Selective LLM-Guided Regularization for Enhancing Recommendation Models

Zhan Shi*
Santa Clara University
Santa Clara, CA, USA
ashi2@scu.edu

Shanglin Yang*
Google
Mountain View, CA, USA
shanglinyang@google.com

Abstract

Large language models (LLMs) provide rich semantic priors and strong reasoning capabilities, making them promising auxiliary signals for recommendation. However, prevailing approaches either deploy LLMs as standalone recommenders or apply global knowledge distillation, both of which suffer from inherent drawbacks. Standalone LLM recommenders are costly, biased, and unreliable across large regions of the user–item space, while global distillation forces the downstream model to imitate LLM predictions even when such guidance is inaccurate. Meanwhile, recent studies show that LLMs excel particularly in re-ranking and challenging scenarios, rather than uniformly across all contexts. We introduce *Selective LLM-Guided Regularization* (S-LLMR), a model-agnostic and computation-efficient framework that activates LLM-based pairwise ranking supervision only when a trainable gating mechanism-informed by user history length, item popularity, and model uncertainty predicts the LLM to be reliable. All LLM scoring is done offline, transferring knowledge without increasing inference cost. Experiments across multiple datasets show that this selective strategy consistently improves overall accuracy and yields substantial gains in cold-start and long-tail regimes, outperforming global distillation baselines.

Keywords

Recommender Systems, Large Language Models, Regularization, Cold-Start, Long-Tail, Knowledge Transfer

ACM Reference Format:
Zhan Shi and Shanglin Yang. 2026. Selective LLM-Guided Regularization for Enhancing Recommendation Models. In *The Nineteenth ACM International Conference on Web Search and Data Mining (WSDM Companion '26), February 22–26, 2026, Boise, ID, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3779211.3795736

1 Introduction

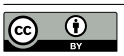
Recommendation systems underpin modern digital platforms by enabling content discovery, personalization, and user engagement across domains such as e-commerce, entertainment, and online media. Classical approaches—including collaborative filtering (CF), matrix factorization (MF), neural recommenders, and graph-based models achieve strong performance when user interaction histories are sufficiently dense. However, their effectiveness deteriorates in sparse regimes, such as cold-start users, long-tail items, and scenarios where user preferences are weakly expressed.

Large language models (LLMs) have emerged as powerful auxiliary knowledge sources for recommendation, offering rich semantic priors and strong reasoning capabilities that enable preference inference even from minimal user interaction data [10]. This makes them particularly promising in cold-start and sparsely observed regions where traditional recommenders tend to underperform. However, existing approaches to leveraging LLM signals remain fundamentally limited. Directly deploying LLMs as recommenders is prohibitively expensive and prone to issues such as position bias and hallucinated predictions. Meanwhile, global knowledge transfer methods [15, 20] require the downstream model to imitate LLM outputs uniformly across the entire user–item space, regardless of whether the LLM is reliable for a given instance. Recent attempts to distill LLM knowledge into classical models partially alleviate these issues, but they often depend on fine-tuned LLMs and still struggle to deliver consistent gains across different architectures or datasets.

Empirical motivation. Beyond high-level intuition, recent evaluations of LLM-based recommenders report *localized strengths* (notably on short histories and re-ranking) alongside *systematic weaknesses* including strong candidate position bias and occasional hallucinations [7]. These phenomena imply that LLM signals are *contextually reliable* rather than uniformly trustworthy. Our design follows directly from this evidence: instead of global imitation, we *selectively* invoke LLM guidance under reliability conditions predicted by a lightweight, learnable gating mechanism.

We propose Selective LLM-Guided Regularization (S-LLMR), a training framework that treats LLM knowledge as a conditional regularizer rather than a global supervisory signal. Instead of enforcing uniform imitation of LLM predictions, S-LLMR incorporates LLM-generated soft rankings only in regions where LLMs exhibit empirical advantages. This selective integration ensures that LLM guidance is beneficial rather than disruptive. We prompt an LLM using a compact representation of each user's recent interaction history to generate soft relevance scores over candidate items. All scoring is performed offline, introducing no inference-time overhead. A gating function controls whether LLM supervision is activated for a given user–item pair. This gate identifies regions where LLM signals are empirically reliable. With the gate active, we apply a weighted pairwise ranking loss that encourages the recommender to align its relative item ordering with LLM soft rankings, while automatically suppressing the influence of unreliable LLM predictions. Extensive experiments across multiple datasets and diverse recommendation backbones show that S-LLMR consistently surpasses global distillation baselines, delivering substantial improvements

*These authors contributed equally to this work.



WSDM Companion '26, Boise, ID, USA

ACM ISBN 979-8-4007-2358-2/2026/02
https://doi.org/10.1145/3779211.3795736

in sparse regimes such as cold-start and long-tail scenarios. The main contributions of this paper are:

- We introduce a gated LLM-based regularization paradigm that selectively incorporates LLM signals, avoiding the drawbacks of global distillation and remaining fully model-agnostic.
- We design S-LLMR, which combines a reliability-aware gating mechanism with an LLM-guided pairwise ranking loss for targeted knowledge transfer.
- Extensive experiments across multiple backbones show consistent AUC improvements, with especially strong gains in cold-start and long-tail scenarios.

2 Related Work

Classical collaborative filtering (CF) forms the foundation of modern recommender systems. Matrix factorization (MF) [9] models user–item affinities through latent factors and has been widely adopted due to its scalability and strong generalization ability. Neural extensions such as Neural Collaborative Filtering (NCF) [4] leverage multilayer perceptrons to capture nonlinear preference interactions. Graph-based recommenders, including NGCF [23], LightGCN [3], and PinSage [24] leverage user–item signals through graph structures to improve high-order connectivity modeling.

Despite their strong performance in dense regimes, these models degrade significantly under *cold-start* [16] and *long-tail* [14] conditions.

Recent hybrid approaches such as UniSRec [5] unify textual and collaborative filtering signals to improve robustness, but still rely on large-scale metadata and do not exploit LLM reasoning. Existing approaches either use LLMs as direct recommenders, e.g., RankLLM [17], or distill LLM outputs into recommendation models in a global manner, such as SLMRec [11] and LLM-CF [19]. However, these methods do not account for the empirical finding that LLM signals are only *locally reliable*—being highly beneficial in semantic or sparse contexts, but noisy or misleading in others [6, 8].

To address this gap, we adopt a different perspective that LLM outputs should be treated as *conditionally reliable* auxiliary signals, rather than unconditional ground truth. Our work operationalizes this idea by introducing a selective LLM integration framework equipped with a lightweight, learnable gating mechanism to determine when LLM guidance should be trusted. This allows the model to avoid global distillation while selectively leveraging LLM strengths in the contexts where they are most effective.

3 Method

As shown in Figure 1, the **Selective LLM-Guided Regularization for Recommendation** (S-LLMR) is a model-agnostic training framework that selectively leverages large language models (LLMs) to regularize classical recommender models only in regions where LLM predictions are empirically reliable. Formally, given a user u and item i, a base recommender produces a predicted relevance score $s_{u,i}$, while the LLM provides a soft preference score $s^{LLM}_{u,i}$. Our goal is to integrate LLM guidance selectively through pairwise ranking supervision with a gating signal. There are three main modules included in the pipeline.

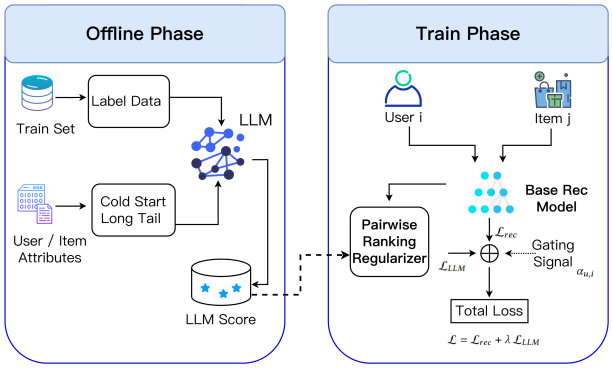


Figure 1: Illustration of our selective LLM-guided regularization framework. Left: In the offline phase, the LLM is prompted to produce soft relevance scores Right: In the training phase, a base recommender produces prediction scores, and the LLM signals are incorporated through a pairwise ranking regularizer whose contribution is controlled by a gating function.

3.1 LLM-Generated Soft Rankings

For each user u, we construct a succinct textual summary of the user's recent interaction history and query an LLM with a prompt of the form:

> "Given that the user recently interacted with items $\{i_1, i_2, \dots\}$, rank the following candidate items by their likelihood of matching the user's preferences."

The LLM returns a soft score $s^{LLM}_{u,i} \in [0, 1]$ for each candidate item, computed via normalized logits or temperature-scaled soft ranking. All LLM scoring is performed *offline*, and therefore introduces *no inference-time overhead*. To improve supervision coverage in sparse regions, we additionally construct two synthetic candidate sets:

- **Cold-start user candidates:** Users with short histories (<=3) are paired with diverse sampled items to elicit LLM judgments for user-item combinations not present in training data.
- **Long-tail items (bottom 10% popularity):** Items whose popularity falls in the lowest 10% of the catalog are paired with sampled users so that the LLM can evaluate these underrepresented items and provide supervision where collaborative filtering is weakest.

These augmented LLM-scored pairs expand the offline supervision table and cover precisely the settings where classical recommenders lack sufficient signals.

3.2 LLM-Guided Pairwise Ranking Regularizer

Motivated by the empirical observation that LLMs are more reliable for re-ranking than for pointwise supervision, we adopt a pairwise ranking regularizer. Given LLM soft scores, we impose an auxiliary pairwise ranking constraint that encourages the recommender to follow the ordering implied by the LLM whenever appropriate. For user u, if $s^{LLM}_{u,i} > s^{LLM}_{u,j}$ for two items (i, j), the model is encouraged to produce $s_{u,i} > s_{u,j}$ with a margin.

The pairwise LLM loss is defined as:

$$\mathcal{L}_{LLM} = \sum_{(u,i,j)\in\mathcal{P}} \alpha_{u,i,j} \max\big(0,\ m - (s_{u,i} - s_{u,j})\big),$$

where $\alpha_{u,i,j}$ is a selective gating weight defined later.

User-consistent pair construction. To ensure semantic alignment, pairs are constructed *within* individual users. In each batch, we sample one or more users, extract items associated with those users, filter valid LLM scores, sort them by LLM ranking, and form ordered pairs (i, j) where $s^{LLM}_{u,i} > s^{LLM}_{u,j}$. This avoids mixing signals from unrelated users.

Adaptive pair selection. Because batches may contain varying numbers of users or valid LLM entries, we employ an adaptive strategy: given a target maximum of K pairs, the effective number $\tilde{K}$ is adjusted based on batch structure. The algorithm selects up to $\tilde{K}$ highest-confidence pairs ranked by their LLM score difference, ensuring (i) at least one pair when possible, and (ii) avoidance of over-regularization.

Overall, this regularizer enables selective, reliability-aware knowledge transfer: the recommender follows LLM rankings when they are trustworthy, while naturally resisting noisy or inconsistent supervision.

3.3 Selective Gating Mechanism

LLM supervision is not uniformly reliable. We therefore define a per-pair gate $\alpha_{u,i} \in [0, 1]$ that scales the contribution of the LLM regularizer.

Signals. We compute: (i) a cold-start indicator $\text{Cold}(u) = \mathbb{1}[|\mathcal{H}(u)| < \tau_u]$, (ii) a long-tail indicator $\text{Tail}(i) = \mathbb{1}[\text{pop}(i) < \tau_i]$, and (iii) a continuous uncertainty score $q_{u,i} \in [0, 1]$ from the base model (e.g., predictive entropy or ensemble variance normalized to $[0, 1]$).

Learnable gate. Let $z_{u,i} = \big[\text{Cold}(u),\ \text{Tail}(i),\ q_{u,i}\big] \in \mathbb{R}^3$. We use a one-layer gating network

$$\alpha_{u,i} = \sigma\big(\mathbf{w}^\top z_{u,i} + b\big),$$

with parameters $\theta_g = \{\mathbf{w}, b\}$ learned jointly by back-propagation from the full objective (Sec. 3.4). For pairwise supervision, we set $\alpha_{u,i,j} = \frac{1}{2}(\alpha_{u,i} + \alpha_{u,j})$.

Uncertainty instantiations. We consider (a) *confidence-based* $q_{u,i} = 1 - \max_c p_\theta(c|u, i)$, (b) *entropy-based* $q_{u,i} = \text{H}(p_\theta(\cdot|u, i))$, or (c) *dropout/ensemble variance*. We select the best on validation.

The gating parameters θ_g are learned jointly with the backbone through back-propagation from the LLM regularization loss. When LLM-guided pairs reduce the hinge loss, gradients increase $\alpha_{u,i}$; when LLM signals are unhelpful, the gate is driven downward. This allows the model to automatically learn when LLM supervision is reliable without manual thresholds—focusing LLM influence on cold-start, long-tail, and high-uncertainty cases.

Algorithm 1 S-LLMR Training with Learnable Gating

Require: Base model θ, gating params θ_g, LLM table $\mathbf{T}_{\text{LLM}}$, margins m, weights λ

while not converged **do**
  Sample minibatch of users $\mathcal{B}$ and their interactions
  for each $u \in \mathcal{B}$ **do**
    Compute base scores $s_{u,i}$ for items in batch
    Build user-consistent ordered pairs $\mathcal{P}_u = \{(i, j) : s^{\text{LLM}}_{u,i} > s^{\text{LLM}}_{u,j}\}$ from $\mathbf{T}_{\text{LLM}}$
    For each (u, i) compute signals: $\text{Cold}(u)$, $\text{Tail}(i)$, $q_{u,i}$
    Gate: $\alpha_{u,i} = \sigma(\mathbf{w}^\top[\text{Cold}(u), \text{Tail}(i), q_{u,i}] + b)$
  end for
  $\mathcal{L}_{\text{rec}} \leftarrow$ base loss (e.g., BCE/BPR/InfoNCE)
  $\mathcal{L}_{\text{LLM}} \leftarrow \sum_{(u,i,j)\in\cup_u\mathcal{P}_u} \frac{\alpha_{u,i}+\alpha_{u,j}}{2} \cdot \max(0, m - (s_{u,i} - s_{u,j}))$
  Update θ, θ_g by SGD on $\mathcal{L} = \mathcal{L}_{\text{rec}} + \lambda\mathcal{L}_{\text{LLM}}$
end while

3.4 Training Objective and Optimization Procedure

Algorithm 1 summarizes the full optimization procedure, including the construction of user-consistent LLM ranking pairs, computation of gating signals, and joint gradient updates of the base recommender and gating parameters. This design ensures that LLM knowledge is injected in a targeted and reliability-aware manner without interfering with the core training dynamics of the underlying model. The full training objective is:

$$\mathcal{L} = \mathcal{L}_{rec} + \lambda\mathcal{L}_{LLM},$$

with λ controlling the strength of the regularizer.

4 Experimental Setup

4.1 Backbone Models

To evaluate the model-agnostic nature of S-LLMR, we integrate it with six widely adopted and architecturally diverse recommendation backbones: DeepFM [1], xDeepFM [12], AutoInt [18], DCNv1 [21], DCNv2 [22], and DIN [25]. These models span a broad spectrum of interaction modeling strategies, including factorization-machine style feature crossing (DeepFM), vector-wise compressed interactions (xDeepFM), self-attentive feature learning (AutoInt), explicit cross layers (DCNv1/DCNv2), and attention over user behavior sequences (DIN). This diversity enables a comprehensive assessment of how selectively injected LLM signals generalize across different inductive biases.

To contextualize S-LLMR within the landscape of LLM-assisted recommendation, we compare against three representative paradigms:

- **KD Distillation Baseline** Following prior work [10, 15], the soft logits from a fine-tuned LLaMA2-7B model into each backbone, representing the standard global LLM-to-recommender imitation approach.
- **KAR (Knowledge-Augmented Recommendation)** KAR [13] aligns user and item representations with LLM-derived open-world knowledge, capturing the representation-enrichment paradigm of using LLMs in recommendation.
- **LLM-CF** LLM-CF [20] distills LLM world knowledge and reasoning ability into collaborative filtering, formulating

Table 1: Dataset statistics for the three Amazon domains.

Metric	Sports	Beauty	Toys
#Users	35,598	22,363	19,412
#Items	18,357	12,101	11,924
#Reviews	379,086	262,826	218,722
Cold-start interactions	190,756	119,854	103,314
(% of interactions)	50.3%	45.6%	47.2%
Long-tail items	3,659	2,400	2,326
(% of items)	19.9%	19.8%	19.5%

LLM-derived semantic signals as soft preference labels. This approach represents the state of the art in LLM-enhanced CF models.

Together, these baselines cover the three dominant LLM-for-recommendation paradigms: global distillation, representation alignment, and LLM-assisted collaborative filtering. Our comparison highlights the conceptual distinction and empirical advantages of *selective* over *global* LLM integration.

4.2 Datasets

We evaluate S-LLMR on three domains of the Amazon Review dataset [2], consistent with widely used recommendation benchmarks. Dataset statistics are shown in Table 1. We use the **Sports & Outdoors**, **Beauty**, and **Toys & Games** subsets.

Across all domains, the data exhibit significant sparsity: nearly half of all interactions originate from cold-start users, and roughly 20% of items fall into the long-tail. These characteristics make the datasets particularly suitable for evaluating algorithms designed to improve performance in sparse regimes—precisely where LLM-based semantic guidance is expected to be most beneficial.

4.3 Offline LLM Scoring Pipeline

To obtain LLM-derived soft preference signals without adding inference-time overhead, we generate all scores $s_{u,i}^{LLM}$ offline through a lightweight pipeline. For each user u, we extract a recent history $\mathcal{H}_L(u)$ (last $L = 10$ interactions) and sample M candidate items from a top-K popularity pool. Each tuple $(u, \mathcal{H}_L(u), C(u))$ is converted into a concise natural-language prompt and sent to **GPT-4o-mini**, which returns structured (`item_id`, `score`) pairs in $[0, 1]$. Returned scores are normalized, missing values default to 0.5, and all results are stored as a lookup table $(u, i) \mapsto s_{u,i}^{LLM}$. During training, these offline scores are used exclusively by the selective regularizer and never affect the base model's loss or inference cost. This design provides flexible control over the number of scored users and candidates, enabling an efficient balance between LLM query cost and supervision coverage.

4.4 Training Protocol

All models are trained using the Adam optimizer with a learning rate of 10^{-3}, a batch size of 128, and an embedding dimension of 64, following common practice in CTR and implicit-feedback recommendation. For S-LLMR, we set the regularization weight to $\lambda = 0.1$, and select the final value based on validation AUC. No additional hyperparameter tuning is performed unless explicitly noted.

Table 2: AUC performance across three Amazon domains using six backbone architectures. For each domain, the highest AUC within a backbone group is bolded. Across all models and datasets, S-LLMR consistently achieves the strongest performance.

Backbone	Framework	AUC↑		
		Sports	Beauty	Toys
DeepFM	None	0.7990	0.7853	0.7681
	KD	0.8043	0.7959	0.7713
	KAR	0.7991	0.7870	0.7698
	LLM-CF	0.8137	0.8044	0.7881
	Ours	**0.8176**	**0.8101**	**0.7961**
xDeepFM	None	0.8158	0.8065	0.7836
	KD	0.8169	0.8104	0.7865
	KAR	0.8161	0.8101	0.7898
	LLM-CF	0.8196	0.8113	0.7947
	Ours	**0.8240**	**0.8183**	**0.7985**
AutoInt	None	0.8003	0.7949	0.7630
	KD	0.8012	0.7961	0.7635
	KAR	0.8039	0.7939	0.7683
	LLM-CF	0.8088	0.8090	0.7754
	Ours	**0.8161**	**0.8145**	**0.7849**
DCNv1	None	0.8023	0.8146	0.7621
	KD	0.8040	0.8147	0.7652
	KAR	0.8024	0.8165	0.7651
	LLM-CF	0.8092	0.8182	0.7702
	Ours	**0.8190**	**0.8189**	**0.7960**
DCNv2	None	0.8110	0.8028	0.7774
	KD	0.8112	0.8057	0.7827
	KAR	0.8087	0.8003	0.7759
	LLM-CF	0.8131	0.8033	0.7812
	Ours	**0.8150**	**0.8177**	**0.7927**
DIN	None	0.7986	0.7861	0.7586
	KD	0.8023	0.7934	0.7652
	KAR	0.7971	0.7861	0.7620
	LLM-CF	0.8089	0.7967	0.7783
	Ours	**0.8100**	**0.8010**	**0.7829**

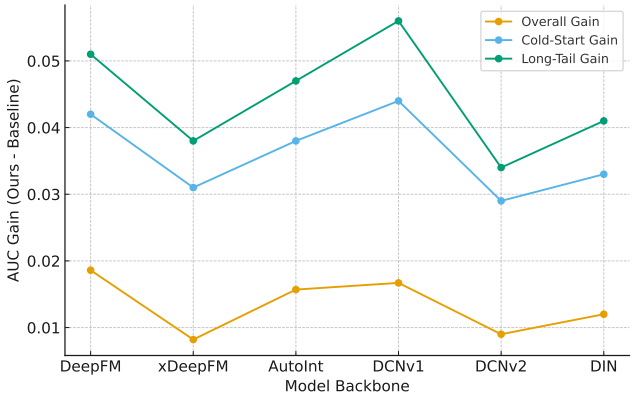


Figure 2: AUC improvements in cold-start and long-tail regimes across backbones. S-LLMR delivers the strongest boosts for cold-start users and long-tail items—often exceeding general relative improvement.

Table 3: Ablation study on DCNv2: We compare global vs. gated LLM regularization, and pointwise vs. pairwise LLM supervision.

Method	Sports			Beauty			Toys		
	Overall	Cold	LongTail	Overall	Cold	LongTail	Overall	Cold	LongTail
Global vs. Gated LLM Regularization									
DCNv2	0.811	0.8140	0.7677	0.8028	0.8006	0.7702	0.7774	0.7868	0.7402
DCN + Global LLM Regularization	0.8114	0.8140	0.7640	0.7911	0.7797	0.7330	0.7887	0.7858	0.7390
DCN + Gated LLM Regularization (Ours)	**0.8150**	**0.8170**	**0.7877**	**0.8177**	**0.8063**	**0.7716**	**0.7927**	**0.7917**	**0.7502**
Pointwise vs. Pairwise LLM Supervision									
DCNv2 (Backbone)	0.811	0.8140	0.7677	0.8028	0.8006	0.7702	0.7774	0.7868	0.7402
DCN + LLM Pointwise MSE	0.8069	0.8119	0.7571	0.7999	0.7922	0.7445	0.7705	0.7671	0.7391
DCN + Pairwise Ranking (Ours)	**0.8150**	**0.8170**	**0.7877**	**0.8177**	**0.8063**	**0.7716**	**0.7927**	**0.7917**	**0.7502**

4.5 Evaluation Protocol

We adopt the standard *full-ranking* evaluation setting, where each test interaction is ranked against all items with which the user has not interacted in the training or validation sets. Since our goal is to assess both global predictive accuracy and robustness in sparse regions, we report AUC as the sole evaluation metric. To further evaluate model performance under challenging conditions, we report AUC on two key sub-populations:

- **Cold-start users**: test interactions belonging to users with fewer than k historical interactions, i.e., $|\mathcal{H}(u)| < k$. We set $k = 3$ in our experiments.
- **Long-tail items**: items in the bottom 20% of the popularity distribution based on training data. We first identify long-tail item IDs from the training set and then select the corresponding interactions from the test set to form the long-tail subset.

These stratified subsets isolate the effect of S-LLMR in sparse and semantically challenging regimes, enabling a clearer understanding of how selective LLM guidance improves recommendation quality under conditions where traditional models typically struggle.

5 Results

Our results show that S-LLMR consistently improves AUC across all backbones and domains, and delivers the largest gains in cold-start and long-tail scenarios.

5.1 Overall Performance Across Backbones

The overall performance is shown in Table 2. Across all six backbone models including DeepFM, xDeepFM, AutoInt, DCNv1, DCNv2, and DIN. S-LLMR achieves the strongest AUC scores on every Amazon domain. The improvements over non-LLM baselines (None, KD, KAR) are consistent and sizable, and our method further surpasses the LLM-CF approach by margins of 0.003–0.01 AUC depending on the model and dataset. Architectures that struggle more with semantic sparsity, such as AutoInt and DCNv1, exhibit particularly large gains: AUC improvements reach 0.007–0.01 on Sports and exceed 0.02 on the Toys domain. These results validate that selectively incorporating LLM signals rather than distilling them globally allows the recommender to capitalize on LLM strengths while avoiding the noise and positional bias present in many LLM outputs, yielding reliable improvements across heterogeneous architectures and domains.

5.2 Effectiveness in Sparse and Hard Regimes

As shown in Figure 2, across all datasets, S-LLMR delivers the strongest boosts for cold-start users and long-tail items often exceeding the general improvement trend. This pattern confirms that the selective gating mechanism effectively activates LLM guidance where collaborative-filtering signals are weakest. Cold-start gains demonstrate that the method leverages LLM semantic priors to compensate for short interaction histories, while long-tail gains highlight improved robustness on niche items that lack sufficient popularity-based signals. Together, these results indicate that the primary benefit of S-LLMR lies in its ability to reinforce the recommender precisely in the regions where traditional models fail, rather than merely improving global accuracy.

5.3 Ablation Study on Module Effectiveness

Across all three domains and evaluation subsets shown in Table 3, the ablations demonstrate that our gated selective LLM-guided regularization is the only strategy that consistently improves performance in both overall and sparse regimes. Applying LLM loss globally often degrades long-tail accuracy and substantially harms Beauty-domain performance, highlighting that LLM predictions are not uniformly reliable. Pointwise (BCE/MSE) LLM supervision also fails to deliver meaningful improvements and frequently underperforms the backbone. In contrast, our selective gating mechanism combined with a pairwise ranking loss yields the strongest gains across all settings—most notably on cold-start and long-tail subsets, where AUC improvements reach +0.02 to +0.04 over the backbone and up to +0.05 over global or pointwise LLM methods. These results confirm that (i) LLM signals must be used selectively, and (ii) ranking-based supervision is the most effective way to transfer LLM semantic knowledge without amplifying LLM noise.

6 Conclusion

This paper proposes S-LLMR, a selective LLM-guided regularization framework that injects LLM semantic signals into classical recommenders in a reliability-aware manner. Instead of globally imitating LLM predictions, S-LLMR activates LLM-based pairwise supervision only where LLMs show clear advantages, such as cold-start users, long-tail items, and high-uncertainty cases. Experiments across six recommenders and three Amazon domains show consistent AUC improvements, with especially large gains in sparse and semantically challenging settings. Ablation studies further reveal that global LLM regularization can harm performance, while selective pairwise regularization reliably improves robustness.

References

[1] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: A Factorization-Machine Based Neural Network for CTR Prediction. In *IJCAI*.

[2] Ruining He and Julian McAuley. 2016. Ups and Downs: Modeling the Visual Evolution of Fashion Trends with One-Class Collaborative Filtering. In *Proceedings of the 25th International Conference on World Wide Web*. 507–517.

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

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

[5] Yupeng Hou, Shanlei Mu, Wayne Xin Zhao, Yaliang Li, Bolin Ding, and Ji-Rong Wen. 2022. Towards universal sequence representation learning for recommender systems. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*. 585–593.

[6] Lei Huang et al. 2023. A Survey on Hallucination in Large Language Models. *arXiv:2311.05232* (2023).

[7] Chumeng Jiang, Jiayin Wang, Weizhi Ma, Charles LA Clarke, Shuai Wang, Chuhan Wu, and Min Zhang. 2025. Beyond Utility: Evaluating LLM as Recommender. In *Proceedings of the ACM on Web Conference 2025*. 3850–3862.

[8] Xiang Jiang et al. 2024. Beyond Utility: Evaluating LLM as Recommender. *arXiv:2411.00331* (2024).

[9] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009), 30–37.

[10] Lei Li, Zhenhua Sun, et al. 2023. LLM4Rec: Large Language Models for Recommendation. *arXiv preprint arXiv:2306.10997* (2023).

[11] Xinyu Li et al. 2025. SLMRec: Distilling Large Language Models into Small Models for Sequential Recommendation. *ICLR* (2025).

[12] Jianxun Lian, Xiaohuan Li, Yujing Zhang, Guangzhong Sun, and Xing Xie. 2018. xDeepFM: Combining Explicit and Implicit Feature Interactions for Recommender Systems. In *KDD*.

[13] Xi Lin, Bowen Du, et al. 2024. Towards Open-World Recommendation with Knowledge Augmentation from Large Language Models. *arXiv preprint arXiv:2306.10933* (2024).

[14] Jing Qin. 2021. A survey of long-tail item recommendation methods. *Wireless Communications and Mobile Computing* 2021, 1 (2021), 7536316.

[15] Kan Ren and et al. Zhang. 2024. LLM-Distill: Distilling Large Language Models into Recommendation Models. *arXiv preprint arXiv:2402.03852* (2024).

[16] Martin Saveski and Amin Mantrach. 2014. Item cold-start recommendations: learning local collective embeddings. In *Proceedings of the 8th ACM Conference on Recommender systems*. 89–96.

[17] Sahel Sharifymoghaddam et al. 2025. RankLLM: A Python Package for Reranking with LLMs. *SIGIR* (2025).

[18] Weiping Song, Chence Shi, Zhiping Xiao, Zhijian Duan, Yewen Xu, Ming Zhang, and Jian Tang. 2019. AutoInt: Automatic Feature Interaction Learning via Self-Attentive Neural Networks. In *CIKM*.

[19] Z. Sun et al. 2024. Large Language Models Enhanced Collaborative Filtering. *arXiv:2403.17688* (2024).

[20] Zhongxiang Sun, Zihua Si, Xiaoxue Zang, Kai Zheng, Yang Song, Xiao Zhang, and Jun Xu. 2024. Large language models enhanced collaborative filtering. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*. 2178–2188.

[21] Ruoxi Wang, Bin Fu, Gang Fu, and Mingliang Wang. 2017. Deep & Cross Network for Ad Click Predictions. In *ADKDD*.

[22] Ruoxi Wang, Rakesh Shivanna, Derek Zhiyuan Cheng, Sagar Jain, Dong Lin, Michael Bendersky, and Marc Najork. 2021. DCN V2: Improved Deep & Cross Network and Practical Lessons for Web-scale Learning to Rank Systems. In *WWW*.

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

[24] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec. 2018. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 974–983.

[25] Guorui Zhou, Chengru Song, Xiaoqiang Zhu, Ying Fan Ma, Ying Yan, Xiangnan He, et al. 2018. Deep Interest Network for Click-Through Rate Prediction. In *KDD*.