A Head Start Matters: Dynamic-Calibrated Representation Alignment and Uniformity for Recommendations

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

Abstract

The Bayesian personalized ranking (BPR) loss is a commonly used objective in training recommender systems, upon which various auxiliary graph-based self-supervised contrastive learning tasks are designed for improved model robustness. Previous research has also shown that the unsupervised contrastive loss shapes the learned representations from the perspectives of *alignment* and *uniformity*, and representations with lower supervised alignment and/or uniformity loss contribute to better model performance. Despite the progress, no one neither explores how the two representation qualities evolve along the learning trajectory, nor associates the behaviors with the combination of supervised and unsupervised representation alignment and uniformity (RAU). In this work, we first observe that different methods trades of alignment and uniformity to varying degrees, and hypothesize that optimizing over supervised RAU loss alone is not sufficient for an optimal trade-off. Then, by analyzing how BPR loss relates to the unsupervised contrastive loss where the supervised RAU loss stems from, we migrate the relation to propose our framework which aligns embeddings from both supervised and unsupervised perspectives while promoting user/item embedding uniformity on the hypersphere. Within the framework, we design a 0-layer embedding perturbation to the neural network on the user-item bipartite graph for minimal yet sufficient data augmentation, discarding the traditional ones such as edge drop. Extensive experiments on three datasets show that our framework improves model performance and quickly converges to user/item embeddings.

[1]University of Notre Dame [2]Brandeis University.

Preliminary work. Under review by *MFPL'23: The Many Facets of Preference-based Learning Workshop @ ICML'23*, Honolulu, Hawaii, USA. Do not distribute.

1. Introduction

The development of recommender systems (RSs)has been widely explored to assist information filtering that alleviates the data overload issue among multiple fields (McAuley et al., 2015; Covington et al., 2016). The goal of a recommender system is to abstract historical data and predict future interactions given current observations. Based on the modeling of the preference, RSs are categorized into content-based models (Lops et al., 2011; Tay et al., 2018), and collaborative filtering (CF) based models (Schafer et al., 2007; Chen et al., 2020a; Yang et al., 2022; Wang et al., 2019; He et al., 2020b; Wu et al., 2021; Lee et al., 2021; Lin et al., 2022), which performs representation learning on the users and items based on the observed user-item interaction histories. On the other hand, Bayesian personalized ranking (BPR) loss, which encourages the posterior probabilities of the observed interactions to be higher than their unobserved counterparts, is widely used for CF models, including matrix factorization (Koren et al., 2009) and other graph-based methods (Ying et al., 2018; Wang et al., 2019; Yu et al., 2019; Sun et al., 2020; He et al., 2020b). Recently inspired by the resurgence of contrastive learning (CL) in deep representation learning (Jaiswal et al., 2020), multiple recent works (Wu et al., 2021; Yu et al., 2021b; Xia et al., 2021; Lin et al., 2022) design one/several unsupervised auxiliary contrastive task(s) to jointly optimized for improved model performance and robustness. Moving beyond, some other works shift their attention toward how the CL loss influences the learned representations (Wang & Isola, 2020; Wang et al., 2022). In spite of the success, no one has explored the reason why in general, directly optimizing over supervised RAU loss yields better results than the BPR loss, let alone exploiting the empirical observation and extending it beyond the supervised side from both the supervised and unsupervised perspectives. They either only consider the unsupervised representation uniformity (Yu et al., 2022), or only the supervised alignment of the observed user-item pairs, along with controlled attention towards improving embedding uniformity (Wang et al., 2022).

With the above identified research gaps, in this work we first compare the representation qualities of the existing models with respect to alignment and uniformity losses, after which

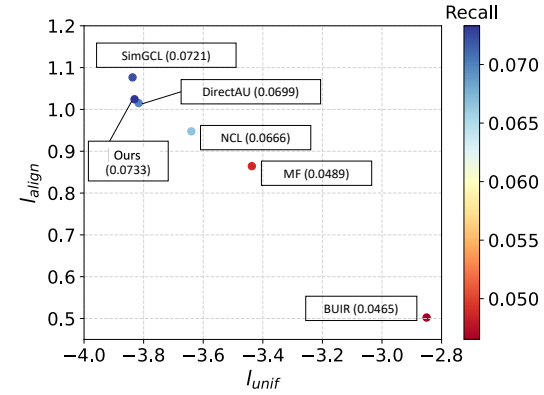
a further inspection of the embedding learning trajectories of the leading models in terms of the two losses shows that optimizing merely the supervised RAU loss is not sufficient for an ideal trade-off between the two properties. Then, we analyze the relationship between the BPR loss and the unsupervised CL loss to explain why models optimized over the supervised RAU loss work better empirically, and further hypothesize that the unsupervised alignment of the embeddings between the contrastive views is able to fill up the deficiency of the supervised RAU loss. Based on the hypothesis, we propose our framework along with two variants that similarly suffice the requirements. In addition, we design a 0-layer embedding perturbation to perform minimal yet sufficient data augmentation without specifying and tuning among all the graph augmentation types such as edge drop. Empirical results show that our proposed model outperforms existing models in terms of with better initial and final representation qualities, as well as remarkably rapid convergence rate. In summary, the main contributions of this work are:

- We analyze the relationship between the BPR loss and unsupervised contrastive loss where the supervised RAU loss stems from, based on which we reach the explanation of why generally models optimized with supervised RAU loss perform better.
- Based on empirical observations, we show that optimizing over merely supervised RAU loss is neither enough to make the initial embeddings a head start in terms of performance, nor direct the model to a converged point with an ideal trade-off between the two properties. In light of these, we propose a novel framework that jointly optimizes the RAU loss with regards to both supervised and unsupervised aspects.
- We conduct comprehensive experiments on three benchmark datasets to show that our model outperforms other methods at the very start of the training process, which leads to a more stabilized learning trajectory with a better ultimate balance between alignment and uniformity, contributing to not only performance improvement but also fast convergence speed.

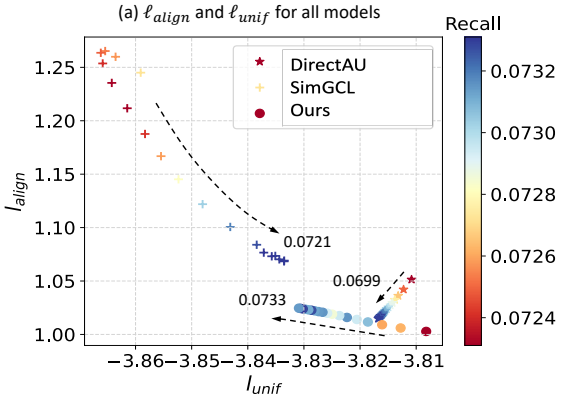
2. Methodology

2.1. Motivation

A recent study (Wang & Isola, 2020) identifies alignment and uniformity as critical properties of representations, which are closely related to unsupervised contrastive loss. The contrastive loss is commonly used as an auxiliary loss, along with the BPR loss, to enhance model performance and robustness (Ye et al., 2019; He et al., 2020a; Chen et al., 2020b; Caron et al., 2020). As a result, prior works (Yu et al., 2022; Wang et al., 2022) focus on improving performance from the perspective of alignment and uniformity. To understand how the two properties affect the model performance, we show the values of the two associated losses calculated from different models' learned embeddings with each model's best performance on Yelp2018 (Wang et al., 2019) - a benchmark dataset for recommendation - in Figure 1(a). From the subfigure, we see that each model subtly trades off between the two losses, leading to different levels of performance with respect to recall. Models like BUIR, MF, and NCL yield relatively smaller alignment losses, while their corresponding uniformity losses are high. In contrast, models like DirectAU, SimGCL, including our method partially sacrifice embedding alignment to favor smaller uniformity loss. Although (Gao et al., 2021; Wang et al., 2022) propose that embeddings with smaller alignment and/or uniformity losses usually lead to better performance, one might still wonder how each model balances the two along the training trajectory. Furthermore, since DirectAU directly minimizes the two losses in the training process without strongly favoring either of them, it is questionable whether this objective is sufficient for the model to converge to a status where the embeddings ideally trade off one property for the other.



(a) ℓ_{align} and ℓ_{unif} for all models



(b) ℓ_{align} and ℓ_{unif} for SimGCL, DirectAU and our method.

Figure 1. The alignment and uniformity losses of different models. (a) Each point represents the corresponding embeddings of the converged model; (b) The arrows point to the converging directions, and the recalls@20 of the converged points are denoted aside.

To answer the above questions, we then take a closer look at the learning trajectories with respect to the two losses of the leading models SimGCL (Yu et al., 2022) and DirectAU (Wang & Isola, 2020), including our model in Figure 1 (b). Through the subfigure, we see that the learning curves of the three models differ: DirectAU directly optimizes the weighted combination of alignment and uniformity altogether, while the others (SimGCL and ours) start from a favor point of one metric, i.e., one with low alignment loss or low uniformity loss, and then sacrifice the advantaged metric for the improvement of the disadvantaged one. Since each model is already under its best hyperparameters setting and the end of the trajectory shows the converging point, we posit that simply optimizing the weighted combination of losses for the two embedding properties as DirectAU may not be sufficient to converge to a point that balances both properties effectively. In addition, it is not obvious how to determine the favorable metric explicitly, since the training behaviors are implicitly affected by the objective function. Hence, identifying an optimal trade-off point between alignment and uniformity, or which property to prioritize over the other at the initial training procedure, is crucial in guiding the model toward a desirable performance outcome.

2.2. Comparison between $\mathcal{L}_{\text{BPR}}$ and $\mathcal{L}_{\text{cl}}$

In this subsection, we compare the BPR loss ($\mathcal{L}$BPR) with the contrastive loss ($\mathcal{L}$cl) to demonstrate how the addition of unsupervised RAU loss to supervised RAU loss implicitly addresses the problem mentioned in the previous subsection. For simplicity, we omit the superscript in the notation for the negative sample distribution of a certain node u. Specifically for a positive user-item pair (u, v), the part it contributes to $\mathcal{L}_{\text{BPR}}$ for the recommendation task is:

$$\mathcal{L}_{\text{BPR}}(u,v) = -\sum_{\{\mathbf{z}_i^-\}_{i=1}^M \overset{\text{i.i.d.}}{\sim} p_{\text{neg}}} \log\sigma(s(\mathbf{z}_u,\mathbf{z}_v) - s(\mathbf{z}_u,\mathbf{z}_i^-)), \tag{1}$$

where $s(\cdot)$ is a similarity function. Let $\mathcal{V}'$ and $\mathcal{V}''$ denote the node set under two different augmented views. For a positive pair created for node u with respect to the two views, the contrastive loss is defined as:

$$\mathcal{L}_{\text{cl}}(u) = -\log\frac{e^{s(\mathbf{z}_{u'},\mathbf{z}_{u''})/\tau}}{e^{s(\mathbf{z}_{u'},\mathbf{z}_{u''})/\tau} + \sum_{\{\mathbf{z}_i^-\}_{i=1}^M \overset{\text{i.i.d.}}{\sim} p_{\text{neg}'}} e^{s(\mathbf{z}_{u'},\mathbf{z}_i^-)/\tau}}, \tag{2}$$

where $p_{\text{neg}'}$ is the negative sample embedding distribution from view $\mathcal{V}'$ with respect to node u, i.e., $\{\mathbf{z}_k \sim p'_{\text{neg}} | k \in \mathcal{V}', k \neq u\}$. Setting aside the difference between positive and negative pairs, we see that $\mathcal{L}_{\text{BPR}}(u,v)$ pulls its positive user-item pairs closer via pairwise similarity ranking loss, while $\mathcal{L}_{\text{cl}}(u)$ does so to its view-view (user-user or item-item) pairs via the softmax function. In other words, $\mathcal{L}_{\text{BPR}}(u,v)$ normalizes through the sigmoid function, while $\mathcal{L}_{\text{cl}}(u)$ normalizes through the softmax function. However, unlike $\mathcal{L}_{\text{BPR}}(u,v)$ which utilizes pairwise signals, $\mathcal{L}_{\text{cl}}$ implicitly promotes the uniformity through the uniformed normalization of all batched views in the softmax function. Since supervised RAU extends from unsupervised $\mathcal{L}_{\text{cl}}$, we believe that the implicit uniformity promotion in $\mathcal{L}_{\text{DirectAU}}$ contributes to the more superior empirical performance than $\mathcal{L}_{\text{BPR}}$. Despite the extra uniformity promotion in $\mathcal{L}_{\text{DirectAU}}$, it statically weighs the supervised RAU along the learning trajectory, and neglects the unsupervised RAU. Given the merits of unsupervised contrastive learning stated in (Wu et al., 2021), we suspect that incorporating unsupervised RAU with supervised RAU dynamically calibrates the training trajectory and adjusts the weighing of the two properties via the introduced inductive bias during the training process. This dynamic calibration not only enables the model to implicitly uncover the optimal initial favorable property, but aslo filters the noise residing in the original data, which greatly affects models that directly optimize over supervised RAU loss. In the experiment section, we empirically verify that the addition of unsupervised RAU loss indeed effectively calibrates the learning process.

2.3. Our Proposed Model

In light of the above motivation, in this part, we introduce our model named **D**ynamic-calibrated Representation **A**lignment and Un**I**formity for **R**ecommendation (*DAIR*), which utilizes the inductive bias brought by unsupervised contrastive learning (RAU) to calibrate the weighing along the training trajectory. To enhance representation properties, we first substitute the BPR loss with the supervised RAU loss for supervised RAU, followed by the integration of an additional unsupervised RAU loss for the training process calibration. Trivially, adding the graph-augmented contrastive loss as SGL suffices the requirements. Generating graph-augmented contrastive views, however, entails the selection of augmentation type among node drop, edge drop, and random walk, as well as determining the augmented ratio, resulting in extra hyperparameter tuning efforts. Inspired by (Gao et al., 2021) which finds that the last dropout layer performs minimal data augmentation, we propose the 0-layer embedding perturbation to create the contrastive views upon the framework of LightGCN. Specifically, we perturb the 0-layer initialized learnable embeddings with a d-dimensional random noise Δ. Formally, the augmented view is created as follows:

$$\mathbf{z}_{u'}^{(0)} = \mathbf{z}_u^{(0)} + \Delta', \mathbf{z}_{u''}^{(0)} = \mathbf{z}_u^{(0)} + \Delta'', \tag{3}$$

where Δ', Δ'' both subject to $\|\Delta\|_2 = \epsilon$, $\Delta = \bar{\Delta} \odot \text{sign}(\mathbf{z}_u^{(0)})$, and $\bar{\Delta} \in \mathcal{R}^d \sim U(0,1)$. We obtain the final

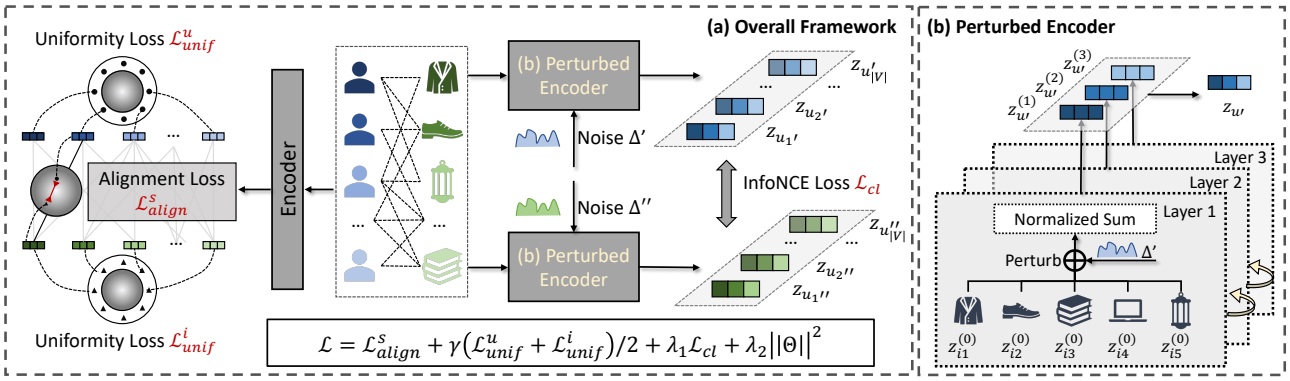


Figure 2. (a) The overall framework of our DAIR. For supervised signals, it aligns the embeddings of the observed user-item pairs, and promotes their uniformity on the hypersphere; for unsupervised signals, it maximizes the agreement of the same node of different views through InfoNCE loss. (b) 0-layer embeddings are perturbed by randomly sampled noise.

embedding following the propagation rule of LightGCN:

$$\mathbf{z}_u = \frac{1}{L+1}\sum_{i=0}^{L}\tilde{\mathbf{A}}^i\mathbf{z}_u^{(0)}, \tag{4}$$

where $\tilde{\mathbf{A}} \in \mathcal{R}^{|\mathcal{V}|\times|\mathcal{V}|}$ is the normalized undirected adjacency matrix of the bipartite graph, and L is the number of layers. We choose InfoNCE (Oord et al., 2018) loss as the contrastive loss between the two views, which has been shown effective in many self-supervised representation learning works (Bachman et al., 2019; Chen et al., 2020b; Wu et al., 2021). Specifically, the contrastive InfoNCE loss is defined as:

$$\mathcal{L}_{cl} = \sum_{u\in\mathcal{V}} -\log\frac{\exp\left(s\left(\mathbf{z}'_u, \mathbf{z}''_u\right)/\tau\right)}{\sum_{v\in\mathcal{V}, v\neq u}\exp\left(s\left(\mathbf{z}'_u, \mathbf{z}''_v\right)/\tau\right)}, \tag{5}$$

where $s(\cdot)$ is the cosine similarity function, and τ is the hyperparameter temperature in the softmax function. The total loss of *DAIR* is defined as the weighted summation of the above losses plus a regularization term:

$$\mathcal{L} = \mathcal{L}^s_{\text{align}} + \gamma\mathcal{L}^s_{\text{uniform}} + \lambda_1\mathcal{L}_{\text{cl}} + \lambda_2\|\Theta\|^2, \tag{6}$$

and our framework is shown in Figure 2. The rationale of our framework lies in that we rely on self-supervised contrastive learning to calibrate the learning trajectory directed by the supervised RAU loss. We circumvent the requirement of classical graph augmentation by the 0-layer embedding perturbation, inspired by the last-layer dropout augmentation previously proven effective.

2.4. Theoretical Comparison with SimGCL

Compared with SimGCL, our method only adds the noise at the 0-layer embeddings, while SimGCL adds the noise to every aggregated layer, starting from layer 1 to layer L. In this part, we aim to theoretically analyze why our 0-level embedding perturbation improves upon SimGCL and performs minimal yet sufficient data augmentation. Formally, let $\mathbf{Z}^{(0)}$ be the embedding matrix at layer 0, i.e., the initialized learnable embedding matrix of all nodes. SimGCL creates one contrastive view $\mathbf{Z}'_{\text{sim}}$ following the propogation rule as follows:

$$\begin{aligned}\mathbf{Z}'_{\text{sim}} =& \frac{1}{L}\left[\left(\tilde{\mathrm{A}}\mathbf{Z}^{(0)} + \Delta^{(1)}\right)\right.\\ &\left.+\left(\tilde{\mathrm{A}}\left(\tilde{\mathrm{A}}\mathbf{Z}^{(0)} + \Delta^{(1)}\right) + \Delta^{(2)}\right) + \ldots\right]\\ =& \frac{1}{L}\left[\left(\hat{A}\mathbf{Z}^{(0)} + \Delta^{(1)}\right) + \right.\\ &\left.\left(\hat{A}^2\mathbf{Z}^{(0)} + \hat{A}\Delta^{(1)} + \Delta^{(2)}\right) + \ldots\right]\\ =& \frac{1}{L}\sum_{i=1}^{L}\hat{A}^i\mathbf{Z}^{(0)} + \frac{1}{L}\sum_{i=1}^{L}\left(I + \sum_{j=1}^{L-i}\hat{A}^j\right)\Delta^{(i)},\end{aligned} \tag{7}$$

where $\Delta^{(i)}$ is the noise generated at layer i. The first term in Eq. 7 is the weighted sum of the propagated embeddings via the rule of LightGCN, where each layer has a weight $1/L$. The second term in Eq. 7 is essentially the average of the summation of the propagated noise up to $L-i$ layer, where i is the layer at which the noise is generated, and we denote it as ϵ_{sim}. For example, for noise that is randomly generated at layer l, the term it contributes to the summation is $\sum_{i=1}^{l}\hat{\Delta}^{(i)}$, where $\hat{\Delta}^{(i)} = \Delta^{(i)} + \hat{A}\Delta^{(i)} + \ldots + \hat{A}^{L-i}\Delta^{(i)}$. Therefore, the noise SimGCL adds to the final embedding is actually the average of the summation of multiple propagated noise to different extents, according to the layer number where the noise is generated. In contrast, our 0-layer embedding perturbation modifies the final $\mathbf{Z}'_{\text{DAIR}}$:

$$\begin{aligned}\mathbf{Z}'_{\text{DAIR}} &= \frac{1}{L}\left[\hat{A}(\mathbf{Z}^{(0)} + \Delta) + \hat{A}^2(\mathbf{Z}^{(0)} + \Delta) + \ldots\right]\\ &= \frac{1}{L}\sum_{i=1}^{L}\hat{A}^i\mathbf{Z}^{(0)} + \frac{1}{L}\sum_{i=1}^{L}\hat{A}^i\Delta,\end{aligned} \tag{8}$$

where Δ is the generated uniform noise at layer 0. Similarly, the first term in Eq. 8 is the embeddings obtained via the original LightGCN model. The second term is the averaged

propagated noise up to layer i, and we denote it as ϵ_{DAIR}. In comparison with ϵ_{DAIR}, we believe there exists redundant information in ϵ_{sim} that reduces the effectiveness of the added noises. Since Δ and $\Delta^{(i)}$ follow the same distribution, in ϵ_{DAIR} the coefficient $\hat{A}^i$ within the inner iteration in front of Δ actually contains the information included in the convolutional coefficient $I + \sum_{j=1}^{L-i} A^j$ in front of $\Delta^{(i)}$ in ϵ_{sim}. In other words, ϵ_{DAIR} already contains information from multiple levels of the propagated noise, while ϵ_{sim} repeats the propagation for every noise generated at each layer, given that they essentially follow the same uniform distribution. This redundancy brings extra learning complexity into the contrastive learning task, and is eliminated by our 0-layer embedding perturbation by only generating the noise at the 0-th layer. Since the generated noise is propagated to each level's extent for data augmentation, and each level's propagated noise has its own contribution to the averaged term, our perturbation makes sure each level's information is augmented while simplifying the information atoms, making the embedding learning process easier.

2.5. Model Variants and Time Complexity

Model Variants. To empirically prove that inductive bias from unsupervised contrastive learning calibrates the learning process in a positive way, we add several variants for comparison. We denote the variant which creates the contrastive views through classic graph augmentation as *DAIR-SGL*. Furthermore, (Wang & Isola, 2020) compares the self-supervised effect among $\mathcal{L}_{\text{cl}}$, the combination of $\mathcal{L}_{\text{align}}$ and $\mathcal{L}_{\text{uniform}}$, as well as the combination of three losses. Experiments show that the combination of $\mathcal{L}_{\text{align}}$ and $\mathcal{L}_{\text{uniform}}$ sometimes brings about more desirable effects to the model with respect to mean squared error. Therefore, here we denote the model variant which replaces $\mathcal{L}_{\text{cl}}$ with $\mathcal{L}_{\text{align}} + \gamma_p \mathcal{L}_{\text{uniform}}$ as *DAIR-AU*. Specifically, the contrastive loss is defined as:

$$\begin{aligned}\mathcal{L}_{\text{cl}}^{au} = &\mathop{\mathbb{E}}_{(\mathbf{z}_{u'}, \mathbf{z}_{u''}) \sim p_{\text{cl}}} \|f(\mathbf{z}_{u'}) - f(\mathbf{z}_{u''})\|^2 + \\ \gamma_p \log &\mathop{\mathbb{E}}_{(\mathbf{z}_{u'}, \mathbf{z}_{v'}) \sim p'_{\text{data}}, u \neq v} e^{-2\|f(\mathbf{z}_{u'}) - f(\mathbf{z}_{v'})\|^2}/2 \quad + \\ \gamma_p \log &\mathop{\mathbb{E}}_{(\mathbf{z}_{u''}, \mathbf{z}_{v''}) \sim p''_{\text{data}}, u \neq v} e^{-2\|f(\mathbf{z}_{u''}) - f(\mathbf{z}_{v''})\|^2}/2,\end{aligned} \tag{9}$$

where p_{cl} is the distribution of the same node in two augmented views, $(\mathbf{u}', \mathbf{v}')$ and $(\mathbf{u}'', \mathbf{v}'')$ are node pairs within the same view.

3. Experiments

In this part, we aim to show the superiority of our framework compared with other baselines from the perspectives of model performance and convergence speed. In addition, we perform an ablation study to show the necessity of jointly optimizing both the supervised and unsupervised RAU losses. We refer the readers of results of the ablation study to Appendix E.3 Our codes can be accessed through this anonymous link.[1]

3.1. 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. We split the public training set with the ratio 8:2 for training and validation, and the model is tested on the test set. Each model's performance is evaluated by the metrics Recall@K and NDCG@K, and $K = 20$, and each reported result is the average over 5 repeated experiments under the same hyperparameter setting. We reproduce the results of DirectAU under the public split, and the results of other methods are copied from the paper for SimGCL.

We first compare our model and its two variants with methods built upon vanilla LightGCN (He et al., 2020b) with auxiliary CL tasks, namely SGL (Wu et al., 2021), NCL (Lin et al., 2022), and SimGCL (Yu et al., 2022). We further select BPRMF (Rendle et al., 2012), Mult-VAE (Liang et al., 2018), BUIR (Lee et al., 2021), and DirectAU (Wang et al., 2022) as other baselines that boost the performance from the perspective of framework or objective modification. More details regarding the experimental settings are outlined in Appendix D

3.2. Comparison with CL-based Methods

We compare our model with the CL-based methods, including SGL, SimGCL, and NCL, and the overall performance of CL-based methods with two layer settings are shown in Table 4. Additional layer settings are provided in Appendix E.1. 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 hyperparameter settings reported in the original paper or the one we find via grid search.

From the table, we observe that:

- Adding CL as the auxiliary task empirically improves the performance of LightGCN, regardless of the augmentation types.

- We credit the improvement of SGL to the fact that the unsupervised contrastive learning loss improves the embedding uniformity. Similarly, SimGCL improves the embedding uniformity through layer-wise noise perturbation. The superiority of SimGCL over SGL can be attributed to the layer-wise perturbation mechanism, which preserves some essential collaborative

[1] https://tinyurl.com/DAIR

Table 1. 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.

Method	Yelp2018		Amazon-book		Douban-book	
	Recall	NDCG	Recall	NDCG	Recall	NDCG
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%)
DAIR-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%)
DAIR-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%)
DAIR	**0.0730(14.2%)**	**0.0614(17%)**	**0.0536(30.7%)**	**0.0432(35.8)**	**0.1776(27.6%)**	**0.1597(34.4%)**

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 that is inconsistent with the downstream task.

- In comparison, our model consistently outperforms other CL-based methods. The fact that both DAIR and its variants yield better performance proves our hypothesis, which is adding unsupervised RAU (whether via directly optimizing the RAU losses or through the unsupervised contrastive loss) to the supervised RAU helps the model better trade-off between embedding alignment and uniformity.

- Our model and *DAIR-AU*, both of which rely on our 0-layer embedding perturbation for the CL task, generally perform better than *DAIR-SGL*, in that the perturbation-based augmentation minimally hurts the essential collaborative signals while providing necessary unsupervised signals.

3.3. Comparison with Other Methods

In this part, we compare our model with methods that improve performance from other perspectives such as structure modification and objective function substitution. The results are shown in Table 2. According to the table, we see 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. BUIR yields better performance than LightGCN, possibly due to the inductive bias injected by stochastic data augmentation. DirectAU outperforms other baselines in that it directly considers the supervised RAU losses, which is proven to be effective from the previous statement. In comparison, our model including its two variants outperforms the baselines mainly in that they all consider supervised and unsupervised RAU, although they are distinct from each other in the manner of how they receive the unsupervised signals. It is also noted that the performance of *Bias-SGL* is slightly inferior to that of *DAIR-AU* and our proposed model. This discrepancy can be attributed to the limitations of the graph augmentation approach discussed previously, and further supports the validity of our 0-level embedding perturbation method.

Table 2. Performance comparison between different other models with our model and its variants. The best performance is in bold and the runner-ups are underlined.

Method	Yelp2018		Amazon-book		Douban-book	
	Recall	NDCG	Recall	NDCG	Recall	NDCG
BPRMF (Rendle et al., 2012)	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
DAIR-SGL	0.0718	0.0600	0.0507	0.0408	0.1746	0.1574
DAIR-AU	0.0729	0.0611	**0.0540**	**0.0436**	0.1779	0.1602
DAIR	**0.0730**	**0.0614**	0.0538	0.0434	**0.1804**	**0.1628**

4. Conclusion

In this paper, we revisit the representation alignment and uniformity problem for the recommendation task and investigate the progressive curves of the two related losses along the model's learning trajectories. We demonstrate the significance of representations' properties with respect to *alignment* and *uniformity* from both supervised and unsupervised perspectives, and such properties are essential in the whole training processes that determine the final model performance. We propose an inductive bias-calibrated RAU mechanism to combine both the supervised and unsupervised RAU and dynamically calibrates the trade-off of the two properties, leading to decent initial and superior final performance. Additionally, we design a 0-layer embedding perturbation for minimal yet sufficient data augmentation for the unsupervised contrastive task jointly trained with the supervised RAU task. Extensive experiments show that the combination of the supervised and unsupervised RAU equips our model with performance improvement, stable learning properties, and fast convergence speed.

References

Bachman, P., Hjelm, R. D., and Buchwalter, W. Learning representations by maximizing mutual information across views. 2019.

Berg, R. v. d., Kipf, T. N., and Welling, M. Graph convolutional matrix completion. *arXiv preprint arXiv:1706.02263*, 2017.

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

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

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

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

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

Cohn, H. and Kumar, A. Universally optimal distribution of points on spheres. *Journal of the American Mathematical Society*, 20(1):99–148, 2007.

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

Gao, C., Zheng, Y., Li, N., Li, Y., Qin, Y., Piao, J., Quan, Y., Chang, J., Jin, D., He, X., et al. A survey of graph neural networks for recommender systems: Challenges, methods, and directions. *ACM Transactions on Recommender Systems*, 2022.

Gao, T., Yao, X., and Chen, D. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*, 2021.

Giorgi, J., Nitski, O., Wang, B., and Bader, G. Declutr: Deep contrastive learning for unsupervised textual representations. *arXiv preprint arXiv:2006.03659*, 2020.

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

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

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

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

Jaiswal, A., Babu, A. R., Zadeh, M. Z., Banerjee, D., and Makedon, F. A survey on contrastive self-supervised learning. *Technologies*, 9(1):2, 2020.

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Koren, Y., Bell, R., and Volinsky, C. Matrix factorization techniques for recommender systems. *Computer*, 42(8): 30–37, 2009.

Lee, D., Kang, S., Ju, H., Park, C., and Yu, H. Bootstrapping user and item representations for one-class collaborative filtering. In *SIGIR*, pp. 317–326, 2021.

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

Lin, Z., Tian, C., Hou, Y., and Zhao, W. X. Improving graph collaborative filtering with neighborhood-enriched contrastive learning. In *WWW*, 2022.

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

Lops, P., De Gemmis, M., and Semeraro, G. Content-based recommender systems: State of the art and trends. *Recommender systems handbook*, pp. 73–105, 2011.

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

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

Rendle, S., Freudenthaler, C., Gantner, Z., and Schmidt-Thieme, L. Bpr: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618*, 2012.

Schafer, J. B., Frankowski, D., Herlocker, J., and Sen, S. Collaborative filtering recommender systems. pp. 291–324, 2007.

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

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

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., and Bengio, Y. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

Wang, C., Yu, Y., Ma, W., Zhang, M., Chen, C., Liu, Y., and Ma, S. Towards representation alignment and uniformity in collaborative filtering. In *SIGKDD*, 2022.

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

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

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

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

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

Wu, Z., Wang, S., Gu, J., Khabsa, M., Sun, F., and Ma, H. Clear: Contrastive learning for sentence representation. *arXiv preprint arXiv:2012.15466*, 2020.

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

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

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

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

Yu, J., Yin, H., Gao, M., Xia, X., Zhang, X., and Viet Hung, N. Q. Socially-aware self-supervised tri-training for recommendation. In *SIGKDD*, pp. 2084–2092, 2021a.

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

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

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

A. Related Work

In this part, we briefly review the works that are closely related to this paper, namely GNN-based recommender systems and contrastive learning for recommendation.

A.1. GNN-based Recommender Systems

Recently, the advances of graph neural networks (Kipf & Welling, 2016; Hamilton et al., 2017; Veličković et al., 2017) 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 et al., 2017) 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 modeling high-order connectivity in an explicit manner. 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. Beyond the previous work, we relate the properties of the representations and the downstream task and design the framework that learns representations well align with the supervised signals while preserving high uniformity.

A.2. Contrastive Learning for Recommendation

Unsupervised 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 (Wu et al., 2020; Giorgi et al., 2020), 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. Specifically for the recommendation task, 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 identify invariant views which filter irrelevant noises with respect to the downstream task, and improve model robustness by pulling them together. 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 embedding properties, and is free from the requirement of traditional graph augmentation as most of the previous work.

B. Preliminaries

In this section, we first formalize the graph-based collaborative filtering problem, and concisely introduce the vanilla LightGCN (He et al., 2020b). Then, we present the measurements of RAU.

B.1. 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. For each user $u \in \mathcal{U}$, let $\boldsymbol{z}_u \in \mathbb{R}^d$ be its learned embedding, and let $\boldsymbol{z}_v \in \mathbb{R}^d$ be the learned embedding for each item $v \in \mathcal{I}$. To extract collaborative signals, the interaction matrix R is usually 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.

Graph neural network (GNN) is one of the most widely adopted methods for representation learning for it captures high-order connectivity information. In general, at each layer of a GNN, the neighborhood information is aggregated and combined, and information received at each layer is summarized via a readout function at last:

$$\begin{aligned}\mathbf{z}_v^{(l)} &= \mathrm{COM}^{(l)}\left(\mathbf{z}_v^{(l-1)}, \mathrm{AGG}^{(l)}\left(\left\{\mathbf{z}_u^{(l-1)}, \forall u \in N_v\right\}\right)\right), \\ \mathbf{z}_v &= \mathrm{READOUT}([\mathbf{z}_v^{(0)}, \mathbf{z}_v^{(1)}, ..., \mathbf{z}_v^{(L)}]),\end{aligned} \tag{10}$$

where $\mathrm{COM}(\cdot), \mathrm{AGG}(\cdot), \mathrm{READOUT}(\cdot)$ are neighbor combination, neighbor aggregation, and readout function respectively, N_v is the neighbor set of node v, $\mathbf{z}_v^{(l)}$ is the embedding of node v at layer l, and L is the number of layers. LightGCN (He et al., 2020b) is empirically proven to be effective in capturing collaborative signals. It aggregates and reads out the information via a simple weighted sum. Specifically, the aggregation and readout function is defined as follows:

$$\mathbf{z}_u^{(l+1)} = \sum_{v \in N_u} \frac{1}{\sqrt{|N_u|}\sqrt{|N_v|}} \mathbf{z}_v^{(l)}, \quad \mathbf{z}_u = \sum_{l=0}^{L} a_l \mathbf{z}_u^{(l)}, \tag{11}$$

where a_l is the readout coefficient for each layer-l's embedding, and is usually set to $1/(L+1)$. After learning the embedding of each node, the preference score of item v to user u can be either directly calculated via $\hat{y}_{u,v} = \mathbf{z}_u^T \mathbf{z}_v$, or through a preference function $\hat{y}_{u,v} = f_p(\mathbf{z}_u, \mathbf{z}_v)$. The BPR loss is widely adopted as the objective for training the models. It encourages the similarity scores of the observed interaction pairs to be higher than the unobserved ones. Formally, it is defined as:

$$\mathcal{L}_{\text{BPR}} = -\sum_{u=0}^{|\mathcal{U}|} \sum_{v \in N_u} \sum_{k \notin N_u} \log \sigma(\hat{y}_{u,v} - \hat{y}_{u,k}), \tag{12}$$

where $\sigma(\cdot)$ is the sigmoid function. We also note that the learnable parameters Θ in LightGCN are the 0-layer initialized embeddings, i.e., $\Theta = \{\mathbf{z}_u^{(0)}, \mathbf{z}_v^{(0)} | \forall u \in \mathcal{U}, \forall i \in \mathcal{I}\}$.

B.2. Representation Alignment and Uniformity

Recent study (Wang & Isola, 2020) identifies two critical properties of the representations - alignment and uniformity - that are closely related to unsupervised contrastive loss. Formally, the unsupervised contrastive loss of two views is defined as:

$$\mathcal{L}_{\text{cl}}(f; \tau, M) \triangleq \mathop{\mathbb{E}}_{\substack{(\mathbf{z}, \mathbf{z}^+) \sim p_{\text{pos}} \\ \{\mathbf{z}_i^-\}_{i=1}^{M} \overset{\text{i.i.d.}}{\sim} p_{\text{neg}}}} \left[-\log \frac{e^{f(\mathbf{z})^\top f(\mathbf{z}^+)/\tau}}{e^{f(\mathbf{z})^\top f(\mathbf{z}^+)/\tau} + \sum_i e^{f(\mathbf{z}_i^-)^\top f(\mathbf{z})/\tau}} \right], \tag{13}$$

where $\mathbf{z}$ is the embedding of one view, τ is the hyperparameter temperature which tunes the level of matching, M is the number of negative samples for one positive pair, p_{pos} is the distribution for positive node pairs from two augmented views, and p_{neg} is the distribution for negative sampling. (Gao et al., 2021) empirically proves that under contrastive loss, representations that align positive pairs and distribute evenly in the hypersphere lead to better model performance. Specifically, the alignment loss is defined as the expected distance between the positive pairs over p_{pos}:

$$\ell_{\text{align}} \triangleq \mathop{\mathbb{E}}_{(\mathbf{z}, \mathbf{z}^+) \sim p_{\text{pos}}} \|f(\mathbf{z}) - f(\mathbf{z}^+)\|^2, \tag{14}$$

where $f(\cdot)$ is the $L2$ normalization. Based on the Gaussian potential kernel (Cohn & Kumar, 2007), the uniformity loss is defined as the logarithm of the expected pairwise Gaussian potential:

$$\ell_{\text{uniform}} \triangleq \log \mathop{\mathbb{E}}_{(\mathbf{z}_u, \mathbf{z}_v) \sim p_{\text{data}}} e^{-2\|f(\mathbf{z}_u) - f(\mathbf{z}_v)\|^2}, \tag{15}$$

where p_{data} is the distribution over the pairwise node embeddings, and the uniformity loss measures how well the embeddings distribute uniformly on the hypersphere. Based on the relationship between the contrastive loss and the two properties, (Wang et al., 2022) extends the unsupervised contrastive learning towards a supervised loss named DirectAU that directly minimizes the representation alignment of the observed user-item pairs as well as the uniformity of the user/item embeddings. Specifically, the two losses DirectAU optimizes are:

$$\mathcal{L}_{\text{align}}^s = \mathop{\mathbb{E}}_{(\mathbf{z}_u, \mathbf{z}_v) \sim p_{\text{pos}}^s} \|f(\mathbf{z}_u) - f(\mathbf{z}_v)\|^2, \tag{16}$$

$$\mathcal{L}^s_{\text{uniform}} = \log \mathop{\mathbb{E}}_{(\mathbf{z}_u, \mathbf{z}_{u'}) \sim p_{\text{user}}} e^{-2\|f(\mathbf{z}_u) - f(\mathbf{z}_{u'})\|^2}/2 \quad + \log \mathop{\mathbb{E}}_{(\mathbf{z}_v, \mathbf{z}_{v'}) \sim p_{\text{item}}} e^{-2\|f(\mathbf{z}_v) - f(\mathbf{z}_{v'})\|^2}/2, \tag{17}$$

where p^s_{pos} is the observed user-item distribution, p_{user} and p_{item} are user and item embedding distributions respectively. The supervised loss for DirectAU is then defined as the weighted combination of the two losses:

$$\mathcal{L}_{\text{DirectAU}} = \mathcal{L}^s_{\text{alignment}} + \gamma \mathcal{L}^s_{\text{uniform}}, \tag{18}$$

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

C. Time Complexity

Since the calculation of the uniformity loss and the InfoNCE loss both involves pairwise embedding distance, their runtime is of the same order. If we treat all other nodes in a batch other than itself as the negative samples, within one batch, the runtime of *DAIR* is $O(2B^2d)$, where B is the batch size and d is the embedding dimension. The time mainly depends on the calculation of the supervised uniformity loss, the unsupervised uniformity loss for *DAIR-AU*, and the InfoNCE loss for *DAIR* and *DAIR-SGL*. Although theoretically, *DAIR* is no faster than the previous CL-based methods, we show in the experiment section that *DAIR* converges much faster than the other methods, therefore requiring less time to achieve desirable model performance.

D. Experimental Details

D.1. Dataset Statistics

Table 3. Statistics of the datasets.

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

D.2. Baselines

- **BPRMF** (Rendle et al., 2012) learns embeddings by randomly sampling negative items coupled with positive items to optimize the BPR loss.
- **Mult-VAE** (Liang et al., 2018) is based on a variational auto-encoder 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 contrasting learning 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 node views where different uniform noises are added to each layer of the aggregated embeddings.
- **BUIR** (Lee et al., 2021) exploits bootstrapping to maintain two encoders that learn from each other and have one approximate the higher level features learned from 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 with respect to the two properties.

D.3. Hyperparameters

For all the baselines, we either refer to the best hyperparameter 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 λ_2 as $1e{-}4$. The batch size is set to 2048 and we use Adam optimizer with a learning rate $1e{-}3$. Following the original setting for SGL and SimGCL, we set the temperature τ as 0.2 and keep it the same for BiasuAU and its variants for a fair comparison. More detailed hyperparameter settings are provided in the appendix.

E. Additional Experiments

E.1. Additional CL-based Comparison

Table 4 further shows the comparison of our framework with other CL-based methods under the three layer settings. We note that NCL requires the encoder layer larger than 1 to calculate the structure-contrastive loss, therefore the results for NCL with 1-layer are omitted. Results show that our DAIR still outperforms other methods by a large margin. We credit this to the auxiliary CL task, which calibrates the model learning process even under relatively under-fitted models.

Table 4. 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.

	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%)
	DAIR-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%)
	DAIR-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%)
	DAIR	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%)
	DAIR-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%)
	DAIR-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%)
	DAIR	**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%)
	DAIR-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%)
	DAIR-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%)
	DAIR	**0.0730(14.2%)**	**0.0614(17%)**	**0.0536(30.7%)**	**0.0432(35.8)**	**0.1776(27.6%)**	**0.1597(34.4%)**

E.2. Convergence Speed Comparison

In this part, we aim to 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 3. For comparison purposes, we keep the number of epochs as 50. 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. Consequently, the

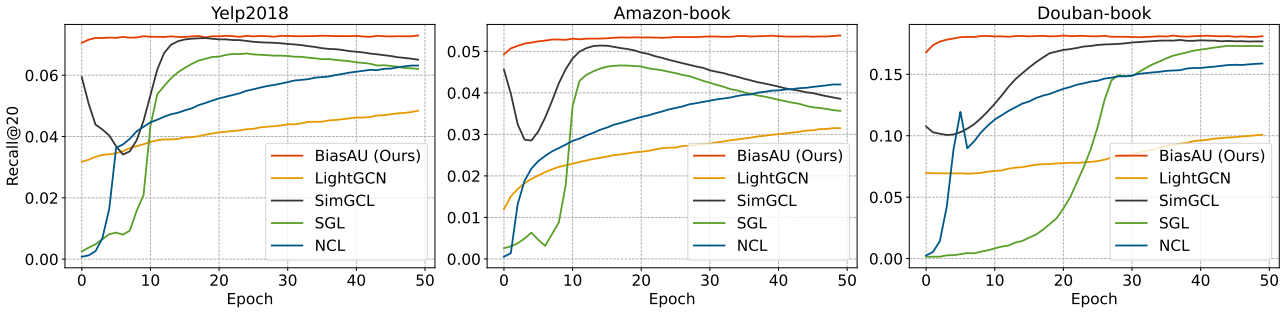


Figure 3. The learning curve w.r.t. recall@20 for the dataset of Yelp2018, Amazon-book, and Douban-book. All curves are plotted based on the corresponding model's best performance setting, and only the previous 50 epochs are shown.

performance showing up in the first few epochs is inadequate to estimate the final performance of the aforementioned methods, inevitably introducing more efforts for fine-tuning. In comparison, our model has decent performance in the first few epochs, and reveals stable training properties, avoiding the overshooting problem as observed in SimGCL when they have identical learning rates. We credit this property to the combination of supervised and unsupervised RAU losses, which quickly identifies the favorable property and yields lightly learned yet superior embeddings. Given the ideal trade-off point, we expect its surrounding points also yield similar if not better performance, leading to a steady increase in performance as opposed to significant fluctuations.

E.3. Ablation Study

In this part, we perform an ablation study to demonstrate the significance of combining supervised and unsupervised RAU. Specifically, we replace the supervised RAU loss with the BPR loss and denote this variant as *BiasBPR*. We remove the unsupervised RAU task, which degenerates our model to DirectAU. Each of the ablated variants is tuned to their best performance on each of the datasets, and the compared results are shown in Table 5. Clearly, removing/replacing either of the modules causes performance decrement. We attribute that to BiasBPR in its lack of consideration for supervised representation uniformity. Despite this, our approach demonstrates improved performance compared to vanilla LightGCN, affirming the efficacy of our 0-level embedding perturbation. As previously stated, DirectAU underperforms our model due to its lack of unsupervised RAU. Both variants fail to integrate supervised and unsupervised RAU, resulting in subpar performance compared to our model. This highlights the significance of a joint optimization objective.

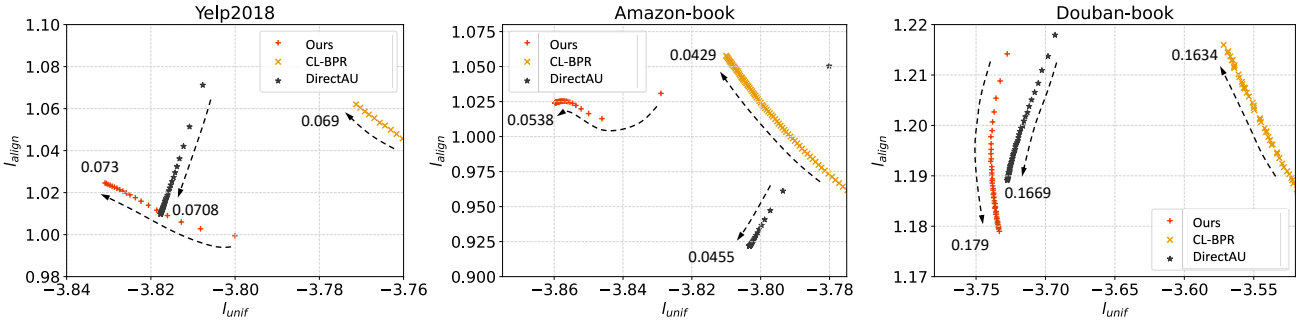


Figure 4. The learning trajectories of BiasBPR, DirectAU, and our model on the three datasets w.r.t. alignment and uniformity losses. The denoted numbers represent the final converged recall@20 and the arrows point to the converging directions.

Table 5. Performance comparison of our model with its ablated versions on Yelp2018, Amazon-book, and Douban-book.

Method	Yelp2018		Amazon-book		Douban-book	
	Recall	NDCG	Recall	NDCG	Recall	NDCG
BiasBPR	0.0690	0.0573	0.0429	0.0335	0.1634	0.1430
DirectAU	0.0708	0.0592	0.0455	0.0364	0.1669	0.1497
Ours	**0.0730**	**0.0614**	**0.0538**	**0.0434**	**0.1804**	**0.1628**

To further elaborate on the difference, we plot the learning trajectories of the two variants with respect to the corresponding two losses, shown in Figure 4. From the figure, we see that the start point of our model generally has smaller RAU losses. For the dataset Yelp2018 and Amazon-book, it chooses to favor the alignment first and then sacrifice the alignment for better uniformity; for the dataset Douban-book, our model prefers low uniformity loss at first and then moves for better alignment. Compared with our model, BiasBPR always favors alignment first and then sacrifices alignment for better uniformity. DirectAU optimizes the two losses altogether but is not able to land at a better point at last without the calibration from the unsupervised RAU loss. This figure serves as additional evidence to support our hypothesis, which assumes that unsupervised RAU loss calibrates the learning process through the inductive bias for better initial and final status.