

# SAILRec: Steering LLM Attention to Dual-Side Semantically Aligned Collaborative Embeddings for Recommendation

Xi Wu, Jiale Wang, Zihan Wang, Yichen Gao,  
Xiaocui Yang, Shi Feng<sup>†</sup>, Daling Wang, Yifei Zhang

School of Computer Science and Engineering, Northeastern University,  
Shenyang 110819, China

wux2@mails.neu.edu.cn, wangjiale12356@mail.ustc.edu.cn, wzh1998921@gmail.com,  
gaoyc3@mails.neu.edu.cn, {yangxiaocui, fengshi, wangdaling, zhangyifei1}@cse.neu.edu.cn

## Abstract

Recent LLM-based recommenders enhance language models with collaborative embeddings from user-item interactions, but making such embeddings available does not ensure their proper use during inference. Through a diagnostic attention analysis, we find that the utilization of collaborative embeddings is depth-dependent and alignment-sensitive, suggesting that LLMs need to balance their internal semantic knowledge with external collaborative knowledge. To address this issue, we propose SAILRec, an LLM-based recommender that improves this balance through dual-side semantic alignment and hierarchical attention steering. The former aligns item-side embeddings with item-text semantics and user-side embeddings with codebook-based semantic profiles, while the latter suppresses premature shallow-layer collaborative interference and strengthens collaborative evidence in deeper decision layers. Experiments on MovieLens-1M and Amazon-Book show that SAILRec consistently outperforms representative baselines, with ablation and masking analyses validating its key designs. Our implementation is available at <https://github.com/XuPhoenixyear/SAILRec>.

## 1 Introduction

Large language models (LLMs) bring strong semantic understanding and instruction-following abilities to recommendation (Li et al., 2024; Deldjoo et al., 2026; Wu et al., 2024), but they do not naturally model collaborative knowledge hidden in interaction histories. Recent methods therefore inject collaborative embeddings derived from collaborative filtering models (Collab.) into LLMs as soft tokens, as in CoLLM (Zhang et al., 2025), but the collaborative knowledge encoded in these embeddings remains separated from the semantic space of LLMs. SeLLa-Rec (Wang et al., 2026b) shows that semantic alignment can mitigate this

gap through item-side distillation and contrastive alignment, indicating that collaborative knowledge should be made semantically compatible with the LLM before being injected. Nevertheless, existing methods mainly focus on constructing and injecting collaborative embeddings, while paying less attention to how LLMs actually balance and use these collaborative signals with their internal semantic knowledge during inference.

To examine how LLMs use collaborative embeddings during inference, we conduct a diagnostic experiment on MovieLens-1M. Since scaled dot-product attention controls value aggregation from different token groups and is widely used to probe Transformer behavior (Vaswani et al., 2017; Clark et al., 2019; Abnar and Zuidema, 2020), we use attention scores as diagnostic signals. Prior layer-wise analyses suggest that shallow Transformer layers mainly encode lexical, local syntactic, and contextual features, while deeper layers integrate higher-level semantic and task-relevant information (Tenney et al., 2019). Based on this view, we adopt a CoLLM-style injection structure and measure answer-position attention to collaborative embeddings on the user-side, collaborative embeddings on the item-side, and historical embeddings of the item-title, as shown in Figure 1. We do not assume that a specific attention curve is inherently optimal. Instead, we use it to diagnose how collaborative embeddings are internally utilized. The results reveal two observations. First, collaborative embedding utilization is depth-dependent. Their attention remains low in shallow layers and becomes more active in middle-to-deep layers, suggesting that collaborative knowledge is more likely to contribute to preference matching and decision-stage prediction than to early textual encoding. Second, semantic alignment significantly changes the utilization pattern of collaborative embeddings and is associated with better recommendation performance. These observations suggest that effective

<sup>†</sup>Corresponding author.

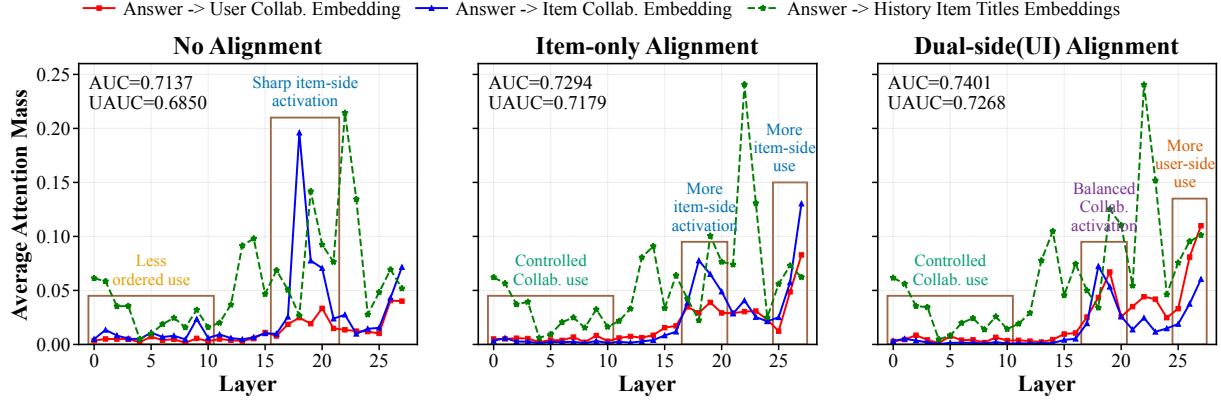


Figure 1: Mean attention from the answer position to key token groups under different semantic alignment settings on MovieLens-1M. Collaborative embedding utilization is depth-dependent and alignment-sensitive. No alignment shows less ordered early use and sharp item-side activation, item-side-only alignment improves late-layer accessibility of collaborative signals, and dual-side alignment achieves better performance with more balanced collaborative use and increased top-layer user-side attention.

use of collaborative embeddings requires addressing two questions: whether they are understandable to the LLM semantic space, and when they should affect Transformer inference.

Based on these findings, we propose SAILRec, short for **Steering Attention to collaborative embeddings In LLM Recommendation**. It is an LLM-based recommender that improves collaborative embedding utilization. SAILRec addresses the above two questions from complementary aspects. First, it performs dual-side semantic alignment to improve the semantic accessibility of user-side and item-side collaborative embeddings. The item side is aligned with LLM semantic representations of item texts, while the user side is aligned with codebook-based semantic profiles from historical interactions. Second, SAILRec introduces hierarchical attention steering to control when collaborative knowledge is used across Transformer layers. This mechanism follows the tendency that collaborative embeddings become more active in middle-to-deep layers. It suppresses premature collaborative interference in shallow layers, preserves natural interaction in middle layers, and strengthens collaborative evidence in deep layers for preference matching. By combining both designs, SAILRec makes external collaborative knowledge more accessible to the LLM semantic space and more appropriately involved in recommendation decisions. Our contributions can be summarized as follows:

- We provide a diagnostic analysis showing that injected collaborative embeddings are utilized in a depth-dependent and alignment-sensitive manner, motivating the need to improve both their

semantic accessibility and layer-wise usage.

- We propose SAILRec, which combines dual-side semantic alignment and hierarchical attention steering to improve the semantic accessibility and layer-wise utilization of Collab. embeddings.
- We conduct experiments on two public datasets, showing that SAILRec outperforms representative baselines and that ablation and masking analyses validate its key designs.

## 2 Related Work

### 2.1 LLM as Recommender

LLM-only recommendation verbalizes user histories, candidate items, and task objectives, enabling LLMs to perform preference prediction, ranking, or next-item prediction through prompting, instruction tuning, or lightweight adaptation (Geng et al., 2022; Gao et al., 2023). TALLRec (Bao et al., 2023) aligns LLMs with recommendation tasks through Alpaca tuning and rec-tuning, while RecRanker (Luo et al., 2025) trains LLMs as top- $K$  rankers with user sampling, position-shifted prompts, and hybrid pointwise, pairwise, and list-wise ranking. RecLM (Jiang et al., 2025) introduces recommendation instruction tuning with reinforcement learning rewards to capture diverse user preferences. Later studies move from task tuning to structural adaptation. CTR-Sink (Li et al., 2026) introduces behavior-level sink tokens and attention biases to aggregate fragmented behavioral semantics in long interaction sequences. HatLLM (Cui et al., 2025) applies hierarchical attention masking to textual interaction histories, emphasizing intra-

item semantics in shallow layers and cross-item relations in deeper layers. Beyond recommendation, PASTA (Zhang et al., 2024a) performs post-hoc attention steering during LLM inference by redirecting selected attention heads toward specified textual content. In contrast, SAILRec explicitly injects external user- and item-side collaborative embeddings, semantically aligns them with the LLM representation space, and applies depth-aware attention steering to regulate when collaborative evidence influences recommendation. Further comparisons are provided in Appendix B.6.

## 2.2 LLM with Collaborative Knowledge

To mitigate the limited ability of LLM-only recommenders to capture collaborative signals, recent studies integrate knowledge from collaborative filtering models (Collabs.) into LLM-based recommendation. LLaRA (Liao et al., 2024) maps ID representations into behavioral tokens through a projector, and uses curriculum prompt tuning to progressively combine textual semantics with behavioral patterns. BIGRec (Bao et al., 2025) does not explicitly inject collaborative embeddings into the internal representation of LLMs, but it incorporates collaborative information in a two-step grounding process and shows that LLMs still have limited ability to absorb statistical collaborative signals. CoLLM (Zhang et al., 2025) directly maps representations learned by external collaborative models into the LLM input space, making collaborative knowledge available as soft prompts. SeLLa-Rec (Wang et al., 2026b) further identifies the semantic gap between the collaborative space and the LLM semantic space, and alleviates this gap through semantic knowledge distillation. Beyond input-level injection, recent methods further extend this line of work to parameter and objective coupling (Liu et al., 2025; Lin et al., 2026). However, this line of work still pays limited attention to how collaborative embeddings are hierarchically utilized inside LLMs across Transformer depths, which is the central focus of SAILRec.

## 3 Methods

### 3.1 Task Formulation

We study click-through rate prediction in LLM-based recommendation, a standard recommendation task widely studied in conventional recommender systems (Cheng et al., 2016; Guo et al., 2017). Let  $\mathcal{U}$  and  $\mathcal{I}$  be the user and item sets. For

a user  $u \in \mathcal{U}$  and a candidate item  $i \in \mathcal{I}$ , let  $\mathcal{H}_u$  denote the historical interaction sequence and  $y_{u,i} \in \{0, 1\}$  denote the binary label, where 1 indicates a click or positive preference. The task is to estimate the click probability  $\hat{y}_{u,i} \in [0, 1]$  according to  $\mathcal{H}_u$  and  $i$ .

We construct a natural language prompt with the historical item list, the target item title, and two kinds of special tokens for collaborative embeddings (Collab. tokens). Prompt details are provided in Appendix A.1. At the answer position, the LLM produces vocabulary logits  $\mathbf{o}_{u,i} \in \mathbb{R}^{|\mathcal{V}|}$ , where  $\mathcal{V}$  is the vocabulary. We take the logits of Yes and No, denoted by  $o_{\text{Yes}}$  and  $o_{\text{No}}$ , and compute

$$\hat{y}_{u,i} = \frac{\exp(o_{\text{Yes}})}{\exp(o_{\text{Yes}}) + \exp(o_{\text{No}})}. \quad (1)$$

The recommendation objective is optimized with binary cross-entropy:

$$\mathcal{L}_{\text{rec}} = -y_{u,i} \log \hat{y}_{u,i} - (1 - y_{u,i}) \log (1 - \hat{y}_{u,i}). \quad (2)$$

### 3.2 Overview

The overall architecture of SAILRec is shown in Figure 2. Starting from recommendation data, a collaborative filtering model (Collab.) first learns user and item collaborative embeddings. These embeddings are then passed through two warm-up mapping layers and inserted into the prompt together with textual tokens. The resulting mixed embeddings are fed into the LLM, where layer-wise attention steering controls the use of collaborative information before the model predicts ‘Yes’ or ‘No’. In this pipeline, dual-side semantic alignment provides reliable collaborative embeddings, and hierarchical attention steering regulates their utilization inside the LLM.

### 3.3 Dual-side Alignment

**Collaborative filtering model (Collab.) construction.** Before dual-side semantic alignment, SAILRec first trains a Collab. to provide task-specific collaborative embeddings for users and items. Specifically, we use matrix factorization (MF) as the collaborative model. For a user  $u \in \mathcal{U}$  and an item  $i \in \mathcal{I}$ , MF learns a user embedding  $\mathbf{e}_u^c \in \mathbb{R}^{d_c}$  and an item embedding  $\mathbf{e}_i^c \in \mathbb{R}^{d_c}$ , where  $d_c$  denotes the dimension of collaborative embeddings. After training, the MF parameters are frozen. The learned user embedding  $\mathbf{e}_u^c$  and item embedding  $\mathbf{e}_i^c$  serve as the inputs for subsequent user-side and item-side semantic alignment.

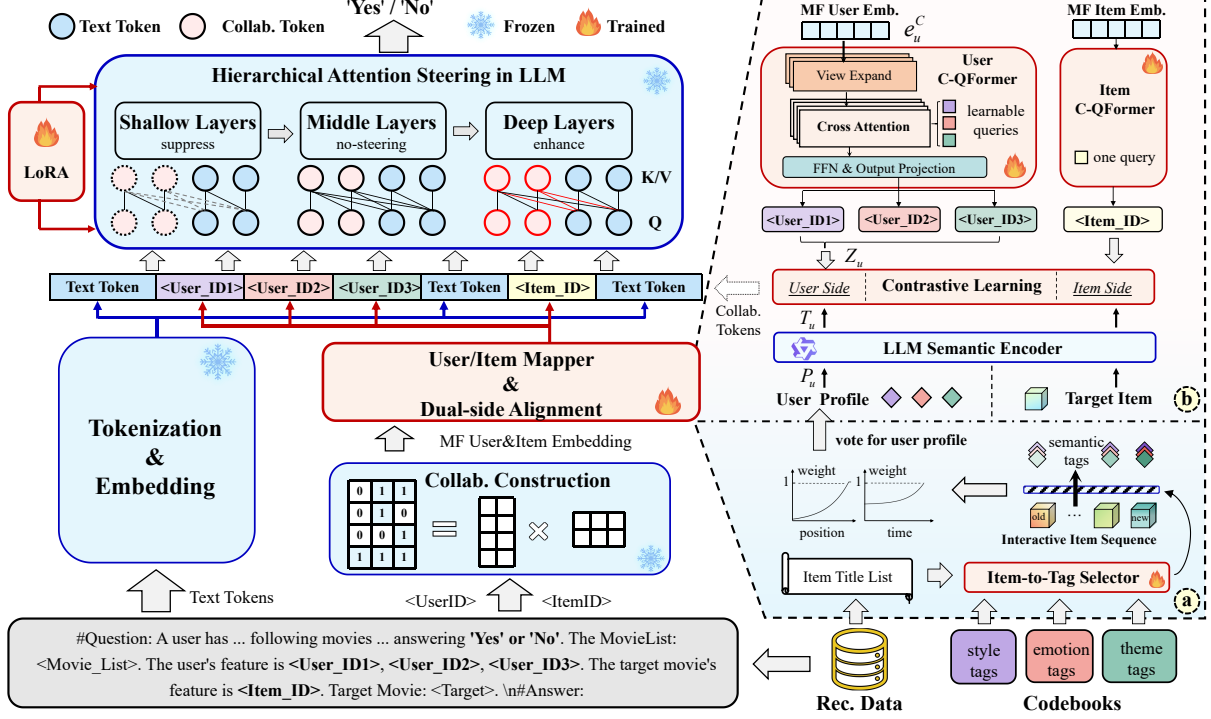


Figure 2: The overall architecture of SAILRec. The left part illustrates the main recommendation pipeline with collaborative embedding injection and hierarchical attention steering. The right part shows the dual-side warm-up process for user and item collaborative embedding mappers. Marker (a) denotes the construction of user-side alignment targets from historical interactions and semantic codebooks, while marker (b) denotes the dual-side contrastive learning process for user-side and item-side semantic alignment.

**User-side semantic alignment.** Since anonymous user IDs have no direct linguistic meaning to the LLM, SAILRec needs structured semantic anchors to describe user preferences. Moreover, user preference is not only determined by item topics, but also by stylistic form, affective response, and high-level value orientation. Motivated by this, SAILRec derives semantic user profiles from historically interacted items with three phrase-level codebooks,  $\mathcal{C}^A$ ,  $\mathcal{C}^B$ , and  $\mathcal{C}^C$ , corresponding to style, emotion, and theme, respectively. Each codebook  $\mathcal{C}^r = \{c_1^r, \dots, c_{K_r}^r\}$  contains candidate tags for semantic direction  $r \in \{A, B, C\}$ . We use the frozen LLM semantic encoder  $E_{LLM}(\cdot)$  to encode item titles and codebook tags. Examples of semantic tags in the codebooks are provided in Appendix A.2. Then a lightweight item-to-tag selector is trained to select the tag whose LLM semantic representation is most compatible with the item-title representation in each semantic direction. For user  $u$  with history  $\mathcal{H}_u = (i_1^u, \dots, i_T^u)$ , each historical item is weighted by positional and temporal decay. Specifically, we compute  $w_j = w_j^{\text{pos}} w_j^{\text{time}}$  and normalize it to  $\tilde{w}_j = w_j / \sum_{\ell=1}^T w_\ell$ . Then, for each semantic

direction, SAILRec aggregates historical tags by weighted voting and selects the top-scoring tag:

$$S_u^r(c) = \sum_{j=1}^T \tilde{w}_j \mathbb{I}[\phi^r(i_j^u) = c], \quad (3)$$

$$p_u^r = \arg \max_{c \in \mathcal{C}^r} S_u^r(c).$$

The final semantic user profile is  $\mathcal{P}_u = \{p_u^A, p_u^B, p_u^C\}$ . The LLM semantic representations of these three profile tags are used as the alignment targets for user-side collaborative knowledge.

After constructing user-side semantic targets, SAILRec trains a user-side mapping module to align low-dimensional collaborative embeddings with the high-dimensional LLM semantic space. We design a lightweight variant of Q-Former (Li et al., 2023), named Collaborative Q-Former (C-QFormer), which uses learnable queries and cross-attention to extract collaborative information from MF embeddings. Given the frozen MF user embedding  $e_u^c \in \mathbb{R}^{d_c}$ , the user-side C-QFormer  $g_u(\cdot)$  maps it into three high-dimensional collaborative embeddings, denoted as  $\mathbf{Z}_u = g_u(e_u^c) = [\mathbf{z}_u^A, \mathbf{z}_u^B, \mathbf{z}_u^C] \in \mathbb{R}^{3 \times d_L}$ , where  $d_L$

is the LLM hidden size. We use special textual placeholders, namely  $\langle \text{User\_ID1} \rangle$ ,  $\langle \text{User\_ID2} \rangle$ , and  $\langle \text{User\_ID3} \rangle$ , to indicate the insertion positions of the three user-side collaborative embeddings. After the corresponding mapped embeddings are inserted into these positions, we refer to them as Collab. tokens for brevity.

For the semantic user profile  $\mathcal{P}_u = \{p_u^A, p_u^B, p_u^C\}$ , the frozen LLM semantic encoder provides target embeddings  $\mathbf{T}_u = [\mathbf{t}_u^A, \mathbf{t}_u^B, \mathbf{t}_u^C]$ , where  $\mathbf{t}_u^r = E_{\text{LLM}}(p_u^r)$  and  $r \in \{A, B, C\}$  corresponds to style, emotion, and theme. We then apply slot-wise InfoNCE alignment (van den Oord et al., 2018). For a training batch  $\mathcal{B}$ , the alignment loss of semantic direction  $r$  is

$$\mathcal{L}_u^r = -\frac{1}{|\mathcal{B}|} \sum_{u \in \mathcal{B}} \log \frac{\exp(\text{sim}(\mathbf{z}_u^r, \mathbf{t}_u^r)/\tau)}{\sum_{v \in \mathcal{B}} \exp(\text{sim}(\mathbf{z}_u^r, \mathbf{t}_v^r)/\tau)}, \quad (4)$$

where  $\text{sim}(\cdot, \cdot)$  denotes cosine similarity and  $\tau$  is the temperature. The final user-side alignment objective  $\mathcal{L}_u$  is obtained by averaging  $\mathcal{L}_u^r$  over the three semantic directions  $r \in \{A, B, C\}$ . Through this process, C-QFormer maps implicit user collaborative preferences into structured Collab. tokens aligned with three semantic directions, providing semantically structured user-side collaborative information for LLM-based recommendation.

**Item-side semantic alignment.** Item-side semantic alignment is simpler because the LLM already contains world knowledge about items. We therefore use the LLM semantic representation of each item title as the alignment target. Given the frozen MF item embedding, the item-side C-QFormer uses one learnable query to map it into a high-dimensional item-side Collab. token. This token is then aligned with the corresponding item semantic representation using the InfoNCE loss, making item-side collaborative knowledge compatible with LLM item semantics.

### 3.4 Hierarchical Attention Steering

SAILRec introduces hierarchical attention steering inside the LLM by explicitly partitioning the  $L$  Transformer layers into shallow, middle, and deep groups. The shallow group refers to early Transformer layers, the middle group refers to intermediate layers, and the deep group refers to top layers close to the output. Let  $\mathcal{C}$  denote the key positions of user-side and item-side Collab. tokens; at layer  $\ell$ , SAILRec steers attention by adding a bias

only to logits associated with these key positions. The attention weights are computed as

$$\mathbf{A}^\ell = \text{softmax} \left( \frac{\mathbf{Q}^\ell (\mathbf{K}^\ell)^\top}{\sqrt{d_h}} + \mathbf{M} + \Delta^\ell \right), \quad (5)$$

where  $\mathbf{M}$  is the causal mask. The bias matrix  $\Delta^\ell$  has the same shape as the attention logits, and its entry  $\Delta_{q,k}^\ell$  modifies only the attention path from query position  $q$  to key position  $k$  when  $k$  is a Collab. token position.

For shallow layers, SAILRec suppresses premature attention to Collab. tokens by assigning a negative bias to their key positions:

$$\Delta_{q,k}^\ell = \begin{cases} b_s, & k \in \mathcal{C}, \\ 0, & k \notin \mathcal{C}, \end{cases} \quad b_s < 0. \quad (6)$$

This reduces the influence of collaborative embeddings in early layers, allowing the model to focus on lexical, syntactic, and contextual understanding.

For middle layers, SAILRec keeps the original attention structure unchanged by setting  $\Delta_{q,k}^\ell = 0$  for all positions. This allows textual semantics and collaborative information to interact naturally without additional steering.

For deep layers, SAILRec strengthens the attention to Collab. tokens by assigning a positive bias to their key positions:

$$\Delta_{q,k}^\ell = \begin{cases} b_d(\ell), & k \in \mathcal{C}, \\ 0, & k \notin \mathcal{C}, \end{cases} \quad b_d(\ell) > 0. \quad (7)$$

Here,  $b_d(\ell)$  is a depth-dependent schedule, meaning that the positive bias changes with the layer depth. This design increases the influence of collaborative evidence near the decision stage. In this way, SAILRec regulates only the attention paths toward Collab. tokens and explicitly models their layer-wise utilization during LLM inference.

### 3.5 Training Strategy

SAILRec adopts a three-stage training strategy:

**Stage 1:** We train an MF model on recommendation data to obtain low-dimensional user and item collaborative embeddings, and then freeze it as the collaborative knowledge provider.

**Stage 2:** We respectively warm up the user-side and item-side C-QFormers with contrastive alignment, using semantic user profiles and LLM item-title representations as targets.

**Stage 3:** We perform supervised fine-tuning (SFT) based on Low-Rank Adaptation (LoRA) (Hu

et al., 2022) for click-through rate prediction. In this stage, the MF model remains frozen, while LoRA parameters and both C-QFormers are jointly optimized under hierarchical attention steering.

## 4 Experiment

### 4.1 Experimental Settings

**Datasets.** Following the experimental settings and data processing protocols of CoLLM (Zhang et al., 2025) and SeLLa-Rec (Wang et al., 2026b), we evaluate SAILRec on two public datasets, MovieLens-1M (Harper and Konstan, 2015) and Amazon-Book (Ni et al., 2019). MovieLens-1M is suitable for evaluating recommendation performance in a relatively dense movie preference scenario, while Amazon-Book contains more diverse user interests, stronger long-tail item distributions, and sparser interactions. Detailed data processing procedures are provided in Appendix A.3.

**Baselines.** We compare SAILRec with three groups of representative baselines. The first group includes collaborative filtering models (Collabs.), namely MF (Koren et al., 2009), LightGCN (He et al., 2020), and SASRec (Kang and McAuley, 2018), which learn user and item representations directly from interaction data. The second group includes LLM-only recommenders, namely TALLRec (Bao et al., 2023), GDRT (Wang et al., 2026a), and HatLLM (Cui et al., 2025). These methods rely on the semantic understanding, instruction following, and attention modeling abilities of LLMs for preference prediction. The third group includes collaborative-enhanced LLM recommenders, including CoLLM (Zhang et al., 2025), BinLLM (Zhang et al., 2024b), CoRA (Liu et al., 2025), CKF (Zhao et al., 2025), TokenRec (Qu et al., 2025), SeLLa-Rec (Wang et al., 2026b), and TCA4Rec (Lin et al., 2026). These methods inject collaborative knowledge learned from collaborative models or ID representations into LLMs to compensate for the limited modeling of user IDs and item IDs. Detailed descriptions of the baselines are provided in Appendix A.4.

**Metrics.** We evaluate model performance using *Area Under the ROC Curve* (AUC) (Fawcett, 2006), *User-level Area Under the ROC Curve* (UAUC), *Normalized Discounted Cumulative Gain* (NDCG) (Järvelin and Kekäläinen, 2002), and *Mean Average Precision* (MAP). These metrics can measure model performance from different per-

spectives (Davis and Goadrich, 2006; Wu et al., 2023; Yue et al., 2007). Detailed implementation is provided in Appendix A.5.

**Other Settings.** We use MF as the collaborative model, with the collaborative embedding dimension set to 256. The LLM backbone is Qwen2-7B (Yang et al., 2024), whose hidden size is 3584. For hierarchical attention steering, SAILRec sets the shallow-layer ratio to 30%, the deep-layer ratio to 20%, the shallow-layer bias to  $-1.0$ , the maximum deep-layer bias to  $2.0$ , and uses a triangular schedule for deep-layer bias. More details are provided in Appendix A.6.

### 4.2 Overall Performance

Table 1 reports the overall performance. SAILRec achieves the best results on both datasets across all metrics. On MovieLens-1M, it improves over the strongest baseline by 1.43, 2.77, 1.00, and 1.69 percentage points on AUC, UAUC, NDCG, and MAP, respectively. On Amazon-Book, the corresponding gains are 2.02, 1.96, 0.77, and 1.05 percentage points. These results show that SAILRec consistently outperforms conventional collaborative filtering models, LLM-only recommenders, and collaborative-enhanced LLM recommenders. The larger gains on UAUC and MAP indicate that SAILRec improves not only global positive-negative discrimination, but also user-level preference modeling and ranking quality. The consistent improvements on both MovieLens-1M and Amazon-Book further suggest that the method is effective in both relatively dense movie recommendation and sparser book recommendation scenarios. Compared with CoLLM, which directly injects unaligned collaborative embeddings, and SeLLa-Rec, which mainly performs item-side alignment, SAILRec achieves stronger results, suggesting that collaborative knowledge should be made semantically accessible on both user and item sides. Compared with HatLLM, which steers attention over textual interaction sequences in an LLM-only setting, SAILRec further shows that attention control is also effective for injected collaborative embeddings when combined with semantic alignment. Overall, the gains indicate that dual-side semantic alignment and hierarchical attention steering jointly enable more effective use of collaborative information inside LLMs.

| Method                                         | MovieLens-1M  |               |               |               | Amazon-Book   |               |               |               |
|------------------------------------------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|
|                                                | AUC           | UAUC          | NDCG          | MAP           | AUC           | UAUC          | NDCG          | MAP           |
| <i>Collaborative filtering models</i>          |               |               |               |               |               |               |               |               |
| MF                                             | 0.6429        | 0.6126        | 0.8422        | 0.7093        | 0.7098        | 0.5630        | 0.8221        | 0.7420        |
| LightGCN                                       | 0.5831        | 0.6493        | 0.8532        | 0.7320        | 0.7001        | 0.5608        | 0.8218        | 0.7415        |
| SASRec                                         | 0.7010        | 0.6742        | 0.8581        | 0.7411        | 0.6646        | 0.5629        | 0.8208        | 0.7393        |
| <i>LLM-only recommenders</i>                   |               |               |               |               |               |               |               |               |
| TALLRec                                        | 0.7039        | 0.6837        | 0.8701        | 0.7598        | 0.7198        | 0.6524        | 0.8624        | 0.7959        |
| GDRT                                           | 0.7249        | 0.6871        | 0.8689        | 0.7537        | 0.7607        | 0.6280        | 0.8505        | 0.7800        |
| HatLLM                                         | 0.7330        | 0.7058        | 0.8762        | 0.7645        | 0.7397        | 0.6147        | 0.8475        | 0.7754        |
| <i>Collaborative-enhanced LLM recommenders</i> |               |               |               |               |               |               |               |               |
| CoLLM                                          | 0.7137        | 0.6850        | 0.8757        | 0.7647        | 0.7924        | 0.7197        | 0.8896        | 0.8351        |
| BinLLM                                         | 0.7271        | 0.6980        | 0.8833        | 0.7779        | 0.8031        | 0.7234        | 0.8922        | 0.8384        |
| CoRA                                           | 0.7284        | 0.6956        | 0.8744        | 0.7614        | 0.8156        | 0.7201        | 0.8930        | 0.8393        |
| CKF                                            | 0.7320        | 0.7120        | 0.8884        | <u>0.7830</u> | 0.8048        | 0.7246        | 0.8929        | 0.8398        |
| TokenRec                                       | 0.7272        | 0.7036        | 0.8798        | 0.7702        | 0.7975        | 0.6027        | 0.8360        | 0.7603        |
| SeLLa-Rec                                      | <u>0.7428</u> | <u>0.7133</u> | <u>0.8885</u> | 0.7825        | <u>0.8191</u> | <u>0.7310</u> | <u>0.8940</u> | <u>0.8419</u> |
| TCA4Rec                                        | 0.7353        | 0.7123        | 0.8838        | 0.7733        | 0.7703        | 0.6208        | 0.8464        | 0.7747        |
| <b>Ours (SAILRec)</b>                          | <b>0.7571</b> | <b>0.7410</b> | <b>0.8985</b> | <b>0.7999</b> | <b>0.8393</b> | <b>0.7506</b> | <b>0.9017</b> | <b>0.8524</b> |

Table 1: Overall recommendation performance on MovieLens-1M and Amazon-Book. The best results are in bold, and the second-best results are underlined.

| Model                           | MovieLens-1M  |               |               | Amazon-Book   |               |               |
|---------------------------------|---------------|---------------|---------------|---------------|---------------|---------------|
|                                 | AUC           | UAUC          | NDCG          | AUC           | UAUC          | NDCG          |
| <i>Model Structure Variants</i> |               |               |               |               |               |               |
| SAIL-w/o-S                      | 0.7546        | 0.7268        | 0.8923        | 0.8220        | 0.7194        | 0.8896        |
| SAIL-w/o-UI                     | 0.7322        | 0.7236        | 0.8894        | 0.8116        | 0.7391        | 0.8989        |
| SAIL-w/o-I                      | 0.7346        | 0.7277        | 0.8971        | 0.8267        | 0.7441        | 0.8991        |
| SAIL-w/o-U                      | 0.7434        | 0.7205        | 0.8932        | 0.8288        | 0.7369        | 0.8980        |
| SAIL-w/o-C                      | 0.7475        | 0.7295        | 0.8953        | 0.8310        | 0.7367        | 0.8967        |
| <i>Tuning Method Variants</i>   |               |               |               |               |               |               |
| SAIL-LoRA                       | 0.7283        | 0.7116        | 0.8872        | 0.7649        | 0.6928        | 0.8812        |
| SAIL-LoRA-Map                   | 0.7507        | 0.7225        | 0.8980        | 0.8301        | 0.7375        | 0.8983        |
| Full-SAILRec                    | <b>0.7571</b> | <b>0.7410</b> | <b>0.8985</b> | <b>0.8393</b> | <b>0.7506</b> | <b>0.9017</b> |

Table 2: Performance comparison for SAILRec and variants. Here, S, U, I, and C denote attention steering, user-side alignment, item-side alignment, and codebooks.

### 4.3 Performance on Warm/Cold Data

Table 3 compares representative methods on warm and cold subsets. SAILRec shows stronger advantages on warm data, where richer interaction histories provide more reliable collaborative signals and allow dual-side semantic alignment and hierarchical attention steering to better exploit user-item interaction patterns. This trend is particularly evident on the warm subsets, suggesting that SAILRec benefits from sufficiently informative collaborative representations rather than relying solely on textual semantics. On cold data, SAILRec achieves strong performance on Book-Cold, whereas its advantage diminishes on Movie-Cold, where it does not outperform the strongest baselines. The re-

duced advantage under sparse interaction histories suggests that collaborative embeddings may become less informative when historical evidence is limited. These results suggest that cold-start prediction still relies heavily on textual semantics, and that collaborative enhancement should be introduced selectively rather than uniformly amplified. Overall, the warm/cold comparison highlights the importance of balancing semantic and collaborative evidence according to the reliability of available interaction signals.

### 4.4 Ablation Study

Table 2 reports the ablation results of SAILRec. SAIL-w/o-S removes the hierarchical attention steering mechanism. Its performance drop shows that aligned collaborative embeddings are not sufficient by themselves, since the LLM still benefits from explicit layer-wise control over when collaborative evidence should be emphasized. SAIL-w/o-UI removes semantic alignment on both user and item sides, while SAIL-w/o-U and SAIL-w/o-I remove user-side and item-side alignment, respectively. The degradation of these variants indicates that collaborative embeddings need to be made semantically compatible with the LLM before being injected into the prompt. The weaker performance of SAIL-w/o-U further shows the importance of user-side alignment, because anonymous user IDs do not provide linguistic cues and require explicit

| Method    | MovieLens-1M  |               |               |               | Amazon-Book   |               |               |               |
|-----------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|
|           | Cold          |               | Warm          |               | Cold          |               | Warm          |               |
|           | AUC           | UAUC          | AUC           | UAUC          | AUC           | UAUC          | AUC           | UAUC          |
| MF        | 0.5022        | 0.5241        | 0.7039        | 0.6381        | 0.5147        | 0.5097        | 0.7819        | 0.5842        |
| HatLLM    | 0.6957        | 0.6423        | 0.7401        | 0.7198        | 0.7159        | 0.6182        | 0.7405        | 0.6013        |
| CoLLM     | 0.6920        | 0.6060        | 0.7236        | 0.7012        | 0.7282        | 0.6746        | 0.8184        | 0.7375        |
| CoRA      | <b>0.7101</b> | <u>0.6542</u> | 0.7447        | <u>0.7392</u> | 0.7991        | 0.7172        | <u>0.8220</u> | <u>0.7421</u> |
| SeLLa-Rec | <u>0.7047</u> | <b>0.6724</b> | <u>0.7866</u> | 0.7330        | <u>0.8046</u> | <u>0.7282</u> | 0.8206        | 0.7410        |
| SAILRec   | 0.7038        | 0.6344        | <b>0.7900</b> | <b>0.7701</b> | <b>0.8184</b> | <b>0.7300</b> | <b>0.8449</b> | <b>0.7654</b> |

Table 3: Recommendation performance on warm and cold subsets of MovieLens-1M and Amazon-Book. The best results are in bold and the second-best results are underlined.

| Method     | MovieLens-1M  |               |               |               | Amazon-Book   |               |               |               |
|------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|
|            | AUC           | UAUC          | NDCG          | MAP           | AUC           | UAUC          | NDCG          | MAP           |
| SAIL-S-N-E | <b>0.7571</b> | <b>0.7410</b> | 0.8985        | <b>0.7999</b> | 0.8393        | <b>0.7506</b> | <b>0.9017</b> | <b>0.8524</b> |
| SAIL-S-E-N | 0.7487        | 0.7377        | 0.8948        | 0.7943        | 0.8344        | 0.7401        | 0.8983        | 0.8477        |
| SAIL-E-S-N | 0.7480        | 0.7238        | 0.8949        | 0.7933        | 0.8367        | 0.7418        | 0.8986        | 0.8482        |
| SAIL-E-N-S | 0.7461        | 0.7352        | <b>0.9008</b> | 0.7996        | <b>0.8399</b> | 0.7415        | 0.8992        | 0.8490        |
| SAIL-N-E-S | 0.7484        | 0.7335        | 0.8939        | 0.7932        | 0.8324        | 0.7386        | 0.8984        | 0.8476        |
| SAIL-N-S-E | 0.7505        | 0.7266        | 0.8954        | 0.7943        | 0.8357        | 0.7307        | 0.8948        | 0.8434        |
| SAIL-S     | 0.7503        | 0.7274        | 0.8963        | 0.7966        | 0.8281        | 0.7449        | 0.9000        | 0.8520        |
| SAIL-N     | 0.7546        | 0.7268        | 0.8923        | 0.7894        | 0.8220        | 0.7194        | 0.8896        | 0.8359        |
| SAIL-E     | 0.7449        | 0.7351        | 0.8965        | 0.7974        | 0.8290        | 0.7447        | 0.9008        | 0.8510        |

Table 4: Performance comparison of different layer-wise attention steering schedules. S, N, and E denote suppression, no steering, and enhancement, respectively.

preference anchors for grounding. SAIL-w/o-C removes the codebook-based user profiling strategy and instead uses the embedding encoded from the whole historical interaction sequence as the user-side alignment reference. Its lower performance suggests that a single sequence-level representation is less effective than structured user profiles built from semantic tags, as the codebook design offers clearer and more stable anchors for aligning user-side collaborative preferences.

For training variants, SAIL-LoRA trains only the LoRA parameters in the final stage and freezes both mapping modules, while SAIL-LoRA-Map further updates the mapping modules but does not jointly optimize them with LoRA from the beginning of the final recommendation stage. Their weaker results indicate that collaborative mappings should be adapted together with the LLM to fit the recommendation objective.

#### 4.5 Steering Strategy Analysis

To further examine how collaborative embeddings should be used across Transformer depths, we compare different layer-wise attention steering sched-

ules in Table 4. We divide the Transformer layers into shallow, middle, and deep groups, which account for 30%, 50%, and 20% of all layers, respectively, and then test different steering operations on these groups. Here, S, N, and E denote suppression, no steering, and enhancement, respectively. For example, SAIL-S-N-E applies suppression in shallow layers, no additional steering in middle layers, and enhancement in deep layers. The single-strategy variants SAIL-S, SAIL-N, and SAIL-E apply the same steering operation to all layers.

The results show that the default SAIL-S-N-E schedule achieves the most consistent performance across the two datasets. Although SAIL-E-N-S obtains slightly higher NDCG on MovieLens-1M and AUC on Amazon-Book, its UAUC is less stable, suggesting that steering strategies should consider both user-level discrimination and ranking quality. The comparison supports the layer-wise utilization pattern in our diagnostic analysis. Schedules with shallow-layer suppression generally outperform those that enhance or leave collaborative embeddings uncontrolled in early layers, indicating that collaborative knowledge should not interfere

too early with textual encoding. Comparing SAIL-S-N-E with SAIL-S-E-N shows that deep-layer enhancement is more effective than middle-layer enhancement, suggesting that collaborative evidence is more useful near the decision stage. The weaker results of SAIL-S, SAIL-N, and SAIL-E show that uniform steering is insufficient. Overall, effective collaborative enhancement requires selective layer-wise access rather than uniform attention adjustment. Appendix B.4 and Appendix B.5 further provide masking and Qwen3-8B backbone evidence supporting the effectiveness of SAILRec.

## 5 Conclusion

In this paper, we present SAILRec, an LLM-based recommender that improves the use of collaborative embeddings through dual-side semantic alignment and hierarchical attention steering. SAILRec aligns user-side and item-side collaborative signals with the LLM semantic space and regulates their influence across Transformer layers to better balance semantic and collaborative knowledge. Experiments on MovieLens-1M and Amazon-Book show consistent improvements over representative baselines, while ablations and masking analyses validate the proposed designs.

## Limitations

This work has two main limitations. First, SAILRec uses a predefined attention steering schedule, which may not be optimal for all samples or LLM backbones. Future work may develop sample-adaptive steering with continuous layer-wise bias functions, allowing customized Collaborative token biases across Transformer layers. Second, the codebook-based user profiling scheme relies on predefined semantic dimensions and requires limited domain adaptation. Although most semantic concepts were reusable when adapting the codebooks from MovieLens-1M to Amazon-Book, a small subset of domain-specific tags required removal or supplementation. Future work could explore automatic codebook construction and adaptive semantic spaces to improve cross-domain transferability. In addition, although our intervention analyses support that model predictions rely on Collaborative tokens, these continuous representations offer limited direct semantic interpretability. More interpretable collaborative tokenization and semantic grounding remain worthwhile directions.

## Acknowledgments

This work is supported by the National Natural Science Foundation of China (Project Nos. 62272092, 62676083 and 62606078), the General Project of the China Postdoctoral Science Foundation (Project No. 2026M791611), the Liaoning Provincial Doctoral Research Start-up Fund Project (Project No. 2026-BS-0112), and the Fundamental Research Funds for the Central Universities (Project No. N25XQD004).

## References

- Samira Abnar and Willem Zuidema. 2020. [Quantifying attention flow in transformers](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4190–4197, Online. Association for Computational Linguistics.
- Keqin Bao, Jizhi Zhang, Wenjie Wang, Yang Zhang, Zhengyi Yang, Yanchen Luo, Chong Chen, Fuli Feng, and Qi Tian. 2025. [A bi-step grounding paradigm for large language models in recommendation systems](#). *ACM Trans. Recomm. Syst.*, 3(4):53:1–53:27.
- Keqin Bao, Jizhi Zhang, Yang Zhang, Wenjie Wang, Fuli Feng, and Xiangnan He. 2023. [TALLRec: An effective and efficient tuning framework to align large language model with recommendation](#). In *Proceedings of the 17th ACM Conference on Recommender Systems, RecSys ’23*, page 1007–1014, New York, NY, USA. Association for Computing Machinery.
- Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, Rohan Anil, Zakaria Haque, Lichan Hong, Vihan Jain, Xiaobing Liu, and Hemal Shah. 2016. [Wide & deep learning for recommender systems](#). In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems, DLRS 2016*, page 7–10, New York, NY, USA. Association for Computing Machinery.
- Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. [What does BERT look at? An analysis of BERT’s attention](#). In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy. Association for Computational Linguistics.
- Yu Cui, Feng Liu, Jiawei Chen, Canghong Jin, Xingyu Lou, Changwang Zhang, Jun Wang, Yuegang Sun, and Can Wang. 2025. [HatLLM: Hierarchical attention masking for enhanced collaborative modeling in LLM-based recommendation](#). *Preprint*, arXiv:2510.10955.
- Jesse Davis and Mark Goadrich. 2006. [The relationship between precision-recall and ROC curves](#). In *Proceedings of the 23rd International Conference on*

- Machine Learning*, ICML '06, page 233–240, New York, NY, USA. Association for Computing Machinery.
- Yashar Deldjoo, Zhankui He, Julian McAuley, Anton Korikov, Scott Sanner, Arnau Ramisa, René Vidal, Maheswaran Sathiamoorthy, Atoosa Kasrizadeh, Silvia Milano, and Francesco Ricci. 2026. [Recommendation with generative models](#). *Foundations and Trends in Information Retrieval*, 20(1-2):1–176.
- Tom Fawcett. 2006. [An introduction to ROC analysis](#). *Pattern Recognition Letters*, 27(8):861–874. ROC Analysis in Pattern Recognition.
- Yunfan Gao, Tao Sheng, Youlin Xiang, Yun Xiong, Haofen Wang, and Jiawei Zhang. 2023. [Chat-REC: Towards interactive and explainable LLMs-augmented recommender system](#). *Preprint*, arXiv:2303.14524.
- Shijie Geng, Shuchang Liu, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. 2022. [Recommendation as language processing \(RLP\): A unified pretrain, personalized prompt & predict paradigm \(P5\)](#). In *Proceedings of the 16th ACM Conference on Recommender Systems*, RecSys '22, page 299–315, New York, NY, USA. Association for Computing Machinery.
- Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. [DeepFM: A factorization-machine based neural network for CTR prediction](#). In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 1725–1731.
- F. Maxwell Harper and Joseph A. Konstan. 2015. [The MovieLens datasets: History and context](#). *ACM Trans. Interact. Intell. Syst.*, 5(4):19:1–19:19.
- Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. [LightGCN: Simplifying and powering graph convolution network for recommendation](#). In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, page 639–648, New York, NY, USA. Association for Computing Machinery.
- Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. [LoRA: Low-rank adaptation of large language models](#). In *International Conference on Learning Representations*.
- Kalervo Järvelin and Jaana Kekäläinen. 2002. [Cumulated gain-based evaluation of IR techniques](#). *ACM Trans. Inf. Syst.*, 20(4):422–446.
- Yangqin Jiang, Yuhao Yang, Lianghao Xia, Da Luo, Kangyi Lin, and Chao Huang. 2025. [RecLM: Recommendation instruction tuning](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15443–15459, Vienna, Austria. Association for Computational Linguistics.
- Wang-Cheng Kang and Julian McAuley. 2018. [Self-attentive sequential recommendation](#). In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 197–206.
- Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. [Matrix factorization techniques for recommender systems](#). *Computer*, 42(8):30–37.
- Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. [BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models](#). In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 19730–19742. PMLR.
- Lei Li, Yongfeng Zhang, Dugang Liu, and Li Chen. 2024. [Large language models for generative recommendation: A survey and visionary discussions](#). In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 10146–10159, Torino, Italia. ELRA and ICCL.
- Zixuan Li, Binzong Geng, Jing Xiong, Yong He, Yuxuan Hu, Jian Chen, Dingwei Chen, Xiyu Chang, Ngai Wong, Liang Zhang, Linjian Mo, Chengming Li, Chuan Yuan, and Zhenan Sun. 2026. [CTR-Sink: Attention sink for language models in click-through rate prediction](#). In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2, KDD '26*, page 2979–2990, New York, NY, USA. Association for Computing Machinery.
- Jiayi Liao, Sihang Li, Zhengyi Yang, Jiancan Wu, Yancheng Yuan, Xiang Wang, and Xiangnan He. 2024. [LLaRA: Large language-recommendation assistant](#). In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, page 1785–1795, New York, NY, USA. Association for Computing Machinery.
- Fake Lin, Binbin Hu, Zhi Zheng, Xi Zhu, Ziqi Liu, Zhiqiang Zhang, Jun Zhou, and Tong Xu. 2026. [Token-level collaborative alignment for LLM-based generative recommendation](#). In *Proceedings of the ACM Web Conference 2026*, WWW '26, page 6909–6919, New York, NY, USA. Association for Computing Machinery.
- Yuting Liu, Jinghao Zhang, Yizhou Dang, Yuliang Liang, Qiang Liu, Guibing Guo, Jianzhe Zhao, and Xingwei Wang. 2025. [CoRA: collaborative information perception by large language model's weights for recommendation](#). In *Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence and Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence and Fifteenth Symposium on Educational Advances in Artificial Intelligence, AAAI'25/IAAI'25/EAAI'25*, pages 12246–12254. AAAI Press.

- Sichun Luo, Bowei He, Haohan Zhao, Wei Shao, Yanlin Qi, Yinya Huang, Aojun Zhou, Yuxuan Yao, Zongpeng Li, Yuanzhang Xiao, Mingjie Zhan, and Linqi Song. 2025. [RecRanker: Instruction tuning large language model as ranker for top-k recommendation](#). *ACM Trans. Inf. Syst.*, 43(5):113:1–113:31.
- Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. [Justifying recommendations using distantly-labeled reviews and fine-grained aspects](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 188–197, Hong Kong, China. Association for Computational Linguistics.
- Haohao Qu, Wenqi Fan, Zihuai Zhao, and Qing Li. 2025. [TokenRec: Learning to tokenize ID for LLM-based generative recommendations](#). *IEEE Transactions on Knowledge and Data Engineering*, 37(10):6216–6231.
- Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. [BERT rediscovers the classical NLP pipeline](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.
- Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. [Representation learning with contrastive predictive coding](#). *Preprint*, arXiv:1807.03748.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. [Attention is all you need](#). In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.
- Bohao Wang, Jiawei Chen, Feng Liu, Changwang Zhang, Jun Wang, Canghong Jin, Chun Chen, and Can Wang. 2026a. Does LLM focus on the right words? Mitigating context bias in LLM-based recommenders. In *Proceedings of the ACM Web Conference 2026*, WWW '26, page 6688–6699, New York, NY, USA. Association for Computing Machinery.
- Zihan Wang, Jinghao Lin, Xiaocui Yang, Yongkang Liu, Shi Feng, Daling Wang, Yifei Zhang, and Ge Yu. 2026b. [Enhancing LLM-based recommendation with semantic-aligned collaborative knowledge](#). In *Database Systems for Advanced Applications*, volume 16535 of *Lecture Notes in Computer Science*, pages 405–421, Singapore. Springer Nature Singapore.
- Le Wu, Xiangnan He, Xiang Wang, Kun Zhang, and Meng Wang. 2023. [A survey on accuracy-oriented neural recommendation: From collaborative filtering to information-rich recommendation](#). *IEEE Transactions on Knowledge and Data Engineering*, 35(5):4425–4445.
- Likang Wu, Zhi Zheng, Zhaopeng Qiu, Hao Wang, Hongchao Gu, Tingjia Shen, Chuan Qin, Chen Zhu, Hengshu Zhu, Qi Liu, Hui Xiong, and Enhong Chen. 2024. [A survey on large language models for recommendation](#). *World Wide Web*, 27(5):60.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. [Qwen3 technical report](#). *Preprint*, arXiv:2505.09388.
- An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, and 43 others. 2024. [Qwen2 technical report](#). *Preprint*, arXiv:2407.10671.
- Yisong Yue, Thomas Finley, Filip Radlinski, and Thorsten Joachims. 2007. [A support vector method for optimizing average precision](#). In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '07, page 271–278, New York, NY, USA. Association for Computing Machinery.
- Qingru Zhang, Chandan Singh, Liyuan Liu, Xiaodong Liu, Bin Yu, Jianfeng Gao, and Tuo Zhao. 2024a. [Tell your model where to attend: Post-hoc attention steering for LLMs](#). In *International Conference on Learning Representations*.
- Yang Zhang, Keqin Bao, Ming Yan, Wenjie Wang, Fuli Feng, and Xiangnan He. 2024b. [Text-like encoding of collaborative information in large language models for recommendation](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9181–9191, Bangkok, Thailand. Association for Computational Linguistics.
- Yang Zhang, Fuli Feng, Jizhi Zhang, Keqin Bao, Qifan Wang, and Xiangnan He. 2025. [CoLLM: Integrating collaborative embeddings into large language models for recommendation](#). *IEEE Trans. on Knowl. and Data Eng.*, 37(5):2329–2340.
- Chuang Zhao, Xing Su, Ming He, Hongke Zhao, Jianping Fan, and Xiaomeng Li. 2025. [Collaborative knowledge fusion: A novel method for multi-task recommender systems via LLMs](#). *IEEE Transactions on Knowledge and Data Engineering*, 37(9):5017–5033.

## A Experimental Details

### A.1 Prompt

The prompt used for supervised fine-tuning (SFT) in SAILRec training stage 3 is illustrated in Figure 3.

| Prompt Used for SFT in Training Stage 3                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p><b>System prompt:</b> A user has given high ratings to the following movies. Given the movielist, the user's feature and a target movie's feature and Title. Please use all available information to identify whether the user will like the target movie by answering 'Yes.' or 'No.'</p> <p><b>User prompt:</b> The MovieList: {Item_List_History}. The user's feature is &lt;User_ID1&gt;, &lt;User_ID2&gt;, &lt;User_ID3&gt;. The target movie's feature is &lt;Item_ID&gt;. Targetmovie: {target}.</p> <p><b>Assistant answer:</b> 'Yes.' / 'No.'</p> |

Figure 3: Prompt used for training SAILRec.

## A.2 Codebooks

To construct semantic alignment targets for user-side collaborative embeddings, we build three phrase-level semantic codebooks from complementary preference perspectives: *style/form*, *emotion/ton*e, and *theme/value orientation*. For MovieLens-1M, candidate tags are drawn from semantically meaningful words in the Qwen2-7B vocabulary, preferring single-token expressions when sufficient and multi-word phrases otherwise. Tags are screened by their semantic relation to item descriptions: broadly applicable “semantic hub” expressions with limited preference discrimination are removed, followed by manual filtering for clarity, redundancy, and domain appropriateness. The resulting three codebooks each contain 220 tags and are fixed across users and experimental runs.

For Amazon-Book, we adapt rather than rebuild the MovieLens-1M codebooks. Domain-general tags are retained, movie-specific expressions are removed, and book-specific expressions are added using the same screening procedure, resulting in 240 tags per dimension. Overall, 620 of the 660 MovieLens-1M tags are retained (93.9%), indicating substantial cross-domain reuse with limited domain-specific adaptation. More generally, transferring the codebooks follows the same retain–remove–add procedure over the target item corpus. Partial examples are shown in Figure 4 and Figure 5.

Each codebook entry is converted into a high-dimensional semantic target using the frozen Qwen2-7B backbone. Specifically, a selected tag is inserted into four slot-specific prompt templates, and each prompt is encoded by Qwen2-7B. We extract the last hidden-layer representation at the final non-padding token, producing a 3584-dimensional

| Codebook Tags (Movie)                                                                                                                                                                                                                                                                                                                                                                                                        |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p><b>Style(220):</b> serially accretive, nonlinear, episodic, fragmentary, recursive, cyclical, layered, braided, polyphonic ...</p> <p><b>Emotion(220):</b> melancholic, wistful, elegiac, tender, radiant, serene, tranquil, hushed, somber, brooding, aching ...</p> <p><b>Theme(220):</b> existential, nihilistic, humanist, utopian, dystopian, posthuman, secular, sacral, spiritual, materialist, idealistic ...</p> |

Figure 4: Examples of phrase-level codebook tags for MovieLens-1M. The codebooks cover style, emotion, and theme, with 220 tags in each semantic dimension.

| Codebook Tags (Book)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p><b>Style(240):</b> character-forward, voice-led, argument-driven, parable-like, fable-like, diary-like, journal-like, chapter mosaic, story cycle, document bundle ...</p> <p><b>Emotion(240):</b> simmering tension, hushed sorrow, private shame, aching nostalgia, gentle amusement, tender urgency, survival anxiety, restorative calm, fragile hope ...</p> <p><b>Theme(240):</b> social mobility, private faith, public duty, moral education, identity formation, work and dignity, personal transformation ...</p> |

Figure 5: Examples of phrase-level codebook tags for Amazon-Book. The book-domain codebooks are adapted from the movie-domain codebooks and expanded to 240 tags for each semantic dimension.

vector for each prompt. The four vectors are averaged and then  $\ell_2$ -normalized to obtain the semantic target used by the user-side C-QFormer. Thus, the alignment target is not a separately learned embedding table, but a fixed semantic representation produced by the same frozen language-model space used throughout target construction.

To examine whether the alignment benefit can be obtained from arbitrary semantic targets, we construct two control variants on MovieLens-1M. **RandomTag** randomly assigns targets from the same three codebooks while breaking the correspondence between a user’s interaction history and its semantic profile, thereby preserving the target space and alignment architecture. **RandomText** instead uses randomly sampled meaningful textual tokens from the Qwen2-7B vocabulary as alignment targets. The results are reported in Table 5.

RandomText performs substantially worse than the user-specific targets, showing that aligning

| Target               | AUC           | UAUC          | NDCG          | MAP           |
|----------------------|---------------|---------------|---------------|---------------|
| RandomText           | 0.7335        | 0.7080        | 0.8868        | 0.7813        |
| RandomTag            | 0.7484        | 0.7194        | 0.8942        | 0.7911        |
| <b>User-specific</b> | <b>0.7571</b> | <b>0.7410</b> | <b>0.8985</b> | <b>0.7999</b> |

Table 5: Performance of different semantic alignment targets on MovieLens-1M. RandomText uses randomly sampled meaningful textual targets, while RandomTag samples from the same codebooks but breaks the user-target correspondence. User-specific denotes the semantic targets used by SAILRec.

collaborative embeddings to arbitrary meaningful locations in the LLM representation space is insufficient. RandomTag performs better than RandomText because its targets remain within the same recommendation-oriented semantic codebooks, but it still underperforms the user-specific targets across all four metrics. In particular, UAUC decreases from 0.7410 to 0.7194 under RandomTag and to 0.7080 under RandomText. These results indicate that both the semantic structure of the target space and the correspondence between user behavior and the selected targets matter for alignment. They do not, however, imply that the three adopted semantic dimensions are uniquely optimal; rather, they show that arbitrary target matching cannot explain the gains of SAILRec.

### A.3 Datasets

MovieLens-1M (Harper and Konstan, 2015) is a movie rating dataset released by GroupLens and is one of the most widely used public benchmarks in recommender system research. It contains about one million user-movie ratings collected between 2000 and 2003, involving about 6K users and 4K movies. Each rating ranges from 1 to 5. Following CoLLM and SeLLa-Rec, we convert ratings into binary labels with a threshold of 3, where ratings greater than 3 are treated as positive interactions and the remaining ratings are treated as negative interactions. We keep the interactions from the last 20 months and split them chronologically into training, validation, and test sets with a ratio of 10:5:5. This temporal split evaluates recommendation performance under a realistic setting where future interactions are predicted from earlier histories.

Amazon-Book (Ni et al., 2019) is derived from the book category of Amazon Product Reviews. The original review data contain user ratings, review texts, timestamps, and product metadata,

| Dataset      | #Train  | #Valid | #Test  | #User  | #Item  |
|--------------|---------|--------|--------|--------|--------|
| MovieLens-1M | 33,891  | 10,401 | 7,331  | 839    | 3,256  |
| Amazon-Book  | 727,468 | 25,747 | 25,747 | 22,967 | 34,154 |

Table 6: Statistics of datasets.

with ratings ranging from 1 to 5. Compared with MovieLens-1M, Amazon-Book contains more users and items, sparser interactions, stronger long-tail distributions, and richer semantic information from book titles and categories. Following prior settings, we use interactions from 2017 and keep users with more than 20 interactions to ensure data quality. Review scores are converted into binary labels with a threshold of 4. The data are then sorted by timestamp and split chronologically into training, validation, and test sets with a ratio of 11:0.5:0.5.

After preprocessing, both datasets are organized into click-through rate prediction samples, where each instance consists of a user historical interaction sequence, a candidate item, and a binary label. The processed statistics are shown in Table 6.

### A.4 Baselines

We compare three groups of representative baselines.

**Collaborative filtering models.** MF (Koren et al., 2009) learns low-dimensional user and item representations through matrix factorization and estimates preference by their latent interaction. LightGCN (He et al., 2020) models high-order collaborative signals by neighborhood aggregation over the user-item graph while removing feature transformation and nonlinear activation. SASRec (Kang and McAuley, 2018) applies self-attention to historical interaction sequences to capture dynamic user preferences for sequential recommendation.

**LLM-only recommenders.** TALLRec (Bao et al., 2023) converts recommendation samples into instruction data and instruction-tunes LLMs for direct preference prediction. GDRT (Wang et al., 2026a) addresses context bias by applying Group DRO over token-relevance groups, encouraging the model to rely more robustly on informative interaction signals. HatLLM (Cui et al., 2025) introduces hierarchical attention masking, emphasizing intra-item semantic modeling in shallow layers and cross-item relations in deeper Transformer layers.

**Collaborative-enhanced LLM recommenders.** CoLLM (Zhang et al., 2025) maps user and item embeddings from an external collaborative model

into the LLM input space as soft prompts. BinLLM (Zhang et al., 2024b) represents collaborative information as text-like binary sequences that can be directly processed by LLMs. CoRA (Liu et al., 2025) injects collaborative signals into the parameter space through low-rank incremental weights rather than additional input tokens. CKF (Zhao et al., 2025) combines personalized collaborative mapping with Multi-LoRA to separate task-shared and task-specific knowledge. TokenRec (Qu et al., 2025) quantizes collaborative representations into discrete ID tokens and performs generative retrieval. SeLLa-Rec (Wang et al., 2026b) reduces the semantic gap between collaborative and LLM representations through semantic distillation, contrastive alignment, and hybrid projection. TCA4Rec (Lin et al., 2026) converts item-level collaborative logits into token-level distributions and uses them as soft supervision for generative recommendation.

### A.5 Evaluation Metrics

Under the click-through rate prediction setting, we use the predicted positive-class probability as the candidate item score. Specifically, for each user-item pair  $(u, i)$ , the LLM produces logits for Yes and No at the answer position. Let  $o_{\text{Yes}}$  and  $o_{\text{No}}$  denote the corresponding logits. The prediction score is computed by a binary softmax:

$$s_{u,i} = \frac{\exp(o_{\text{Yes}})}{\exp(o_{\text{Yes}}) + \exp(o_{\text{No}})}. \quad (8)$$

This score is used for all evaluation metrics. In implementation, the logits of Yes and No are first extracted from the vocabulary distribution, normalized with a binary softmax, and the probability of Yes is used as the positive-class score.

AUC evaluates the overall discrimination ability between positive and negative instances over all test samples. Let  $\mathcal{D}$  denote the test set, and let  $\mathcal{D}^+$  and  $\mathcal{D}^-$  denote the positive and negative subsets, respectively. AUC is computed as

$$\text{AUC} = \frac{1}{|\mathcal{D}^+||\mathcal{D}^-|} \sum_{p \in \mathcal{D}^+} \sum_{n \in \mathcal{D}^-} \psi(s_p, s_n), \quad (9)$$

where

$$\psi(a, b) = \begin{cases} 1, & a > b, \\ \frac{1}{2}, & a = b, \\ 0, & a < b. \end{cases}$$

Here,  $\psi(a, b)$  assigns a score of 1 when a positive instance receives a higher prediction score than a

negative instance,  $\frac{1}{2}$  when the two scores are tied, and 0 otherwise. In our implementation, global AUC is computed with `roc_auc_score` over all test labels and prediction scores.

UAUC evaluates ranking performance at the user level. For each user  $u$ , we collect the candidate set  $\mathcal{D}_u = \{(i, y_{u,i}, s_{u,i})\}$  from the test set and compute AUC within this user. Users with only one candidate sample or with only one label class are excluded, because AUC is undefined in these cases. Let  $\mathcal{U}_{\text{auc}}$  denote the remaining valid users. UAUC is then computed as the unweighted mean of per-user AUC:

$$\text{UAUC} = \frac{1}{|\mathcal{U}_{\text{auc}}|} \sum_{u \in \mathcal{U}_{\text{auc}}} \text{AUC}_u. \quad (10)$$

Different from impression-weighted user AUC, each valid user contributes equally in this metric. Therefore, UAUC reduces the influence of user activity imbalance and better reflects personalized discrimination ability across users.

NDCG evaluates whether positive items are ranked near the top of each user’s candidate list. For each valid user  $u$ , candidate items are sorted in descending order according to  $s_{u,i}$ . Let  $y_{u,r} \in \{0, 1\}$  denote the binary relevance label at rank  $r$  after sorting, and let  $n_u$  be the number of candidates for user  $u$ . The discounted cumulative gain is

$$\text{DCG}_u = \sum_{r=1}^{n_u} \frac{y_{u,r}}{\log_2(r+1)}. \quad (11)$$

Let  $P_u = \sum_{r=1}^{n_u} y_{u,r}$  denote the number of positive items for user  $u$ . The ideal discounted cumulative gain is

$$\text{IDCG}_u = \sum_{r=1}^{P_u} \frac{1}{\log_2(r+1)}. \quad (12)$$

The user-level NDCG is then

$$\text{NDCG}_u = \frac{\text{DCG}_u}{\text{IDCG}_u}. \quad (13)$$

For consistency with the user-level evaluation protocol, we restrict NDCG and MAP to users containing both positive and negative samples, thereby excluding degenerate ranking cases. The final NDCG is computed as the unweighted mean over all valid users:

$$\text{NDCG} = \frac{1}{|\mathcal{U}_{\text{rank}}|} \sum_{u \in \mathcal{U}_{\text{rank}}} \text{NDCG}_u, \quad (14)$$

where  $\mathcal{U}_{\text{rank}}$  denotes the set of users used for ranking evaluation.

MAP measures the average precision of positive items in the ranked candidate list. For each valid user  $u$ , after sorting candidate items by prediction score, precision at rank  $r$  is defined as

$$\text{Prec}_u(r) = \frac{1}{r} \sum_{t=1}^r y_{u,t}. \quad (15)$$

The average precision for user  $u$  is

$$\text{AP}_u = \frac{1}{P_u} \sum_{r=1}^{n_u} y_{u,r} \text{Prec}_u(r), \quad (16)$$

where  $P_u$  is the number of positive items for this user. The final MAP is the unweighted mean of user-level average precision:

$$\text{MAP} = \frac{1}{|\mathcal{U}_{\text{rank}}|} \sum_{u \in \mathcal{U}_{\text{rank}}} \text{AP}_u. \quad (17)$$

Thus, AUC measures global discrimination over all samples, while UAUC, NDCG, and MAP evaluate user-level personalized ranking quality by first computing metrics within each user and then averaging over valid users.

## A.6 Training and Hyperparameter Details

SAILRec uses Qwen2-7B as the backbone LLM. For fair comparison, all LLM-based baseline methods are also evaluated with Qwen2-7B as the backbone. In the second training stage, SAILRec performs dual-side semantic alignment with contrastive learning. The learning rate is set to  $1 \times 10^{-3}$ , the batch size is set to 128, and the temperature in the InfoNCE loss is set to 0.07.

In the third training stage, SAILRec performs supervised fine-tuning (SFT) for click-through rate prediction. The learning rate is selected from the range between  $8 \times 10^{-5}$  and  $2 \times 10^{-4}$ . We set the weight decay to 0.01,  $\beta_2$  in Adam to 0.95, the warmup ratio to 0, and the maximum sequence length to 768. For MovieLens-1M, the batch size is set to 256 and the model is trained for 2 epochs. For Amazon-Book, the batch size is set to 2000 and the model is trained for 1 epoch. All experiments are conducted on NVIDIA A6000 GPUs. The total training time is about 7 hours on MovieLens-1M and about 92 hours on Amazon-Book. A stage-wise training-cost comparison with CoLLM and SeLLa-Rec on Amazon-Book is reported in Table 7. For each run, checkpoint selection is based

solely on overall validation performance. For statistical reliability, SAILRec and SeLLa-Rec are additionally evaluated over three matched random seeds in Appendix B.1.

| Method    | Pretraining<br>/ Alignment | Rec.<br>/ Mapping | Total  |
|-----------|----------------------------|-------------------|--------|
| CoLLM     | ~90.9 h                    | ~92 h             | ~183 h |
| SeLLa-Rec | ~90.9 h                    | ~92 h             | ~183 h |
| SAILRec   | <20 min                    | ~92 h             | ~92 h  |

Table 7: Stage-wise training cost on Amazon-Book using an NVIDIA A6000 GPU. For SAILRec, the alignment cost includes both user-side and item-side Stage-2 alignment, while the recommendation stage jointly optimizes LoRA and the C-QFormers. Shared MF pretraining is excluded for all methods.

## B Additional Experimental Results and Analysis

### B.1 Statistical Significance

To assess statistical reliability, we rerun SAILRec and the strongest baseline, SeLLa-Rec, using three matched random seeds and the same validation-based checkpoint-selection protocol. We report the mean and sample standard deviation across runs and conduct two-sided paired  $t$ -tests between the matched runs of the two methods.

SAILRec consistently outperforms SeLLa-Rec across all metrics on both datasets, with all paired tests yielding  $p < 0.05$ . The relatively small standard deviations further indicate that the improvements are stable across the evaluated random seeds.

### B.2 Additional Analysis of Collab. tokens

We further analyze how SAILRec uses collaborative embeddings inside the LLM. Figure 6 shows the layer-wise attention patterns on MovieLens-1M. The attention to Collab. tokens remains low in shallow layers, which is consistent with our design to avoid premature interference with the understanding of historical item tokens. It then increases in middle and deep layers, indicating that collaborative evidence is gradually introduced near the decision stage. Meanwhile, the attention to Collab. tokens does not exceed that to historical item-title tokens, suggesting that SAILRec enhances collaborative evidence without overusing it. For the three user-side Collab. tokens, their attention curves follow similar depth-wise trends, which also matches the hierarchical steering design.

| Dataset      | Method          | AUC                                 | UAUC                                | NDCG                                | MAP                                 |
|--------------|-----------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|
| MovieLens-1M | SeLLa-Rec       | 0.7453 $\pm$ 0.0031                 | 0.7109 $\pm$ 0.0065                 | 0.8875 $\pm$ 0.0022                 | 0.7827 $\pm$ 0.0025                 |
|              | SAILRec         | <b>0.7575<math>\pm</math>0.0012</b> | <b>0.7403<math>\pm</math>0.0012</b> | <b>0.8993<math>\pm</math>0.0012</b> | <b>0.7986<math>\pm</math>0.0015</b> |
|              | <i>p</i> -value | 0.0300                              | 0.0111                              | 0.0085                              | 0.0175                              |
| Amazon-Book  | SeLLa-Rec       | 0.8181 $\pm$ 0.0020                 | 0.7320 $\pm$ 0.0010                 | 0.8941 $\pm$ 0.0001                 | 0.8420 $\pm$ 0.0002                 |
|              | SAILRec         | <b>0.8394<math>\pm</math>0.0024</b> | <b>0.7515<math>\pm</math>0.0011</b> | <b>0.9029<math>\pm</math>0.0012</b> | <b>0.8538<math>\pm</math>0.0014</b> |
|              | <i>p</i> -value | 0.0008                              | 0.00006                             | 0.0062                              | 0.0044                              |

Table 8: Statistical reliability over three matched random seeds. Results are reported as mean  $\pm$  sample standard deviation. The *p*-values are obtained from two-sided paired *t*-tests between SAILRec and SeLLa-Rec.

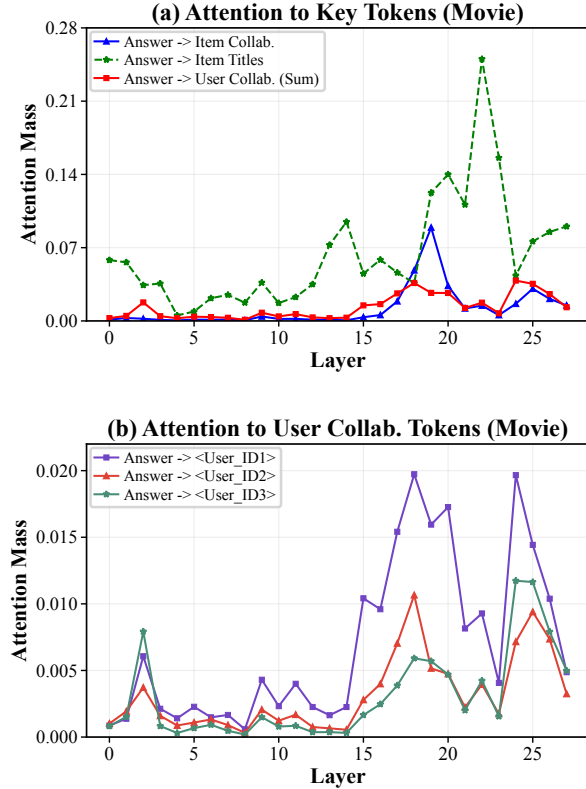


Figure 6: Layer-wise answer attention to semantic and collaborative tokens in SAILRec on MovieLens-1M, with user-side attention further decomposed into three embeddings.

Figure 7 reports the layer-wise attention patterns on Amazon-Book. Similar to MovieLens-1M, the attention to Collab. tokens stays low in shallow layers and becomes more active in middle-to-deep layers. This pattern suggests SAILRec does not force collaborative evidence to dominate the whole inference process, but introduces it mainly when preference matching becomes more relevant. A difference is that Amazon-Book shows sharper attention peaks around the decision-related layers, especially for summed user-side Collab. tokens. This may be related to the stronger sparsity and semantic diversity of book interactions, where user-side pref-

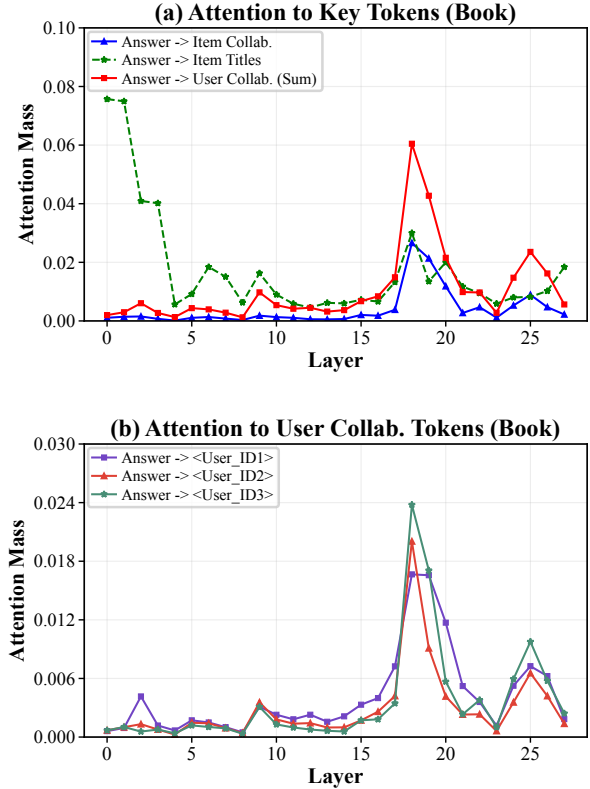


Figure 7: Layer-wise answer attention to semantic and collaborative tokens in SAILRec on Amazon-Book. The upper panel compares attention to item titles, item-side Collab. tokens, and summed user-side Collab. tokens, while the lower panel decomposes the attention to the three user-side Collab. tokens.

erence abstraction becomes more important than relying only on item titles. In contrast, MovieLens-1M exhibits stronger and more persistent attention to historical item-title tokens, which is reasonable because movie titles and genres often provide more compact semantic cues. In both datasets, the three user-side Collab. tokens follow similar depth-wise trends, but their magnitudes differ, suggesting that they participate in the same inference stage while capturing complementary preference semantics.

Figure 8 further visualizes the learned user-side

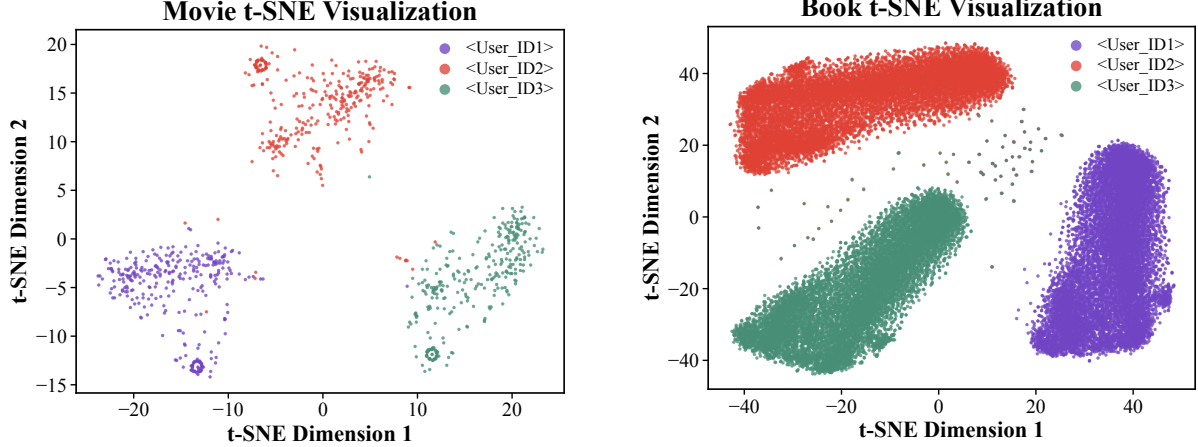


Figure 8:  $t$ -SNE visualization of the three user-side Collab. tokens on MovieLens-1M and Amazon-Book. Different colors denote the three semantic directions learned by the user-side C-QFormer.

| User-side Alignment |        |         |               |
|---------------------|--------|---------|---------------|
| Stage               | MRR    | R@1 (%) | Slot Acc. (%) |
| Stage 2             | 0.8759 | 81.73   | 79.42         |
| Stage 3             | 0.5994 | 51.08   | 69.42         |
| Chance              | –      | 0.42    | 33.33         |

| Item-side Alignment |        |          |          |
|---------------------|--------|----------|----------|
| Stage               | MRR    | R@10 (%) | R@50 (%) |
| Stage 2             | 0.0311 | 6.98     | 17.50    |
| Stage 3             | 0.1562 | 32.50    | 57.15    |
| Chance              | –      | 0.029    | –        |

Table 9: Quantitative evaluation of semantic alignment on Amazon-Book. Stage 2 denotes the representations after semantic alignment, while Stage 3 denotes those after subsequent recommendation tuning.

Collab. tokens with  $t$ -SNE. The visualization suggests that the three user-side Collab. tokens occupy distinguishable regions in the projected space on both datasets. This pattern indicates that the user-side C-QFormer does not simply generate identical or fully redundant representations for the three placeholders. Instead, the mapped embeddings tend to preserve different representation patterns that correspond to the three semantic directions used in user-side alignment. The separation is more compact on Amazon-Book, which may be related to its larger data scale and richer phrase-level preference space.

Table 9 quantitatively examines how the learned Collab. tokens correspond to their semantic alignment targets across training stages. After Stage 2, the user-side tokens exhibit strong semantic correspondence, achieving 81.73% R@1 compared with

a 0.42% random retrieval level and 79.42% slot accuracy compared with a 33.33% chance level. After Stage 3 recommendation tuning, user-side correspondence decreases but remains substantially above chance, with 51.08% R@1 and an MRR of 0.5994. In contrast, item-side correspondence becomes stronger during Stage 3, with item-title R@10 increasing from 6.98% to 32.50%, far above the 0.029% random retrieval level. These results suggest that semantic alignment primarily provides a meaningful initialization for collaborative representations, which are then adapted to the downstream recommendation objective rather than being constrained to maintain fixed final-state cosine correspondence. Therefore, the strength of final representation similarity should not be interpreted as a monotonic indicator of recommendation performance.

### B.3 Sensitivity of Hierarchical Attention Steering

We further examine the sensitivity of hierarchical attention steering to the bias magnitude and the partition of Transformer layers. Unless otherwise specified, all experimental settings remain identical to the default SAILRec configuration. We use S, N, and E to denote the shallow-suppression, neutral/no-steering, and deep-enhancement stages, respectively.

**Sensitivity to steering bias.** We first vary the magnitude of the attention bias while keeping the default layer partition fixed at 30% shallow, 50% neutral, and 20% deep layers. The notation  $SxEy$  denotes a shallow-layer bias of  $-x$  and a maximum

| Setting     | MovieLens-1M  |               |               |               | Amazon-Book   |               |               |               |
|-------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|
|             | AUC           | UAUC          | NDCG          | MAP           | AUC           | UAUC          | NDCG          | MAP           |
| S0E2        | 0.7562        | 0.7236        | 0.8920        | 0.7907        | 0.8230        | 0.7394        | 0.8970        | 0.8459        |
| S1E0        | 0.7557        | 0.7324        | 0.8929        | 0.7917        | 0.8328        | 0.7402        | 0.8943        | 0.8486        |
| <b>S1E2</b> | <b>0.7571</b> | <b>0.7410</b> | <b>0.8985</b> | <b>0.7999</b> | <b>0.8393</b> | <b>0.7506</b> | <b>0.9017</b> | <b>0.8524</b> |
| S2E2        | 0.7550        | 0.7352        | 0.8944        | 0.7932        | 0.8301        | 0.7459        | 0.8999        | 0.8501        |
| S1E3        | 0.7558        | 0.7288        | 0.8915        | 0.7892        | 0.8290        | 0.7366        | 0.8936        | 0.8433        |

Table 10: Sensitivity to the steering-bias magnitude.  $SxEy$  denotes a shallow-layer bias of  $-x$  and a maximum deep-layer bias of  $+y$ . The layer partition is fixed at 30% shallow, 50% neutral, and 20% deep layers.

| Setting            | MovieLens-1M  |               |               |               | Amazon-Book   |               |               |               |
|--------------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|
|                    | AUC           | UAUC          | NDCG          | MAP           | AUC           | UAUC          | NDCG          | MAP           |
| S0-N80-E20         | 0.7562        | 0.7236        | 0.8920        | 0.7907        | 0.8290        | 0.7399        | 0.8996        | 0.8493        |
| <b>S30-N50-E20</b> | <b>0.7571</b> | <b>0.7410</b> | <b>0.8985</b> | <b>0.7999</b> | <b>0.8393</b> | <b>0.7506</b> | <b>0.9017</b> | <b>0.8524</b> |
| S60-N20-E20        | 0.7558        | 0.7288        | 0.8915        | 0.7892        | 0.8371        | 0.7416        | 0.8989        | 0.8483        |
| S80-N0-E20         | 0.7561        | 0.7343        | 0.8919        | 0.7928        | 0.8390        | 0.7400        | 0.8989        | 0.8488        |
| S30-N70-E0         | 0.7557        | 0.7324        | 0.8929        | 0.7917        | 0.8377        | 0.7479        | 0.9007        | 0.8509        |
| S30-N20-E50        | 0.7559        | 0.7304        | 0.8938        | 0.7923        | 0.8346        | 0.7421        | 0.8981        | 0.8471        |
| S30-N0-E70         | 0.7549        | 0.7212        | 0.8904        | 0.7871        | 0.8357        | 0.7454        | 0.9000        | 0.8497        |

Table 11: Sensitivity to the shallow-neutral-deep layer partition.  $Sx-Ny-Ez$  denotes the percentage of Transformer layers assigned to each stage. The steering biases are fixed to the default setting of  $-1$  for shallow layers and a maximum of  $+2$  for deep layers.

deep-layer bias of  $+y$ , while the neutral layers receive no additional bias. The results are shown in Table 10.

The default S1E2 configuration provides the most consistent performance across both datasets and all metrics. Removing shallow suppression (S0E2) mainly degrades UAUC and ranking metrics, indicating that unrestricted access to collaborative tokens in early layers can interfere with semantic representation formation. Removing deep enhancement (S1E0) also causes consistent degradation, suggesting that stronger access to collaborative evidence in later layers is beneficial. Meanwhile, stronger suppression (S2E2) or enhancement (S1E3) does not yield further gains. These results indicate that hierarchical steering benefits from a moderate bias magnitude rather than monotonically stronger attention intervention.

**Sensitivity to layer partition.** We next vary the proportions of shallow, neutral, and deep layers while fixing the steering biases to the default configuration, i.e., a shallow bias of  $-1$  and a maximum deep bias of  $+2$ . The notation  $Sx-Ny-Ez$  denotes that  $x\%$ ,  $y\%$ , and  $z\%$  of Transformer layers are assigned to the shallow, neutral, and deep stages, respectively. Table 11 reports the results.

The 30%-50%-20% partition achieves the most consistent performance across the two datasets.

| Setting           | AUC           | UAUC          | NDCG          | MAP           |
|-------------------|---------------|---------------|---------------|---------------|
| Learnable Bias    | 0.7531        | 0.7248        | 0.8948        | 0.7935        |
| <b>Fixed S1E2</b> | <b>0.7571</b> | <b>0.7410</b> | <b>0.8985</b> | <b>0.7999</b> |

Table 12: Comparison between learnable and fixed attention biases on MovieLens-1M.

Eliminating the shallow stage (S0-N80-E20) reduces performance, particularly on UAUC and MAP, while extending shallow suppression to 60% or 80% of the network also becomes less effective. Similarly, removing the deep enhancement stage (S30-N70-E0) causes a consistent decline, whereas assigning 50% or 70% of the layers to deep enhancement does not improve performance. These observations suggest that effective steering depends on preserving sufficient unmodified intermediate layers between early suppression and late enhancement. Overall, the sensitivity results support the adopted 30%-50%-20% partition and S1E2 bias as a robust configuration across the evaluated settings, rather than implying that they are universally optimal.

**Learnable attention bias.** We further compare the fixed steering bias with a naive learnable variant, where the attention biases toward Collab. tokens are directly optimized together with the recommendation objective without imposing the pre-

| Method              | MovieLens-1M  |               |               |               | Amazon-Book   |               |               |               |
|---------------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|
|                     | AUC           | UAUC          | NDCG          | MAP           | AUC           | UAUC          | NDCG          | MAP           |
| SAIL-mask-U         | 0.7248        | 0.7008        | 0.8797        | 0.7670        | 0.5553        | 0.5694        | 0.8281        | 0.7477        |
| SAIL-mask-I         | 0.7074        | 0.6613        | 0.8637        | 0.7415        | 0.8168        | 0.7136        | 0.8866        | 0.8306        |
| SAIL-mask-UI        | 0.6941        | 0.6758        | 0.8687        | 0.7494        | 0.5361        | 0.5452        | 0.8178        | 0.7337        |
| SAIL-mask-Shallow   | 0.6750        | 0.6364        | 0.8483        | 0.7198        | 0.6944        | 0.6532        | 0.8594        | 0.7921        |
| SAIL-mask-Middle    | 0.7166        | 0.6700        | 0.8601        | 0.7397        | 0.7178        | 0.6600        | 0.8662        | 0.8012        |
| SAIL-mask-Deep      | 0.7541        | 0.6840        | 0.8689        | 0.7569        | 0.8247        | 0.7374        | 0.8954        | 0.8431        |
| <b>Full-SAILRec</b> | <b>0.7571</b> | <b>0.7410</b> | <b>0.8985</b> | <b>0.7999</b> | <b>0.8393</b> | <b>0.7506</b> | <b>0.9017</b> | <b>0.8524</b> |

Table 13: Inference-time hard attention masking results. SAIL-mask-U, SAIL-mask-I, and SAIL-mask-UI block attention to user-side, item-side, and both-side Collab. tokens, respectively. The remaining variants block attention to all Collab. tokens in the corresponding layer groups.

| Path         | $\lambda$ | AUC           | UAUC          | NDCG          | MAP           |
|--------------|-----------|---------------|---------------|---------------|---------------|
| User         | -1        | 0.8348        | 0.7377        | 0.8955        | 0.8438        |
|              | -2        | 0.8039        | 0.7195        | 0.8887        | 0.8339        |
|              | -5        | 0.6080        | 0.5839        | 0.8356        | 0.7581        |
|              | -8        | 0.6003        | 0.5827        | 0.8342        | 0.7561        |
|              | -10       | 0.5547        | 0.5737        | 0.8291        | 0.7492        |
| Item         | -1        | 0.8379        | 0.7392        | 0.8963        | 0.8452        |
|              | -2        | 0.8363        | 0.7292        | 0.8926        | 0.8397        |
|              | -5        | 0.8341        | 0.7241        | 0.8904        | 0.8367        |
|              | -8        | 0.8337        | 0.7225        | 0.8908        | 0.8371        |
|              | -10       | 0.8342        | 0.7220        | 0.8897        | 0.8359        |
| User+Item    | -1        | 0.8323        | 0.7304        | 0.8918        | 0.8391        |
|              | -2        | 0.7934        | 0.6959        | 0.8782        | 0.8188        |
|              | -5        | 0.5937        | 0.5671        | 0.8282        | 0.7477        |
|              | -8        | 0.5837        | 0.5636        | 0.8260        | 0.7449        |
|              | -10       | 0.5843        | 0.5587        | 0.8255        | 0.7439        |
| Full-SAILRec | -         | <b>0.8393</b> | <b>0.7506</b> | <b>0.9017</b> | <b>0.8524</b> |

Table 14: Finite-bias intervention on user-side and item-side Collab. token paths on Amazon-Book.  $\lambda$  denotes an additional negative attention bias applied at inference time.

| Layers       | $\lambda$ | AUC           | UAUC          | NDCG          | MAP           |
|--------------|-----------|---------------|---------------|---------------|---------------|
| Shallow      | -1        | 0.8344        | 0.7386        | 0.8965        | 0.8451        |
|              | -2        | 0.8240        | 0.7217        | 0.8896        | 0.8352        |
|              | -5        | 0.7743        | 0.6908        | 0.8774        | 0.8171        |
|              | -8        | 0.7489        | 0.6789        | 0.8726        | 0.8106        |
|              | -10       | 0.7456        | 0.6757        | 0.8700        | 0.8073        |
| Middle       | -1        | 0.8390        | 0.7416        | 0.8980        | 0.8473        |
|              | -2        | 0.8332        | 0.7336        | 0.8942        | 0.8426        |
|              | -5        | 0.7645        | 0.6815        | 0.8737        | 0.8128        |
|              | -8        | 0.7404        | 0.6626        | 0.8671        | 0.8031        |
|              | -10       | 0.7399        | 0.6700        | 0.8694        | 0.8063        |
| Deep         | -1        | 0.8376        | 0.7451        | 0.8986        | 0.8485        |
|              | -2        | 0.8377        | 0.7476        | 0.9000        | 0.8503        |
|              | -5        | 0.8377        | 0.7467        | 0.8991        | 0.8491        |
|              | -8        | 0.8378        | 0.7436        | 0.8981        | 0.8476        |
|              | -10       | 0.8377        | 0.7470        | 0.9000        | 0.8504        |
| Full-SAILRec | -         | <b>0.8393</b> | <b>0.7506</b> | <b>0.9017</b> | <b>0.8524</b> |

Table 15: Finite-bias intervention on different Transformer layer groups on Amazon-Book. The additional bias is applied to both user-side and item-side Collab. token paths within the selected layers.

defined S1E2 magnitude. As shown in Table 12, the learnable variant underperforms the fixed S1E2 configuration on all four metrics, with a particularly clear decrease in UAUC. This suggests that unconstrained bias learning does not automatically recover an effective depth-wise steering pattern, and that the fixed hierarchical bias provides a useful structural prior for optimization.

#### B.4 Inference-time Attention Bias Masking

To examine whether the model relies on the injected collaborative information, we first perform inference-time attention masking while keeping all Full-SAILRec parameters fixed. For each variant, the attention logits corresponding to selected Collab. token paths are set to  $-\infty$  during the forward

pass. As shown in Table 13, all variants underperform Full-SAILRec. However, hard masking completely removes an attention path that remains available during training and may therefore introduce a substantial distribution shift. We consequently interpret these results as a stress test of model reliance rather than as an unbiased estimate of the causal importance of each attention path.

To examine whether the degradation is specific to this extreme intervention, we additionally replace the  $-\infty$  mask with finite negative biases on Amazon-Book, where the hard-masking effects are particularly pronounced. Specifically, an additional bias  $\lambda \in \{-1, -2, -5, -8, -10\}$  is applied to the selected attention logits without retraining the model. A more negative  $\lambda$  progressively re-

duces, rather than abruptly removes, access to the corresponding Collab. tokens.

Table 14 reveals a clear graded response rather than a degradation that appears only under hard masking. Suppressing user-side attention progressively reduces performance, with AUC decreasing from 0.8393 to 0.8039 at  $\lambda = -2$  and to 0.6080 at  $\lambda = -5$ . Simultaneously suppressing both sides produces a similar trend, reaching 0.7934 and 0.5937 at the same bias levels. The degradation largely saturates after  $\lambda = -5$ , suggesting that sufficiently strong finite suppression already removes most of the usable direct access to these paths. The additional decline under the  $-\infty$  intervention, e.g., from approximately 0.584 at  $\lambda = -10$  to 0.5361 for SAIL-mask-UI, also suggests that the hard-mask result partly reflects the more extreme intervention.

Item-side suppression produces a smaller AUC change on Amazon-Book, but generally degrades UAUC and ranking quality as the negative bias increases. For example, UAUC decreases from 0.7506 to 0.7292 at  $\lambda = -2$  and to 0.7220 at  $\lambda = -10$ . Thus, the two sides exhibit asymmetric but complementary roles: user-side collaborative information is particularly important for preference discrimination in this dataset, whereas item-side information still contributes to personalized ranking.

We further apply finite negative biases separately to the shallow, middle, and deep layer groups. Here, the additional bias is imposed on both user-side and item-side Collab. token paths within the selected layer group, while all other attention paths remain unchanged.

Table 15 shows markedly different sensitivity across Transformer depth. Shallow- and middle-layer suppression produces clear dose-dependent degradation. At  $\lambda = -10$ , AUC decreases to 0.7456 and 0.7399, respectively, compared with 0.8393 for Full-SAILRec. In contrast, finite suppression of direct deep-layer access has a substantially smaller effect: even at  $\lambda = -10$ , AUC remains 0.8377, while the other metrics also remain close to the full model.

This result does not imply that deep-layer steering is unnecessary. The sensitivity experiments in Appendix B.3 show that removing deep enhancement during model construction degrades recommendation performance. Instead, the inference-time results suggest a distinction between *steering information integration* and *direct token access at the final model state*. Collaborative information

introduced earlier can propagate through ordinary hidden states and residual connections, making prediction less dependent on continuously attending to the original Collab. tokens in deep layers. Consequently, a large deep-layer attention value should not be interpreted as a proportional measure of causal importance.

Overall, the finite-bias results refine the interpretation of Table 13. The consistent graded degradation under finite interventions supports the conclusion that the model relies on collaborative attention paths and that the effects are not solely artifacts of the  $-\infty$  mask. At the same time, the additional degradation caused by hard masking suggests that its magnitude can overstate the functional importance of a path. We therefore use attention visualization to characterize information routing, and masking interventions as complementary evidence of model reliance, rather than equating attention magnitude with causal importance.

## B.5 Performance on Qwen3-8B

To examine backbone generalization, we further conduct experiments with Qwen3-8B (Yang et al., 2025) in the no-think mode. For controlled comparison, we reuse the Qwen2-7B training hyperparameters, including the learning rate range, LoRA configuration, batch setting, and attention steering schedule. As shown in Table 16, performance on Qwen3-8B is generally lower than on Qwen2-7B, although the decline is not uniform. This may result from insufficient backbone-specific tuning, as Qwen3-8B may have different representation spaces, logit distributions, and layer-wise attention dynamics. Its stronger semantic prior may also reduce the use of external heterogeneous collaborative information, especially without semantic alignment. Thus, the lower scores suggest the need for backbone-specific tuning and semantic alignment, rather than failure of the method.

Despite the drop, the relative results still support SAILRec. Here, SAIL-w/o-S removes hierarchical attention steering, and SAIL-w/o-UI removes user-side and item-side semantic alignment. SeLLa-Rec improves over CoLLM on all metrics, showing that semantic alignment remains useful for Qwen3-8B. Full-SAILRec achieves the best results across both datasets and all metrics. The comparison with SAIL-w/o-UI indicates that dual-side semantic alignment helps beyond direct collaborative embedding injection, while the comparison with SAIL-w/o-S shows that attention steering improves

| Method       | MovieLens-1M  |               |               |               | Amazon-Book   |               |               |               |
|--------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|
|              | AUC           | UAUC          | NDCG          | MAP           | AUC           | UAUC          | NDCG          | MAP           |
| CoLLM        | 0.6916        | 0.6666        | 0.8665        | 0.7502        | 0.7520        | 0.6874        | 0.8750        | 0.8131        |
| SeLLa-Rec    | 0.7101        | 0.6938        | 0.8780        | 0.7657        | 0.8036        | 0.7152        | 0.8863        | 0.8310        |
| SAIL-w/o-S   | 0.7176        | 0.6870        | 0.8579        | 0.7397        | 0.8165        | 0.7247        | 0.8920        | 0.8382        |
| SAIL-w/o-UI  | 0.7057        | 0.6973        | 0.8719        | 0.7569        | 0.8042        | 0.7020        | 0.8801        | 0.8220        |
| Full-SAILRec | <b>0.7398</b> | <b>0.7114</b> | <b>0.8861</b> | <b>0.7784</b> | <b>0.8359</b> | <b>0.7407</b> | <b>0.8986</b> | <b>0.8485</b> |

Table 16: Performance comparison on MovieLens-1M and Amazon-Book using Qwen3-8B as the backbone.

layer-wise utilization. On Amazon-Book, Full-SAILRec brings clear UAUC, NDCG, and MAP gains over both ablation variants, suggesting that the two designs are especially helpful in the sparser book domain. Overall, SAILRec remains effective under Qwen3-8B without backbone-specific hyperparameter tuning.

## B.6 Comparison with Attention-Oriented Methods

**Comparison with HatLLM and CTR-Sink.** Although HatLLM (Cui et al., 2025), CTR-Sink (Li et al., 2026), and SAILRec all exploit attention mechanisms for recommendation, they operate on different information sources. HatLLM applies hierarchical attention masking to textual interaction histories, emphasizing intra-item semantics in shallow layers and cross-item relations in deeper layers. CTR-Sink introduces behavior-level sink tokens and attention biases to improve the aggregation of behavioral semantics. In contrast, SAILRec explicitly injects external MF-derived user and item collaborative embeddings, semantically aligns them with the LLM space, and regulates their utilization across Transformer depth. We additionally reproduce CTR-Sink under the same MovieLens-1M CTR task and evaluation protocol as SAILRec. As shown in Table 17, SAILRec consistently outperforms both attention-oriented recommendation methods across all metrics.

| Method   | AUC           | UAUC          | NDCG          | MAP           |
|----------|---------------|---------------|---------------|---------------|
| CTR-Sink | 0.6962        | 0.6557        | 0.8528        | 0.7290        |
| HatLLM   | 0.7330        | 0.7058        | 0.8762        | 0.7645        |
| SAILRec  | <b>0.7571</b> | <b>0.7410</b> | <b>0.8985</b> | <b>0.7999</b> |

Table 17: Performance comparison with attention-oriented recommendation methods on MovieLens-1M.

**Comparison with PASTA.** PASTA (Zhang et al., 2024a) is a general post-hoc attention-steering method that reweights selected attention heads to-

ward user-specified textual spans during LLM inference. SAILRec differs in both the steering target and objective: rather than steering attention toward existing textual content, it controls access to externally injected collaborative tokens that have been semantically aligned with the LLM representation space. Moreover, PASTA performs head-level post-hoc intervention, whereas SAILRec imposes a depth-aware shallow–middle–deep schedule to regulate when collaborative evidence is incorporated for recommendation. Thus, SAILRec uses attention steering as part of a recommendation-specific collaborative integration framework rather than as a general inference-time control mechanism.