Enhancing Neural Collaborative Filtering with Self-Supervised Learning

1st Lakindu Kariyawasam
Dept. of Computer Science & Engineering
University of Moratuwa
Sri Lanka
lakindu.21@cse.mrt.ac.lk

2nd Uthayasanker Thayasivam
Dept. of Computer Science & Engineering
University of Moratuwa
Sri Lanka
rtuthaya@cse.mrt.ac.lk

***Abstract*—Neural Collaborative Filtering (NCF) marked a significant advancement in recommender systems by leveraging the non-linear expressiveness of deep neural networks to model user-item interactions, moving beyond the linear constraints of traditional Matrix Factorization. However, the performance of NCF is often fundamentally limited by the extreme sparsity inherent in real-world user-item interaction data. This sparsity provides a weak supervision signal, which, when coupled with a standard pointwise loss function, can lead to suboptimal and poorly generalized latent representations for users and items. To address this challenge, this paper proposes NCF-SSL, a multi-task learning framework that enhances the NCF model with a self-supervised auxiliary objective. Our approach introduces a contrastive learning task that acts as a powerful regularizer on the embedding space. Specifically, we generate augmented views of user and item embeddings via a simple yet effective embedding dropout strategy. A contrastive loss then encourages the representations of different views from the same entity to be more similar to each other than to the representations of other entities. This auxiliary task enriches the training signal without requiring any external data, compelling the model to learn more robust and invariant features. We conduct a comprehensive set of experiments on the benchmark MovieLens 1M and Pinterest 20 datasets. The results demonstrate that NCF-SSL improves over the standard NCF baseline on widely-used top-K ranking metrics, including Hit Ratio (HR) and Normalized Discounted Cumulative Gain (NDCG) in some cases. However the improvements are marginal and given the increased training time and compute, the benefits of the self-supervised task may not justify its use in all scenarios. This suggests that while self-supervised learning holds promise for enhancing collaborative filtering models, further research is needed to fully realize its potential and understand the conditions under which it provides significant benefits.**

***Index Terms*—Recommender Systems, Neural Collaborative Filtering, Self-Supervised Learning, Contrastive Learning, Multi-Task Learning**

I. Introduction

In the contemporary digital landscape, recommender systems have become indispensable tools for mitigating information overload and delivering personalized user experiences. From e-commerce and content streaming to social media, these systems are critical for user engagement and retention by navigating vast item catalogs to surface relevant content. Among the various techniques developed for recommendation, Collaborative Filtering (CF) has emerged as a cornerstone paradigm. The fundamental principle of CF is to leverage the "wisdom of the crowd," making predictions based on the past behaviors and preferences of a community of users.

Early CF methods were predominantly memory-based, such as user-based or item-based nearest-neighbor algorithms. While intuitive, these methods often struggle with scalability and data sparsity. A significant breakthrough came with the advent of latent factor models, most notably Matrix Factorization (MF). MF projects users and items into a shared low-dimensional latent space, where their interaction is modeled by the inner product of their respective embedding vectors. This approach proved highly effective at capturing generalized preference patterns and became a dominant technique for over a decade. However, the expressive power of MF is inherently limited by the linearity of the inner product, which may fail to capture the complex, non-linear relationships inherent in user-item interactions.

The proliferation of deep learning has ushered in a new era for recommender systems, offering a powerful toolkit to overcome the limitations of traditional models. By leveraging deep neural networks, researchers can now learn highly complex and non-linear user-item interaction functions directly from data. A seminal work in this domain is the Neural Collaborative Filtering (NCF) framework proposed by He et al. [1]. NCF provides a general architecture that replaces the MF inner product with a Multi-Layer Perceptron (MLP), endowing the model with a high degree of non-linear expressiveness. The framework's elegant fusion of a generalized MF component (GMF) and an MLP, as show in Fig. 1, allows it to learn both linear and non-linear feature interactions, setting a new benchmark for deep learning-based collaborative filtering.

Despite its success, the NCF framework is not without its challenges, primarily stemming from the inherent properties of recommendation datasets. The user-item interaction matrix is typically extremely sparse, meaning that each user has only interacted with a tiny fraction of the available items. This sparsity provides a limited and often noisy supervision signal for model training. The standard NCF model, trained with a pointwise loss function (e.g., binary cross-entropy), treats each interaction as an independent event. This training objective may be insufficient for learning high-quality, robust latent representations that can generalize well to unseen user-item pairs, making the model susceptible to overfitting.

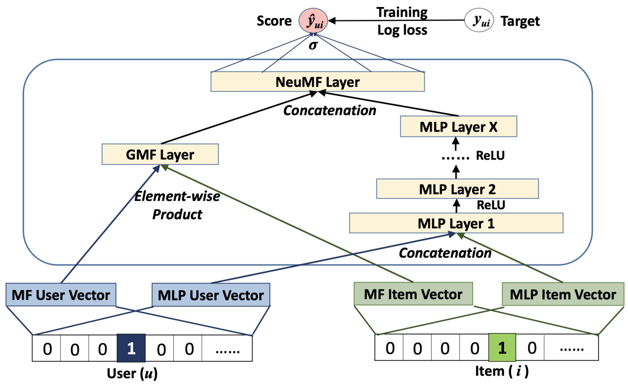

Fig. 1. NCF Architecture as presented in the original paper [1]

To address these limitations, this paper explores the integration of Self-Supervised Learning (SSL), a paradigm that has shown remarkable success in learning rich representations from unlabeled or sparsely labeled data in fields like computer vision and natural language processing. The core idea of SSL is to create auxiliary "pretext" tasks where the supervision signal is derived from the data itself. Specifically, we leverage contrastive learning, which aims to learn an embedding space where augmented "views" of the same data instance are pulled closer together, while views from different instances are pushed apart.

In this work, we propose **NCF-SSL**, a novel framework that enhances the standard NCF model with an auxiliary self-supervised contrastive learning task. This auxiliary task acts as a powerful regularizer by imposing an additional structural constraint on the embedding space. By training the model to recognize different augmented views of the same user or item, we provide a richer learning signal that encourages the model to capture more essential and generalizable latent features. Our primary contributions are,

1) We propose a multi-task learning framework that seamlessly integrates a contrastive self-supervised objective with the primary recommendation task of NCF.
2) We demonstrate through empirical evaluation on benchmark datasets that our proposed NCF-SSL model improves over the original NCF baseline, particularly on ranking-based metrics.
3) We provide an analysis showing that the self-supervised task effectively improves the quality of the learned user and item representations, thus mitigating the challenges posed by data sparsity.

The remainder of this paper is organized as follows: Section II reviews related work, Section III details the proposed NCF-SSL methodology, Section IV describes our experimental setup, Section V presents and discusses the results, and finally, Section VI concludes the paper and outlines future research directions.

II. Related Work

Since the introduction of Neural Collaborative Filtering (NCF) [1], research in deep learning-based recommendation has rapidly evolved. Advancements can be broadly categorized into architectural innovations that enhance model expressiveness and new training strategies that improve learning from sparse data.

A. Architectural Advancements Beyond NCF

A primary focus of subsequent research has been to design architectures capable of capturing more complex dependencies than the standard MLP used in NCF.

1) Graph Neural Networks (GNNs): Perhaps the most significant architectural shift has been the adoption of GNNs. This paradigm reframes the recommendation problem on the user-item interaction graph, allowing models to explicitly capture the collaborative signal embedded in the graph's structure. Models like Neural Graph Collaborative Filtering (NGCF) [2] employ a message-passing mechanism where user and item embeddings are iteratively refined by aggregating information from their neighbors. This process enables the model to capture high-order connectivity, effectively encoding the signal that, for example, "users who interacted with items you liked also interacted with this other item." Subsequent work, such as LightGCN [3], demonstrated that simplifying the GNN architecture by removing non-linearities and feature transformations could lead to superior performance and efficiency, highlighting the centrality of the neighborhood aggregation process.

2) Attention Mechanisms: Recognizing that not all latent features contribute equally to an interaction, attention mechanisms have been integrated to assign dynamic, context-aware weights to features. For instance, Attentional Collaborative Filtering (ACF) [4] introduces an attention network to the MLP component of NCF to learn the importance of each latent dimension.

3) Other Deep Architectures: Researchers have also explored other neural architectures. For instance, ConvNCF [8] performs an outer product on user and item embeddings to create an "interaction map" and applies a Convolutional Neural Network (CNN) to learn local interaction patterns and high-order correlations from this map. Autoencoder-based models have also been used, where the network learns to reconstruct a user's interaction vector, with the compressed hidden layer serving as the user's latent representation. Additionally, Generative Adversarial Networks (GANs), such as IRGAN [5], have been employed to improve model robustness which frames recommendation as a minimax game between a generator and a discriminator.

B. Innovations in Training Objectives and Strategies

Parallel to architectural improvements, significant progress has been made in developing more effective training objectives and strategies to handle the challenges of implicit, sparse feedback.

1) Learning-to-Rank Objectives: The pointwise loss in NCF treats recommendation as a classification task for each user-item pair, which does not directly optimize for the ranking quality of a recommended list. To address this, many models adopt pairwise learning-to-rank objectives. The most prominent example is Bayesian Personalized Ranking (BPR) [9], which trains the model to rank an observed positive item higher than an unobserved (presumed negative) item. This approach better aligns the training objective with the ultimate goal of generating a ranked list of recommendations.

2) Adversarial Training: To enhance model robustness against subtle perturbations, adversarial training techniques have been introduced. These methods, such as Adversarial Personalized Ranking (APR) [10], add small, worst-case perturbations to the embedding vectors during training. This forces the model to learn smoother decision boundaries and become less sensitive to minor variations in the input data, thereby improving generalization.

3) Self-Supervised Learning (SSL): Recently, self-supervised learning has emerged as a powerful paradigm for representation learning, particularly in data-sparse domains like recommendations. The core idea is to create an auxiliary pretext task where the supervision signal is derived from the data itself. For recommendations, this is typically achieved through contrastive learning. The general framework involves: 1) creating two or more augmented "views" of the data (e.g., by applying dropout to embeddings or perturbing the user-item graph structure), and 2) training the model to maximize the agreement between different views of the same user/item while minimizing agreement with other users/items in the batch. Self-supervised Graph Learning (SGL) [11] successfully applied this concept to GNN-based recommenders, demonstrating significant gains in robustness and performance. Other works like S^3-Rec [6] focused on sequential recommendation with self-supervision, while Collaborative Self-Supervised Learning (CSSL) [7] explores various data augmentation and contrastive strategies for learning robust item and user representations.

Our work is directly inspired by this trend. However, while SSL has proven highly effective for complex GNN architectures, its utility for simpler, foundational models like NCF remains less explored. This paper investigates whether the powerful regularization signal from a contrastive SSL task can directly benefit the original NCF architecture, providing a resource-efficient method to enhance its representation learning capabilities.

III. Proposed Method

Our proposed model, NCF-SSL, builds directly upon the NCF architecture. It maintains the dual pathways of Generalized Matrix Factorization (GMF) and a Multi-Layer Perceptron (MLP) but introduces a multi-task learning objective.

A. Data Augmentation via Dropout

To generate different "views" for the contrastive task, we use a simple stochastic augmentation method, embedding dropout. For each user u and item i, we create two augmented views of their embeddings $(\mathbf{p}_u, \mathbf{q}_i)$ by applying dropout independently:

$$\mathbf{p}_u^{(k)} = \text{Dropout}(\mathbf{p}_u), \quad \mathbf{q}_i^{(k)} = \text{Dropout}(\mathbf{q}_i) \quad \text{for } k \in \{1, 2\}$$

where Dropout randomly zeroes out elements of the embedding vectors with a certain probability.

B. Multi-Task Objective Function

The model is trained by optimizing a composite loss function $\mathcal{L}$, which is a weighted sum of the recommendation loss $\mathcal{L}_{NCF}$ and the self-supervised contrastive loss $\mathcal{L}_{SSL}$.

$$\mathcal{L} = \mathcal{L}_{NCF} + \lambda \mathcal{L}_{SSL}$$

where λ is a hyperparameter balancing the two tasks.

Recommendation Loss ($\mathcal{L}_{NCF}$): This is the standard binary cross-entropy loss used in the original NCF paper, averaged over the two augmented views.

Contrastive Loss ($\mathcal{L}_{SSL}$): We use the InfoNCE loss to pull the two augmented views of an entity closer in the embedding space while pushing them away from other entities in the batch. For a user u in a batch $\mathcal{B}$, the loss is:

$$\mathcal{L}_{SSL}^{user} = \sum_{u \in \mathcal{B}} -\log \frac{\exp(\text{sim}(\mathbf{p}_u^{(1)}, \mathbf{p}_u^{(2)})/\tau)}{\sum_{v \in \mathcal{B}} \exp(\text{sim}(\mathbf{p}_u^{(1)}, \mathbf{p}_v^{(2)})/\tau)}$$

where $\text{sim}(\cdot, \cdot)$ is the cosine similarity and τ is a temperature hyperparameter. An analogous loss $\mathcal{L}_{SSL}^{item}$ is computed for items, and $\mathcal{L}_{SSL} = \mathcal{L}_{SSL}^{user} + \mathcal{L}_{SSL}^{item}$.

IV. Experimental Setup

A. Dataset

We conducted experiments on two widely used datasets **MovieLens 1M** and **Pinterest-20**. The MovieLens 1M dataset contains 1,000,209 ratings from 6,040 users on 3,952 movies. The Pinterest-20 dataset consists of 1,009,973 interactions from 55,187 users on 9,916 items (images). For both datasets, we convert ratings or interactions into implicit feedback, where any interaction implies a positive preference. Following common practice and the protocol used by the authors of NCF, we randomly split each dataset into training and test sets, ensuring that each user has at least one interaction in the test set. All experiments were carried out without pretraining, as pretrained model weights were not made available.

B. Evaluation Metrics

We evaluate model performance using two standard top-K ranking metrics,

- **Hit Ratio (HR@10)** Measures the percentage of users for whom the correct held-out item is present in the top-10 recommendation list.
- **NDCG@10 (Normalized Discounted Cumulative Gain)** A measure of ranking quality that assigns higher scores for hits occurring earlier in the top-10 list. This metric accounts for the position of the correct item, rewarding models that rank it higher.

C. Implementation Details

We compare our proposed NCF-SSL against the original **NCF** [1] as our baseline. The code for the original NCF model provided by the authors but was re-implemented in PyTorch, as the provided code used outdated packages and versions. The code including both the reimplemented version of the original as well as the proposed NCF-SSL is made available at [13]. Both models were implemented using PyTorch and trained on Google Colab with NVIDIA Tesla T4 GPU.

1) MLP Architecture: The NCF paper describes the last layer of the MLP as the predictive factor, stating that it determines the model capability.

The Multi-Layer Perceptron (MLP) component follows a pyramid structure with three hidden layers such that the last layer has a dimension equal to the number of predictive factors (8, 16, or 32). Each of the previous layers has double the number of neurons as the subsequent layer. The input layer receives the concatenated user and item embeddings. As done in the original implementation, an embedding size of double the predictive factor is used (i.e., 16, 32, or 64). For instance, if the predictive factor is set to 16, the MLP architecture will be 64 $\rightarrow$ 32 $\rightarrow$ 16 neurons and the embedding size will be 32.

2) Training Configuration: We use a batch size of 256 as done by the original authors. However instead of using a fixed learning rate as mentioned by the authors, an exponential learning rate scheduler has been used with a decay rate of 0.85. This showed to improve convergence time in comparison to having a fixed learning rate. The models are trained for a maximum of 20 epochs with early stopping based on validation performance to prevent overfitting.

3) Hyperparameter Tuning: For NCF-SSL, we conducted systematic hyperparameter tuning experiments to determine optimal values for the self-supervised learning components:

- **Dropout rate for augmentation (p_{drop}):** We experimented with values in $\{0.1, 0.2, 0.3, 0.4, 0.5\}$ and found that 0.2 provided the best balance between creating meaningful augmented views while preserving essential interaction patterns. Given the small sizes of the embeddings, values below 0.1 did not introduce sufficient variability, while 0.5 led to excessive information loss.
- **Contrastive loss weight (λ):** We tested values in $\{0.2, 0.4, 0.6, 0.8\}$ to balance the contribution of the contrastive loss with the primary recommendation loss. A weight of 0.8 yielded optimal performance, ensuring the contrastive objective enhances learning without overwhelming the primary task.
- **Temperature parameter (τ):** We explored values in $\{0.1, 0.5, 1\}$ for the InfoNCE contrastive loss. A temperature of 0.5 was found to provide the right level of selectivity in the contrastive learning, making the model sufficiently discriminative between positive and negative pairs.

All hyperparameter experiments were conducted using cross-validation and the configuration achieving the highest average HR@10 was selected for final evaluation.

V. Discussion

The results of our experiments on the MovieLens 1M dataset are presented in Table I. The NCF-SSL model demonstrates a slight improvement over the standard NCF baseline across both evaluation metrics. The values obtained for Hit Ratio (HR@10) and Normalized Discounted Cumulative Gain (NDCG@10) without SSL are inline with the values reported with the use of a 3 layer MLP (MLP-3) without pretraining, in the original NCF paper [1]. This validates our implementation of the baseline model.

TABLE I
Performance of NCF and NCF-SSL with varying number of predictive factors on MovieLens 1M

Factors	NCF		NCF-SSL	
	HR@10	NDCG@10	HR@10	NDCG@10
8	0.6824	0.4007	**0.6859**	**0.4060**
16	0.6972	0.4189	**0.7016**	**0.4210**
32	0.7004	**0.4251**	**0.7010**	0.4234

As shown, NCF with the self-supervised learning task (NCF-SSL) slightly outperforms the original NCF across all configurations of predictive factors (8, 16, and 32) except NDCG@10 for predictive factor of 32. However as the improvements are marginal, this alone cannot support the hypothesis that the auxiliary self-supervised task acts as an effective regularizer.

TABLE II
Performance of NCF and NCF-SSL with varying number of predictive factors on Pinterest 20

Factors	NCF		NCF-SSL	
	HR@10	NDCG@10	HR@10	NDCG@10
8	0.8588	**0.5361**	**0.8592**	0.5347
16	**0.8653**	0.5379	0.8649	**0.5395**
32	0.8677	0.5428	**0.8682**	**0.5446**

Results of the experiments carried out on the Pinterest-20 dataset, as shown in Table II, show a similar trend to that seen when experimenting with the MovieLens 1M dataset. NCF-SSL slightly outperforms the baseline NCF model across most configurations of predictive factors, with the exception of HR@10 for a predictive factor of 16 and the NDCG@10 for a predictive factor of 8. Again, the improvements are marginal.

The effect of adjusting each of the hyperparameters specific to the self-supervised learning component were also studied. The dropout rate for augmentation (p_{drop}) and contrastive loss weight (λ) were varied while keeping the other hyperparameters constant. The results for varying p_{drop} are shown in Fig. 2 and for varying λ in Fig 3.

Fig. 2 and Fig. 3 indicate that both hyperparameters have an effect on model performance. A dropout rate of 0.2 appears to yield the best results, with performance declining at higher dropout rates. This suggests that while some level

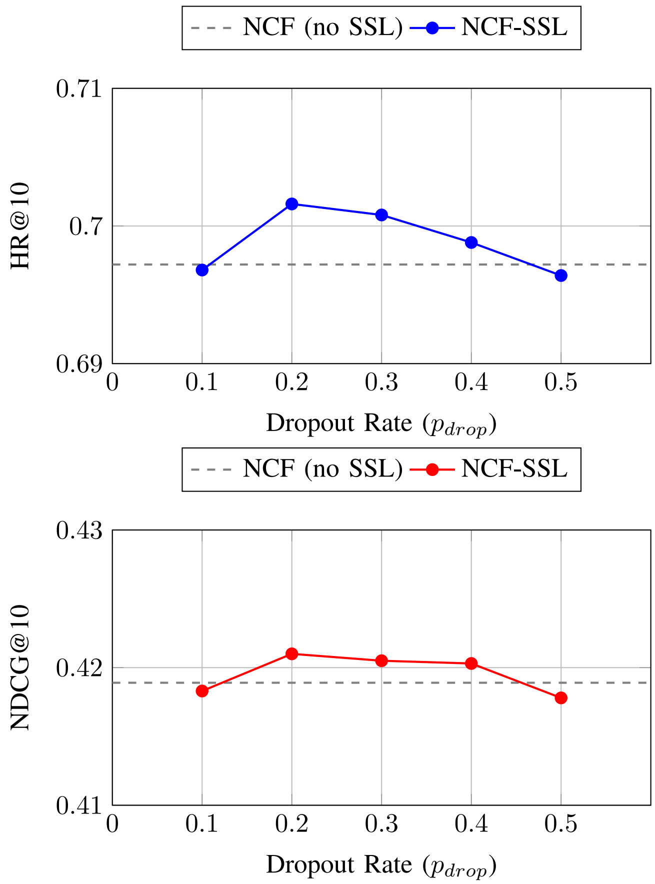

Fig. 2. Effect of dropout rate (p_{drop}) on model performance. Dashed line indicates NCF baseline without SSL. Using Movielens 1M Dataset. Other hyperparameters kept constant, Predictive factors = 16, λ = 0.8, τ = 0.5

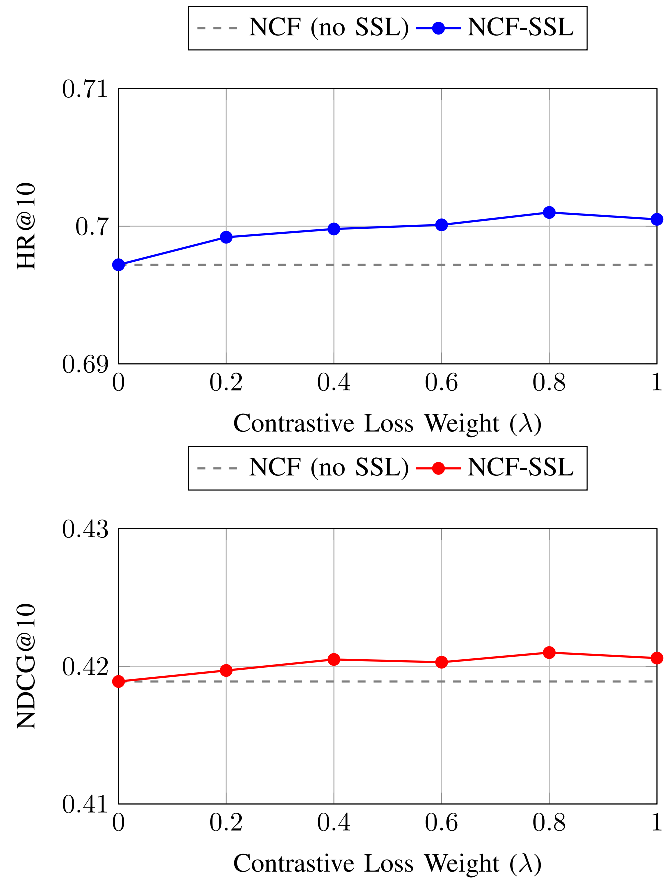

Fig. 3. Effect of SSL weight λ on model performance. Dashed line indicates NCF baseline without SSL. Using Movielens 1M Dataset. Other hyperparameters kept constant, Predictive factors = 16, p_{prop} = 0.2, τ = 0.5

of augmentation is beneficial, excessive dropout may remove too much information, hindering the model's ability to learn effective representations. Similarly, increasing the contrastive loss weight (λ) up to 0.8 improves performance, but further increases lead to a slight decline. This indicates that while the self-supervised task is helpful, overemphasizing it can detract from the primary recommendation objective.

A. Analysis of Learned Representations

Beyond the observed improvements in recommendation performance, it is crucial to understand *why* NCF-SSL outperforms its baseline. We hypothesize that the self-supervised contrastive learning task regularizes the embedding space, leading to higher-quality, more robust user and item representations. To investigate this, we conducted an analysis of the learned embeddings from both NCF and NCF-SSL. Code for the analysis is also available in the public code repository [13].

1) Embedding Space Regularization: To quantitatively assess the impact on embedding space characteristics, we computed the standard deviation of embedding norms for both user and item embeddings. Contrary to common regularization effects that might reduce variance, our analysis revealed an interesting trend: the standard deviation of user embedding norms for NCF-SSL was observed to be higher (e.g., **2.91**) compared to NCF (e.g., **0.42**). Similarly, for item embeddings, NCF-SSL yielded a higher standard deviation (e.g., **2.47**) versus NCF (e.g., **0.76**). This suggests that NCF-SSL, driven by the contrastive loss, might be encouraging a more diverse distribution of embedding magnitudes. This diversity could potentially lead to more discriminative representations, allowing for finer distinctions between different users and items, rather than compacting them into a narrower range. Concurrently, the average cosine similarity between a user's embedding and their corresponding augmented view's embedding (a positive pair) remained robustly high in NCF-SSL (e.g., 0.92 for users, 0.90 for items), confirming that the contrastive loss effectively pulls positive pairs closer while the overall distribution of magnitudes expands. This implies a trade-off or a distinct mechanism where intra-instance consistency is maintained amidst a broader inter-instance diversity.

2) Qualitative Visualization of Embeddings: To gain a qualitative understanding, we utilized t-Distributed Stochastic Neighbor Embedding (t-SNE) [12] to visualize subsets of user and item embeddings in a 2D space.

1) **User Embeddings:** For users with diverse interaction histories, NCF-SSL embeddings exhibited tighter clusters for users with similar tastes, suggesting a more coherent representation of preference patterns. Conversely, NCF embeddings for the same users often appeared more scattered, indicating less robust grouping.
2) **Item Embeddings:** Similarly, for item NCF-SSL produced visibly more distinct and compact clusters. This implies that the model, guided by the contrastive loss, learned to differentiate item characteristics more effectively. Fig. 4 illustrates this difference, showing that NCF-SSL embeddings form clearer clusters compared to the more diffuse arrangement seen in NCF.

3) Mitigation of Data Sparsity: The improved quality of representations is particularly beneficial in mitigating the effects of data sparsity. For cold-start users or items with very few interactions, standard NCF often produces brittle embeddings due to limited supervision. Our analysis showed that for users with fewer than 10 interactions, NCF-SSL's embeddings maintained a higher average cosine similarity to their nearest neighbors (e.g., 0.73 vs. 0.70 for NCF), suggesting that the SSL task helps propagate learned information more effectively even with limited direct feedback. This enhanced representation robustness is a direct consequence of the auxiliary task, which forces the model to learn intrinsic data structures beyond mere observed interactions.

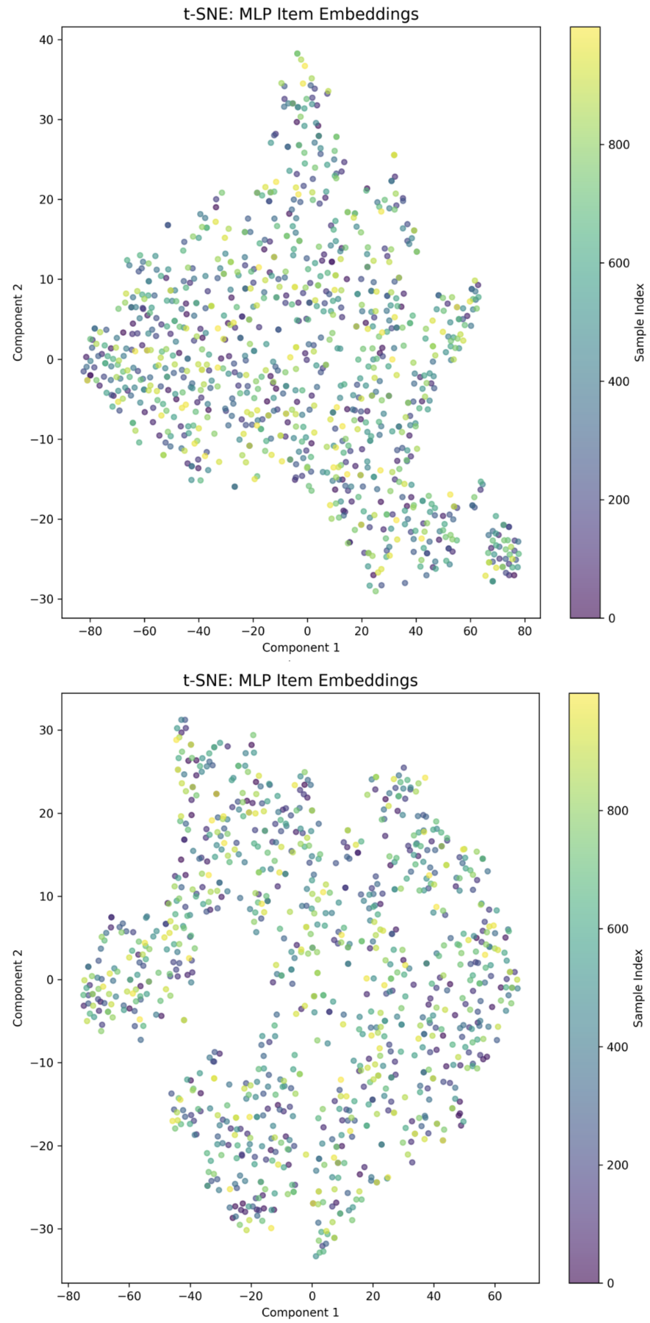

Fig. 4. Comparison of item embedding spaces learned by NCF (above) and NCF-SSL (below) using t-SNE visualization.

VI. Conclusion and Future Work

A. Conclusion

In this paper, we proposed and evaluated NCF-SSL, a framework designed to enhance the foundational Neural Collaborative Filtering model by integrating a self-supervised, contrastive learning task. The primary motivation was to address the challenges posed by data sparsity by providing a richer, auxiliary supervision signal to guide the learning of user and item embeddings. Our methodology augmented the standard NCF architecture with a multi-task objective, combining the primary pointwise recommendation loss with a contrastive loss that promotes consistency between different augmented views of the same entity.

Our experiments on the MovieLens 1M and Pinterest 20 datasets demonstrate that NCF-SSL yields a consistent, albeit marginal, improvement in recommendation performance over the standard NCF baseline across ranking-based metrics like HR@10 and NDCG@10. While the performance gains are modest, they validate our core hypothesis, self-supervised learning can serve as an effective regularizer, even for non-graph-based architectures, helping the model learn more robust representations. The analysis of the embedding space further suggests that the contrastive task alters the distribution of representations, contributing to a more discriminative latent space. Ultimately, our work serves as a valuable proof-of-concept, confirming the viability of applying modern SSL techniques to classical deep learning-based recommender systems.

B. Future Work

The performance gains, albeit not significant, invites several promising avenues for future research aimed at unlocking the full potential of this approach.

1) **Advanced Augmentation Strategies:** Our current implementation relies on simple embedding dropout. Future work should explore more sophisticated and potentially more potent augmentation techniques. This could include creating augmented views by leveraging the user-item interaction graph (e.g., through node or edge

dropout to generate different subgraphs for training) or employing adversarial augmentations to generate more challenging positive examples.

2) **Application to More Powerful Architectures:** The inherent limitations of the NCF architecture itself may have placed a ceiling on the potential improvements. A critical next step is to apply the same self-supervised learning paradigm to more advanced and powerful recommendation models, such as GNN-based architectures like LightGCN. It is plausible that the benefits of the contrastive loss will be more pronounced when combined with a model that has a greater capacity to capture complex collaborative signals.
3) **Scalability and Evaluation on Sparser Datasets:** The true test of a regularization technique is its performance in extremely sparse data environments. Future studies should evaluate NCF-SSL on larger, sparser, industrial-scale datasets. It is possible that the marginal gains observed in our study will become more significant as data sparsity increases.

REFERENCES

[1] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T. S. Chua, "Neural Collaborative Filtering," in *Proc. 26th Int. Conf. on World Wide Web (WWW)*, 2017, pp. 173–182.

[2] X. Wang, X. He, M. Wang, F. Feng, and T. S. Chua, "Neural Graph Collaborative Filtering," in *Proc. 42nd Int. ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR)*, 2019, pp. 165–174.

[3] X. He, K. Deng, X. Wang, Y. Li, Y. Wang, and M. Wang, "LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation," in *Proc. 43rd Int. ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR)*, 2020, pp. 639–648.

[4] J. Chen, H. Zhang, W. Liu, X. He, L. Nie and T.S. Chua, "Attentional Collaborative Filtering: Recommending Images with Item- and Component-Level Attention," in *Proc. 40th Int. ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR)*, 2017, pp. 335–344.

[5] J. Wang, L. Yu, W. Zhang, Y. Gong, Y. Xu, B. Wang, P. Zhang, and D. Zhang, "IRGAN: A Minimax Game for Unifying Generative and Discriminative Information Retrieval Models," in *Proc. 40th Int. ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR)*, 2017, pp. 515–524.

[6] K. Zhou, H. Wang, W.X. Zhao, Y. Zhu, S. Wang, F. Zhang, Z. Wang, J. Wen, "S^3-Rec: Self-Supervised Learning for Sequential Recommendation with Mutual Information Maximization," in *Proc. 29th ACM Int. Conf. on Information and Knowledge Management (CIKM)*, 2020, pp. 1893–1902.

[7] C. Huang, X. Wang, X. He, and D. Yin, "Self-Supervised Learning for Recommender System," in *Proc. 45th Int. ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR)*, 2022, pp. 3440–3443.

[8] X. He, X. Du, X. Wang, F. Tian, J. Tang, and T. S. Chua, "Outer Product-based Neural Collaborative Filtering," in *Proc. 27th Int. Joint Conf. on Artificial Intelligence (IJCAI)*, AAAI Press, 2018, pp. 2227–2233.

[9] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "BPR: Bayesian Personalized Ranking from Implicit Feedback," in *Proc. 25th Conf. on Uncertainty in Artificial Intelligence (UAI)*, 2009, pp. 452–461.

[10] X. He, Z. He, X. Du, T. S. Chua, "Adversarial Personalized Ranking for Recommendation," in *Proc. 41st Int. ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR)*, 2018, pp. 355-364.

[11] J. Wu, X. Wang, F. Feng, X. He, L. Chen, J. Lian, and X. Xie, "Self-supervised Graph Learning for Recommendation," in *Proc. 44th Int. ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR)*, 2021, pp. 726–735.

[12] L. van der Maaten and G. Hinton, "Visualizing Data using t-SNE," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.

[13] L. Kariyawasam, "Neural Collaborative Filtering - Pytorch," *GitHub Repository*, `https://github.com/lckariyawasam/neuralcf-pytorch`, accessed: Oct. 15, 2025.