Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 8, 871-881
2025
Publisher: Learning Gate
DOI: 10.55214/2576-8484.v9i8.9471

Optimizing collaborative filtering recommender systems with the GRPO reinforcement learning algorithm

Hwi Jeon[1], Chan-Ho Lee[2], Jong-Geun Choi[3], Hyuk-Jin Kwon[4*]
[1,2,3,4]Department of Defense AI Applications, Seoul National University of Science and Technology, Seoul 01811, Republic of Korea; hwi042595@seoultech.ac.kr (H.J.) cksgh1211@seoultech.ac.kr (C.H.L.) choijon1@seoultech.ac.kr (J.G.C.) kwonhj@seoultech.ac.kr (H.J.K.).

Abstract: Collaborative filtering recommender systems primarily focus on short-term prediction accuracy but exhibit limitations concerning long-term user satisfaction and content diversity. In this paper, we reinterpret user-item interaction data as reinforcement learning with verifiable rewards and introduce the Group Relative Policy Optimization (GRPO) reinforcement learning algorithm, originally proposed in the large language model domain, to collaborative filtering model fine-tuning for the first time. GRPO directly updates policies without separate critic networks, balancing exploration and exploitation while optimizing long-term user engagement. In experiments conducted on Amazon review datasets covering baby products, video games, and industrial & scientific categories, the GRPO-optimized model achieved up to 15.16% improvement in Recall@10 compared to baseline models. Additionally, we revealed that user embeddings from graph-based collaborative filtering architectures positively contribute to GRPO algorithm optimization, whereas positional embeddings from sequential collaborative filtering architectures impede optimization performance. These findings empirically validate the effectiveness of the GRPO algorithm as a robust approach for recommender system model optimization.

Keywords: *Collaborative filtering, GRPO, Recommender system, Reinforcement learning, RLVR.*

1. Introduction

Recommender systems are core technologies used in various commercial platforms, such as e-commerce and streaming services, to support decision-making by providing personalized content based on users' past behavior. Recommender systems are classified into collaborative filtering, content-based filtering, and hybrid methods that integrate the two. Collaborative filtering methods are rely on the theoretical assumption that users exhibiting similar behaviors share similar preferences. Collaborative filtering methods are further divided into sequential collaborative filtering methods and graph-based collaborative filtering methods, with the former learning users' temporal interaction relationships and the latter learning structural relationships between users and items.

Representative sequential collaborative filtering methods include FPMC [1] which integrates Markov chains and matrix factorization to capture long-term and short-term preferences; RNN and LSTM-based session models; GRU4Rec [2] which addresses long-term dependency issues prior to the advent of the Transformer architecture; and SASRec [3] which introduced the self-attention mechanism from the Transformer architecture.

Meanwhile, graph-based collaborative filtering methods represent interactions between users and items as a bipartite graph structure and reinforce the sparsity of the bipartite graph through multi-layer propagation. For example, NGCF [4] learned local neighborhood information and global collaborative signals using a Graph Convolutional Network to capture higher-order

History: Received: 3 June 2025; Revised: 18 June 2025; Accepted: 23 June 2025; Published: 15 August 2025
* Correspondence: kwonhj@seoultech.ac.kr

connectivity. LightGCN [5] improves efficiency and performance by removing the nonlinear activation of NGCF.

However, when sequential collaborative filtering methods and graph-based collaborative filtering methods are used independently, only their respective advantages are utilized. Therefore, recent studies [6, 7] that integrate the two methods have attracted attention. In particular, the latest model GSAU [8] aligns and homogenizes sequential and graph-based modules in a single embedding space and combines their complementary characteristics to contribute to improved recommendation performance. As shown in Figure 1, the GSAU model achieved significant improvements in performance metrics compared to the traditional sequential collaborative filtering model SASRec and the graph-based collaborative filtering model LightGCN.

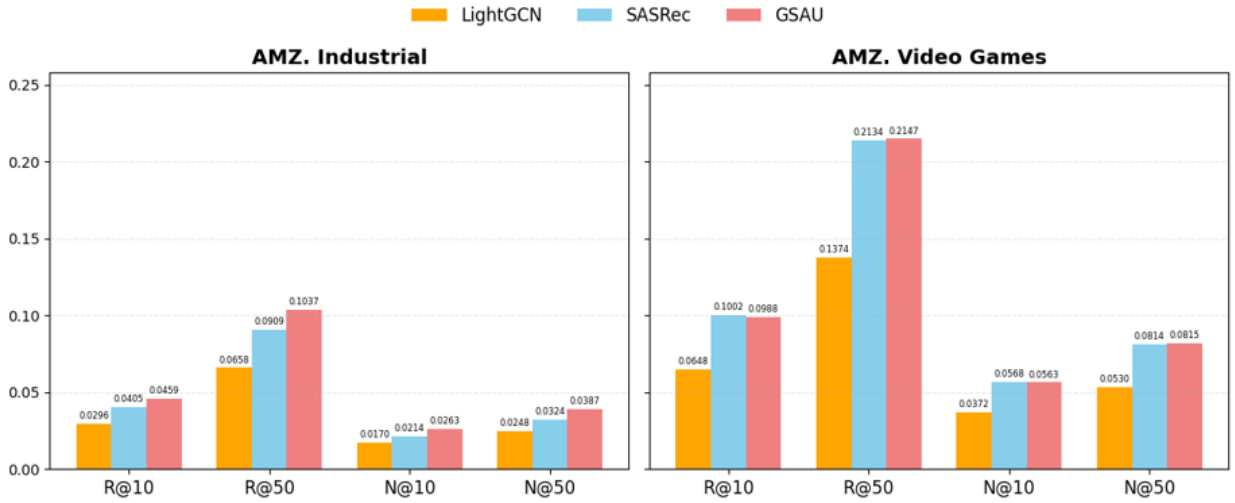

Figure 1.
Comparison of Collaborative Filtering Model Performance Using Amazon Industrial & Scientific and Video Games Data in 2023.

Along with research on integrating collaborative filtering models, research on recommender systems utilizing reinforcement learning [9, 10] is also continuing. This is because existing collaborative filtering-based research has focused only on one-time and immediate predictions, optimizing for short-term performance, and as a result, only recommending items from similar clusters, ultimately reducing user engagement in the long term. Reinforcement learning-based approaches redefine the problem as optimizing user interactions from a long-term perspective, maximizing the diversity of recommendations and user satisfaction through exploration and reinforcement. For example, Chen, et al. [11] successfully applied reinforcement learning agents to YouTube's recommender system, proving its practicality. However, to address policy instability caused by distribution variability in off-policy learning, subsequent research [12] introduced the actor-critic structure.

In this paper, we propose a methodology for applying Deepseek's Group Relative Policy Optimization (GRPO) [13] which was recently proposed in the LLM field as a pure policy-based approach without a separate critic network, to recommender systems. We experimentally verified performance improvements by applying the GRPO algorithm to sequential collaborative filtering, graph-based collaborative filtering, and integrated sequential and graph-based collaborative filtering models. The main contributions of this paper are as follows:

- We are the first to apply the Deepseek GRPO algorithm to a collaborative filtering-based recommender system, demonstrating a significant improvement in recommendation performance.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 8: 871-881, 2025
DOI: 10.55214/2576-8484.v9i8.9471

- We performed policy optimization for each embedding structure of collaborative filtering models and experimentally analyzed and verified the differences in performance.

2. Related Work

2.1. Collaborative Filtering Integration

Researchers have attempted to integrate sequential and graph-based methods to effectively capture user's short-term and long-term preferences. SR-GNN [6] learns sequential relationships between items within a session using a graph neural network and predicts the next item by combining global collaborative signals and session interests based on the last node embedding using attention. FGNN [14] improves recommendation accuracy by adding a Weighted Graph Attention Layer that reflects the directionality and interaction frequency between neighboring nodes, based on the structure of SR-GNN. TAGNN [15] adds a Target Attentive module to the graph within a session, re-weighting to focus more on past behaviors associated with the target item.

While previous studies attempted to integrate sequential and graph structures within a session, COTREC [16] separated the session into internal and external graphs and applied contrastive learning between the two encoders to effectively improve data sparsity issues. GCE-GNN [17] proposed a dual network structure separated into session graphs and global graphs. The former learns sequential relationships between items within the current session, while the latter learns sequential relationships between items globally, demonstrating consistent performance advantages across all benchmark datasets. GAG [18] added user embeddings as global features to both graphs, simultaneously reflecting sequential relationships between items within the current session and global user preferences. However, sequential and graph integration models are optimized for predicting the next item, and thus have limitations in reflecting multi-stage decision-making processes such as long-term user satisfaction and content diversity.

2.2. Reinforcement Learning and Recommender Systems

2.2.1. Reinforcement Learning

Reinforcement learning is a machine learning paradigm in which an agent learns the optimal policy that maximizes the expected value of cumulative rewards through interaction with the environment. Reinforcement learning algorithms are broadly categorized into value-based methods and policy-based methods. The former includes Q-Learning [19] which approximates the optimal action-value function using the Bellman equation, and DQN [20] which extends this to deep neural networks. The latter includes the policy gradient method proposed by Sutton, et al. [21], which directly parameterizes the policy and maximizes the expected cumulative reward using gradient ascent. However, early policy gradient methods relied on Monte Carlo estimation, resulting in high variance and significantly reduced learning stability. To address these issues, TRPO [22] limited the policy gradient update rate using a KL divergence constraint, ensuring monotonic improvement. Furthermore, PPO [23] introduced a clipped surrogate objective function to suppress excessive policy changes, achieving learning stability similar to TRPO while improving computational efficiency.

2.2.2. Reinforcement Learning for LLMs

A notable development in recent reinforcement learning research is that algorithms developed in different fields are being successfully applied to other fields with structurally similar characteristics. OpenAI's GPT-o1 [24] utilized the PPO algorithm to optimize a human feedback-based reward model, thereby enhancing its ability to generate responses aligned with user preferences. A particularly notable development is the GRPO algorithm proposed in DeepSeek-R1. The GRPO algorithm demonstrated that it can effectively improve policy performance without a separate reward model by utilizing only Reinforcement Learning with Verifiable Rewards (RLVR). Yu, et al. [25] proposed a methodology that utilizes the degree of agreement with expert reference answers as a binary signal, and Wu, et al. [26] achieved performance improvements by setting accuracy and visual quality metrics as binary rewards in

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 8: 871-881, 2025
DOI: 10.55214/2576-8484.v9i8.9471

the RLVR-World environment.

2.2.3. Reinforcement Learning for Recommender Systems

Reinforcement learning in recommender systems probabilistically models the uncertainty of user behavior and optimizes long-term user satisfaction and engagement by designing multidimensional reward functions such as click-through rates, purchase conversion rates, and dwell time. This approach differs from traditional supervised learning-based recommendation methods, which focus on short-term prediction accuracy.

PRL Xin, et al. [27] converts past interaction data into state-cumulative reward prompts and replaces the traditional policy evaluation step of value function learning with cross-entropy-based supervised learning, thereby significantly improving recommendation performance without the need for a separate critic network. CSA [28] improves the stability of policy learning by integrating single-state data augmentation and contrastive learning in sparse reward environments. Similarly, MECRL [29] addresses the data scarcity issue in sequential recommendation environments by generating virtual state transitions using an environment model and significantly improving the stability of policy learning and recommendation accuracy through contrastive learning.

However, reinforcement learning research in the existing recommender system field has mainly developed using value-based methods or limited forms of policy gradient methods, and the application of next-generation algorithms such as GRPO, which has recently gained attention in LLM research, has not yet been sufficiently studied. In particular, considering the excellent performance of GRPO in RLVR environments proven in LLM research, it is necessary to verify the applicability and effectiveness of GRPO in the recommender systems field, which has similar structural characteristics.

3. Methodology

In this section, we describe how to perform policy gradient updates by applying the GRPO algorithm to existing collaborative filtering models. The rationale for applying the GRPO algorithm is as follows:

- Reinforcement Learning with Verifiable Rewards. The GRPO algorithm uses a binary reward structure of acceptance or rejection for each token during the LLM learning process. This can be interpreted similarly to user interactions such as clicks or non-clicks in recommender systems.
- High-dimensional action space. DeepSeek-R1 selects and generates tokens from a vast vocabulary. This can be interpreted as similar to the structure of selecting one recommendation from a vast pool of recommendations in a recommender system.

Based on these commonalities, this section proposes a method for implementing an effective recommender system by interpreting user interactions in collaborative filtering as verifiable binary reward signals and utilizing the GRPO algorithm to learn the optimal policy from a vast pool of recommendation candidates.

3.1. Collaborative Filtering Model

3.1.1. Sequential Collaborative Filtering Model

Sequential collaborative filtering models analyze historical user behavior in chronological order. Let $u \in \mathcal{U}$ be a user who interacts with item $i_t^u \in \mathcal{I}$ at time t. The chronologically ordered sequence of items with which user u has interacted is denoted by $\mathcal{S}^u = \left(i_1^u, i_2^u, \ldots, i_{|\mathcal{S}^u|}^u\right)$. For every item $i \in \mathcal{I}$, we define a trainable item embedding matrix $M \in R^{|\mathbb{I}| \times d}$ whose rows are d-dimensional vectors. We further introduce a position embedding matrix $P \in R^{|\mathbb{S}^{\mathbb{u}}| \times d}$ that encodes the position of each item in the sequence. By selecting the rows of M that correspond to $\mathcal{S}^u$ and adding the associated position embeddings, we obtain the sequential embeddings $E_{sequential}$ as follows:

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 8: 871-881, 2025
DOI: 10.55214/2576-8484.v9i8.9471

$$E_{sequential} = \begin{bmatrix} M_{i_1^u} + P_1 \\ M_{i_2^u} + P_2 \\ \vdots \\ M_{i_{|S^u|-1}^u} + P_{|S^u|-1} M_{i_{|S^u|}^u} + P_{|S^u|} \end{bmatrix} \in R^{|\mathbb{S}^{\mathbb{u}}| \times d} \quad (1)$$

The models that generate embeddings using the above architecture are GRU4Rec, which uses gated recurrent unit, and SASRec, which replaces recurrent layers with Self-Attention.

3.1.2. Graph-based Collaborative Filtering Model

Graph-based collaborative filtering models analyze the structural relationships between users and items via graph propagation. Given a bipartite graph $\mathcal{G} = (\mathcal{U} \cup \mathcal{I}, \mathcal{E})$ with user set $\mathcal{U} = (u_1, \ldots, u_{|\mathcal{U}|})$ and item set $\mathcal{I} = (i_1, \ldots, i_{|\mathcal{I}|})$ the observed interaction set $\mathcal{E} \subseteq \mathcal{U} \times \mathcal{I}$ is used for training. First, we define an initial embedding matrix $M^{(0)} \in R^{(|\mathbb{U}|+|\mathbb{I}|) \times d}$ that stores trainable d-dimensional vectors for all nodes. We then construct the adjacency matrix $A \in (0,1)^{(|\mathcal{U}|+|\mathcal{I}|) \times (|\mathcal{U}|+|\mathcal{I}|)}$, with $A_{u,i} = 1$ if $(u,i) \in \mathcal{E}$, 0 otherwise. The adjacency matrix is normalized as $\tilde{A} = D^{-1/2} A D^{-1/2}$ where D is the diagonal degree matrix. By propagating embeddings through K layers, we obtain $(M^{(0)}, \ldots, M^{(K)})$. The final graph embedding is generated as follows:

$$\boldsymbol{E_{graph}} = \sum_{k=0}^{K} \alpha_k \boldsymbol{M^{(k)}} \in R^{(|\mathbb{U}|+|\mathbb{I}|) \times d} \quad (2)$$

The models that generate embeddings using the above architecture are GraphSAGE [30] and LightGCN

3.1.3. Integrated Collaborative Filtering Model

The integration of sequential and graph modules is achieved by jointly learning both the temporal interaction sequence $\mathcal{S}^u = \left(i_1^u, i_2^u, \ldots, i_{|\mathcal{S}^u|}^u\right)$ and the bipartite graph $\mathcal{G} = (\mathcal{U} \cup \mathcal{I}, \mathcal{E})$ in a single embedding space. To store trainable d-dimensional vectors for every node $v \in \mathcal{U} \cup \mathcal{I}$, we define an initial embedding matrix $M^{(0)} \in R^{(|\mathbb{U}|+|\mathbb{I}|) \times d}$. The sequential encoder then computes representations $e_v^{sequential}$ and the graph encoder computes e_v^{graph}. By applying an alignment and uniformity loss over the two views of each node $\left(e_v^{sequential}, e_v^{graph}\right)$, we obtain the final integrated embeddings as follows:

$$E_{integrated} = \beta\, e_v^{graph} + (1 - \beta)\, e_v^{sequential} \quad (3)$$

The model that generates embeddings using the above architecture is GSAU.

3.2. Applying the GRPO Algorithm in Recommender Systems

We fine-tune the embeddings from pre-trained collaborative filtering models by applying the GRPO algorithm. When GRPO is applied, the sequential model state is defined as $s_{sequential} \coloneqq e_v^{sequential} = \left(e_{item}, e_{position}\right)$, the graph model state as $s_{graph} \coloneqq e_v^{graph} = (e_{item}, e_{user})$, and the integrated model state as $s_{integrated} \coloneqq e_v^{integrated} = \left(e_{item}, e_{position}, e_{user}\right)$. Given a state s, we sample G candidate items $(o_1, \ldots, o_G)$ from the old policy $\pi_{\theta_{\text{old}}}$; for each o, we observe a verifiable binary reward $r_t = r(s,o) \in (0,1)$ and estimate the current policy's success probability p by $p = \frac{1}{G}\sum_{t=1}^{G} r_t$. We then define the GRPO advantage estimate $A(s,o)$ as follows:

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 8: 871-881, 2025
DOI: 10.55214/2576-8484.v9i8.9471

$$A(s,o) = \begin{cases} \sqrt{\frac{1-p}{p}} & \text{if } r(s,o) = 1 \\ -\sqrt{\frac{p}{1-p}} & \text{if } r(s,o) = 0 \end{cases} \quad (4)$$

Next, we compute the importance ratio $\rho_t = \frac{\pi_{\theta_{new}}(o_t|s)}{\pi_{\theta_{old}}(o_t|s)}$, and apply the PPO clipping to each candidate o_t to obtain gradient optimization objective as follows:

$$J_{\text{GRPO}}(\theta) = \frac{1}{G}\sum_{t=1}^{G}\left[\min\left(\rho_t A(s,o_t),\ \text{clip}(\rho_t, 1-\varepsilon, 1+\varepsilon)\, A(s,o_t)\right) - \beta\, D_{\text{KL}}\left(\pi_{\theta_{\text{old}}}|\pi_{\text{ref}}\right)\right] \quad (5)$$

Here, $\varepsilon \in (0,1)$ is a hyperparameter that bounds the importance ratio within $[1-\varepsilon,\ 1+\varepsilon]$, and $\beta > 0$ is the KL-regularization coefficient controlling the divergence between the new policy and the reference policy. Through this process, GRPO repeatedly amplifies the reference policy's success probability p and enhances convergence stability during training, thereby achieving model optimization [31].

4. Results

4.1. Experimental Setup

4.1.1. Datasets

For model evaluation, we utilized datasets from Amazon recommendation reviews Hou, et al. [32] in the fields of baby products, video games, and industrial and scientific. We intentionally selected datasets with varying statistics regarding the number of users and items to comprehensively evaluate the models' performance.

Table 1 represents the statistics of preprocessed datasets. In previous studies [3, 5] as part of core data filtering, users and items with fewer than interactions were removed only once. However, in recent studies [8, 33] the core data filtering method is repeatedly executed until there are no more items to remove. Therefore, in this study, users and items with fewer than five interactions were repeatedly removed to preprocess the data.

Table 1.
Statistics of the dataset after preprocessing. Avg. Len denotes the average sequence length of users.

Datasets	#Users	#Items	Interactions	Avg. Len
Baby Products	155.434	36.845	1.287.653	8.28
Video Games	98.906	26.354	857.505	8.67
Industrial & Scientific	54.567	27.229	446.046	8.17

4.1.2. Baseline

In this study. the GRPO algorithm was applied to three baseline models. Among sequential collaborative filtering models. SASRec was used as the baseline model; among graph-based collaborative filtering models. LightGCN was used; and among sequential and graph integrated collaborative filtering models. GSAU was used.

4.1.3. Evaluation Setting

We grouped all interaction data by user and sorted each user's interaction history chronologically. The last interaction data in the sorted data was assigned to the evaluation set. the previous interaction data was assigned to the validation set. and the rest was used as the training set. In the evaluation

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 8: 871-881, 2025
DOI: 10.55214/2576-8484.v9i8.9471

phase. instead of the sampling method used in previous studies. we performed a full-ranking evaluation [33, 34] with all items as candidates. Additionally. we masked past items that had already been exposed during the training phase to exclude them from the evaluation. For quantitative comparison and evaluation. we used Recall and Normalized Discounted Cumulative Gain (NDCG) scores. which are widely used as evaluation metrics in recommender systems.

4.1.4. Implementation Details

The hyperparameters of the three baseline collaborative filtering models were set identically for fair comparison. The batch size is 2048. the random seed is 2020. and the epoch is 200. For GRPO learning. the epoch is 1.000. and the specific hyperparameter settings are shown in Table 2.

Table 2.
Collaborative Filtering Models and GRPO Learning Hyperparameters.

Datasets	Learning rate	Dropout	Embedding dimension d	Sample candidates G	β	ε
Baby Products	0.0001	0.5	64	4096	0.01	0.1
Video Games	0.0001	0.5	64	4096	0.01	0.1
Industrial & Scientific	0.0001	0.5	64	4096	0.01	0.1

4.2. Performance Comparison

The experimental results show that optimizing the collaborative filtering model using the GRPO algorithm yields different performance depending on the model architecture. As shown in Table 3. $GSAU - GRPO_{ourmethod}$. in which GRPO is applied to the integrated sequential-graph model GSAU. consistently improves recommendation performance across all datasets. On the Industrial and Scientific and Video Games datasets. $GSAU - GRPO_{ourmethod}$ yields modest relative gains of approximately 1.5%-3%. Most notably. on the Baby Products dataset it achieves substantial improvement of about 13%-15%. with the largest performance improvement of about 15.16% in the Recall@10. Likewise. $LightGCN - GRPO_{ourmethod}$. which optimized the graph-based model LightGCN with GRPO. showed a slight performance improvement of approximately 0.07%-0.81% on the Industrial and Scientific and Video Games datasets. while exhibiting no meaningful change on the Baby Products dataset.

In contrast. $SASRec - GRPO_{ourmethod}$. in which GRPO is applied to the sequential model SASRec. suffers performance degradation on every dataset. In particular. on the Video Games dataset. NDCG@10 suffers the most significant drop. decreasing by 38.3%. This suggests that the policy gradient update process for item and positional embeddings in sequential collaborative filtering models may differ from that in graph and integrated models.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 8: 871-881, 2025
DOI: 10.55214/2576-8484.v9i8.9471

Table 3.
Comparison of top recommendation performance between our method models and baseline models and GRPO algorithms applied to Amazon industrial and scientific. video games. and baby products review datasets. R stands for recall. and N stands for NDCG evaluation metrics. Superior performance is indicated in bold.

Method	Baby Products				Video Games				Industrial & Scientific			
	R@10	**R@50**	**N@10**	**N@50**	**R@10**	**R@50**	**N@10**	**N@50**	**R@10**	**R@50**	**N@10**	**N@50**
SASRec	0.0394	0.0976	0.0219	0.0344	0.1005	0.2133	0.0569	0.0815	0.0408	0.0940	0.0232	0.0346
$SASRec - GRPO_{ourmethod}$	0.0306	0.0858	0.0164	0.0282	0.0639	0.1529	0.0351	0.0543	0.0389	0.0976	0.0202	0.0329
LightGCN	0.0209	0.0487	0.0122	0.0182	0.0648	0.1374	0.0372	0.0530	0.0296	0.0658	0.0170	0.0248
$LightGCN - GRPO_{ourmethod}$	0.0209	0.0487	0.0122	0.0182	0.0649	0.1375	0.0372	0.0530	0.0297	0.0660	0.0171	0.0250
GSAU	0.0322	0.0805	0.0177	0.0281	0.0967	0.2139	0.0546	0.0801	0.0459	0.1037	0.0263	0.0387
$GSAU - GRPO_{ourmethod}$	0.0366	0.0927	0.0200	0.0321	0.0995	0.2184	0.0555	0.0813	0.0468	0.1063	0.0267	0.0395

Table 4.
Performance differences between policy gradient update targets. Bold text indicates the highest performance. and underlined text indicates the second highest performance.

Method	Baby Products				Video Games				Industrial & Scientific			
	R@10	**R@50**	**N@10**	**N@50**	**R@10**	**R@50**	**N@10**	**N@50**	**R@10**	**R@50**	**N@10**	**N@50**
GSAU	0.0322	0.0805	0.0177	0.0281	0.967	0.2139	0.0546	0.0801	0.0459	0.1037	0.0263	0.0387
e_{item}	0.0364	0.0921	0.0199	0.0319	0.0993	0.2181	0.0554	0.0812	0.0467	0.1062	0.0267	0.0395
$e_{item} \cdot e_{user}$	0.0366	0.0927	0.0200	0.0321	0.0995	0.2184	0.0555	0.0813	0.0468	0.1063	0.0267	0.0395
$e_{item} \cdot e_{position}$	0.0314	0.0776	0.0173	0.0272	0.0875	0.1969	0.0486	0.0723	0.0451	0.1019	0.0257	0.0380
$e_{item} \cdot e_{user} \cdot e_{position}$	0.0317	0.0776	0.0174	0.0273	0.0876	0.1972	0.0487	0.0725	0.0451	0.1021	0.0258	0.0381

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 8: 871-881, 2025
DOI: 10.55214/2576-8484.v9i8.9471

4.3. Ablation Study

In this section. we conducted the ablation study to verify the performance changes according to the collaborative filtering model architecture when updating the GRPO policy. We selectively used item embeddings. user embeddings. and positional embeddings. which constitute the integrated collaborative filtering model GSAU. for policy optimization. As shown in Table 4. when only item embeddings were used for policy gradient updates. and when both item embeddings and user embeddings were used for policy gradient updates. consistent performance improvements were observed across all datasets—baby products. video games. and industrial and scientific—compared to the baseline model GSAU. In particular. the greatest performance improvement was observed when both item embeddings and user embeddings were used for policy gradient updates. suggesting that user embeddings are the most significant positive factor in GRPO reinforcement learning.

In contrast. when item embeddings. user embeddings. and positional embeddings were all used in policy gradient updates. and when item embeddings and positional embeddings were used in policy gradient updates. consistent performance degradation was observed across all datasets compared to the baseline model GSAU. In particular. when item embeddings and positional embeddings were used together for policy gradient updates. the largest performance degradation was observed. suggesting that positional embeddings are the factor that has the greatest negative effect in GRPO reinforcement learning. This suggests that policy gradient updates of positional embeddings may destabilize the advantage estimates in the GRPO calculation process. distorting the importance ratio distribution.

These observations demonstrate that embedding selection is as important as model architecture in optimizing GRPO-based collaborative filtering models. In particular. the significant performance degradation observed in sequential models suggests that updated positional embeddings were a major contributing factor.

5. Conclusion

This paper proposes a new method for integrating the GRPO algorithm into a collaborative filtering-based recommender system. Unlike traditional collaborative filtering models. which primarily focus on short-term prediction accuracy. this paper utilizes reinforcement learning to focus on long-term user satisfaction and prediction accuracy. achieving performance optimization using various real-world datasets. The GRPO reinforcement learning-based model achieved superior performance compared to the baseline model across multiple metrics. Notably. it demonstrated up to a 15% performance improvement on the baby products dataset. Through removal experiments. the role of the GRPO policy gradient update target embeddings was validated. and the most outstanding performance optimization was achieved when both item embeddings and user embeddings were updated via the policy.

Funding:

This study was financially supported by Seoul National University of Science and Technology.

Transparency:

The authors confirm that the manuscript is an honest. accurate. and transparent account of the study; that no vital features of the study have been omitted; and that any discrepancies from the study as planned have been explained. This study followed all ethical practices during writing.

References

[1] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme, "Factorizing personalized markov chains for next-basket recommendation," presented at the Proceedings of the 19th international conference on World wide web (pp. 811-820), 2010.

[2] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, "Session-based recommendations with recurrent neural networks," *arXiv preprint arXiv:1511.06939*, 2015.

[3] W. C. Kang and J. McAuley, "Self-attentive sequential recommendation," presented at the IEEE International Conference on Data Mining (ICDM) (pp. 197-206). IEEE, 2018.

[4] X. Wang, X. He, M. Wang, F. Feng, and T.-S. Chua, "Neural graph collaborative filtering," in *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2019.

[5] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, "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*, 2020.

[6] S. Wu, Y. Tang, Y. Zhu, L. Wang, X. Xie, and T. Tan, "Session-based recommendation with graph neural networks," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 1, pp. 346–353, 2019.

[7] J. Chang et al, "Sequential recommendation with graph neural networks," in *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2021.

[8] Y. Cao et al, "Graph-sequential alignment and uniformity: toward enhanced recommendation systems," in *Companion Proceedings of the ACM on Web Conference*, 2025.

[9] X. Zhao, L. Xia, L. Zou, H. Liu, D. Yin, and J. Tang, "Whole-chain recommendations," in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 2020.

[10] L. Zou, L. Xia, Z. Ding, J. Song, W. Liu, and D. Yin, "Reinforcement learning to optimize long-term user engagement in recommender systems," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019.

[11] M. Chen, A. Beutel, P. Covington, S. Jain, F. Belletti, and E. H. Chi, "Top-k off-policy correction for a REINFORCE recommender system," in *Proceedings of the twelfth ACM International Conference on Web Search and Data Mining*, 2019.

[12] M. Chen, C. Xu, V. Gatto, D. Jain, A. Kumar, and E. Chi, "Off-policy actor-critic for recommender systems," in *Proceedings of the 16th ACM Conference on Recommender Systems*, 2022.

[13] D. Guo *et al.*, "Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning," *arXiv preprint arXiv:2501.12948*, 2025.

[14] R. Qiu, J. Li, Z. Huang, and H. Yin, "Rethinking the item order in session-based recommendation with graph neural networks," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2019.

[15] F. Yu, Y. Zhu, Q. Liu, S. Wu, L. Wang, and T. Tan, "TAGNN: Target attentive graph neural networks for session-based recommendation," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020.

[16] X. Xia, H. Yin, J. Yu, Y. Shao, and L. Cui, "Self-supervised graph co-training for session-based recommendation," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 2021, pp. 2180-2190.

[17] Z. Wang, W. Wei, G. Cong, X.-L. Li, X.-L. Mao, and M. Qiu, "Global context enhanced graph neural networks for session-based recommendation," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020, pp. 169-178.

[18] R. Qiu, H. Yin, Z. Huang, and T. Chen, "Gag: Global attributed graph neural network for streaming session-based recommendation," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020, pp. 669-678.

[19] C. J. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, no. 3, pp. 279-292, 1992.

[20] V. Mnih *et al.*, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.

[21] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," *Advances in Neural Information Processing Systems*, vol. 12, pp. 1057–106, 1999.

[22] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *International Conference on Machine Learning*, 2015: PMLR, pp. 1889-1897.

[23] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[24] L. Ouyang *et al.*, "Training language models to follow instructions with human feedback," *Advances in Neural Information Processing Systems*, vol. 35, pp. 27730–27744, 2002.

[25] D. Yu *et al.*, "Crossing the reward bridge: Expanding rl with verifiable rewards across diverse domains," *arXiv preprint arXiv:2503.23829*, 2025.

[26] J. Wu, S. Yin, N. Feng, and M. Long, "RLVR-world: Training world models with reinforcement learning," *arXiv preprint arXiv:2505.13934*, 2025.

[27] X. Xin, T. Pimentel, A. Karatzoglou, P. Ren, K. Christakopoulou, and Z. Ren, "Rethinking reinforcement learning for recommendation: A prompt perspective," in *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2022, pp. 1347-1357.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 8: 871-881, 2025
DOI: 10.55214/2576-8484.v9i8.9471

[28] Z. Ren *et al.*, "Contrastive state augmentations for reinforcement learning-based recommender systems," in *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2023, pp. 922-931.
[29] C. Li *et al.*, "Model-enhanced contrastive reinforcement learning for sequential recommendation," *arXiv preprint arXiv:2310.16566*, 2023.
[30] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," *Advances in Neural Information Processing Systems*, vol. 30, pp. 1024–1034, 2017.
[31] Y. Mroueh, "Reinforcement Learning with Verifiable Rewards: GRPO's Effective Loss, Dynamics, and Success Amplification," *arXiv preprint arXiv:2503.06639*, 2025.
[32] Y. Hou, J. Li, Z. He, A. Yan, X. Chen, and J. McAuley, "Bridging language and items for retrieval and recommendation," *arXiv preprint arXiv:2403.03952*, 2024.
[33] W. X. Zhao *et al.*, "Recbole: Towards a unified, comprehensive and efficient framework for recommendation algorithms," in *proceedings of the 30th acm International Conference on Information & Knowledge Management*, 2021, pp. 4653-4664.
[34] W. Krichene and S. Rendle, "On sampled metrics for item recommendation," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 1748-1757.

Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 9, No. 8: 871-881, 2025
DOI: 10.55214/2576-8484.v9i8.9471