SIMPLE YET EFFECTIVE MARGIN SEMANTIC EXTRACTION FOR LARGE LANGUAGE MODELS ENHANCED RECOMMENDER SYSTEMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent advances in large language models (LLMs) have introduced rich semantic information into recommender systems. However, LLMs-derived semantics are not inherently compatible with collaborative signals, making it difficult to fully exploit their benefits and potentially causing information distortion. To address this issue, we shift from fusion or alignment to extraction and propose a model-agnostic framework, **MarRec**, which imposes a lightweight margin constraint as an implicit information bottleneck to effectively extract useful semantics. Additionally, we design two instance margin variants and verify that fine-grained constraints can further improve useful semantic extraction. Extensive experiments demonstrate that MarRec outperforms baselines while maintaining high efficiency and showing strong robustness under data sparsity, long-tail, and cold-start scenarios. [1]

1 INTRODUCTION

With the development of large language models (LLMs), recommender systems are undergoing significant technological transformations. Although traditional methods (Rao et al., 2015; Li et al., 2017; Ding et al., 2020; Zhang et al., 2022; Ju et al., 2024) have been widely applied, they still face numerous challenges including data sparsity, long-tail, and cold-start problems. In contrast, LLMs-based recommender systems (Liu et al., 2024; Kong et al., 2024; Wu et al., 2024; Yoo et al., 2026; Shirkavand et al., 2026) offer new possibilities to address these issues, owing to their powerful capabilities in semantic understanding and content generation. In particular, one major direction in LLMs-based recommender systems (Wang et al., 2025a; Xu et al., 2025) leverages LLMs to generate semantic representations, which supplies recommender model with meaningful semantic information.

Traditional recommender models are mainly optimized through user-item interactions and capture implicit preference patterns. Although content-relevant semantics derived from LLMs prove to be powerful for recommender systems, however, only part of the semantics is directly used for recommendation, since content relevance differs from implicit preference. Therefore, the semantics from LLMs are not directly compatible with recommender model, making it difficult to fully exploit the advantages of LLMs (Bao et al., 2023; Wang et al., 2024; Jia et al., 2025; Liu et al., 2025).

Moreover, existing works mainly adopt concatenation (Xi et al., 2024) or alignment (Ren et al., 2024; Wang et al., 2025b) strategies. Due to the aforementioned incompatibility, concatenation fusion tends to introduce irrelevant semantics, whereas alignment forces dense semantics to fit sparse and biased collaborative signals. Consequently, semantics useful for recommendation may be discarded, while irrelevant semantics risk disrupting collaborative signals and degrading performance. We refer to this mismatch as information distortion.

To address this issue, we propose a novel two-stage **Mar**gin semantic extraction framework for **Rec**ommendation (**MarRec**), which introduces a controllable extraction boundary to preserve preference-relevant semantics while filtering out irrelevant semantics that may distort collaborative signals. The first stage introduces a lightweight fixed margin constraint as a controllable extraction boundary, which encourages the recommender model to absorb semantics only when they improve

[1] Our code is available at `https://anonymous.4open.science/r/MarRec-anonymous/`.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

preference modeling, while preventing excessive optimization toward biased collaborative supervision. Since part of the extracted semantics is encoded in non-trainable semantic embeddings, Stage II re-parameterizes these semantics as trainable ID embeddings, enabling standard recommender training to further optimize them as effective prior knowledge. In this way, the margin constraint acts as an **implicit information bottleneck** (Wu et al., 2020; Dubois et al., 2020; Yu et al., 2024; Butakov et al., 2024; Zhu et al., 2025; Zhang et al., 2025a; Ouyang et al., 2025; Kowsher et al., 2026), filtering out redundant semantics and retaining preference-relevant semantics to be injected into the recommender model. As a widely studied information-theoretic framework for representation learning, the information bottleneck provides a theoretical basis for our margin semantic extraction.

Furthermore, beyond a fixed margin shared by all training instances, we design two per-instance margin variants to investigate whether finer-grained boundary control can further improve semantic extraction: (1) **Adaptive Margin** learns an instance-specific margin through a hyperparameter meta-learning network (Liu et al., 2019; Shu et al., 2019), allowing the extraction boundary to adapt to the training state of each instance. (2) **Soft Margin** derives an instance-specific margin from the post-training objective of a recommender model as a data-dependent reference to guide semantic extraction. These variants offer deeper insights into the role of margin, showing that effective semantic extraction depends on controlling how much semantic information should be absorbed per instance.

Our MarRec framework has two notable advantages: *effective semantic extraction* and *lightweight, flexible design*. Firstly, guided by the information bottleneck principle, MarRec adopts a controllable margin constraint to extract useful semantics without relying on complex mutual information estimation. Secondly, MarRec is flexible with auxiliary design: it allows for a simplified auxiliary loss in the first stage to reduce training cost, particularly beneficial when the removed loss is computationally expensive. Since MarRec introduces only minimal additional operations during training and no extra cost during inference, its computational complexity primarily depends on the base recommender model, making it both efficient and scalable.

Overall, the main contributions of our work are as follows:

- We propose a model-agnostic LLMs-enhanced recommender framework that employs a lightweight margin constraint to extract preference-relevant semantics, rather than fusion or alignment.
- We provide theoretical analysis to demonstrate that MarRec aligns with the information bottleneck principle. In addition, we design two instance-level margin variants to further investigate fine-grained constraints and provide deeper insights into the margin constraint mechanism.
- We conduct extensive experiments demonstrating that MarRec outperforms baselines across multiple recommender models in performance while maintaining competitive efficiency, achieves notable robustness in data sparsity, long-tail effect, and cold-start settings, and further verifies that existing concatenation and alignment methods may induce information distortion.

2 Problem Formulation

In this work, we consider a general LLMs-enhanced recommender setting that consists of user-item interactions and semantic embeddings. These semantic embeddings are obtained by first using LLMs to refine raw textual data (e.g., descriptions or comments), and then encoding the processed text with a textual embedding model. Formally, we define the user-item interactions as a binary matrix R, where $R_{ui} = 1$ indicates that user u has interacted with item i, and $R_{ui} = 0$ otherwise. Each user u and item i are associated with semantic embeddings $\mathbf{e}_u^s$, $\mathbf{e}_i^s \in \mathbb{R}^{d_s}$, respectively. The goal is to leverage semantic embeddings with collaborative signals derived from R to learn user and item representations $\mathbf{h}_u$ and $\mathbf{h}_i$, such that the predicted score $\hat{y}_{ui} = \mathbf{h}_u^\top \mathbf{h}_i$ reflects the relevance between user u and item i. Finally, items are ranked by $\hat{y}_{ui}$ to generate recommendations for user u.

3 Methodology

3.1 Overview

We present the overall architecture of MarRec in Figure 1, which enhances recommender models with semantic embeddings derived from LLMs. Instead of concatenating or aligning, MarRec selectively extracts semantic information that improves preference discrimination. Specifically, MarRec adopts

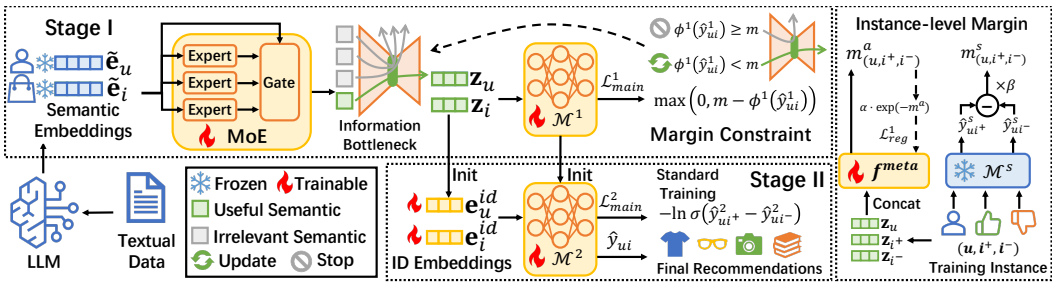

Figure 1: The overall architecture of MarRec. The left shows the pipeline of our two-stage training framework, and the right illustrates the margin constraint with adaptive and soft margins.

a two-stage training framework guided by a margin constraint, which serves as a preference-aware extraction boundary to mitigate the incompatibility between semantic information and collaborative signals. For convenience, we refer to the base recommender model in our framework as base model.

3.2 Stage I: Margin Constraint Training

The first stage aims to extract useful semantics under a controllable margin constraint. Since LLMs-derived embeddings are high-dimensional and contain redundant information, we first project them to match the capacity of the base model, which serves as a compact interface for subsequent margin-based extraction. To avoid excessive information loss, we employ a Mixture-of-Experts (MoE) architecture to enhance the expressive ability of the projection module. This process is formulated as:

$$\mathbf{z}_u = \mathbf{MoE}\left(\tilde{\mathbf{e}}_u^s\right), \quad \mathbf{z}_i = \mathbf{MoE}\left(\tilde{\mathbf{e}}_i^s\right), \tag{1}$$

where $\tilde{\mathbf{e}}_u^s$, $\tilde{\mathbf{e}}_i^s$ are frozen semantic embeddings; **MoE** is the projection module; $\mathbf{z}_u, \mathbf{z}_i \in \mathbb{R}^d$ are the projected semantic embeddings with $d \ll d_s$. We adopt a dense MoE without top-k expert selection. Each expert network and the gating network are implemented as a one-layer linear transformation.

The projected embeddings are then fed into the base model for margin-constrained semantic extraction. The margin constraint imposes a boundary on the main loss and stops an instance from contributing gradients once its preference training objective exceeds the margin threshold. By enforcing an upper bound on the optimization, the constraint serves as an implicit information bottleneck that encourages the base model to extract semantics only when they contribute to preference discrimination. This design mitigates information distortion by preventing the model from continuously absorbing unnecessary semantics to further fit collaborative supervision. This process is formulated as:

$$\mathbf{h}_u^1, \mathbf{h}_i^1 = \mathcal{M}^1\left(\mathbf{z}_u, \mathbf{z}_i, R\right), \quad \hat{y}_{ui}^1 = {\mathbf{h}_u^1}^\top \mathbf{h}_i^1, \tag{2}$$

$$\mathcal{L}_{main}^1 = \sum_{(u,i)\in R} \max\left(0,\ m - \phi^1(\hat{y}_{ui}^1)\right), \tag{3}$$

where $\mathcal{M}^1$, $\mathbf{h}_u^1$, $\mathbf{h}_i^1$, $\hat{y}_{ui}^1$, and $\mathcal{L}_{main}^1$ denote the base model, representations, predicted score, and main loss in the first stage; $\phi^1(\cdot)$ is the training objective applied to $\hat{y}_{ui}^1$; and m is the margin threshold used to stop updating training instance (u, i) when its training objective $\phi^1(\hat{y}_{ui}^1)$ exceeds m.

To implement the training objective that captures relative preferences, we apply $\phi^1(\cdot)$ as a pairwise score difference between an interacted positive item i^+ and a randomly sampled non-interacted negative item i^- for the same user u. Based on this definition, we derive $\phi^1(\cdot)$ and $\mathcal{L}_{main}^1$ as:

$$\phi^1(\hat{y}_{ui}^1) = \phi^1(\hat{y}_{ui^+}^1, \hat{y}_{ui^-}^1) = \hat{y}_{ui^+}^1 - \hat{y}_{ui^-}^1, \tag{4}$$

$$\mathcal{L}_{main}^1 = \sum_{(u,i^+,i^-)} \max\left(0,\ m - \hat{y}_{ui^+}^1 + \hat{y}_{ui^-}^1\right). \tag{5}$$

3.3 Fine-grained Margin Constraint

Building upon the margin constraint, we adopt a fixed margin as the default setting, where m is a hyperparameter applied uniformly across all training instances. However, the fixed-margin strategy may overlook the dynamics of the training process and the diverse patterns exhibited by different instances. To further explore the effectiveness of fine-grained constraints, we design two instance-level margin variants and examine how the extraction boundary influences semantic extraction.

3.3.1 ADAPTIVE MARGIN

To capture instance-specific training dynamics, we propose a meta-learning network to generate an adaptive margin $m^a_{(u,i^+,i^-)}$ for each instance (u, i^+, i^-). This network takes the concatenated representations of the user, positive and negative items as input and outputs a tailored margin value:

$$m^a_{(u,i^+,i^-)} = \boldsymbol{f}^{\text{meta}}\left(\text{Concat}\left(\mathbf{h}^1_u, \mathbf{h}^1_{i^+}, \mathbf{h}^1_{i^-}\right)\right), \tag{6}$$

where $\text{Concat}(\cdot)$ denotes the vector concatenation, and $\boldsymbol{f}^{\text{meta}}(\cdot) : \mathbb{R}^{3d} \to \mathbb{R}^1$ is a linear network.

By assigning a dynamic margin to each instance, the model can adapt to different optimization states. To prevent the margins from collapsing to small values, we introduce the regularization term:

$$\mathcal{L}^1_{reg} = \alpha \cdot \sum_{(u,i^+,i^-)} \exp(-m^a_{(u,i^+,i^-)}), \tag{7}$$

$$\mathcal{L}^1_{main} = \sum_{(u,i^+,i^-)} \max(0, m^a_{(u,i^+,i^-)} - \phi^1(\hat{y}^1_{ui})) + \mathcal{L}^1_{reg}, \tag{8}$$

where $\mathcal{L}^1_{reg}$ is the regularization term, and α is the regularization weight hyperparameter.

3.3.2 SOFT MARGIN

To capture the diverse preference patterns exhibited by different training instances, we derive the soft margin $m^s_{(u,i^+,i^-)}$ from the post-training objective of a pretrained base recommender model $\mathcal{M}^s$. We compute the base model's score difference between positive item and negative item, and scale it by a hyperparameter β to construct a data-dependent reference margin:

$$m^s_{(u,i^+,i^-)} = \beta \cdot (\hat{y}^s_{ui^+} - \hat{y}^s_{ui^-}), \tag{9}$$

where $\hat{y}^s_{ui^+}$ and $\hat{y}^s_{ui^-}$ are the predicted scores of the positive item and negative item, obtained from a frozen recommender model. This model is trained before Stage I with randomly initialized parameters and kept frozen during Stage I. The main loss for soft margin variant is as follows:

$$\mathcal{L}^1_{main} = \sum_{(u,i^+,i^-)} \max(0,\ m^s_{(u,i^+,i^-)} - \phi^1(\hat{y}^1_{ui})). \tag{10}$$

3.4 STAGE II: TRAINING WITH MARGIN INITIALIZATION

In Stage I, the extracted semantics are encoded in base model parameters and non-trainable projected semantic embeddings. Stage II transfers these semantics to standard recommender training by using the projected semantic embeddings to initialize trainable ID embeddings, while also initializing a new base model with model parameters learned in Stage I. The base model architecture is kept the same across both stages to ensure compatibility during parameter initialization and to prevent information loss caused by inconsistencies. In this way, the extracted semantics are re-parameterized as learnable parameters and further optimized by collaborative signals, allowing the recommender model to effectively utilize them as prior knowledge for faster convergence and better performance.

We formulate the procedure of the second stage as follows:

$$\mathbf{e}^{id}_u \xleftarrow{\text{init}} \mathbf{z}_u, \quad \mathbf{e}^{id}_i \xleftarrow{\text{init}} \mathbf{z}_i, \quad \mathcal{M}^2 \xleftarrow{\text{init}} \mathcal{M}^1, \tag{11}$$

$$\mathbf{h}^2_u, \mathbf{h}^2_i = \mathcal{M}^2\left(\mathbf{e}^{id}_u, \mathbf{e}^{id}_i, R\right), \quad \hat{y}^2_{ui} = {\mathbf{h}^2_u}^\top \mathbf{h}^2_i, \tag{12}$$

where $\xleftarrow{\text{init}}$ denotes the parameter initialization operation; $\mathbf{e}^{id}_u$ and $\mathbf{e}^{id}_i$ are trainable ID embeddings for user u and item i; the $\mathcal{M}^2$, $\mathbf{h}^2_u$, $\mathbf{h}^2_i$, and $\hat{y}^2_{ui}$ denote the base model, representations and predicted score in the second stage.

As Stage II does not apply margin constraint, we adopt a more practical training objective defined as:

$$\phi^2(\hat{y}^2_{ui}) = \phi^2(\hat{y}^2_{ui^+}, \hat{y}^2_{ui^-}) = \ln \sigma\left(\hat{y}^2_{ui^+} - \hat{y}^2_{ui^-}\right), \tag{13}$$

$$\mathcal{L}^2_{main} = \sum_{(u,i^+,i^-)} -\ln \sigma\left(\hat{y}^2_{ui^+} - \hat{y}^2_{ui^-}\right), \tag{14}$$

where $\sigma(\cdot)$ denotes the sigmoid function; $\phi^2(\cdot)$ and $\mathcal{L}^2_{main}$ are the training objective and the main loss in the second stage. The overall loss in the second stage, denoted as $\mathcal{L}^2$, is formulated as:

$$\mathcal{L}^2 = \mathcal{L}^2_{main} + \mathcal{L}^2_{aux}, \tag{15}$$

where $\mathcal{L}^2_{aux}$ is the auxiliary loss, which may include regularization or other terms such as contrastive learning, depending on the loss used in the original base model.

The overall loss in Stage I, denoted as $\mathcal{L}^1$, follows a similar form:

$$\mathcal{L}^1 = \mathcal{L}^1_{main} + \mathcal{L}^1_{aux}, \tag{16}$$

where $\mathcal{L}^1_{aux}$ is the auxiliary loss in Stage I. $\mathcal{L}^1_{aux}$ can be simplified by removing components unnecessary for semantic extraction, such as those based on pseudo-noisy signals, thereby reducing training cost, which is beneficial when the removed components are computationally expensive.

4 Theoretical Analysis

In this section, we theoretically demonstrate that MarRec embodies the information bottleneck principle. The statements are presented as theorems, with proofs in Appendix A.

Formally, we follow the standard convention and denote the set of all realizations of a random variable (e.g., embeddings or representations) by an uppercase letter (e.g., $\mathbf{X}$).

Definition 4.1. *The information bottleneck principle aims to learn a compressed representation $\mathbf{Z}$ of the input $\mathbf{X}$ that preserves as much relevant information about the target $\mathbf{Y}$ as possible. This objective can be formalized as minimizing $I(\mathbf{X};\mathbf{Z}) - \gamma I(\mathbf{Z};\mathbf{Y})$, where $I(\cdot;\cdot)$ denotes the mutual information between two variables; $I(\mathbf{X};\mathbf{Z})$ promotes compression; $I(\mathbf{Z};\mathbf{Y})$ preserves task-relevant information; and $\gamma > 0$ controls the trade-off between these two terms.*

In MarRec, Stage I corresponds to the compression term $I(\mathbf{X};\mathbf{Z})$, and Stage II corresponds to the preservation term $I(\mathbf{Z};\mathbf{Y})$, as defined in the information bottleneck principle (Definition 4.1). We next demonstrate the roles of Stage I and Stage II through Theorem 4.2 and Theorem 4.3.

Theorem 4.2. *The margin constraint in Stage I ensures the mutual information $I(\mathbf{E}^s;\mathbf{H}^1)$ is upper bounded by $m + H(\mathbf{H}^1|\mathbf{Y}) - H(\mathbf{H}^1|\mathbf{E}^s)$, thereby achieving compression between the input $\mathbf{E}^s$ and the compressed representation $\mathbf{H}^1$. Furthermore, $H(\mathbf{H}^1|\mathbf{Y})$ reduces task-irrelevant redundancy, and $H(\mathbf{H}^1|\mathbf{E}^s)$ retains essential information from semantic input. (See Appendix A.1 for proof.)*

Theorem 4.3. *The initialization from margin-constrained parameters effectively transfers condensed semantic information into Stage II. As training proceeds without the margin constraint, the model further increases $I(\mathbf{H}^2;\mathbf{Y})$, thereby preserving semantic information critical for the recommendation task. (See Appendix A.2 for proof.)*

Based on Theorem 4.2 and Theorem 4.3, we derive the overall objective of MarRec as Theorem 4.4, which unifies the compression and preservation terms in line with the information bottleneck objective.

Theorem 4.4. *The overall objective of MarRec is $\min\left(I(\mathbf{E}^s;\mathbf{H}^1), m'\right) - I(\mathbf{H}^2;\mathbf{Y})$, where $m' = m + H(\mathbf{H}^1|\mathbf{Y}) - H(\mathbf{H}^1|\mathbf{E}^s)$ is the upper bound of $I(\mathbf{E}^s;\mathbf{H}^1)$, and the final state of $\mathbf{H}^1$ is equal to the initial state of $\mathbf{H}^2$. (See Appendix A.3 for proof.)*

5 Experiment

In this section, we conduct extensive experiments on 4 widely used public datasets and 5 representative base recommender models to validate the effectiveness, efficiency, and underlying mechanism. The details of the datasets, base recommender models, baselines, and settings are provided in Appendix B.1, Appendix B.2, Appendix B.3, and Appendix B.4, respectively. For convenience, we denote the base recommender model as **Base**, and the fixed, adaptive, soft margin variants as **MarRec**, **Adaptive**, **Soft**. We organize our experiments around the following research questions: **Q1**: How effective and efficient is MarRec vs. baselines? **Q2**: How do key components contribute to MarRec? **Q3**: How does MarRec perform under data sparsity? **Q4**: How does MarRec perform under the long-tail effect? **Q5**: How does MarRec handle cold-start situations? **Q6**: What insights arise from optimal margin, α, and β? **Q7**: What is the impact of Stage I training on Stage II?

Table 1: Overall results on Beauty, Sports, and Steam. Time and Mem denote training time (s) and peak GPU memory (MB). Best results are in bold and second-best underlined. Percentages show improvements over the Base method. Yelp and @K results are reported in Appendix D.1.

Dataset		Beauty				Sports				Steam			
Backbone	Methods	Recall@20	NDCG@20	Time	Mem	Recall@20	NDCG@20	Time	Mem	Recall@20	NDCG@20	Time	Mem
NGCF	Base	0.0896	0.0406	2.05	442	0.0755	0.0336	25.12	568	0.1259	0.0755	34.75	508
	KAR	0.1357(51.5%)	0.0599(47.5%)	25.71	1044	0.0763 (1.1%)	0.0322 (-4.2%)	89.63	2492	0.1261 (0.2%)	0.0749 (-0.8%)	105.60	1576
	RLMRec-Con	0.1552(73.2%)	0.0739(82.0%)	73.65	1584	0.0881(16.7%)	0.0394(17.3%)	137.81	3528	0.1340 (6.4%)	0.0810 (6.1%)	96.76	2436
	RLMRec-Gen	0.1606(79.2%)	0.0712(75.4%)	28.89	1254	0.0968(28.2%)	0.0425(26.5%)	91.32	2822	0.1344 (6.8%)	0.0809 (7.2%)	84.02	1470
	PAD	0.1438(60.5%)	0.0642(58.1%)	22.81	3876	0.0774 (2.5%)	0.0335 (-0.3%)	**26.13**	6894	0.1212 (-3.7%)	0.0727 (-3.7%)	**28.60**	2966
	MarRec	0.1760(96.4%)	0.0800(97.0%)	30.54	**838**	0.1063(40.8%)	**0.0470(39.9%)**	46.62	**1388**	**0.1433(13.8%)**	0.0860(13.9%)	115.08	874
	Adaptive	**0.1767(97.2%)**	**0.0812(100.0%)**	**20.33**	1052	**0.1071(41.9%)**	0.0467(39.0%)	98.77	1628	0.1427(13.3%)	**0.0862(14.2%)**	123.73	994
	Soft	0.1746(94.9%)	0.0804(98.0%)	37.07	892	0.1055(39.7%)	0.0466(38.7%)	88.71	1410	0.1426(13.3%)	0.0855(13.3%)	164.22	**858**
GCCF	Base	0.1060	0.0471	7.19	438	0.0511	0.0214	4.99	488	0.1160	0.0697	19.64	474
	KAR	0.1324(24.9%)	0.0563(19.5%)	**7.71**	1032	0.0678(32.7%)	0.0282(31.8%)	**6.92**	2174	0.1205 (3.9%)	0.0725 (4.0%)	**19.53**	1270
	RLMRec-Con	0.1126 (6.2%)	0.0486 (3.2%)	22.57	1580	0.0579(13.3%)	0.0248(15.9%)	59.20	3480	0.1200 (3.5%)	0.0716 (2.7%)	63.56	2410
	RLMRec-Gen	0.1281(20.9%)	0.0538(14.2%)	21.89	1258	0.0638(24.9%)	0.0263(22.9%)	19.27	2770	0.1198 (3.3%)	0.0714 (2.4%)	30.84	1392
	PAD	0.1391(31.2%)	0.0611(29.7%)	35.56	3856	0.0713(39.5%)	0.0292(36.5%)	94.71	6843	0.1149 (-1.0%)	0.0684 (-1.9%)	84.41	2923
	MarRec	**0.1532(44.5%)**	**0.0667(41.6%)**	25.84	880	0.0865(69.3%)	0.0355(65.9%)	28.12	**1308**	0.1291(11.3%)	0.0778(11.6%)	43.66	**798**
	Adaptive	0.1500(41.5%)	0.0663(40.8%)	12.51	1052	0.0806(57.7%)	0.0334(56.1%)	9.24	1456	**0.1319(13.7%)**	**0.0785(12.6%)**	51.68	1012
	Soft	0.1480(39.6%)	0.0641(36.1%)	22.35	**852**	**0.0870(70.3%)**	**0.0361(68.7%)**	52.54	1362	0.1283(10.6%)	0.0772(10.8%)	71.49	814
LightGCN	Base	0.1551	0.0711	20.96	366	0.0845	0.0382	24.55	432	0.1357	0.0827	78.70	442
	KAR	0.1692 (9.1%)	0.0771 (8.4%)	35.07	956	0.0947(12.1%)	0.0417 (9.2%)	76.40	2174	0.1360 (0.2%)	0.0830 (0.4%)	**103.66**	1298
	RLMRec-Con	0.1814(17.0%)	0.0835(17.4%)	63.8	1622	0.1008(19.3%)	0.0449(17.5%)	108.43	3098	0.1400 (3.2%)	0.0856 (3.5%)	119.84	2000
	RLMRec-Gen	0.1569 (1.2%)	0.0713 (0.3%)	**29.92**	1222	0.0961(13.7%)	0.0432(13.1%)	106.52	2678	0.1427 (5.2%)	0.0867 (4.8%)	153.66	1396
	PAD	0.1772(14.3%)	0.0805(13.2%)	51.53	3839	0.0939(11.1%)	0.0412 (7.9%)	58.46	6825	0.1385 (2.1%)	0.0840 (1.6%)	118.91	2907
	MarRec	0.1829(17.9%)	0.0858(20.7%)	44.07	**794**	**0.1142(35.2%)**	0.0496(29.8%)	**56.98**	**1300**	0.1463 (7.8%)	0.0894(8.1%)	107.56	**802**
	Adaptive	**0.1870(20.6%)**	**0.0864(21.5%)**	64.38	1016	0.1120(32.5%)	0.0495(29.6%)	95.14	1564	**0.1471(8.4%)**	**0.0897(8.5%)**	151.65	1040
	Soft	0.1829(17.9%)	0.0852(19.8%)	62.83	806	0.1135(34.3%)	**0.0500(30.9%)**	119.61	1428	0.1466(8.0%)	0.0890 (7.6%)	290.14	896
SGL	Base	0.1563	0.0715	56.86	818	0.0851	0.0382	78.18	1376	0.1348	0.0825	248.21	1204
	KAR	0.1635 (4.6%)	0.0745 (4.2%)	**56.33**	1362	0.0931 (9.4%)	0.0417 (9.2%)	136.77	2980	0.1357 (0.7%)	0.0826 (0.1%)	248.50	1980
	RLMRec-Con	0.1790(14.5%)	0.0831(16.2%)	101.06	1850	0.0998(17.3%)	0.0451(18.1%)	168.23	4034	0.1398 (3.7%)	0.0855 (3.6%)	224.78	2782
	RLMRec-Gen	0.1630 (4.3%)	0.0751 (5.0%)	77.50	1646	0.0958(12.6%)	0.0435(13.9%)	138.55	3638	0.1425 (5.7%)	0.0863 (4.6%)	325.30	2010
	PAD	0.1775(13.6%)	0.0800(11.9%)	102.67	4116	0.0940(10.5%)	0.0412 (7.9%)	124.16	7400	0.1386 (2.8%)	0.0840 (1.8%)	299.69	3391
	MarRec	0.1817(16.3%)	**0.0850(18.9%)**	59.95	852	**0.1142(34.2%)**	0.0502(31.4%)	105.87	**1428**	0.1464(8.6%)	0.0894(8.4%)	250.69	**1204**
	Adaptive	**0.1838(17.6%)**	0.0846(18.3%)	71.06	1038	0.1111(30.6%)	0.0494(29.3%)	**103.35**	1486	0.1457 (8.1%)	0.0891 (8.0%)	**204.08**	**1204**
	Soft	0.1826(16.8%)	**0.0850(18.9%)**	122.52	**820**	0.1141(34.1%)	**0.0503(31.7%)**	254.50	1450	**0.1469(9.0%)**	**0.0896(8.6%)**	498.64	**1204**
SimGCL	Base	0.1572	0.0717	125.47	830	0.0855	0.0384	231.01	1402	0.1354	0.0825	583.59	1206
	KAR	0.1662 (5.7%)	0.0772 (7.7%)	186.86	1414	0.0928 (8.5%)	0.0418 (8.9%)	420.05	3144	0.1362 (0.6%)	0.0826 (0.1%)	803.63	1996
	RLMRec-Con	0.1808(15.0%)	0.0834(16.3%)	144.40	1856	0.1005(17.5%)	0.0451(17.5%)	229.11	4068	0.1399 (3.3%)	0.0857 (3.9%)	362.15	2784
	RLMRec-Gen	0.1606 (2.2%)	0.0744 (3.8%)	231.09	1658	0.0955(11.7%)	0.0430(12.0%)	390.25	3672	0.1416 (4.6%)	0.0860 (4.2%)	734.45	2014
	PAD	0.1760(12.0%)	0.0800(11.6%)	111.79	4131	0.0941(10.1%)	0.0409 (6.5%)	**114.25**	7432	0.1367 (1.0%)	0.0831 (0.7%)	**310.45**	3403
	MarRec	0.1834(16.7%)	0.0849(18.4%)	**86.67**	**830**	**0.1159(35.6%)**	**0.0510(32.8%)**	266.20	**1428**	0.1462(8.0%)	0.0893(8.2%)	520.29	**1206**
	Adaptive	**0.1840(17.1%)**	**0.0868(21.1%)**	99.07	984	0.1122(31.2%)	0.0494(28.7%)	225.57	1490	**0.1483(9.5%)**	**0.0905(9.7%)**	424.09	**1206**
	Soft	0.1825(16.1%)	0.0846(18.0%)	227.80	872	0.1147(34.2%)	0.0505(31.5%)	521.06	**1428**	0.1458 (7.7%)	0.0888 (7.6%)	999.16	**1206**

5.1 Overall Performance Analysis (Q1)

To evaluate the effectiveness and efficiency of MarRec, we compare it with baselines for LLMs-enhanced recommendation in terms of recommendation performance, training time, and peak GPU memory usage. The baselines include the concatenation method KAR (Xi et al., 2024), the contrastive learning alignment method RLMRec (Ren et al., 2024), and the distribution alignment method PAD (Wang et al., 2025b). Since SGL and SimGCL construct contrastive learning loss through noise, we remove the contrastive loss from Stage I. Training time and peak GPU memory are measured over the entire training process. For Soft, this process also includes base model pretraining.

As shown in Table 1 and Table 8, we summarize the following findings:

- MarRec, Adaptive, and Soft consistently outperform the baselines on Recall@20 and NDCG@20, demonstrating the superiority of our margin constraint in effectively extracting useful semantics. The improvements are more pronounced on Beauty and Sports, which have higher-dimensional semantic embeddings, validating that our framework benefits from richer semantic inputs.
- In terms of training time, MarRec or Adaptive achieves the best or second-best result in most cases. Soft requires additional pretraining but remains competitive on lightweight backbones. Moreover, our optimizations for SGL and SimGCL significantly reduce training time, even outperforming the Base model. In terms of memory usage, our variants are substantially more efficient than baselines in most cases, with peak memory generally close to that of Base model. These results demonstrate the strong efficiency of our framework in both time and space complexity.
- In the majority of cases, Adaptive and Soft outperform MarRec in performance or training time, while keeping similar memory usage. This indicates that instance margins can further extract useful semantics in a more fine-grained manner while preserving the lightweight nature of our framework.
- Although PAD achieves competitive performance, it requires substantially higher memory and shows unstable performance and time. In some cases, KAR and PAD even exhibit negative gains over the base model, indicating that concatenation or alignment may distort useful semantics.

5.2 Ablation Study (Q2)

To investigate the contribution of key components in our framework, we conduct ablation studies on the fixed margin variant MarRec. The ablation settings are described as follows: (1) **w/o Stage II** removes Stage II and evaluates Stage I. (2) **w/o Freeze** keeps semantic embeddings trainable during Stage I. (3) **w/o Margin** removes margin constraint and replaces the main loss in Stage I

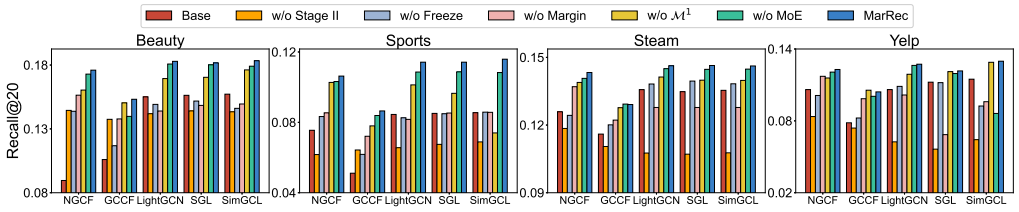

Figure 2: Recall@20 results for Ablation Study (NDCG@20 results are shown in Appendix D.2).

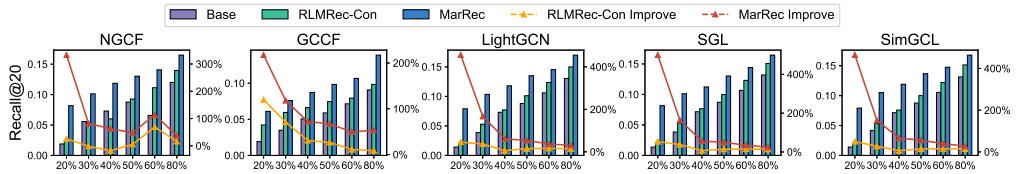

Figure 3: Results for Data Sparsity Analysis on Beauty dataset (full dataset results are provided in Appendix D.3). The x-axis denotes sparsity ratio and the right y-axis shows improvement over Base.

with the same loss used in Stage II. (4) **w/o** $\mathcal{M}^1$ removes $\mathcal{M}^1$ in Stage I and randomly initializes the parameters of $\mathcal{M}^2$ in Stage II. (5) **w/o MoE** replaces **MoE** with a single linear network.

As shown in Figure 2, it can be observed that these key components all contribute positively to the effectiveness of our framework. Among these settings, the three with the worst performance are w/o Stage II, w/o Freeze, and w/o Margin. The large gap between w/o Stage II and MarRec indicates that Stage I indeed extracts semantic information that benefits Stage II. The poor performance of w/o Freeze suggests that not freezing the semantic embeddings may lead to the loss of essential semantics and information distortion. The results of w/o Margin show the crucial role of the margin constraint in extracting preference-relevant semantic information. The w/o $\mathcal{M}^1$ and w/o MoE variants result in relatively minor performance degradation, but their removal still negatively affects the final results, indicating that both **MoE** and $\mathcal{M}^1$ contribute to extracting useful semantic information.

5.3 Data Sparsity Analysis (Q3)

To examine whether our framework can effectively alleviate data sparsity, we randomly remove a portion of interactions from the training set while keeping the resulting interaction matrix connected. To avoid leakage, we replace the user semantic embedding with the average semantic embedding of the remaining interacted items. Considering both performance and stability, we regard RLMRec-Con as the most competitive baseline, and compare our framework with it in the following analysis.

As shown in Figure 3 and Appendix D.3, the base model consistently degrades as the retained ratio decreases, and the performance gaps between different ratios are substantial, indicating that data sparsity poses a significant challenge. In contrast, MarRec achieves the best performance in most cases. Moreover, the relative improvements of MarRec over the base model become larger as the ratio decreases, with the most pronounced gain observed in the sparsest setting. In contrast, RLMRec-Con yields limited improvements and even underperforms the base model in some settings. This observation supports our claim in Section 1 that aligning semantic representations may suffer from semantic information distortion, whereas our margin constraint provides a more robust mechanism for extracting preference-relevant semantics under data sparsity scenarios.

5.4 Long-Tail Effect Analysis (Q4)

To verify whether our framework can effectively mitigate the long-tail effect, we divide users into five equally sized groups (G1 to G5) based on their interaction counts in the training set, sorted in ascending order. Users in G1 have the fewest interactions, while those in G5 have the most.

As illustrated in Figure 4 and Appendix D.4, the base model exhibits a clear long-tail effect on Beauty and Sports datasets in Recall and NDCG, where users with fewer interactions consistently perform worse, indicating the presence of long-tail challenges. On Steam and Yelp datasets, performance initially improves with more interactions, but then declines for the most active users. The decline

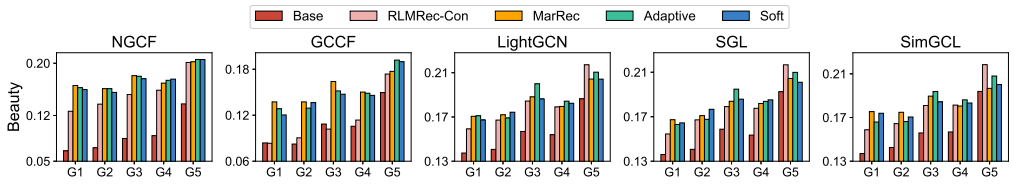

Figure 4: Recall@20 results for Long-Tail Effect Analysis on Beauty dataset (full dataset results are provided in Appendix D.4). G1-G5 denote equally sized user groups with increasing interactions.

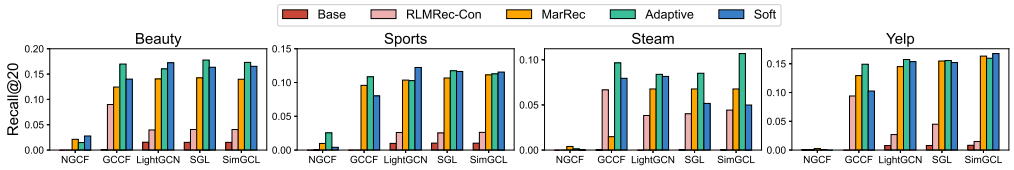

Figure 5: Recall@20 results for Cold-Start Analysis (NDCG@20 results are shown in Appendix D.5).

may be related to overfitting, as the model tends to recommend popular items, which reduces personalization. Influenced by base model, RLMRec-Con and our variants show similar patterns.

In most cases, our variants outperform the base model and RLMRec-Con. Additionally, the performance of our variants across G1 to G4 remains relatively consistent, with their performance in some cases approaching that of users in G5. In contrast, the base models show clear performance drops in G1 to G4, highlighting the difficulty of long-tail recommendation. These results demonstrate that our framework alleviates the long-tail effect by improving performance for users with sparse interactions, while maintaining competitive performance across the full range of activity levels.

5.5 Cold-Start Analysis (Q5)

To simulate the cold-start setting, we split items into training/validation/test sets with a ratio of 8:1:1, such that the items in the validation and test sets never appear in the training set. As shown in Figure 5 and Appendix D.5, the base models achieve zero results in most cases, indicating that methods relying solely on collaborative signals struggle to recommend cold-start items. Notably, RLMRec-Con also yields zero results in some cases and performs poorly overall, which further supports our observation that semantic alignment suffers from information distortion under the cold-start setting. In contrast, our variants obtain substantially higher performance, and consistently outperform RLMRec-Con. These results demonstrate that MarRec effectively mitigates the cold-start issue by extracting and utilizing preference-relevant semantic information, enabling the model to generalize to unseen items.

5.6 Margin Analysis (Q6)

To further investigate the role of margin, we compare the margin, α, and β corresponding to the best performance. We report the average margin m_a from the final epoch for Adaptive and m_s for Soft. It can be observed from Table 2 that the margin values of these variants are close in some cases. The similarity between m, m_a and m_s indicates that the learned or derived margins align well with the empirically best fixed value. Meanwhile, the instance-level margins provide fine-grained adjustments during training, which may help better capture the semantic variation among instances and further enhance recommendation performance. Moreover, the optimal α and β in Adaptive and Soft variants are similar across datasets under the same base model, which helps reduce hyperparameter tuning effort and demonstrates the robustness of our automated margin modeling strategies.

5.7 Stage Impact Analysis (Q7)

To better understand the relationship between Stage I and Stage II, we use the parameters obtained after every 5 epochs in Stage I to initialize Stage II and evaluate the recommendation performance for MarRec. As illustrated in Figure 6 and Appendix D.6, the performance of Stage II improves as Stage I progresses, suggesting that better-learned semantic information in Stage I leads to more effective recommendation in Stage II. Moreover, Stage II achieves competitive performance within the early

Table 2: The optimal fixed margin, adaptive margin, soft margin, α, and β.

Backbone	NGCF				GCCF				LightGCN				SimGCL				SGL			
Dataset	Beauty	Sports	Steam	Yelp	Beauty	Sports	Steam	Yelp	Beauty	Sports	Steam	Yelp	Beauty	Sports	Steam	Yelp	Beauty	Sports	Steam	Yelp
m	0.8	0.24	0.75	0.68	0.46	0.2	0.1	0.48	0.08	0.06	0.02	0.15	0.1	0.06	0.02	0.0006	0.08	0.08	0.02	0.02
m^a	0.26	0.68	0.64	0.97	0.98	0.58	0.36	0.83	0.10	0.36	0.16	0.02	0.49	0.004	0.16	0.14	0.57	0.23	0.18	0.02
m^s	0.67	0.68	0.65	1.16	0.80	0.26	0.85	0.90	0.12	0.10	0.22	0.14	0.24	0.25	0.22	0.12	0.12	0.25	0.17	0.005
α	0.1	0.25	0.45	0.35	0.2	0.2	0.28	0.18	0.1	0.1	0.26	0.15	0.14	0.1	0.25	0.15	0.15	0.12	0.28	0.16
β	0.46	0.18	0.2	0.2	0.1	0.05	0.2	0.1	0.02	0.02	0.05	0.02	0.04	0.05	0.05	0.02	0.02	0.05	0.04	0.001

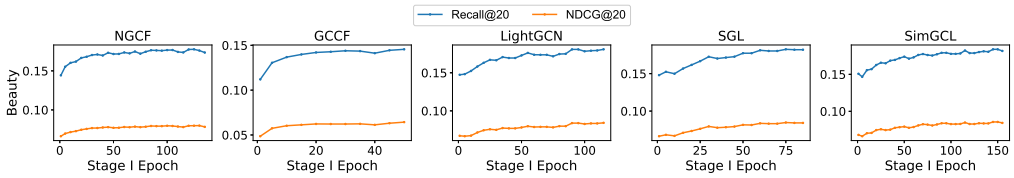

Figure 6: Results for Stage Impact Analysis (full results are shown in Appendix D.6).

epochs of Stage I, indicating that our framework has potential for further time efficiency optimization by balancing training time and recommendation performance in Stage I. These observations further validate that our two-stage design does not introduce additional optimization burden.

6 Related Work

6.1 LLMs-based Recommender System

LLMs-based recommender systems can be broadly categorized into two paradigms: LLMs as Recommender Systems and LLMs enhanced Recommender Systems. The former focuses on enhancing LLMs' generative capabilities to recommend directly. Initial efforts in this paradigm focus on prompt engineering (Hou et al., 2024), pretraining (Geng et al., 2022), or fine-tuning (Zhu et al., 2024) LLMs for recommendation. Recent studies (Kim et al., 2024; Zhang et al., 2025b) incorporate collaborative representations from recommender models into LLMs to complement content semantics with preference signals. The latter aims to enhance recommender models by incorporating semantic knowledge derived from LLMs. KAR (Xi et al., 2024) uses LLMs to infer preferences and factual knowledge, then concatenates them with recommender representations. RLMRec (Ren et al., 2024) employs LLMs to obtain semantic profiles and aligns them with recommender representations via contrastive and generative objectives. PAD (Wang et al., 2025b) pre-trains collaborative and LLMs-derived semantic embeddings, and reduces their distribution discrepancy via maximum mean discrepancy.

6.2 Margin and Information Bottleneck

The margin constraint is inspired by margin-based loss functions, which are commonly used in distance-based metric learning. For example, metric learning methods (Hsieh et al., 2017; Li et al., 2020) apply margin ranking loss to separate positive and negative samples by enforcing a predefined distance gap. In contrast, our margin constraint framework implicitly enforces the information bottleneck. Previous works implement the information bottleneck through mutual information estimation (Fan & Li, 2022; Xie et al., 2024) or variational approximations (Wei et al., 2022; Yan et al., 2024), which introduce additional complexity and cost. MarRec provides a more intuitive and lightweight alternative, effectively preserving preference-relevant semantics.

7 Conclusion

In this paper, we propose a simple-yet-effective model-agnostic framework MarRec for LLMs-enhanced recommender systems to address the incompatibility between semantic and collaborative representations. Extensive experiments demonstrate that MarRec consistently outperforms baselines, while maintaining high efficiency and exhibiting strong robustness in data sparsity, long-tail, and cold-start scenarios. Our work provides a novel perspective on integrating LLMs-generated knowledge with recommender systems and a preliminary exploration in an industrial scenario, detailed in Appendix C.1, further supports its practical value. Nevertheless, a limitation is that MarRec has not been explored on broader recommendation tasks, which we leave for future work.

AI USE STATEMENT

Generative AI tools are used to assist with language polishing and formatting. The semantic embeddings used in our experiments are obtained from previously released LLMs-enhanced recommendation datasets, as described in Appendix B.1. All AI-assisted content and experimental results are reviewed and verified by the authors, who take full responsibility for the paper.

ETHICS STATEMENT

This work uses public recommendation datasets and does not involve new human-subject data collection. We acknowledge and adhere to the ICLR Code of Ethics. No high-risk generative model or sensitive dataset is released as part of this work.

REPRODUCIBILITY STATEMENT

See `https://anonymous.4open.science/r/MarRec-anonymous/` for details.

REFERENCES

Keqin Bao, Jizhi Zhang, Yang Zhang, Wenjie Wang, Fuli Feng, and Xiangnan He. Tallrec: An effective and efficient tuning framework to align large language model with recommendation. In *Proceedings of the 17th ACM conference on recommender systems*, pp. 1007–1014, 2023.

Ivan Butakov, Aleksandr Tolmachev, Sofia Malanchuk, Anna Neopryatnaya, Alexey Frolov, and Kirill Andreev. Information bottleneck analysis of deep neural networks via lossy compression. In *International Conference on Learning Representations*, volume 2024, pp. 40868–40890, 2024.

Lei Chen, Le Wu, Richang Hong, Kun Zhang, and Meng Wang. Revisiting graph based collaborative filtering: A linear residual graph convolutional network approach. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 27–34, 2020.

Wentao Cheng, Zhida Qin, Zexue Wu, Pengzhan Zhou, and Tianyu Huang. Large language models enhanced hyperbolic space recommender systems. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1944–1953, 2025.

Jingtao Ding, Yuhan Quan, Quanming Yao, Yong Li, and Depeng Jin. Simplify and robustify negative sampling for implicit collaborative filtering. *Advances in Neural Information Processing Systems*, 33:1094–1105, 2020.

Yann Dubois, Douwe Kiela, David J Schwab, and Ramakrishna Vedantam. Learning optimal representations with the decodable information bottleneck. *Advances in Neural Information Processing Systems*, 33:18674–18690, 2020.

Jiameng Fan and Wenchao Li. Dribo: Robust deep reinforcement learning via multi-view information bottleneck. In *International conference on machine learning*, pp. 6074–6102. PMLR, 2022.

Shijie Geng, Shuchang Liu, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. Recommendation as language processing (rlp): A unified pretrain, personalized prompt & predict paradigm (p5). In *Proceedings of the 16th ACM conference on recommender systems*, pp. 299–315, 2022.

Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng 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*, pp. 639–648, 2020.

Yupeng Hou, Junjie Zhang, Zihan Lin, Hongyu Lu, Ruobing Xie, Julian McAuley, and Wayne Xin Zhao. Large language models are zero-shot rankers for recommender systems. In *European Conference on Information Retrieval*, pp. 364–381. Springer, 2024.

Cheng-Kang Hsieh, Longqi Yang, Yin Cui, Tsung-Yi Lin, Serge Belongie, and Deborah Estrin. Collaborative metric learning. In *Proceedings of the 26th international conference on world wide web*, pp. 193–201, 2017.

Jian Jia, Yipei Wang, Yan Li, Honggang Chen, Xuehan Bai, Zhaocheng Liu, Jian Liang, Quan Chen, Han Li, Peng Jiang, et al. Learn: Knowledge adaptation from large language model to recommendation for practical industrial application. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 11861–11869, 2025.

Clark M Ju, William Shiao, Zhichun Guo, Yanfang Ye, Yozen Liu, Neil Shah, and Tong Zhao. How does message passing improve collaborative filtering? *Advances in neural information processing systems*, 37:8760–8784, 2024.

Sein Kim, Hongseok Kang, Seungyoon Choi, Donghyun Kim, Minchul Yang, and Chanyoung Park. Large language models meet collaborative filtering: An efficient all-round llm-based recommender system. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1395–1406, 2024.

Xiaoyu Kong, Jiancan Wu, An Zhang, Leheng Sheng, Hui Lin, Xiang Wang, and Xiangnan He. Customizing language models with instance-wise lora for sequential recommendation. *Advances in Neural Information Processing Systems*, 37:113072–113095, 2024.

Md Kowsher, Nusrat Prottasha, Shiyun Xu, Shetu Mohanto, Niloofar Yousefi, Ozlem Garibay, and Chen Chen. Flownib: An information bottleneck analysis of bidirectional vs. unidirectional language models. In *International Conference on Learning Representations*, volume 2026, pp. 11123–11160, 2026.

Dongsheng Li, Chao Chen, Wei Liu, Tun Lu, Ning Gu, and Stephen Chu. Mixture-rank matrix approximation for collaborative filtering. *Advances in Neural Information Processing Systems*, 30, 2017.

Mingming Li, Shuai Zhang, Fuqing Zhu, Wanhui Qian, Liangjun Zang, Jizhong Han, and Songlin Hu. Symmetric metric learning with adaptive margin for recommendation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 4634–4641, 2020.

Qidong Liu, Xian Wu, Yejing Wang, Zijian Zhang, Feng Tian, Yefeng Zheng, and Xiangyu Zhao. Llm-esr: Large language models enhancement for long-tailed sequential recommendation. *Advances in Neural Information Processing Systems*, 37:26701–26727, 2024.

Qidong Liu, Xian Wu, Wanyu Wang, Yejing Wang, Yuanshao Zhu, Xiangyu Zhao, Feng Tian, and Yefeng Zheng. Llmemb: Large language model can be a good embedding generator for sequential recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 12183–12191, 2025.

Shikun Liu, Andrew Davison, and Edward Johns. Self-supervised generalisation with meta auxiliary learning. *Advances in Neural Information Processing Systems*, 32, 2019.

Zhuo Ouyang, Kaiwen Hu, Qi Zhang, Yifei Wang, and Yisen Wang. Projection head is secretly an information bottleneck. In *International Conference on Learning Representations*, volume 2025, pp. 76941–76960, 2025.

Nikhil Rao, Hsiang-Fu Yu, Pradeep K Ravikumar, and Inderjit S Dhillon. Collaborative filtering with graph information: Consistency and scalable methods. *Advances in neural information processing systems*, 28, 2015.

Xubin Ren, Wei Wei, Lianghao Xia, Lixin Su, Suqi Cheng, Junfeng Wang, Dawei Yin, and Chao Huang. Representation learning with large language models for recommendation. In *Proceedings of the ACM on Web Conference 2024*, pp. 3464–3475, 2024.

Reza Shirkavand, Xiaokai Wei, Chen Wang, Zheng Hui, Heng Huang, and Michelle Gong. Catalog-native llm: Speaking item-id dialect with less entanglement for recommendation. In *International Conference on Learning Representations*, volume 2026, pp. 12528–12552, 2026.

Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, and Deyu Meng. Meta-weight-net: Learning an explicit mapping for sample weighting. *Advances in neural information processing systems*, 32, 2019.

Chen Wang, Liangwei Yang, Zhiwei Liu, Xiaolong Liu, Mingdai Yang, Yueqing Liang, and Philip S Yu. Collaborative alignment for recommendation. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pp. 2315–2325, 2024.

Shuyao Wang, Zhi Zheng, Yongduo Sui, and Hui Xiong. Unleashing the power of large language model for denoising recommendation. In *Proceedings of the ACM on Web Conference 2025*, pp. 252–263, 2025a.

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

Yuhao Wang, Junwei Pan, Pengyue Jia, Wanyu Wang, Maolin Wang, Zhixiang Feng, Xiaotian Li, Jie Jiang, and Xiangyu Zhao. Pre-train, align, and disentangle: Empowering sequential recommendation with large language models. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1455–1465, 2025b.

Chunyu Wei, Jian Liang, Di Liu, and Fei Wang. Contrastive graph structure learning via information bottleneck for recommendation. *Advances in neural information processing systems*, 35:20407–20420, 2022.

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

Junda Wu, Cheng-Chun Chang, Tong Yu, Zhankui He, Jianing Wang, Yupeng Hou, and Julian McAuley. Coral: Collaborative retrieval-augmented large language models improve long-tail recommendation. In *Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining*, pp. 3391–3401, 2024.

Tailin Wu, Hongyu Ren, Pan Li, and Jure Leskovec. Graph information bottleneck. *Advances in Neural Information Processing Systems*, 33:20437–20448, 2020.

Yunjia Xi, Weiwen Liu, Jianghao Lin, Xiaoling Cai, Hong Zhu, Jieming Zhu, Bo Chen, Ruiming Tang, Weinan Zhang, and Yong Yu. Towards open-world recommendation with knowledge augmentation from large language models. In *Proceedings of the 18th ACM Conference on Recommender Systems*, pp. 12–22, 2024.

Junsong Xie, Yonghui Yang, Zihan Wang, and Le Wu. Learning fair representations for recommendation via information bottleneck principle. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, pp. 2469–2477, 2024.

Jian Xu, Sichun Luo, Xiangyu Chen, Haoming Huang, Hanxu Hou, and Linqi Song. Rallrec: Improving retrieval augmented large language model recommendation with representation learning. In *Companion Proceedings of the ACM on Web Conference 2025*, pp. 1436–1440, 2025.

Hanqi Yan, Lin Gui, Menghan Wang, Kun Zhang, and Yulan He. Explainable recommender with geometric information bottleneck. *IEEE Transactions on Knowledge and Data Engineering*, 36(7): 3036–3046, 2024.

Hyunsik Yoo, Ting-Wei Li, SeongKu Kang, Zhining Liu, Caizhi Charlie Xu, Qilin Qi, and Hanghang Tong. Continual low-rank adapters for llm-based generative recommender systems. In *International Conference on Learning Representations*, volume 2026, pp. 6618–6642, 2026.

Junliang Yu, Hongzhi Yin, Xin Xia, Tong Chen, Lizhen Cui, and Quoc Viet Hung Nguyen. 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*, pp. 1294–1303, 2022.

Shujian Yu, Xi Yu, Sigurd Løkse, Robert Jenssen, and Jose Principe. Cauchy-schwarz divergence information bottleneck for regression. In *International Conference on Learning Representations*, volume 2024, pp. 40025–40068, 2024.

An Zhang, Wenchang Ma, Xiang Wang, and Tat-Seng Chua. Incorporating bias-aware margins into contrastive loss for collaborative filtering. *Advances in Neural Information Processing Systems*, 35:7866–7878, 2022.

Shuai Zhang, Junfeng Fang, Xuqiang Li, Hongxin Xiang, Jun Xia, Ye Wei, Wenjie Du, and Yang Wang. Iterative substructure extraction for molecular relational learning with interactive graph information bottleneck. In *International Conference on Learning Representations*, volume 2025, pp. 63678–63706, 2025a.

Xinyu Zhang, Linmei Hu, Luhao Zhang, Wentao Cheng, Yashen Wang, Ge Shi, Chong Feng, and Liqiang Nie. Bi-tuning with collaborative information for controllable llm-based sequential recommendation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 19340–19351, 2025b.

Yaochen Zhu, Liang Wu, Qi Guo, Liangjie Hong, and Jundong Li. Collaborative large language model for recommender systems. In *Proceedings of the ACM Web Conference 2024*, pp. 3162–3172, 2024.

Zhiyu Zhu, Zhibo Jin, Jiayu Zhang, Nan Yang, Jiahao Huang, Jianlong Zhou, and Fang Chen. Narrowing information bottleneck theory for multimodal image-text representations interpretability. In *International Conference on Learning Representations*, volume 2025, pp. 22906–22923, 2025.

A PROOFS

Definition A.1. *The mutual information $I(A;B)$ measures the amount of information shared between two variables A and B. It can be derived as: $I(A;B) = H(A) - H(A|B) = H(B) - H(B|A)$, where $H(\cdot)$ denotes entropy; $H(A|B)$ is the conditional entropy of A given B; and $H(B|A)$ is the conditional entropy of B given A.*

Proof. Based on the probability distributions, we derive the relationship between mutual information and entropy as follows:

$$\begin{aligned} I(A;B) &= \iint p(a,b) \log \frac{p(a,b)}{p(a)p(b)} \, da \, db \\ &= \iint p(a,b) \log \frac{p(a|b)}{p(a)} \, da \, db \\ &= \iint p(a,b) \left[\log p(a|b) - \log p(a)\right] da \, db \\ &= -\iint p(a,b) \log p(a) \, da \, db + \iint p(a,b) \log p(a|b) \, da \, db \\ &= -\int p(a) \log p(a) \, da + \int p(b) \int p(a|b) \log p(a|b) \, da \, db \\ &= H(A) - H(A|B), \end{aligned} \tag{17}$$

where, $p(a,b)$ is the joint probability distribution of A and B; $p(a)$ and $p(b)$ are the marginal probability distributions; and $p(a|b)$ is the conditional probability distribution of A given B.

Similarly, $I(A;B) = H(B) - H(B|A)$ can be derived using the same steps by interchanging A and B in Equation (17). □

A.1 PROOF OF THEOREM 4.2

We define $\mathbf{E}^s$ as the collection of all semantic embeddings ($\mathbf{e}_u^s$ and $\mathbf{e}_i^s$); $\mathbf{H}^1$ as the collection of all representations ($\mathbf{h}_u^1$ and $\mathbf{h}_i^1$) produced by the first-stage base model $\mathcal{M}^1$; $\mathbf{Y}$ as the collection of all ground truth labels y_{ui} derived from user-item interactions R.

Proof. The margin constraint in Stage I limits the training objective by enforcing a margin threshold. This constraint can be formulated as bounding the mutual information between the compressed representation $\mathbf{H}^1$ and the target labels $\mathbf{Y}$, i.e., $I(\mathbf{H}^1; \mathbf{Y}) \leq m$.

Based on Definition A.1, we derive the upper bound on the mutual information between the semantic input $\mathbf{E}^s$ and the compressed representation $\mathbf{H}^1$ as follows:

$$\begin{aligned} I(\mathbf{E}^s; \mathbf{H}^1) &= H(\mathbf{H}^1) - H(\mathbf{H}^1|\mathbf{E}^s) \\ &= I(\mathbf{H}^1; \mathbf{Y}) + H(\mathbf{H}^1|\mathbf{Y}) - H(\mathbf{H}^1|\mathbf{E}^s) \\ &\leq m + H(\mathbf{H}^1|\mathbf{Y}) - H(\mathbf{H}^1|\mathbf{E}^s), \end{aligned} \tag{18}$$

where $H(\mathbf{H}^1|\mathbf{Y})$ corresponds to the excess information in $\mathbf{H}^1$ that is irrelevant to the label $\mathbf{Y}$; $H(\mathbf{H}^1|\mathbf{E}^s)$ corresponds to the input information from $\mathbf{E}^s$ preserved in the fixed-capacity $\mathbf{H}^1$.

Since the capacity of $\mathbf{H}^1$ is limited in practice, the difference $H(\mathbf{H}^1|\mathbf{Y}) - H(\mathbf{H}^1|\mathbf{E}^s)$ is also finite. Consequently, the mutual information $I(\mathbf{E}^s; \mathbf{H}^1)$ is upper bounded, thereby achieving information compression. Furthermore, $H(\mathbf{H}^1|\mathbf{Y})$ reduces task-irrelevant redundancy, and $H(\mathbf{H}^1|\mathbf{E}^s)$ retains essential information from input because $\mathbf{H}^1$ is optimized for the target $\mathbf{Y}$ under a fixed capacity.

To further interpret this bound, we note that representation distributions are commonly modeled as Gaussian distributions. Accordingly, suppose that $\mathbf{H}^1|\mathbf{Y} \sim \mathcal{N}(\boldsymbol{\mu}_y, \boldsymbol{\Sigma}_y)$ and $\mathbf{H}^1|\mathbf{E}^s \sim \mathcal{N}(\boldsymbol{\mu}_e, \boldsymbol{\Sigma}_e)$, where $\boldsymbol{\Sigma}_y \preceq \sigma_y^2 \mathbf{I}$ and $\boldsymbol{\Sigma}_e \succeq \sigma_e^2 \mathbf{I}$. For a d-dimensional Gaussian distribution, its entropy is given by:

$$H(\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})) = \frac{1}{2} \log \left((2\pi e)^d \det(\boldsymbol{\Sigma}) \right). \tag{19}$$

Therefore, we have $\det(\boldsymbol{\Sigma}_y) \leq (\sigma_y^2)^d$ and $\det(\boldsymbol{\Sigma}_e) \geq (\sigma_e^2)^d$, which further leads to:

$$\begin{aligned} H(\mathbf{H}^1|\mathbf{Y}) &\leq \frac{d}{2} \log \left(2\pi e \sigma_y^2 \right), \\ H(\mathbf{H}^1|\mathbf{E}^s) &\geq \frac{d}{2} \log \left(2\pi e \sigma_e^2 \right). \end{aligned} \tag{20}$$

Combining the above entropy bounds with $I(\mathbf{H}^1; \mathbf{Y}) \leq m$, we obtain:

$$\begin{aligned} I(\mathbf{E}^s; \mathbf{H}^1) &\leq m + \frac{d}{2} \log \left(2\pi e \sigma_y^2 \right) - \frac{d}{2} \log \left(2\pi e \sigma_e^2 \right) \\ &= m + \frac{d}{2} \log \frac{\sigma_y^2}{\sigma_e^2}. \end{aligned} \tag{21}$$

Under the Gaussian distribution assumption, the upper bound becomes tighter as the label-conditioned variance σ_y^2 decreases, yielding more compact preference-relevant information, or as the semantic-conditioned variance σ_e^2 increases, reflecting that more input information from $\mathbf{E}^s$ is preserved in the fixed-capacity $\mathbf{H}^1$. □

A.2 Proof of Theorem 4.3

Proof. In Stage II, the ID embeddings $\mathbf{e}_u^{id}$, $\mathbf{e}_i^{id}$ and the base model $\mathcal{M}^2$ are initialized using the projected semantic embeddings $\mathbf{z}_u$, $\mathbf{z}_i$ and the base model $\mathcal{M}^1$ from Stage I. Since the base model architecture remains consistent across both stages, the initial state of Stage II can be regarded as a continuation of the final state of Stage I. Therefore, the initial representations $\mathbf{H}^2$ in Stage II retain the condensed semantic information learned in Stage I, where $\mathbf{H}^2$ is the set of all representations ($\mathbf{h}_u^2$ and $\mathbf{h}_i^2$) produced by the second stage base model $\mathcal{M}^2$.

As training proceeds without the margin constraint in Stage II, the optimization objective shifts from compressing semantic inputs to maximizing the predictive signal for user preference, leading to an increase in $I(\mathbf{H}^2; \mathbf{Y})$. Therefore, the learned representations in Stage II preserve semantics relevant to the recommendation. □

A.3 Proof of Theorem 4.4

Proof. The margin constraint in Stage I bounds the mutual information between the compressed representation and the target as: $I(\mathbf{H}^1; \mathbf{Y}) \leq m$. From the proof in Theorem 1, it follows that:

$$I(\mathbf{E}^s; \mathbf{H}^1) \leq m + H(\mathbf{H}^1|\mathbf{Y}) - H(\mathbf{H}^1|\mathbf{E}^s) = m', \tag{22}$$

which can be reformulated as the following equivalent upper bound:

$$\min \left(I(\mathbf{E}^s; \mathbf{H}^1), m' \right). \tag{23}$$

The objective of Stage II is to maximize the mutual information $I(\mathbf{H}^2; \mathbf{Y})$ between the learned representation and the target labels. Therefore, the overall objective of MarRec is formulated as:

$$\min \left(I(\mathbf{E}^s; \mathbf{H}^1), m' \right) - I(\mathbf{H}^2; \mathbf{Y}), \tag{24}$$

which unifies the compression and preservation terms in line with the information bottleneck objective in Definition 4.1. □

B Experimental Settings

Table 3: Dataset statistics. Density represents the proportion of observed interactions relative to the full user-item interaction space. d_s is the dimensionality of the semantic embedding $\mathbf{e}_u^s$, $\mathbf{e}_i^s$.

Dataset	#Users	#Items	#Interactions	Density	d_s	LLMs	Embedding Model
Beauty	10,553	6,086	94,148	0.15%	3072	Llama3-8B	text-embedding-3-large
Sports	22,686	12,301	185,718	0.07%	3072	Llama3-8B	text-embedding-3-large
Steam	23,310	5,237	316,190	0.26%	1536	GPT-3.5-turbo	text-embedding-ada-002
Yelp	11,091	11,010	166,620	0.14%	1536	GPT-3.5-turbo	text-embedding-ada-002

B.1 Datasets

We utilize 4 widely used recommender datasets: Beauty, Sports, Yelp, and Steam, based on the LLMs-enhanced versions processed by two prior works (Ren et al., 2024; Cheng et al., 2025). These datasets are partitioned into training/validation/test sets with a ratio of 8:1:1. As shown in Table 3, these datasets differ in terms of size, interaction density, the dimensionality of their semantic embeddings, and the LLMs used. In addition, the two source works adopt different strategies for refining raw textual data and employ different textual embedding models to encode the textual content generated by LLMs. It is worth noting that Beauty and Sports have higher-dimensional semantic embeddings, which indicates that they potentially contain richer semantic information. The diversity in dataset characteristics enables a more comprehensive evaluation of the generalization and robustness.

B.2 Base Recommender Models

To verify the effectiveness of MarRec across different base recommender models, we adopt 5 representative recommender models as the backbone for $\mathcal{M}^{\mathbf{1}}$, $\mathcal{M}^{\mathbf{2}}$ and soft margin recommender model in our framework. These models differ in architecture and technique, covering both widely-used and advanced approaches.

The details of these models are summarized as follows:

- **NGCF** (Wang et al., 2019): A representative parameterized graph convolutional network (GCN)-based recommender model that propagates high-order collaborative signals on the user-item interaction graph.
- **GCCF** (Chen et al., 2020): A parameterized GCN-based recommender model that removes non-linear activation and adopts a residual structure.
- **LightGCN** (He et al., 2020): A non-parameterized GCN-based recommender model that removes feature transformation and non-linear activation to enhance efficiency and effectiveness.
- **SGL** (Wu et al., 2021): A non-parameterized contrastive learning-based recommender model built on LightGCN, which generates augmented views via random graph perturbations.
- **SimGCL** (Yu et al., 2022): A non-parameterized contrastive learning-based recommender model built on LightGCN, which generates augmented views by injecting noise into the graph propagation.

The non-parameterized models only learn user and item ID embeddings, whereas parameterized models additionally involve learning model-specific parameters, such as the transformation weights.

Therefore, MarRec built on LightGCN, SGL, and SimGCL, only requires the initialization of user and item ID embeddings ($\mathbf{e}_u^{id}$, $\mathbf{e}_i^{id}$), without the need to initialize model-specific parameters of $\mathcal{M}^{\mathbf{2}}$.

B.3 BASELINES

- **KAR** concatenates the dense MoE-projected semantic embeddings to the input of the base model.
- **RLMRec-Con** aligns model outputs with projected semantic embeddings via contrastive learning.
- **RLMRec-Gen** aligns the projected model outputs with semantic embeddings via masked modeling.
- **PAD** aligns collaborative and textual embeddings using multi-kernel maximum mean discrepancy.

In addition, our framework has three variants:

- **MarRec** applies a fixed margin constraint uniformly across all training instances.
- **Adaptive** leverages a meta-learning network to generate instance-specific margins.
- **Soft** derives data-dependent margins from the post-training objective of the base model.

B.4 EXPERIMENTAL SETTINGS

We fix the dimensionality d of ID embeddings to 50, the learning rate to 0.001, the batch size to 4096, the maximum training epochs to 500, and the early stopping patience to 10. For other hyperparameters, we first perform grid search on the base recommender model. To ensure a fair comparison, MarRec shares all common hyperparameters with the corresponding base model, and only the additional hyperparameters are further tuned, including the number of experts N and the margin-related parameters (i.e., margin m, adaptive regularization weight α, or soft scale factor β, depending on the variant used). N is selected from $\{2, 4, 8\}$, m is tuned within the range $[0, 2.0]$, and α, β are tuned within the range $[0, 1.0]$. Additionally, we use graph normalization based on the implementation from NGCF and Adam optimizer for model optimization.

The baselines also share the same hyperparameters as the corresponding base models. Specifically, for KAR, we select the number of experts in the dense MoE from $\{2, 4, 8\}$. For the contrastive learning loss in RLMRec-Con and RLMRec-Gen, we tune the weight from $\{0.001, 0.01, 0.1\}$ and the temperature coefficient from $\{0.1, 0.2, 0.5, 1.0\}$. For RLMRec-Gen, we also tune the mask rate from $\{0.01, 0.05, 0.1, 0.15, 0.2\}$. For PAD, we tune the multi-kernel alignment loss weight from $\{0.1, 0.15, 0.2, 0.25, 0.3\}$.

In addition, we use Recall@K and NDCG@K to evaluate recommendation performance, and measure training time and peak GPU memory usage to assess efficiency. All experiments are conducted on a single NVIDIA RTX 3090 GPU with 24GB of memory. To ensure accurate efficiency measurement, we run each experiment in isolation, ensuring that no other processes are active on the machine. For every method, hyperparameters and the final model state are selected according to the sum of Recall@20 and NDCG@20 on the validation set, and the selected model is then evaluated on test set.

In the cold-start setting, items are randomly partitioned into training/validation/test sets with a ratio of 8:1:1, and each split retains only interactions involving its assigned items. Consequently, validation and test items never occur in the training data. The LLMs-generated item embeddings encode only item-side textual information, and user semantic embeddings are constructed by aggregating item embeddings solely from the interactions available after splitting, following the same leakage-free setting as in the data sparsity setting. Therefore, neither test interactions nor information derived from them is used for training or model selection. All methods consistently use the same preprocessed data and these experimental settings to ensure fair comparisons and prevent data leakage.

C ADDITIONAL EXPERIMENTS

C.1 INDUSTRIAL EXPERIMENT

To evaluate the effectiveness and practicality of our method in industrial scenarios, we conduct a preliminary experiment by integrating the margin constraint into the final ranking stage of an online recommender system. Specifically, under the cold-start setting, we compare four methods: (1) without incorporating semantic embeddings (**Model**), (2) directly concatenating semantic embeddings to the input of the model (**Model+Emb**), (3) aligning the embeddings using contrastive learning (**Model+CL**), and (4) applying our proposed margin constraint for integration (**Model+Margin**).

Table 4: Performance of recommender models under the cold-start scenario in an industrial system.

	Model	Model+Emb	Model+CL	Model+Margin
AUC01	0.5972	0.5982	0.5968	0.6026
AUC11	0.7344	0.7348	0.7354	0.7362
AUC	0.6716	0.6722	0.6719	0.6750

Table 5: Warm-start and semantic-input analysis with LightGCN. R@20 denotes Recall@20, and N@20 denotes NDCG@20.

Methods	Beauty		Sports		Steam		Yelp	
	R@20	N@20	R@20	N@20	R@20	N@20	R@20	N@20
Base	0.1551	0.0711	0.0845	0.0382	0.1357	0.0827	0.1061	0.0641
ID-Standard	0.1544	0.0709	0.0846	0.0383	0.1359	0.0828	0.1065	0.0642
ID-Margin	0.1594	0.0727	0.0863	0.0388	0.1367	0.0833	0.1037	0.0623
Rand-Standard-MoE	0.1147	0.0524	0.0709	0.0322	0.1213	0.0726	0.0883	0.0525
Rand-Margin-MoE	0.1368	0.0619	0.0730	0.0327	0.1321	0.0804	0.1008	0.0602
Rand-Margin-MoE-Frozen	0.1506	0.0701	0.0809	0.0368	0.1342	0.0821	0.1088	0.0653
MarRec	**0.1829**	**0.0858**	**0.1142**	**0.0496**	**0.1463**	**0.0894**	**0.1273**	**0.0768**

Experimental Details. The objective of the model is to predict users' lifetime value (LTV), which serves as the ranking target in the system. The training set consists of **55 million samples**, while the test set contains **8 million samples**. The input to the model includes various features, such as attributes and interactions. For this experiment, we focus on one feature generated by LLMs and encoded using a contrastive learning-based text encoder. In addition, we adopt the industry-standard cold-start setting, where the model is trained without loading any previous checkpoints, and training is conducted only on the training set.

Evaluation Metrics. The final evaluation of LTV prediction includes three metrics: AUC01, AUC11, and AUC. AUC01 is calculated by treating the task as a binary classification problem, where samples with LTV > 0 are labeled as positive and those with LTV = 0 as negative. This metric evaluates the model's ability to distinguish between users with and without any predicted monetary value. AUC11 is computed only on samples with LTV > 0, measuring the model's ranking capability among positive samples. AUC is calculated over all samples and reflects the model's overall ranking performance.

Findings. As shown in Table 4, directly adding semantic embeddings to the input of the industrial model improves its performance under the cold-start scenario. However, the contrastive learning alignment underperforms directly using the semantic embeddings, indicating that such alignment distorts useful signals. In contrast, our proposed margin constraint effectively mitigates this issue and yields notable improvements in an industrial system.

C.2 Warm-Start and Semantic-Input Analysis

To distinguish the contribution of LLMs-derived semantics from generic warm-start and parameter-transfer effects, we construct controlled variants with LightGCN. We choose LightGCN because its simple architecture minimizes backbone-specific factors, enabling a clearer attribution of performance differences to the controlled components. **ID-Standard** and **ID-Margin** pretrain learnable ID embeddings with the same dimensionality as the projected semantic embeddings using the standard objective and the margin objective, respectively, and directly transfer them to Stage II. **Rand-Standard-MoE** and **Rand-Margin-MoE** instead use learnable random embeddings with the same dimensionality as the original semantic embeddings and transfer their MoE-projected embeddings to Stage II. **Rand-Margin-MoE-Frozen** further freezes the random embeddings, matching MarRec except that the input contains no semantic information. Stage II and all other settings remain identical between these variants and MarRec.

As shown in Table 5, ID-Standard remains within 0.5% of Base, indicating that generic ID warm-start alone contributes little to MarRec's improvement. Rand-Standard-MoE underperforms Base on all datasets, showing that the parameter-transfer pathway alone does not provide useful information. Applying the margin objective to random inputs yields only small or inconsistent improvements. The

Table 6: Results of retaining the contrastive learning loss in Stage I for SGL and SimGCL.

Backbone	Variant	Beauty		Sports		Steam		Yelp	
		R@20	N@20	R@20	N@20	R@20	N@20	R@20	N@20
SGL	MarRec	**0.1817**	**0.0850**	**0.1142**	**0.0502**	**0.1464**	**0.0894**	**0.1217**	**0.0721**
	MarRec + Stage I CL	0.1759	0.0803	0.1125	0.0492	0.1423	0.0862	0.0523	0.0273
SimGCL	MarRec	**0.1834**	**0.0849**	**0.1159**	**0.0510**	0.1462	0.0893	**0.1298**	**0.0786**
	MarRec + Stage I CL	0.1816	0.0847	0.1070	0.0468	**0.1470**	**0.0898**	0.1233	0.0747

Table 7: Recall@20 and NDCG@20 results with the non-GCN MF backbone.

Methods	Beauty		Sports		Steam		Yelp	
	R@20	N@20	R@20	N@20	R@20	N@20	R@20	N@20
Base	0.0958	0.0448	0.0329	0.0145	0.0858	0.0517	0.0757	0.0459
KAR	0.1172	0.0510	0.0676	0.0287	0.1014	0.0598	0.0863	0.0518
RLMRec-Con	0.1299	0.0596	0.0690	0.0299	0.1027	0.0614	0.0853	0.0509
RLMRec-Gen	0.0964	0.0448	0.0353	0.0152	0.0887	0.0536	0.0806	0.0487
PAD	0.1115	0.0502	0.0413	0.0178	0.0986	0.0590	0.0806	0.0481
MarRec	**0.1705**	**0.0796**	0.0989	0.0431	**0.1384**	0.0836	0.1148	0.0692
Adaptive	0.1671	0.0761	0.0951	0.0420	0.1382	**0.0841**	**0.1156**	**0.0695**
Soft	0.1680	0.0778	**0.1014**	**0.0441**	0.1382	0.0833	0.1137	0.0680

consistent gap between these content-free variants and MarRec demonstrates that MarRec improves recommendation by extracting useful information from LLMs-derived semantics, rather than through warm-start or parameter transfer alone.

C.3 Stage I Contrastive Loss Analysis

SGL and SimGCL construct contrastive learning losses from noise-perturbed views. We therefore remove the contrastive loss from Stage I to avoid forcing semantic extraction toward pseudo-noisy signals. To verify this adjustment, we introduce **MarRec + Stage I CL**, which retains the contrastive loss in both stages. As shown in Table 6, retaining CL in Stage I degrades Recall@20 and NDCG@20 and causes a substantial performance drop on SGL-Yelp. The only improvement on SimGCL-Steam is marginal. These results confirm that removing pseudo-noisy contrastive supervision from Stage I improves semantic extraction.

C.4 Non-GCN Backbone Analysis

In Section 5, the selected backbones are graph-based recommender models. This choice is motivated by their predominant use in collaborative recommendation, as graph propagation naturally captures high-order collaborative signals from user-item interactions, and also follows the experimental settings adopted in prior LLMs-enhanced recommendation studies. Accordingly, we select five representative graph-based recommender models that cover diverse architectures and learning paradigms. To examine whether MarRec generalizes beyond graph architectures, we further evaluate it with matrix factorization (MF), a non-graph backbone based solely on inner-product scoring. Under the same experimental settings, we implement the baselines and our three variants with MF as the backbone.

As shown in Table 7, MarRec, Adaptive, and Soft achieve the best or second-best Recall@20 and NDCG@20 results across all four datasets, substantially outperforming the LLMs-enhanced baselines. The improvements over Base are also larger than those observed with the GCN backbones. These results demonstrate that MarRec is not tied to graph propagation and generalizes to non-GCN recommender models.

C.5 Hyperparameters Analysis

In this analysis, we investigate the impact of key hyperparameters on our variants and examine whether our framework exhibits robustness under different hyperparameters. Since our variants share the same common hyperparameters with the base model, our analysis focuses on the additional

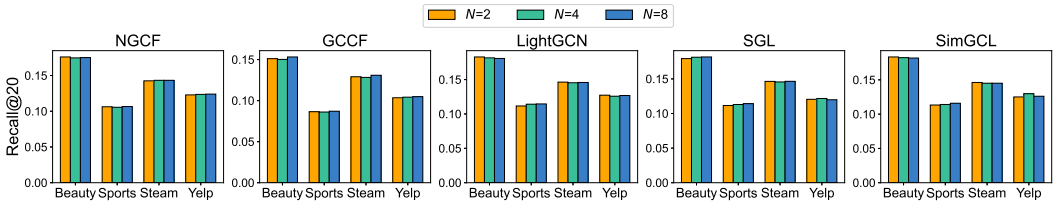

Figure 7: Results for the Hyperparameters Analysis of MoE experts N in MarRec.

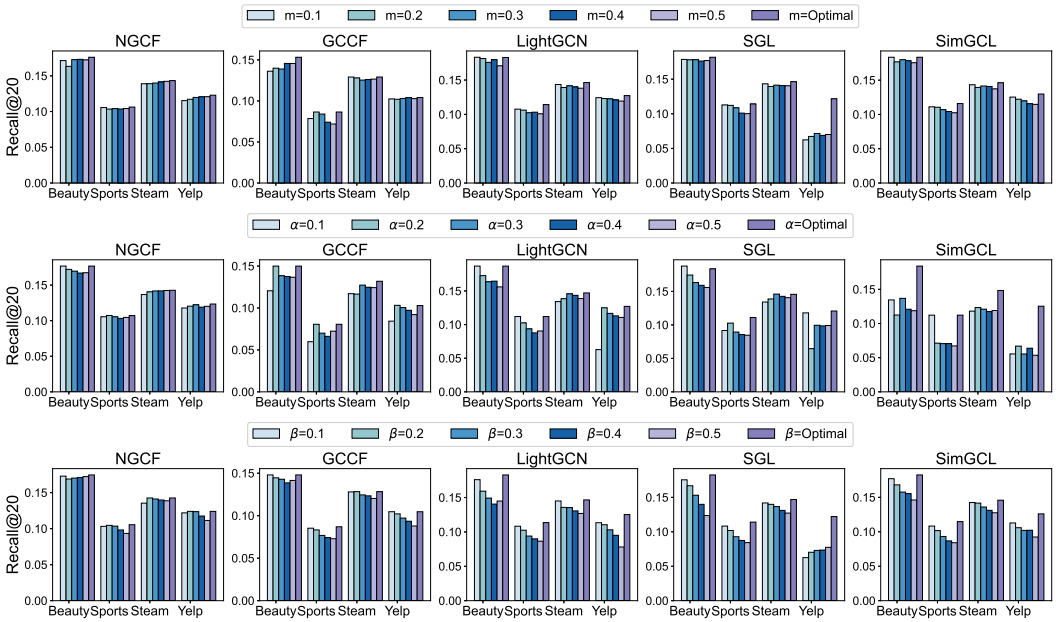

Figure 8: Results for margin m in MarRec, regularization weight α in Adaptive variant, and soft scale factor β in Soft variant. Optimal denotes the best hyperparameter value.

hyperparameters introduced by our framework: the number of MoE experts N, the margin m, the adaptive regularization weight α, and the soft scale factor β.

As shown in Figure 7, the performance of MarRec slightly improves or remains stable as N increases. This indicates that MarRec can preserve valuable semantic information during projection with few additional MoE parameters.

It can be observed from Figure 8 that MarRec achieves similar performance across different margin values, except on the Yelp dataset with the base model SGL. According to the results in Section 5.6, this sensitivity stems from the particularly small optimal margin, which makes performance more susceptible to deviations from the best value. Similar results are also observed for α and β, which is reasonable since α, β act as weights and are therefore expected to have relatively small values.

These results demonstrate the robustness of our variants with respect to their additional hyperparameters, reducing the burden of hyperparameter tuning in practical applications.

D Supplementary Experimental Details

D.1 Yelp, Recall@K, and NDCG@K Results for Overall Performance Analysis

Due to space limitations in the main text, we present the Yelp results for the Overall Performance Analysis (Section 5.1) in Table 8. We also provide additional experimental results under different top-K evaluation settings. Specifically, Figure 9 and Figure 10 show the Recall@K and NDCG@K

Table 8: Overall Performance Analysis (Section 5.1) results on Yelp.

Dataset		Yelp			
Backbone	Methods	Recall@20	NDCG@20	Time	Mem
NGCF	Base	0.1061	0.0622	65.23	512
	KAR	0.1088 (2.5%)	0.0644 (3.5%)	**60.24**	1320
	RLMRec-Con	0.1075 (1.3%)	0.0638 (2.6%)	97.48	1728
	RLMRec-Gen	0.1114 (5.0%)	0.0658 (5.8%)	91.50	1342
	PAD	0.0959 (-9.6%)	0.0555(-10.8%)	74.16	6566
	MarRec	0.1229(15.8%)	0.0736(18.3%)	108.30	**776**
	Adaptive	0.1234(16.3%)	0.0741(19.1%)	154.53	996
	Soft	**0.1240(16.9%)**	**0.0748(20.3%)**	205.45	796
GCCF	Base	0.0784	0.0453	10.18	450
	KAR	0.0898(14.5%)	0.0517(14.1%)	**18.22**	1168
	RLMRec-Con	0.0862(10.0%)	0.0501(10.6%)	51.01	1650
	RLMRec-Gen	0.0860 (9.7%)	0.0498 (9.9%)	41.47	1264
	PAD	0.0977(24.6%)	0.0572(26.3%)	69.73	6513
	MarRec	0.1042(32.9%)	0.0612(35.1%)	50.62	**710**
	Adaptive	0.1029(31.3%)	0.0602(32.9%)	45.13	910
	Soft	**0.1047(33.6%)**	**0.0620(36.9%)**	88.09	774
LightGCN	Base	0.1061	0.0641	55.19	382
	KAR	0.1164 (9.7%)	0.0688 (7.3%)	**77.12**	1116
	RLMRec-Con	0.1230(15.9%)	0.0736(14.8%)	101.72	1564
	RLMRec-Gen	0.1131 (6.6%)	0.0684 (6.7%)	105.55	1262
	PAD	0.1145 (7.9%)	0.0693 (8.1%)	157.60	6500
	MarRec	**0.1273(20.0%)**	0.0768(19.8%)	129.71	**686**
	Adaptive	0.1271(19.8%)	**0.0769(20.0%)**	140.81	912
	Soft	0.1252(18.0%)	0.0762(18.9%)	218.72	758
SGL	Base	0.1123	0.0672	73.30	932
	KAR	0.1110 (-1.2%)	0.0666 (-0.9%)	104.43	1698
	RLMRec-Con	0.1213 (8.0%)	0.0720 (7.1%)	99.10	1938
	RLMRec-Gen	0.1201 (7.0%)	0.0712 (6.0%)	**84.87**	1684
	PAD	0.1169 (4.1%)	0.0698 (3.9%)	195.11	6871
	MarRec	0.1217(8.4%)	0.0721(7.3%)	123.62	**932**
	Adaptive	0.1208 (7.6%)	0.0717 (6.7%)	139.38	**932**
	Soft	**0.1221(8.7%)**	**0.0729(8.5%)**	202.00	**932**
SimGCL	Base	0.1148	0.0698	299.94	948
	KAR	0.1128 (-1.7%)	0.0686 (-1.7%)	590.32	1822
	RLMRec-Con	0.1213 (5.7%)	0.0730 (4.6%)	**234.46**	1976
	RLMRec-Gen	0.1222 (6.5%)	0.0731 (4.7%)	378.78	1700
	PAD	0.1200 (4.5%)	0.0723 (3.6%)	246.31	6886
	MarRec	**0.1298(13.1%)**	**0.0786(12.6%)**	315.50	**948**
	Adaptive	0.1253 (9.2%)	0.0761 (9.0%)	385.98	**948**
	Soft	0.1260(9.8%)	0.0763(9.3%)	687.53	**948**

results. To provide a clearer comparison, we only report the results of Base, RLMRec-Con, and MarRec, as RLMRec-Con achieves the most competitive performance and efficiency among the baselines. Since the Adaptive and Soft variants exhibit comparable trends across different K values, we present the results of the base MarRec variant as a representative. These results further validate the effectiveness of MarRec across a range of K values. As K increases, MarRec consistently achieves larger performance gains over the base model and RLMRec-Con, demonstrating its superior ability to retrieve relevant items under various top-K recommendations.

D.2 Ablation Study NDCG@20 Results

To complement the Ablation Study in Section 5.2, we further report the NDCG@20 results in Figure 11. The trends observed in NDCG@20 are consistent with those in Recall@20, confirming the effectiveness of each key component in our MarRec framework. Notably, removing Stage II, the margin constraint, or freezing of semantic embeddings leads to significant performance drops across different base models, highlighting their critical roles in extracting and preserving recommendation-useful semantics. Meanwhile, the relatively smaller drops observed for the w/o $\mathcal{M}^1$ and w/o MoE variants again demonstrate that while these components contribute positively, MarRec remains effective even with simpler configurations. In addition, these figures clearly show

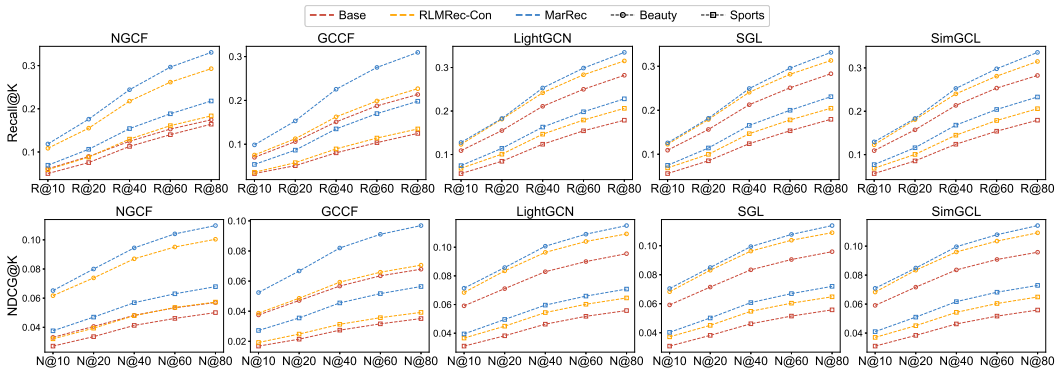

Figure 9: Recall@K (R@K) and NDCG@K (N@K) results on Beauty and Sports datasets.

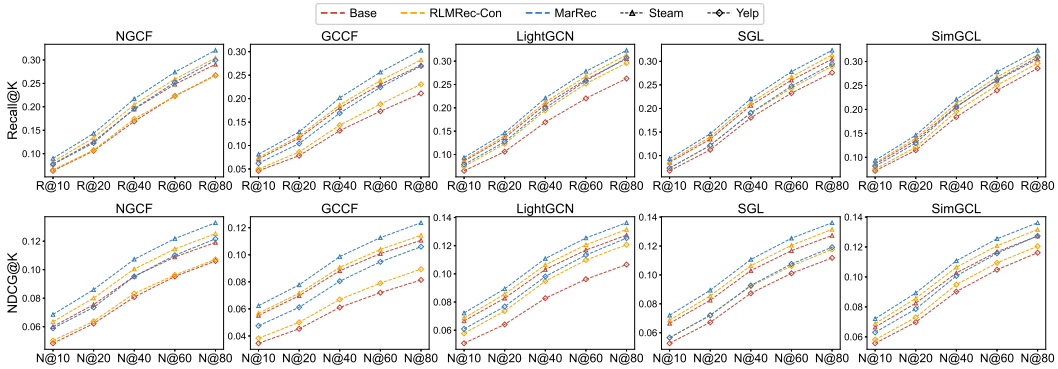

Figure 10: Recall@K (R@K) and NDCG@K (N@K) results on Steam and Yelp datasets.

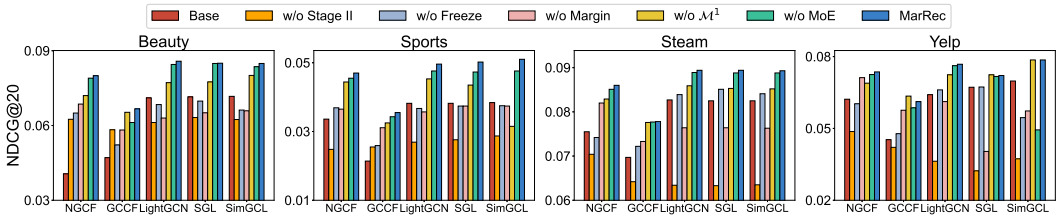

Figure 11: NDCG@20 results for Ablation Study in Section 5.2.

that despite the large performance gap among base models, MarRec also performs consistently well across different base models, which suggests that it is capable of effectively extracting and utilizing semantic information to enhance recommendation performance, even when the underlying base model is relatively weak.

D.3 Data Sparsity Analysis Full Results

To complement the Data Sparsity Analysis in Section 5.3, we present additional results in Figure 12 and Figure 13, showing Recall@20 and NDCG@20 across datasets. These results highlight the performance improvements of MarRec over the base model across all datasets, though the magnitude of improvement varies depending on the characteristics of the datasets.

As shown in these figures, MarRec consistently outperforms the baseline models across all ratios, except for certain cases such as SimGCL on Yelp dataset. This can be attributed to the hyperparameter settings used in this analysis, which are consistent with those in Section 5.1. These hyperparameters

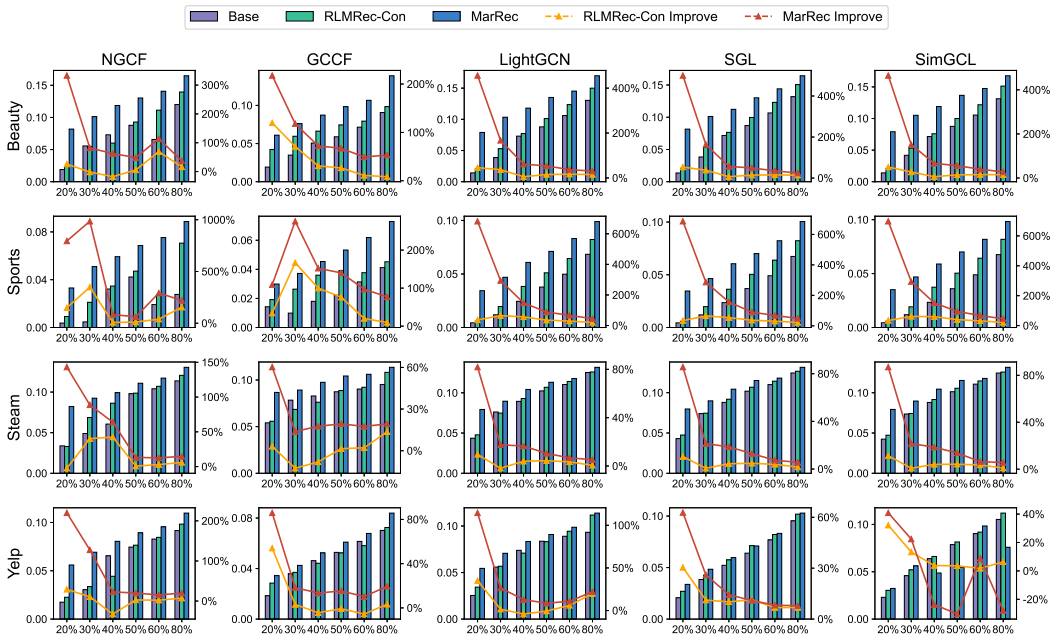

Figure 12: Recall@20 results for Data Sparsity Analysis in Section 5.3. The x-axis denotes sparsity ratio and the right y-axis shows improvement over Base.

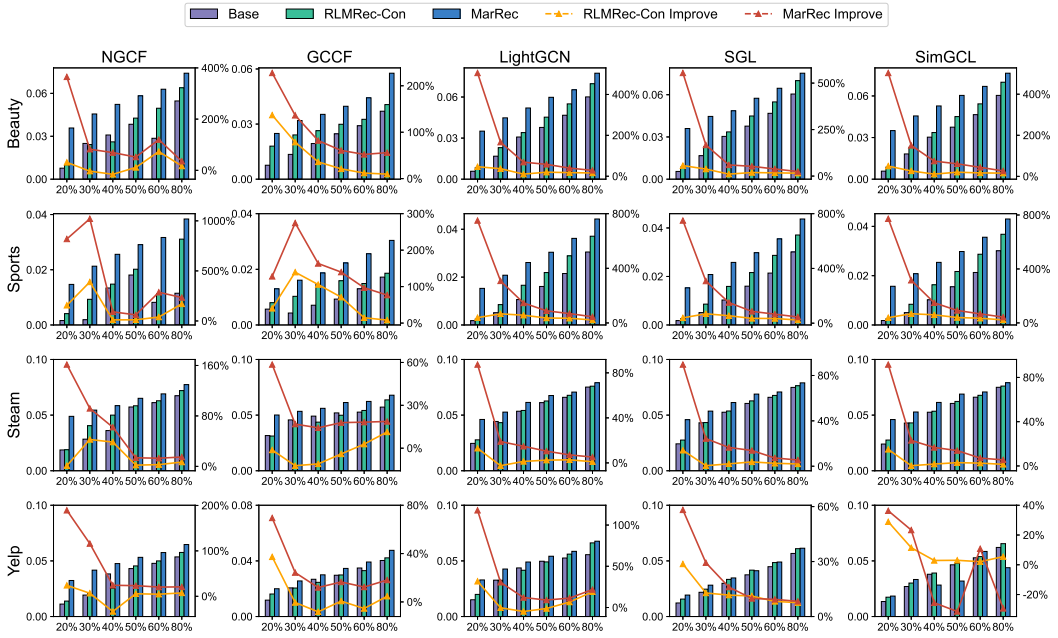

Figure 13: NDCG@20 results for Data Sparsity Analysis in Section 5.3. The x-axis denotes sparsity ratio and the right y-axis shows improvement over Base.

are optimized on the original dataset and may not be suitable for these specific cases. Since these cases are rare, their impact on the overall results can be ignored.

Additionally, on the Beauty and Sports datasets, which have the lowest interaction density, MarRec demonstrates the most significant improvements. This shows that even in scenarios with weak collaborative signals, our margin constraint mechanism can effectively extract valuable semantic information, helping to mitigate the effects of sparse data.

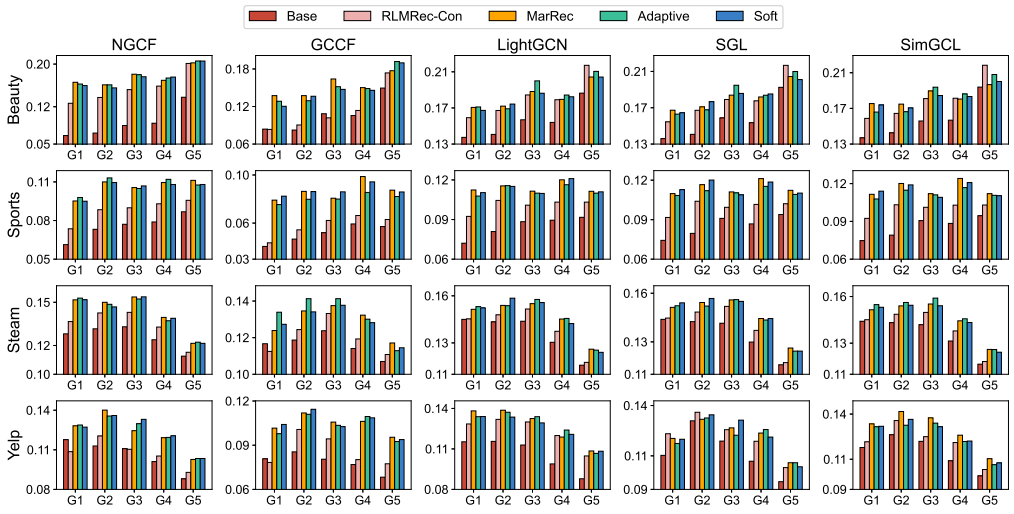

Figure 14: Recall@20 results for Long-Tail Effect Analysis in Section 5.4.

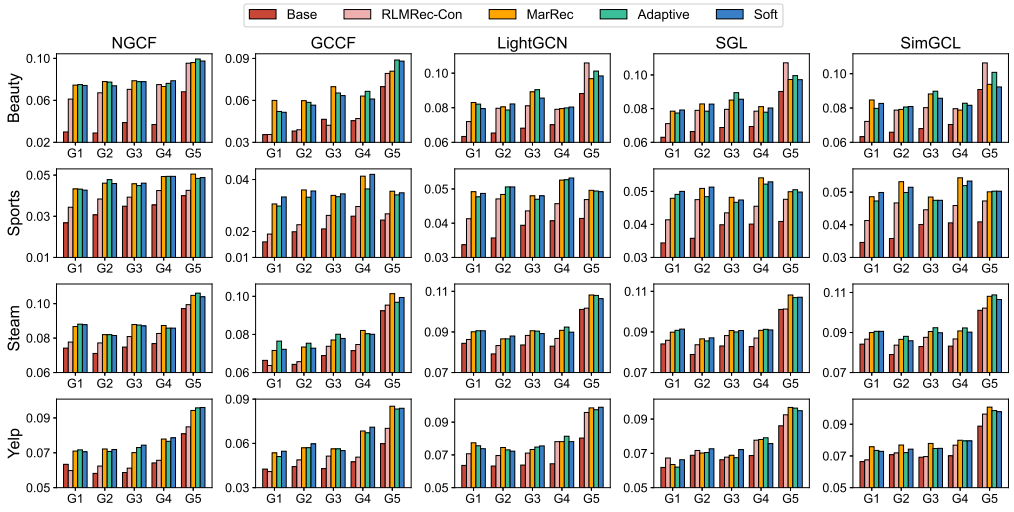

Figure 15: NDCG@20 results for Long-Tail Effect Analysis in Section 5.4.

D.4 Long-Tail Effect Analysis Full Results

The results of the Long-Tail Effect Analysis in Section 5.4 for all datasets are shown in Figure 14 and Figure 15. The results are consistent with the observations in Section 5.4.

D.5 Cold-Start Analysis NDCG@20 Results

The NDCG@20 results of the Cold-Start Analysis in Section 5.5 are shown in Figure 16. It can be observed that the trends in NDCG@20 are consistent with those in Recall@20. Adaptive outperforms MarRec in most cases, indicating that the adaptive margin, which is learned during training, helps the model better adjust to specific data characteristics and improves performance. On the other hand, Soft generally underperforms MarRec, suggesting that the lack of information about cold-start items in the base model leads to inappropriate margins. This highlights the importance of the margin in capturing relevant semantics, as it plays a critical role in improving performance, particularly when dealing with cold-start scenarios.

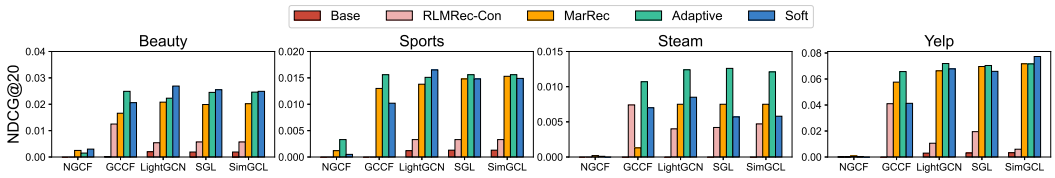

Figure 16: NDCG@20 results for Cold-Start Analysis in Section 5.5.

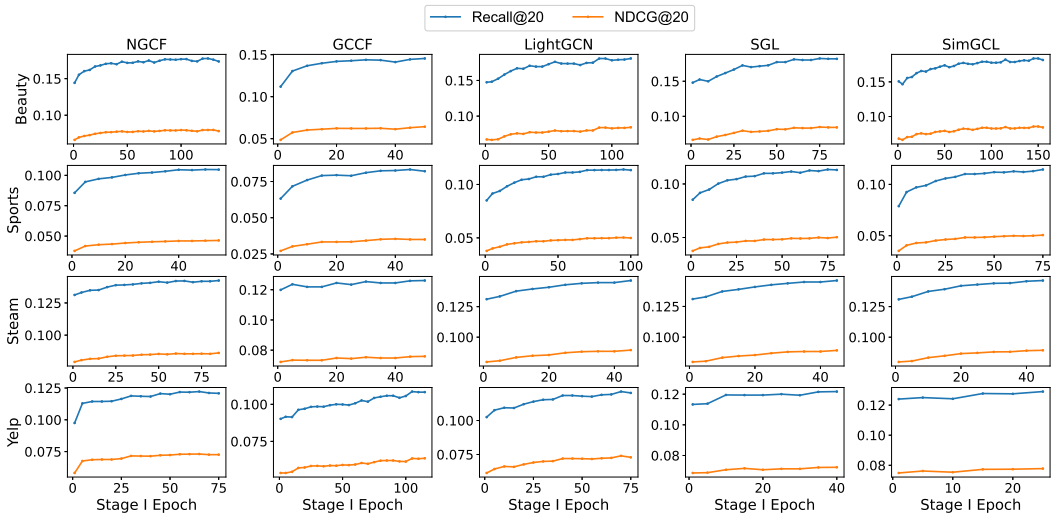

Figure 17: Full Recall@20 and NDCG@20 results for Stage Impact Analysis in Section 5.7.

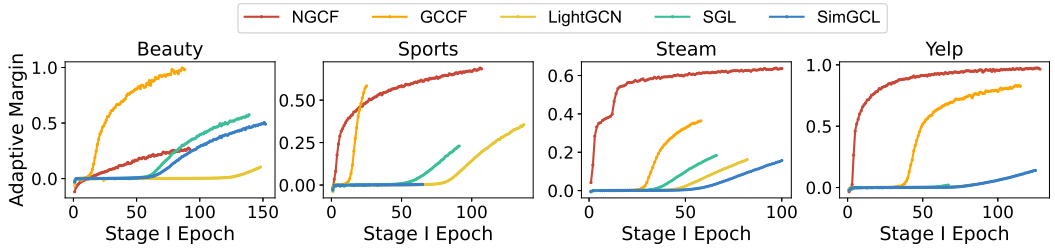

Figure 18: The margin values in the Adaptive variant across Stage I training epochs.

D.6 Stage Impact Analysis Full Results

The complete results for the Stage Impact Analysis in Section 5.7 across datasets are shown in Figure 17. As illustrated in the figure, the impact of Stage I on Stage II is consistent across different models and datasets. This further validates the model-agnostic nature of our framework.

Furthermore, we illustrate the training dynamics of m_a by tracking the average margin per epoch in Stage I for Adaptive. As shown in Figure 18, the adaptive margin values generally increase and tend to stabilize in later epochs. For base models NGCF and GCCF, where our framework achieves notable performance improvements, the margins rise rapidly during early training and plateau afterward, indicating that the model captures discriminative signals early and gradually refines them. In addition, the exponential-like growth pattern suggests that the regularization loss has a stronger impact on m_a, highlighting the potential to improve performance by refining the margin-based loss design.