Squeeze and Excitation: A Weighted Graph Contrastive Learning for Collaborative Filtering

Zheyu Chen
The Hong Kong Polytechnic University
Hong Kong SAR, China
zheyu.chen@connect.polyu.hk

Jinfeng Xu*
The University of Hong Kong
Hong Kong SAR, China
jinfeng@connect.hku.hk

Yutong Wei
Central South University
Changsha, China
8207220511@csu.edu.cn

Ziyue Peng
University of Macau
Macao SAR, China
dc22951@um.edu.mo

ABSTRACT

Contrastive Learning (CL) has recently emerged as a powerful technique in recommendation systems, particularly for its capability to harness self-supervised signals from perturbed views to mitigate the persistent challenge of data sparsity. The process of constructing perturbed views of the user-item bipartite graph and performing contrastive learning between perturbed views in a graph convolutional network (GCN) is called graph contrastive learning (GCL), which aims to enhance the robustness of representation learning. Although existing GCL-based models are effective, the weight assignment method for perturbed views has not been fully explored. A critical problem in existing GCL-based models is the irrational allocation of feature attention. This problem limits the model's ability to effectively leverage crucial features, resulting in suboptimal performance. To address this, we propose a Weighted Graph Contrastive Learning framework (WeightedGCL). Specifically, WeightedGCL applies a robust perturbation strategy, which perturbs only the view of the final GCN layer. In addition, WeightedGCL incorporates a squeeze and excitation network (SENet) to dynamically weight the features of the perturbed views. Our WeightedGCL strengthens the model's focus on crucial features and reduces the impact of less relevant information. Extensive experiments on widely used datasets demonstrate that our WeightedGCL achieves significant accuracy improvements compared to competitive baselines.

CCS CONCEPTS

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

KEYWORDS

Recommendation, Graph Contrastive Learning, Squeeze and Excitation Network.

ACM Reference Format:
Zheyu Chen, Jinfeng Xu, Yutong Wei, and Ziyue Peng. 2025. Squeeze and Excitation: A Weighted Graph Contrastive Learning for Collaborative Filtering. In *The 48th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2025), July 13–17, 2025, Padua, Italy.* ACM, New York, NY, USA, 5 pages. https://doi.org/XXXXXXX.XXXXXXX

*Corresponding author

1 INTRODUCTION

To alleviate the problem of information overload on the web, recommender systems have become an important tool [5, 18, 20, 21], which commonly adopt collaborative filtering (CF) to capture the complex relations between users and items. Attribute to the natural bipartite graph structure of user-item interactions, many existing works have achieved advanced performance by effectively leveraging graph-based recommendation models [4, 15]. For example, NGCF [15] integrates a graph convolutional network (GCN) framework into recommender systems, maintaining both feature transformation and non-linear operations. In contrast, LightGCN [4] raises that these components are unnecessary for recommendation tasks, proposing a lightweight GCN model instead. Numerous subsequent graph-based models [16, 23, 27] have pushed the boundaries of graph-based recommendation systems, offering improved scalability, accuracy, and robustness.

In recent years, contrastive learning (CL) has seen great development in useless representation learning. In the recommendation field, considering the persistent challenge of data sparsity, CL has proven to be a powerful self-supervised learning approach for leveraging unlabeled data.

Recent studies [1, 2, 17, 23] have demonstrated the effectiveness of combining GCN and CL, termed graph contrastive learning (GCL), to improve recommendation performance. GCL typically involves constructing perturbed views of the user-item bipartite graph and applying contrastive learning between these views.

Nevertheless, these existing models [17, 24] often apply uniform weights for all features within each perturbed view, neglecting the varying significance of different features. The equal weight strategy will limit the model's ability to effectively leverage crucial features, making it pay too much attention to less relevant information. This raises a crucial question: ***How to allocate attention to different features dynamically?***

To address this, we propose a tailored and novel framework named **Weighted G**raph **C**ontrastive **L**earning (**WeightedGCL**)[1]. Specifically, WeightedGCL applies a robust perturbation strategy, which only perturbs the final GCN layer's view. In addition, WeightedGCL incorporates a squeeze and excitation network (SENet) [6] to dynamically assign weights to the features of these perturbed views. Our WeightedGCL enhances the attention on the crucial features by assigning greater weight to them. At the same time, it reduces the attention to less relevant information, ensuring that the model's performance is not degraded by this information. In a nutshell, the contributions of our work are as follows:

[1]Code is available at: https://github.com/Zheyu-Chen/WeightedGCL.

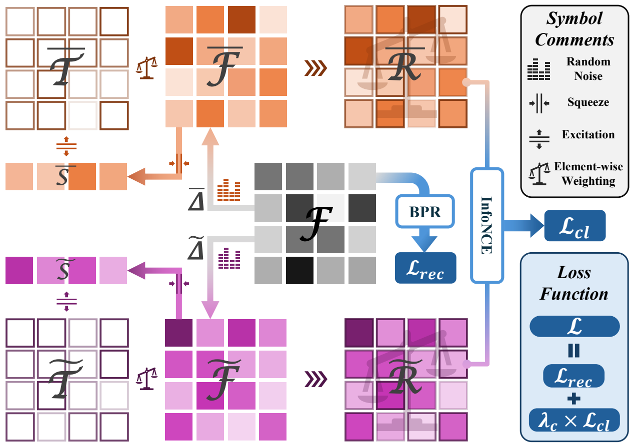

Figure 1: Overall architecture of our WeightedGCL.

- We identify the limitation in existing GCL-based frameworks, in which they assign equal weight to all features and consequently propose a dynamic feature weighting solution.
- We propose a Weighted Graph Contrastive Learning framework, which incorporating the robust perturbation to keep views stable and incorporating the SENet to assign weights dynamically.
- Experiment results on three public datasets demonstrate the effectiveness of our WeightedGCL.

2 METHODOLOGY

In contrastive learning, the construction of contrastive views is crucial. The previous work [23] on GCN-based perturbation applies perturbations across all layers of the network. However, this work introduces noise at earlier layers, which can destabilize the representation learning process. To address this, we propose a robust perturbation strategy that only applies perturbations to the final layer of the GCN. This targeted perturbation avoids the negative impact of early-stage noise, thereby maintaining more stable and meaningful learned representations. Due to the inherent randomness in perturbed views, directly learning the weight matrix proves challenging. To address this challenge, we introduce a SENet, which includes two parts: the squeeze network and the excitation network. The former reduces the dimensions of each perturbed view into a summary statistics matrix, and then the latter transforms the matrix back to the original dimensions, which assign the feature weights dynamically. Therefore, we propose a dynamic feature weighting solution by combining SENet and GCL, named WeightedGCL. Our WeightedGCL enhances the focus on crucial features while effectively mitigating the impact of less relevant information, and achieving performance improvement. The overall architecture of WeightedGCL is depicted in Figure 1.

2.1 Preliminary

Let $\mathcal{U}$ denotes the set of users and $\mathcal{I}$ denotes the set of items. The L represents the number of layers in the GCN. The set of all nodes, including both users and items, is denoted as $\mathcal{N} = \mathcal{U} \cup \mathcal{I}$. Consider a node n within the set $\mathcal{N}$, whose representation is $e_n \in \mathbb{R}^{d\times 1}$ in the view $\mathcal{E} \in \mathbb{R}^{d\times|\mathcal{N}|}$, where d denotes the feature dimension.

2.2 Feature Encoder

GCNs refine node representations by aggregating messages from their neighboring nodes. This process can be formalized as follows:

$$\begin{aligned} e_u^{(l)} &= \mathrm{Aggr}^{(l)}(\{e_i^{(l-1)} : i \in \mathcal{N}_u\}), \\ e_i^{(l)} &= \mathrm{Aggr}^{(l)}(\{e_u^{(l-1)} : u \in \mathcal{N}_i\}), \end{aligned} \tag{1}$$

where $\mathcal{N}_u$ and $\mathcal{N}_i$ denote the neighborhood set of nodes u and i, respectively, and l denotes the l-th layer of GNNs. The node representation aggregation for the entire embeddings $\mathcal{F}$ can be formulated as follows:

$$\mathcal{F} = \frac{1}{L+1}(\mathcal{E}^{(0)} + \mathcal{E}^{(1)} + \ldots + \mathcal{E}^{(L-1)} + \mathcal{E}^{(L)}), \tag{2}$$

where $\mathcal{E}^{(l)}$ denotes the view of nodes in l-th layer.

2.3 Robust Perturbation

We initially construct contrastive views by the robust perturbation strategy, which adds a random noise matrix into the view of the final layer in GCN. Formally:

$$\bar{\mathcal{E}}^{(L)} = \mathcal{E}^{(L)} + \bar{\Delta}, \quad \tilde{\mathcal{E}}^{(L)} = \mathcal{E}^{(L)} + \tilde{\Delta}, \tag{3}$$

$$\begin{aligned} \bar{\mathcal{F}} &= \frac{1}{L+1}(\mathcal{E}^{(0)} + \mathcal{E}^{(1)} + \ldots + \mathcal{E}^{(L-1)} + \bar{\mathcal{E}}^{(L)}), \\ \tilde{\mathcal{F}} &= \frac{1}{L+1}(\mathcal{E}^{(0)} + \mathcal{E}^{(1)} + \ldots + \mathcal{E}^{(L-1)} + \tilde{\mathcal{E}}^{(L)}), \end{aligned} \tag{4}$$

where $\bar{\mathcal{F}}$ and $\tilde{\mathcal{F}}$ denote final representations of two perturbed views. Meanwhile, the $\bar{\Delta}$ and $\tilde{\Delta} \in \mathbb{R}^{d\times|\mathcal{N}|} \sim U(0,1)$ are random noise matrixs. Only perturbing the final layer will maintain the view stability of other front layers. We will conduct an ablation study in Section 3.3 to demonstrate the effectiveness of our strategy.

2.4 Squeeze Network

The squeeze network adopts average pooling, which is beneficial for retaining feature information, to reduce the dimension of the entire perturbed views $\ddot{\mathcal{F}}$ into a summary statistics matrices $\bar{\mathcal{S}}/\tilde{\mathcal{S}} \in \mathbb{R}^{1\times|\mathcal{N}|}$, formally:

$$\mathrm{Squeeze}(\ddot{\mathcal{F}}) = \mathrm{Con}(\frac{1}{d}\sum_{k=1}^{d} \ddot{f}_n^k \mid n \in \mathcal{N}), \tag{5}$$

$$\bar{\mathcal{S}} = \mathrm{Squeeze}(\bar{\mathcal{F}}), \quad \tilde{\mathcal{S}} = \mathrm{Squeeze}(\tilde{\mathcal{F}}), \tag{6}$$

where k represents the k-th feature dimension of perturbed node representation $\ddot{f}_n$, and $\frac{1}{d}\sum_{k=1}^{d} \ddot{f}_n^k$ is the summary statistics matrix of node n.

2.5 Excitation Network

The summary statistics matrix is then expanded back to the original dimensions using the excitation network, formally:

$$\mathrm{Excitation}(\mathcal{S}) = \sigma(W_K(\ldots(W_1 \cdot \mathcal{S} + b_1)\ldots) + b_K), \tag{7}$$

$$\bar{\mathcal{T}} = \mathrm{Excitation}(\bar{\mathcal{S}}), \quad \tilde{\mathcal{T}} = \mathrm{Excitation}(\tilde{\mathcal{S}}), \tag{8}$$

where the resulting matrix $\bar{\mathcal{T}}/\tilde{\mathcal{T}} \in \mathbb{R}^{d\times|\mathcal{N}|}$ serves as the weight matrix corresponding to the perturbed views $\bar{\mathcal{F}}/\tilde{\mathcal{F}}$, and σ denotes the sigmoid function. K is the feedforward network layer number in Eq. 7. The excitation network employs a multi-layer architecture, gradually ascending the dimension according to an equal scale $s = \sqrt[K]{d}$ until the original dimensions are restored, and the $W_1 \in \mathbb{R}^{s\times 1}$, ..., $W_K \in \mathbb{R}^{d\times s^{(K-1)}}$ are the weight matrices for the linear layers. Maintaining a constant ratio facilitates the simplification of the training process. This design results in a more precise generation of the weight matrix, thereby improving the robustness.

Table 1: Performance comparison of baselines, WeightedGCL and variants of WeightedGCL in terms of Recall@K(R@K) and NDCG@K(N@K). The superscript * indicates the improvement is statistically significant where the p-value is less than 0.01.

Model	Amazon				Pinterest				Alibaba			
	R@10	R@20	N@10	N@20	R@10	R@20	N@10	N@20	R@10	R@20	N@10	N@20
MF-BPR	0.0607	0.0956	0.0430	0.0537	0.0855	0.1409	0.0537	0.0708	0.0303	0.0467	0.0161	0.0203
NGCF	0.0617	0.0978	0.0427	0.0537	0.0870	0.1428	0.0545	0.0721	0.0382	0.0615	0.0198	0.0257
LightGCN	0.0797	0.1206	0.0565	0.0689	0.1000	0.1621	0.0635	0.0830	0.0457	0.0692	0.0246	0.0299
UltraGCN	0.0760	0.1155	0.0540	0.0643	0.0967	0.1588	0.0613	0.0808	0.0411	0.0644	0.0227	0.0276
LayerGCN	0.0877	0.1291	0.0647	0.0760	0.1004	0.1620	0.0635	0.0826	0.0448	0.0680	0.0238	0.0285
FKAN-GCF	0.0838	0.1265	0.0602	0.0732	0.1003	0.1614	0.0633	0.0827	0.0441	0.0681	0.0240	0.0290
SGL	0.0898	0.1331	0.0645	0.0777	0.1080	0.1704	0.0701	0.0897	0.0461	0.0692	0.0248	0.0307
NCL	0.0933	0.1381	0.0679	0.0815	0.1033	0.1609	0.0666	0.0833	0.0477	0.0713	0.0259	0.0319
SimGCL	0.0963	0.1336	0.0718	0.0832	0.1051	0.1576	0.0705	0.0871	0.0474	0.0691	0.0262	0.0317
LightGCL	0.0820	0.1278	0.0589	0.0724	0.0881	0.1322	0.0534	0.0673	0.0459	0.0716	0.0239	0.0305
DCCF	0.0903	0.1307	0.0655	0.0781	0.1040	0.1613	0.0661	0.0828	0.0490	0.0729	0.0257	0.0311
RecDCL	0.0927	0.1345	0.0652	0.0780	0.1021	0.1619	0.0663	0.0839	0.0521	0.0768	0.0273	0.0338
BIGCF	0.0948	0.1341	0.0692	0.0810	0.1040	0.1619	0.0680	0.0864	0.0502	0.0744	0.0266	0.0322
WGCL-all pert.	0.0983	0.1378	0.0733	0.0853	0.1163	0.1768	0.0755	0.0961	0.0560	0.0831	0.0305	0.0374
WGCL-w/o pert.	0.0098	0.0159	0.0062	0.0080	0.0103	0.0168	0.0066	0.0086	0.0102	0.0167	0.0050	0.0067
WeightedGCL	**0.0996***	**0.1396***	**0.0741***	**0.0862***	**0.1167***	**0.1793***	**0.0764***	**0.0961***	**0.0596***	**0.0879***	**0.0326***	**0.0397***
Improv.	3.43%	4.49%	3.20%	3.61%	8.06%	5.22%	7.81%	7.13%	21.63%	20.58%	24.42%	24.45%

2.6 Recalibration

Eventually, the weighted views $\bar{\mathcal{R}}/\tilde{\mathcal{R}} \in \mathbb{R}^{d\times|\mathcal{N}|}$ are obtained by scaling the perturbed views $\bar{\mathcal{F}}/\tilde{\mathcal{F}}$ with dynamic weight matrix $\bar{\mathcal{T}}/\tilde{\mathcal{T}}$, formally:

$$\bar{\mathcal{R}} = \bar{\mathcal{T}} \odot \bar{\mathcal{F}}, \quad \tilde{\mathcal{R}} = \tilde{\mathcal{T}} \odot \tilde{\mathcal{F}}, \tag{9}$$

where the $\odot$ represents element-wise multiplication.

2.7 Contrastive Learning

We adopt the InfoNCE [12] loss function to perform contrastive learning between two perturbed views. Formally, the loss function is defined as follows:

$$\mathcal{L}_{cl} = -\sum_{u\in\mathcal{U}} \log \frac{e^{(\bar{r}_u^\top \tilde{r}_u/\tau)}}{\sum_{u'\in\mathcal{U}} e^{(\bar{r}_u^\top \tilde{r}_{u'}/\tau)}} - \sum_{i\in\mathcal{I}} \log \frac{e^{(\bar{r}_i^\top \tilde{r}_i/\tau)}}{\sum_{i'\in\mathcal{I}} e^{(\bar{r}_i^\top \tilde{r}_{i'}/\tau)}}, \tag{10}$$

where $\bar{r}_u$ and $\tilde{r}_{u/u'}$ are representation of user u/u' in contrastive views $\bar{\mathcal{R}}$ and $\tilde{\mathcal{R}}$. Besides, $\bar{r}_i$ and $\tilde{r}_{i/i'}$ are representation of item i/i' in contrastive views $\bar{\mathcal{R}}$ and $\tilde{\mathcal{R}}$. τ is the temperature hyper-parameter.

2.8 Optimization

We utilize LightGCN [4] as the backbone and adopt the Bayesian Personalized Ranking (BPR) loss [14] as our primary optimization objective. The BPR loss is designed to enhance the distinction between predicted preferences for positive and negative items in each triplet $(u, p, n) \in \mathcal{D}$, where $\mathcal{D}$ is the training dataset. The positive item p is an item with which user u has interacted, while the negative item n is randomly selected from the items not interacted with user u. The BPR loss function is formally defined as:

$$\mathcal{L}_{rec} = \sum_{(u,p,n)\in\mathcal{D}} -\log(\sigma(y_{u,p} - y_{u,n})) + \lambda \cdot \|\Theta\|_2^2, \tag{11}$$

where σ denotes the sigmoid function, λ controls the L_2 regularization strength, and Θ denotes model parameters. The $y_{u,p}$ and $y_{u,n}$ are the ratings of user u to the positive item p and negative item n, which calculated by $r_u^\top r_p$ and $r_u^\top r_n$. Ultimately, the total loss function is:

$$\mathcal{L} = \mathcal{L}_{rec} + \lambda_c \cdot \mathcal{L}_{cl}, \tag{12}$$

where λ_c is the balancing hyper-parameter.

3 EXPERIMENT

In this section, we conduct extensive experiments on three real-world datasets to evaluate WeightedGCL, addressing the following research questions:

RQ1: How does the performance of our WeightedGCL compare to advanced recommender systems?

RQ2: How does our Robust Perturbation component influence the overall performance of our WeightedGCL?

RQ3: How do different hyper-parameters influence the performance of our WeightedGCL?

3.1 Experimental Settings

3.1.1 Datasets. To evaluate our WeightedGCL in the recommendation task, we conduct extensive experiments on three widely used datasets: Amazon Books (Amazon) [11], Pinterest [5] and Alibaba. Details can be found in Table 2. These datasets offer a set of user-item interactions, and we use the ratio 8:1:1 for each dataset to randomly split the data for training, validation, and testing.

Table 2: Statistics of datasets.

Datasets	#Users	#Items	#Interactions	Sparsity
Amazon	58,144	58,051	2,517,437	99.925%
Pinterest	55,188	9,912	1,445,622	99.736%
Alibaba	300,001	81,615	1,607,813	99.993%

To ensure the quality of the data, we employ the 15-core setting for Amazon, which ensures a minimum of 15 interactions between users and items. For the Pinterest datasets, users and items with less than five interactions are filtered out. Due to the high sparsity of the Alibaba dataset, we choose to retain all interaction data.

3.1.2 Baselines. To verify the effectiveness of WeightGCL, we select one matrix factorization model **MF-BPR** [14], five advanced GCN-based models (**NGCF** [15], **LightGCN** [4], **UltraGCN** [10], **LayerGCN** [27] and **FKAN-GCF** [19]), and seven state-of-the-art GCL-based models (**SGL**[17], **NCL** [9], **SimGCL** [24], **LightGCL** [1], **DCCF** [13], **RecDCL** [25] and **BIGCF** [26]) for comparison.

3.1.3 Hyper-parameters. To ensure a fair comparison, we initially refer to the optimal hyper-parameter settings as reported in the original papers of the baseline models. Subsequently, we fine-tune all hyper-parameters of the baselines using grid search. For the general settings across all baselines, we apply Xavier initialization [3] to all embeddings. The embedding size is 64, and the batch size is 4096. All models are optimized using the Adam optimizer [8] with a learning rate of 1e-4. We perform a grid search on the balancing hyper-parameter λ_c in {1e-1, 1e-2, 1e-3}, temperature hyper-parameter τ in {0.2, 0.4, 0.6, 0.8}, and the layer number L of GCN in {2, 3, 4}. Empirically, we set the L_2 regularization parameter to 1e-4 for Amazon and Pinterest and 1e-5 for Alibaba. To avoid the over-fitting problem, we set 30 as the early stopping epoch number. Moreover, we set the excitation network boosting granularity as a hyper-parameter, containing 1-4 levels of granularity, and detailed description and analysis are in Section 3.4.

3.1.4 Evaluation Protocols. For fairness, we follow the settings of previous works [5, 15, 22] by adopting two widely-used evaluation protocols for top-K recommendations: Recall@K and NDCG@K[7]. We report the average metrics across all users in the test dataset, evaluating performance at both $K = 10$ and 20.

3.2 Performance Comparison (RQ1)

Detailed experiment results are shown in Table 1. The optimal results are highlighted in bold, while the suboptimal results are indicated with underlines. Based on these results, we observed that: our WeightedGCL outperforms the strongest baselines, achieving 4.49%(R@20), 3.61%(N@20) improvement on the Amazon dataset, achieving 5.22%(R@20), 7.13%(N@20) improvement on the Pinterest dataset, and achieving 20.58%(R@20), 24.45%(N@20) improvement on the Alibaba dataset, which demonstrates the effectiveness of WeightedGCL. Moreover, we can identify that most GCL-based models outperform traditional models, which is a common trend. Our WeightedGCL outperforms all baselines in three datasets for various metrics.

3.3 Ablation Study (RQ2)

To validate the effectiveness of our robust perturbation, we design the following variants:

WGCL-w/o pert. For this variant, we directly removed our robust perturbation component.

WGCL-all pert. For this variant, we utilize all layer's perturbation to replace our robust perturbation component.

Table 1 also demonstrates the significance of perturbation strategies. This ablation study shows that the method's performance without perturbation has a sharp performance degradation, resulting in a completely unusable model, which indicates that the perturbation operation is essential for contrastive learning. Our robust perturbation outperforms other variants, which indicates the robust perturbation component plays a critical role in improving model representation power and model robustness.

3.4 Hyper-parameter Sensitivity Study (RQ3)

To evaluate the hyper-parameter sensitivity of WeightedGCL, we test its performance on three datasets under varying hyper-parameters. Table 3 highlights the impact of various excitation strategies. We denote different ascending granularity as W-G1, W-G2, W-G3, and W-G4, which correspond to varying numbers of layers in the FFN within the excitation network. The larger the number, the finer the granularity, and the more layers.

Table 3: Excitation strategies analysis.

Variant	Amazon		Pinterest		Alibaba	
	R@20	N@20	R@20	N@20	R@20	N@20
W-G1	0.1314	0.0807	0.1713	0.0931	0.0772	0.0359
W-G2	0.1392	0.0859	0.1764	0.0953	0.0793	0.0363
W-G3	0.1391	0.0860	**0.1793**	**0.0961**	**0.0879**	**0.0397**
W-G4	**0.1396**	**0.0862**	0.1790	0.0956	0.0826	0.0371

After analyzing the results in Table 3, we found that our framework performs best at the fourth granularity for the Amazon dataset, and the third granularity is the best for the Pinterest and Alibaba datasets. Although the fourth granularity is not the best choice for the latter two datasets, it is still higher than the performance at the first and second granularities. These observations illustrate two points: from the overall trend, the performance of our incentive network is improving with the increase of granularity; but too high granularity makes it difficult to train due to the increased complexity of the model, which will slightly reduce the performance in some cases. This ablation study demonstrates that the choice of excitation strategies to some extent influences our framework's performance.

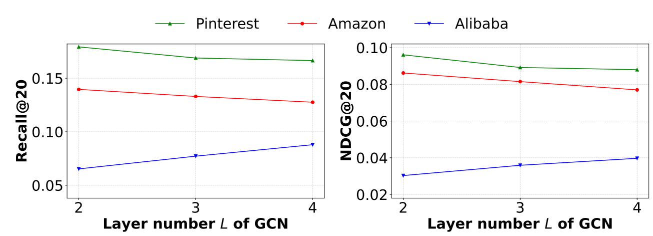

Figure 2: Performance of our WeightedGCL with respect to different layer number L of GCN.

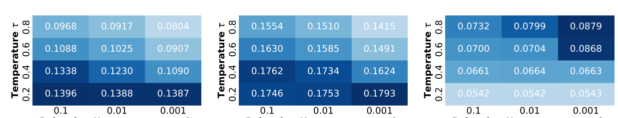

Figure 3: Performance of our WeightedGCL with respect to different hyper-parameter pairs (λ_c, τ) in terms of Recall@20.

As shown in the Figure 2, for the Amazon and Pinterest datasets, the optimal layer number L is 2, and for the Alibaba dataset, the optimal number is 3. Additionally, Figure 3 reveals that for the Amazon dataset, the optimal (λ_c, τ) pair is (1e-1, 0.2); for Pinterest dataset, the optimal pair is (1e-3, 0.2); and for the Alibaba dataset, (1e-3, 0.8) is the optimal pair. The optimal temperature hyper-parameter τ for the Alibaba dataset differs from the other two datasets, attributed to its sparse user-item interactions and the huge number of users. Note that being flexible in choosing the value of hyper-parameters will allow us to adopt our framework to multiple datasets.

4 CONCLUSION

In this paper, we propose WeightedGCL, a novel model that incorporates a tailored robust perturbation strategy and SENet with GCL. Existing GCL-based models assign equal weights for all features within each perturbed view, which limits the model's ability to effectively leverage crucial features. Our WeightedGCL enhances the attention on the crucial features by assigning greater weight to them and reduces the attention to less relevant information. Our experiments on three widely used datasets show that WeightedGCL achieves significant performance improvements compared to existing models. The improvement demonstrates the effectiveness of WeightedGCL and its potential to advance the development of recommendation systems.

REFERENCES

[1] Xuheng Cai, Chao Huang, Lianghao Xia, and Xubin Ren. 2023. LightGCL: Simple Yet Effective Graph Contrastive Learning for Recommendation. In *The Eleventh International Conference on Learning Representations*.

[2] Zheyu Chen, Jinfeng Xu, and Haibo Hu. 2025. Don't Lose Yourself: Boosting Multimodal Recommendation via Reducing Node-neighbor Discrepancy in Graph Convolutional Network. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 1–5.

[3] Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 249–256.

[4] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*. 639–648.

[5] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*. 173–182.

[6] Jie Hu, Li Shen, and Gang Sun. 2018. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 7132–7141.

[7] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (TOIS)* 20, 4 (2002), 422–446.

[8] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[9] Zihan Lin, Changxin Tian, Yupeng Hou, and Wayne Xin Zhao. 2022. Improving graph collaborative filtering with neighborhood-enriched contrastive learning. In *Proceedings of the ACM web conference 2022*. 2320–2329.

[10] Kelong Mao, Jieming Zhu, Xi Xiao, Biao Lu, Zhaowei Wang, and Xiuqiang He. 2021. UltraGCN: ultra simplification of graph convolutional networks for recommendation. In *Proceedings of the 30th ACM international conference on information & knowledge management*. 1253–1262.

[11] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. 2015. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*. 43–52.

[12] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748* (2018).

[13] Xubin Ren, Lianghao Xia, Jiashu Zhao, Dawei Yin, and Chao Huang. 2023. Disentangled contrastive collaborative filtering. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1137–1146.

[14] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*. 452–461.

[15] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural graph collaborative filtering. In *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*. 165–174.

[16] Yinwei Wei, Xiang Wang, Liqiang Nie, Xiangnan He, Richang Hong, and Tat-Seng Chua. 2019. MMGCN: Multi-modal graph convolution network for personalized recommendation of micro-video. In *Proceedings of the 27th ACM international conference on multimedia*. 1437–1445.

[17] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. 2021. Self-supervised graph learning for recommendation. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*. 726–735.

[18] Jinfeng Xu, Zheyu Chen, Jinze Li, Shuo Yang, Hewei Wang, and Edith C-H Ngai. 2024. AlignGroup: Learning and Aligning Group Consensus with Member Preferences for Group Recommendation. *arXiv preprint arXiv:2409.02580* (2024).

[19] Jinfeng Xu, Zheyu Chen, Jinze Li, Shuo Yang, Wei Wang, Xiping Hu, and Edith C-H Ngai. 2024. FourierKAN-GCF: Fourier Kolmogorov-Arnold Network–An Effective and Efficient Feature Transformation for Graph Collaborative Filtering. *arXiv preprint arXiv:2406.01034* (2024).

[20] Jinfeng Xu, Zheyu Chen, Zixiao Ma, Jiyi Liu, and Edith CH Ngai. 2024. Improving Consumer Experience With Pre-Purify Temporal-Decay Memory-Based Collaborative Filtering Recommendation for Graduate School Application. *IEEE Transactions on Consumer Electronics* (2024).

[21] Jinfeng Xu, Zheyu Chen, Shuo Yang, Jinze Li, Hewei Wang, and Edith C-H Ngai. 2024. MENTOR: Multi-level Self-supervised Learning for Multimodal Recommendation. *arXiv preprint arXiv:2402.19407* (2024).

[22] Jinfeng Xu, Zheyu Chen, Shuo Yang, Jinze Li, Wei Wang, Xiping Hu, Steven Hoi, and Edith Ngai. 2025. A Survey on Multimodal Recommender Systems: Recent Advances and Future Directions. *arXiv preprint arXiv:2502.15711* (2025).

[23] Junliang Yu, Xin Xia, Tong Chen, Lizhen Cui, Nguyen Quoc Viet Hung, and Hongzhi Yin. 2023. XSimGCL: Towards extremely simple graph contrastive learning for recommendation. *IEEE Transactions on Knowledge and Data Engineering* (2023).

[24] Junliang Yu, Hongzhi Yin, Xin Xia, Tong Chen, Lizhen Cui, and Quoc Viet Hung Nguyen. 2022. Are graph augmentations necessary? simple graph contrastive learning for recommendation. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*. 1294–1303.

[25] Dan Zhang, Yangliao Geng, Wenwen Gong, Zhongang Qi, Zhiyu Chen, Xing Tang, Ying Shan, Yuxiao Dong, and Jie Tang. 2024. RecDCL: Dual Contrastive Learning for Recommendation. In *Proceedings of the ACM on Web Conference 2024*. 3655–3666.

[26] Yi Zhang, Lei Sang, and Yiwen Zhang. 2024. Exploring the Individuality and Collectivity of Intents behind Interactions for Graph Collaborative Filtering. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1253–1262.

[27] Xin Zhou, Donghui Lin, Yong Liu, and Chunyan Miao. 2023. Layer-refined graph convolutional networks for recommendation. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*. IEEE, 1247–1259.