While we introduce a hypothesis, which states that the sampling variance in labeled training data can compromise the generalizability of learned representations, the hypothesis is only verified empirically. Further analysis can include how the sample variance affects the representations, demonstrated statistically or visually. Additionally, we do not trial how the noise distribution affects the functionality of the devised 0-layer perturbation mechanism. According to SimCSE (Gao, Yao, and Chen 2021), performing dropout along the hidden dimension of the representations is an alternative in augmenting the data. Future research may investigate other efficient and effective data augmentation strategy in CF-based methods, which should not be limited to graph-based ones.

D Broader Impacts

Our research on representation alignment and uniformity draws the model training attention towards the representation side, where the alignment and uniformity property are widely recognized crucial ones determining the representation qualities. However, representations possess more than one properties, and should not be defined by the two only. Each property is related to each downstream task differently, and the relationship varies by dataset as well. We hope this work raise more research attention towards the properties of representations, as well as their relations towards downstream tasks, so as to universally improve models' generalizability.