Large Language Model Enhanced Recommender Systems

Qidong Liu[1,2], Xiangyu Zhao[2], Yuhao Wang[2], Yejing Wang[2], Zijian Zhang[2], Yuqi Sun[1], Xiang Li, Maolin Wang[2], Pengyue Jia[2], Chong Chen, Wei Huang, Feng Tian[1]

[1] Xi'an Jiaotong University
[2] City University of Hong Kong

Aug 3, 2025

Agenda

1 Introduction

Zijian Zhang

2 Knowledge Enhancement

Pengyue Jia

3 Interaction Enhancement

Ziwei Liu

4.1 Model Enhancement 1

Maolin Wang

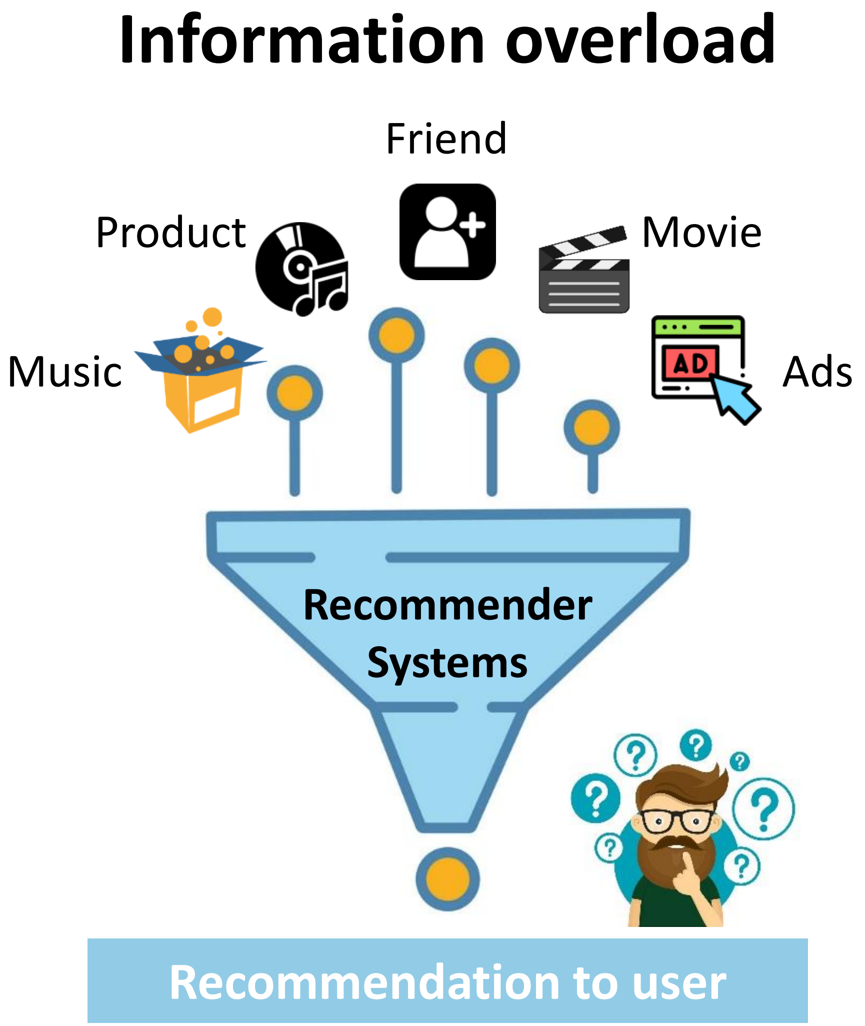
4.2 Model Enhancement 2

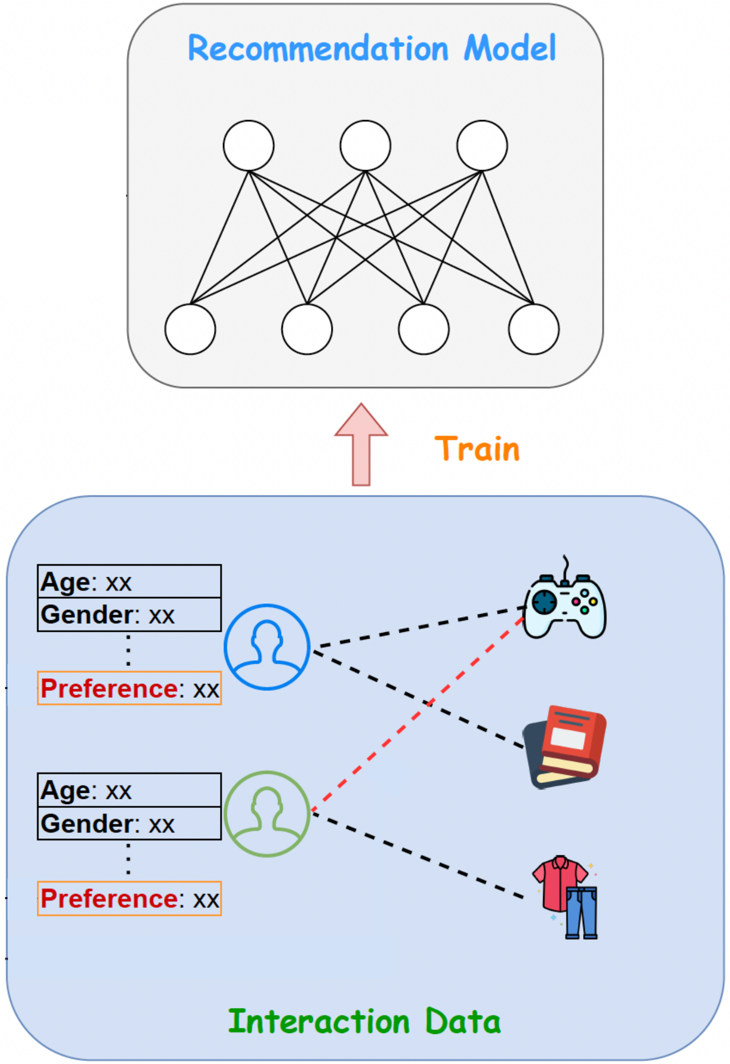
Yuhao Wang

5 Conclusion

Yejing Wang

Background of Recommender Systems

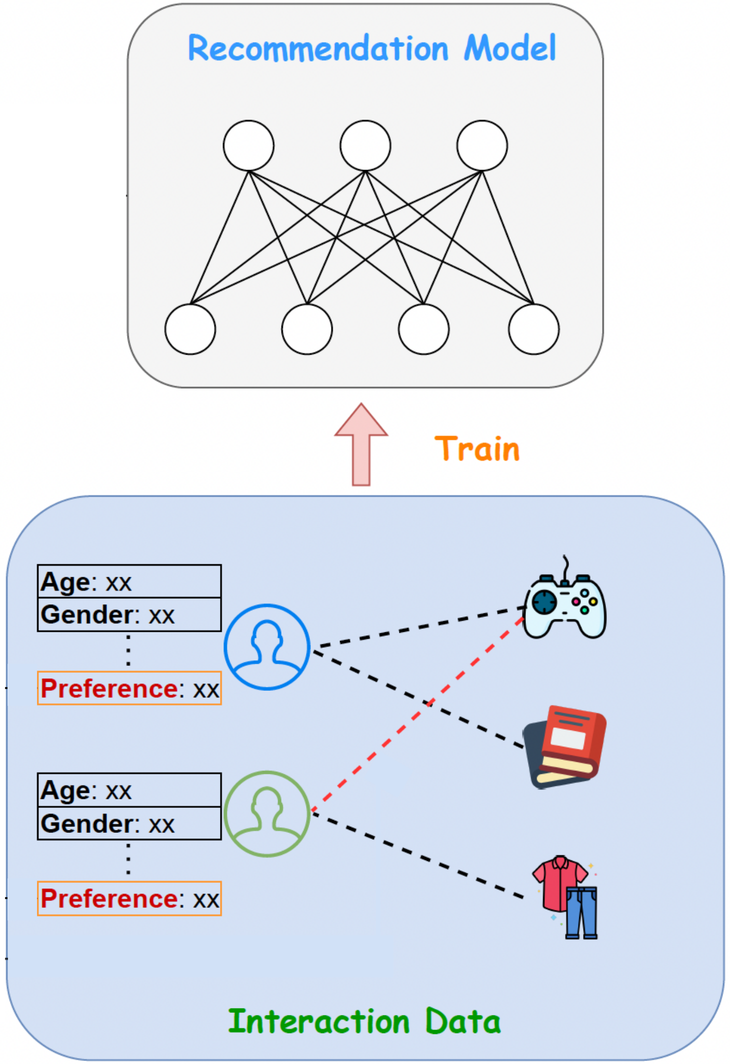




Background of Recommender Systems

Limitations of RS

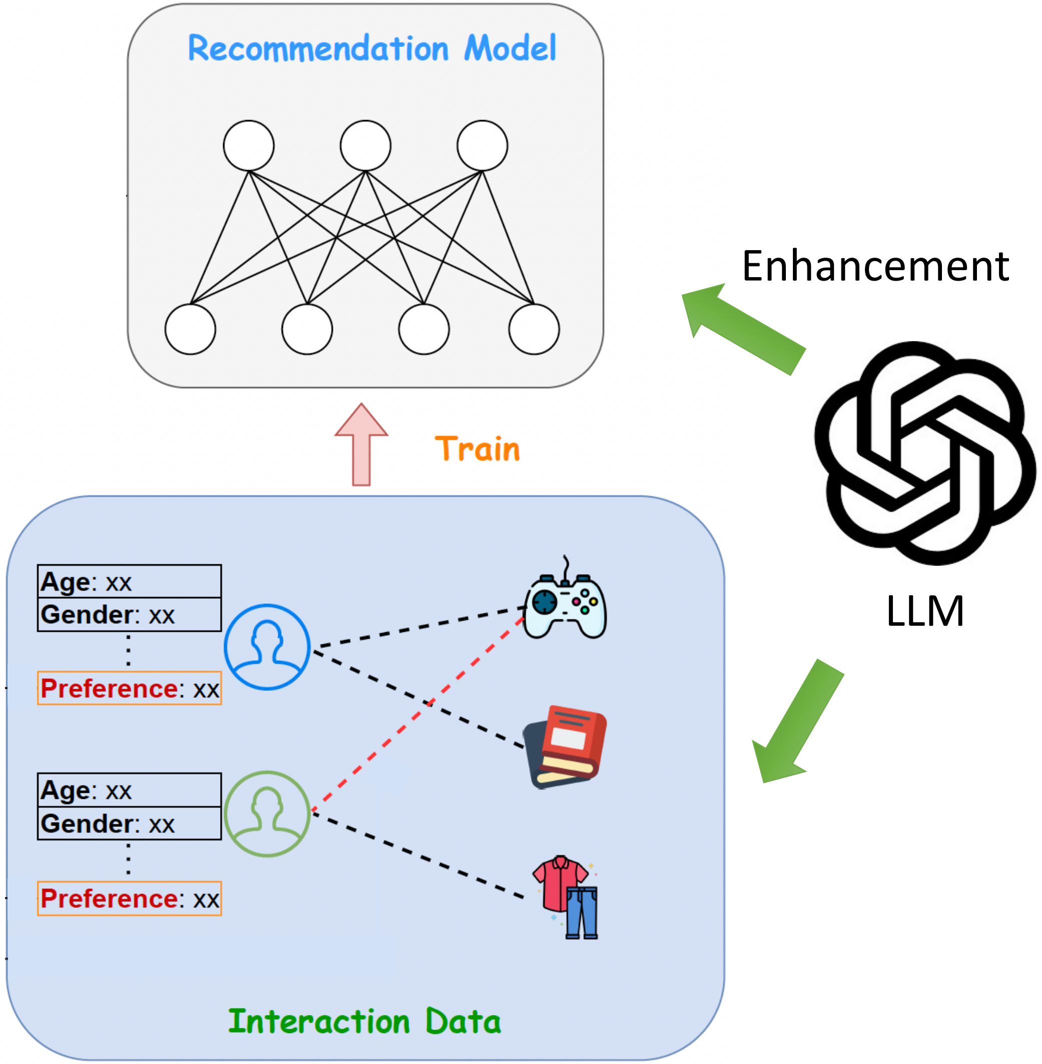
- Feature level: Lack of reasoning and knowledge in feature representations
- Interaction level: Data sparsity in user-item interactions
- Model level: Overlooking semantic understanding



Background of Recommender Systems

Advantages of LLM

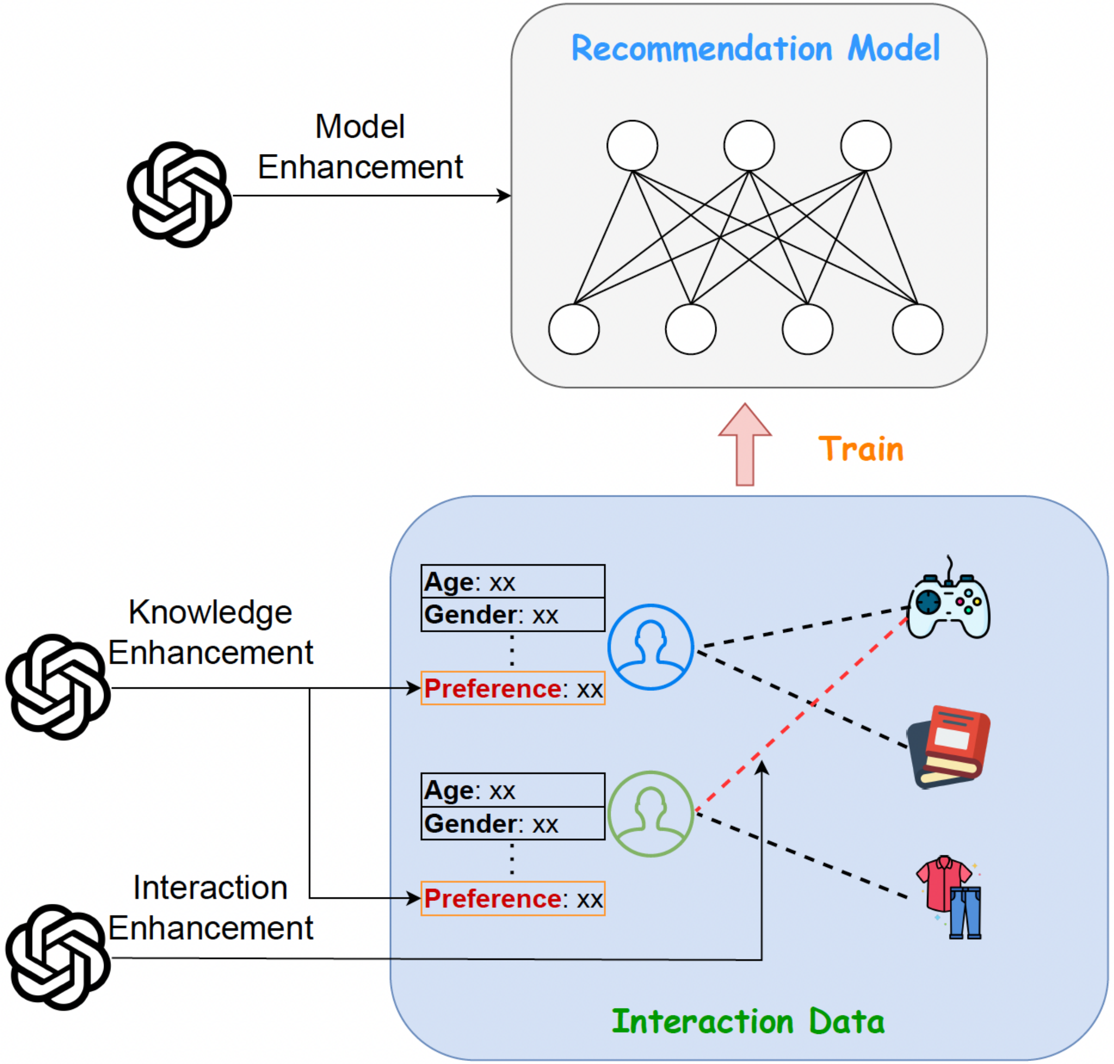
- Feature level: Reasoning abilities and world knowledge to derive textual descriptions
- Interaction level: Approximate users and derive new user-item interactions
- Model level: Analyze interactions from a semantic view



LLM-enhanced RS

LLM-enhanced RS (LLMERS)

- Knowledge enhancement
- Interaction enhancement
- Model enhancement

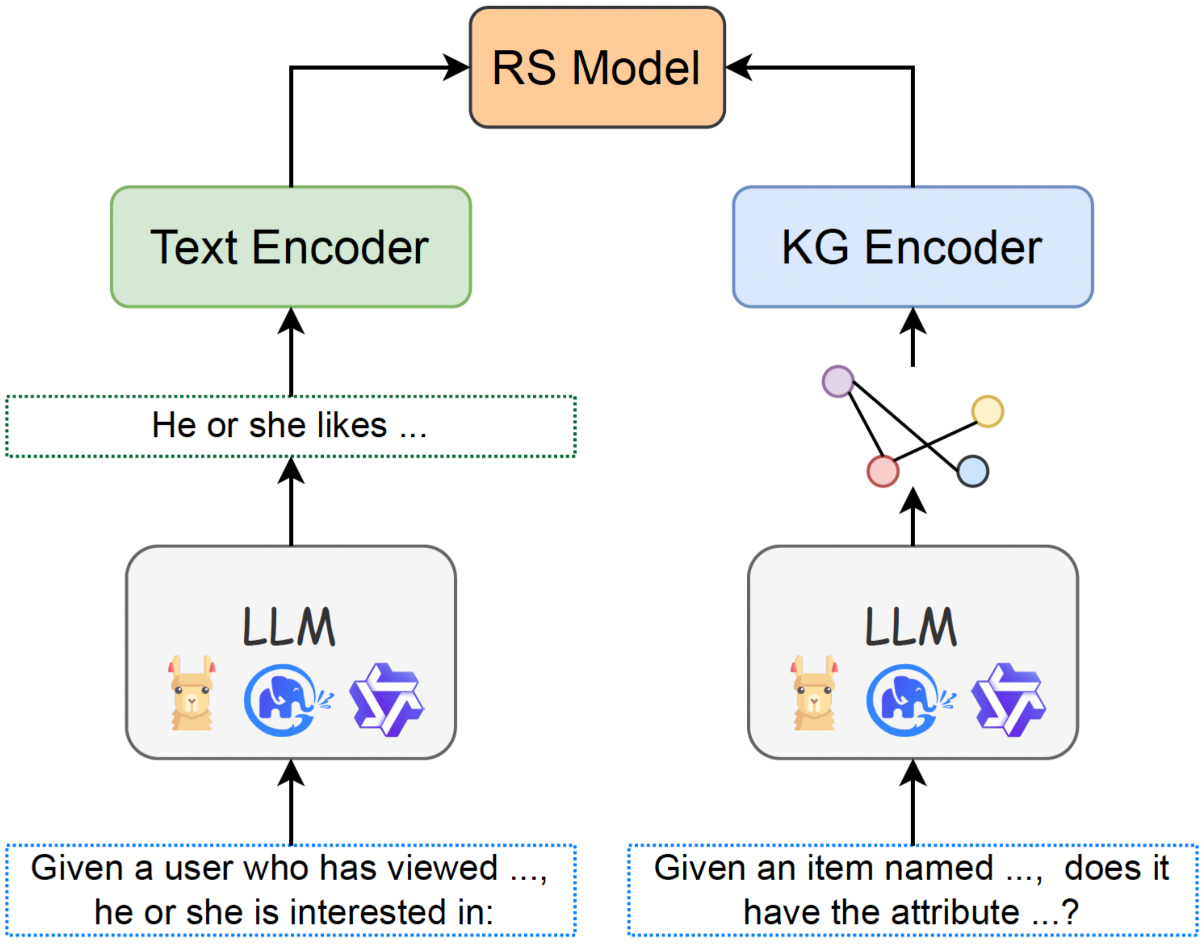


Knowledge enhancement

Knowledge enhancement

- Summary Text: Utilizes LLM to summarize characteristics of items or reason for user preferences.

- Knowledge Graph: Applies LLM to generate or augment structured Knowledge Graphs

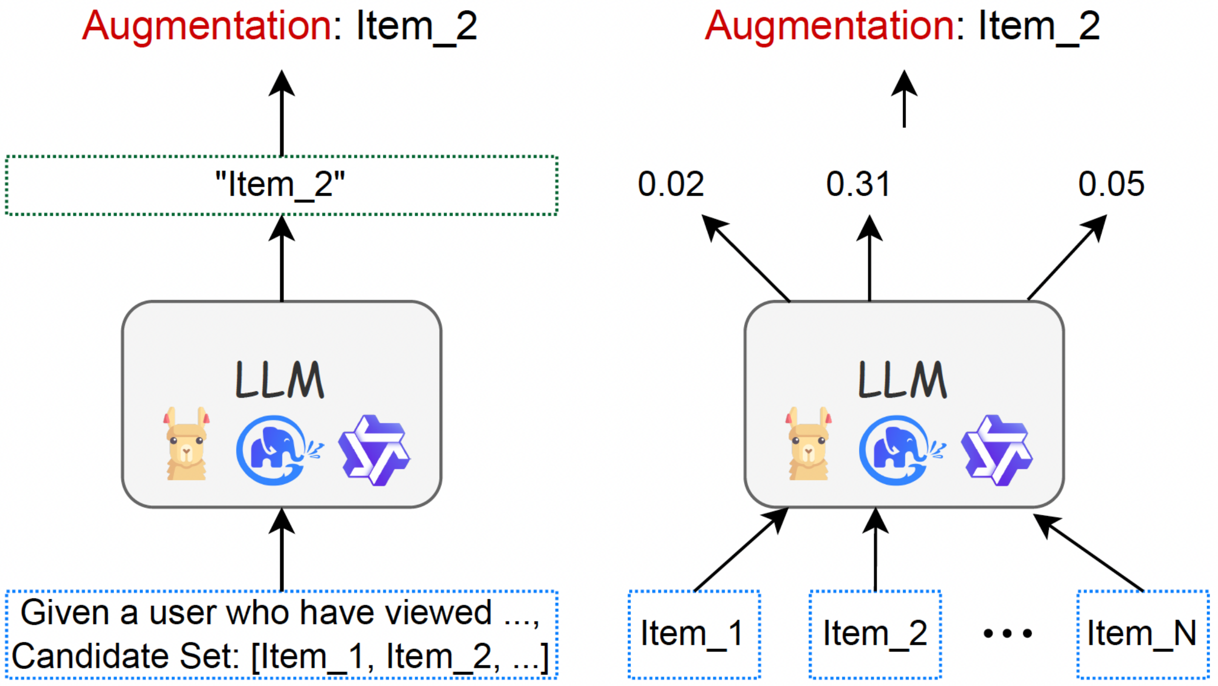


(a) Summary Text (b) Knowledge Graph

Interaction enhancement

Interaction enhancement

- Text-based: LLM outputs names of pseudo-interacted items as augmentation
- Score-based: LLM derives probabilities of possible interactions for augmentation

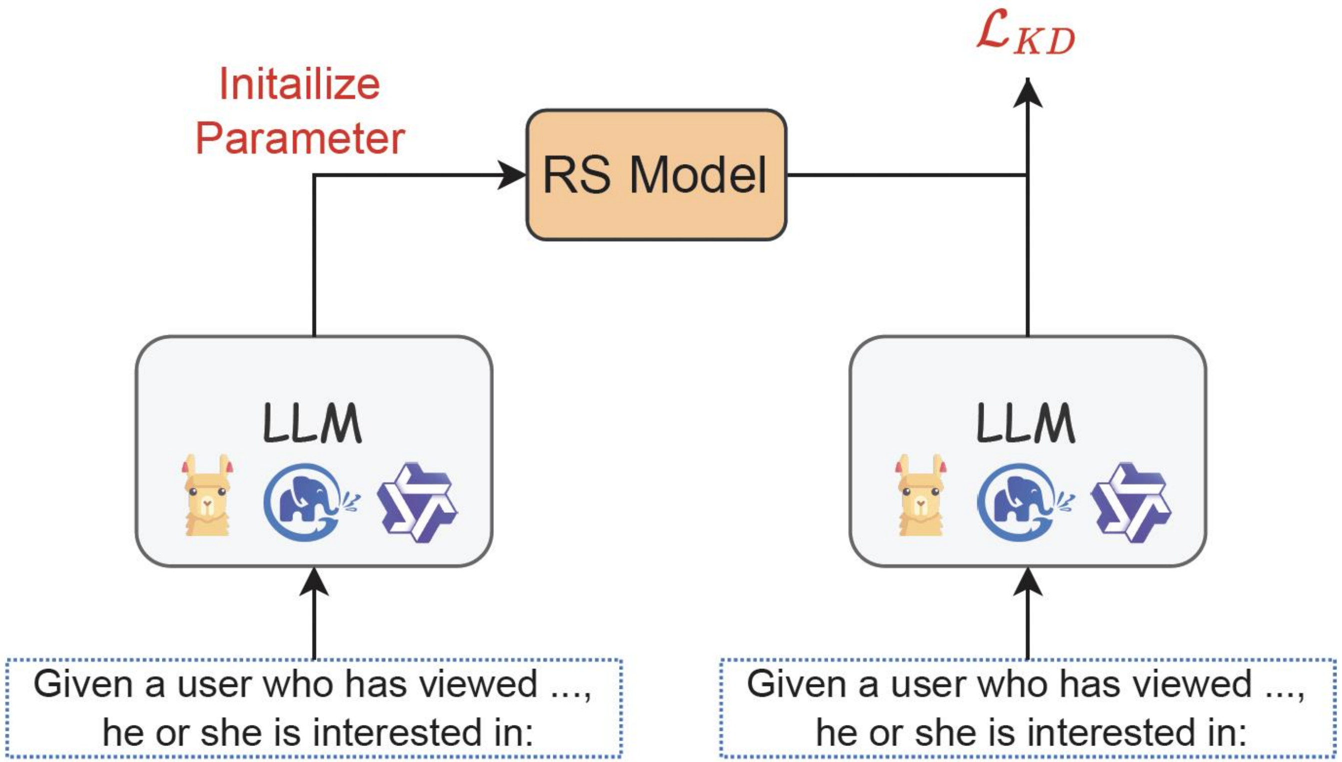


(a) Text-based (b) Score-based

Model enhancement

Model enhancement

- Model Initialization: Pretrain or initialize RS model weights with LLM's semantics before training
- Model Distillation: Transfer powerful abilities of LLM to smaller RS models



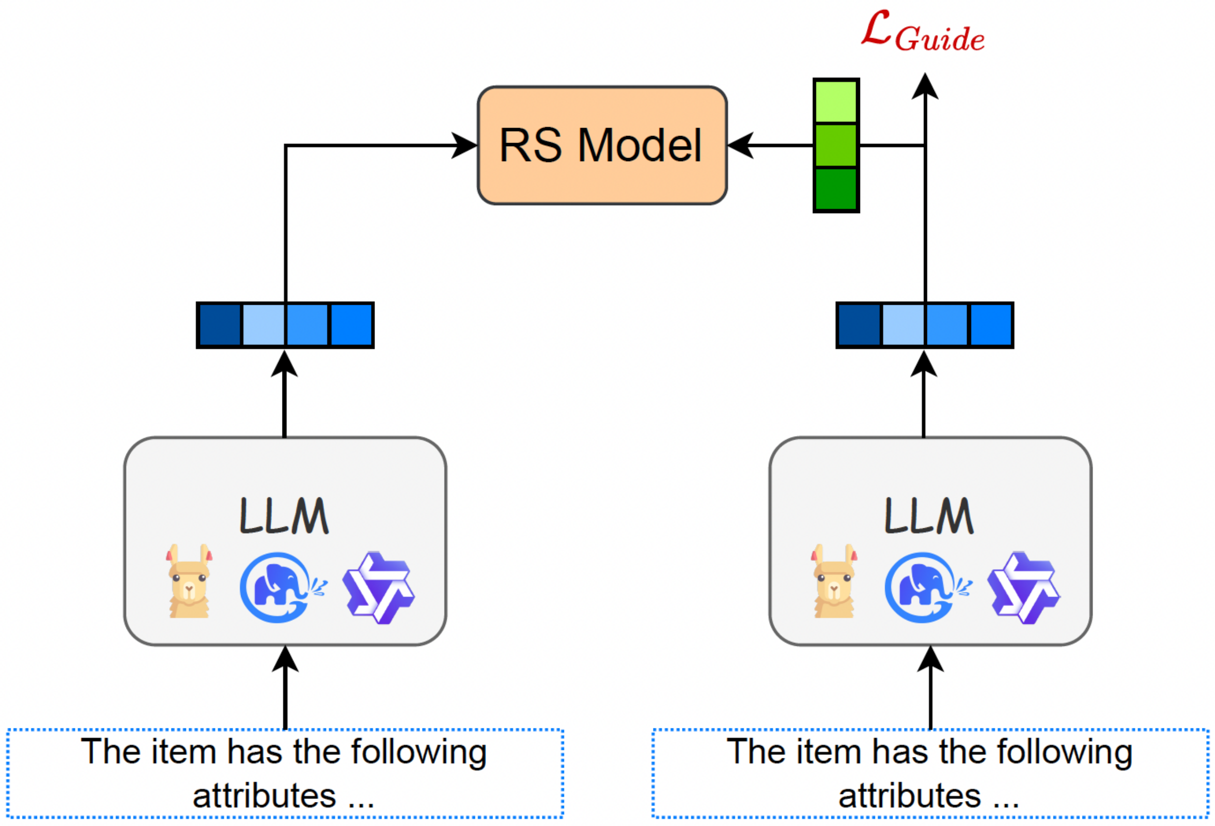
(a) Model Initialization (b) Model Distillation

Model enhancement

Model enhancement

- Embedding Utilization: Directly use LLM-derived embeddings as a semantic supplement for RS

- Embedding Guidance: Use LLM embeddings as guidance for training or parameter synthesis of RS models



(c) Embedding Utilization (d) Embedding Guidance

Agenda

1 Introduction


Zijian Zhang

2 Knowledge Enhancement


Pengyue Jia

3 Interaction Enhancement


Ziwei Liu

4.1 Model Enhancement 1


Maolin Wang

4.2 Model Enhancement 2


Yuhao Wang

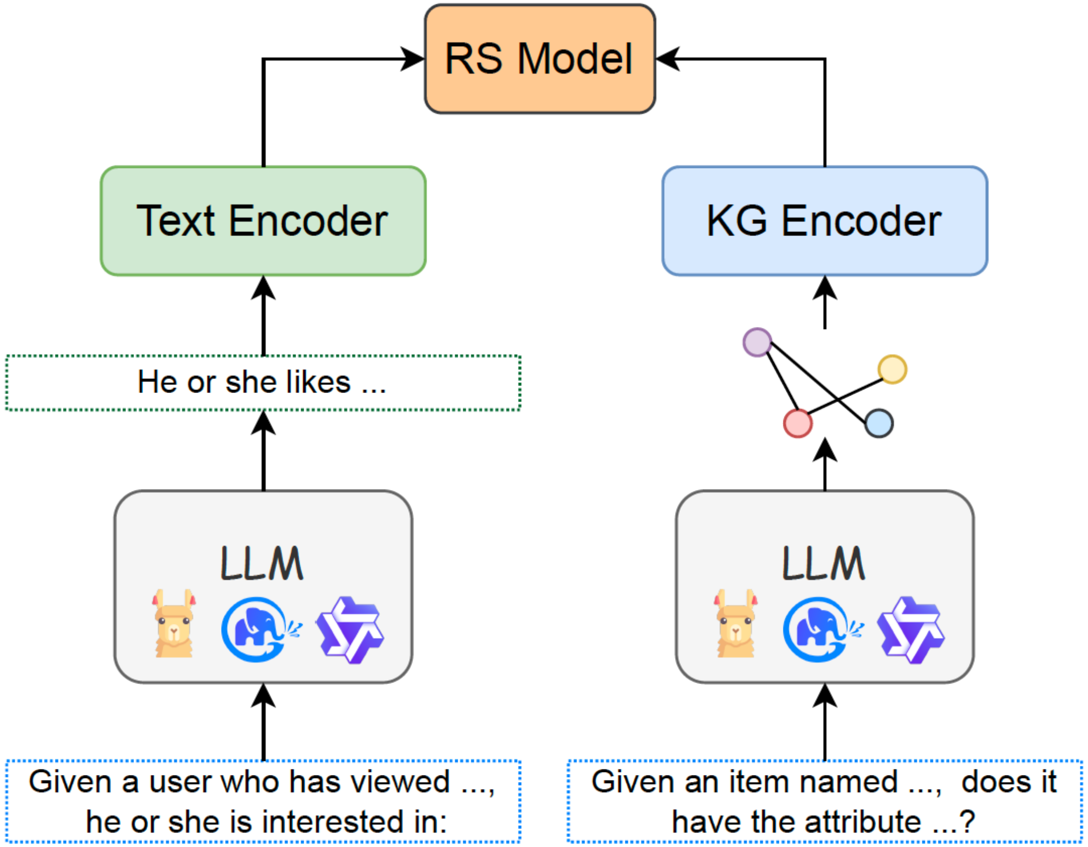
5 Conclusion


Yejing Wang

Knowledge Enhancement

- LLM owns extensive world knowledge and powerful reasoning abilities, which can supplement the RS with external knowledge.

- Categories
  - Summary Text
  - Knowledge Graph
  - Combination

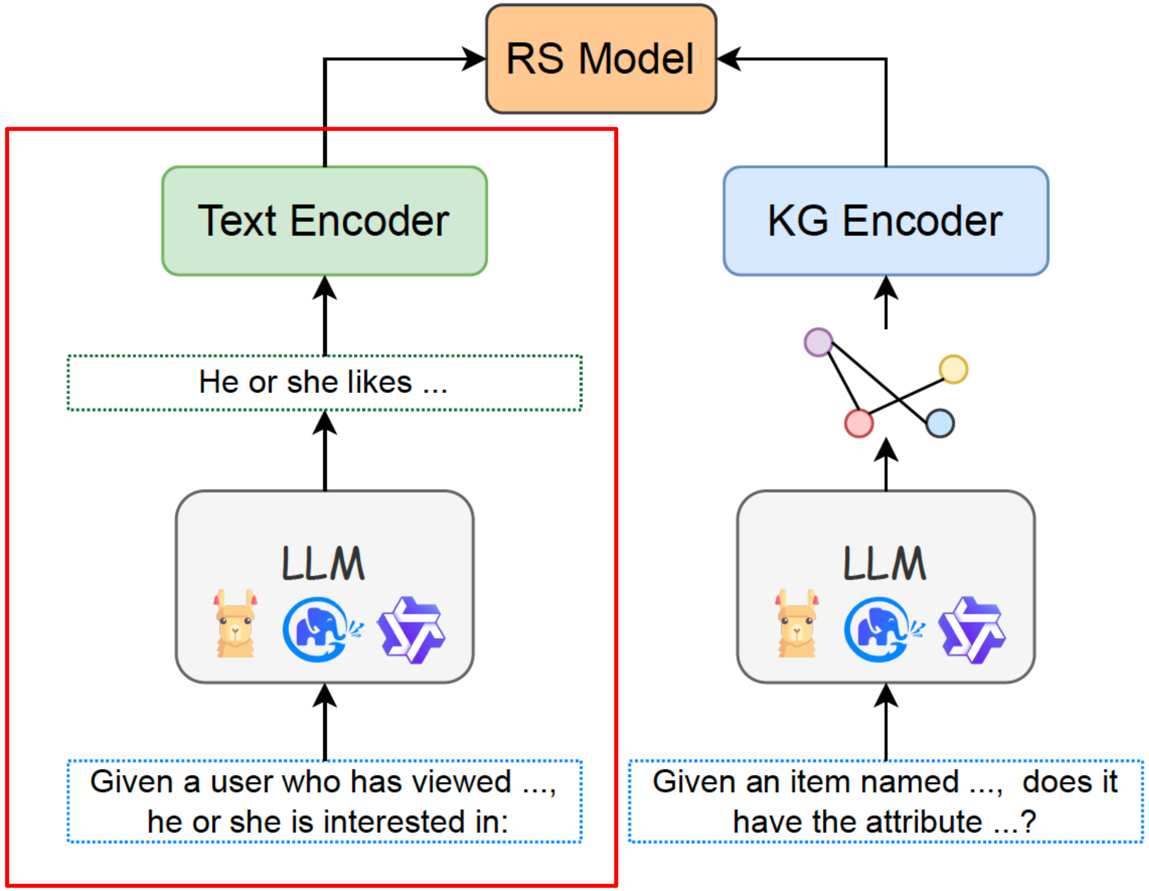


Summary Text

- This subcategory refers to utilizing LLM to summarize the characteristics of items or to reason for the preference of users.

- For example, the prompt for summarization can be “Given a user who has viewed <Browsing History>, please explain what he or she is interested in: ”.

- Categories
  - User Only
  - Item Only
  - User & Item



User Only - LANE

- Background
  - The explainability of recommendation systems is crucial for enhancing user trust and satisfaction.
- Motivation
  - Fine-tuning LLM models for recommendation tasks incurs high computational costs and alignment issues with existing systems, limiting the application potential of proven proprietary/closed-source LLM models, such as GPT-4.
- Key Components
  - Semantic embedding
  - User multi-preference extraction using zero-shot prompting
  - Semantic alignment

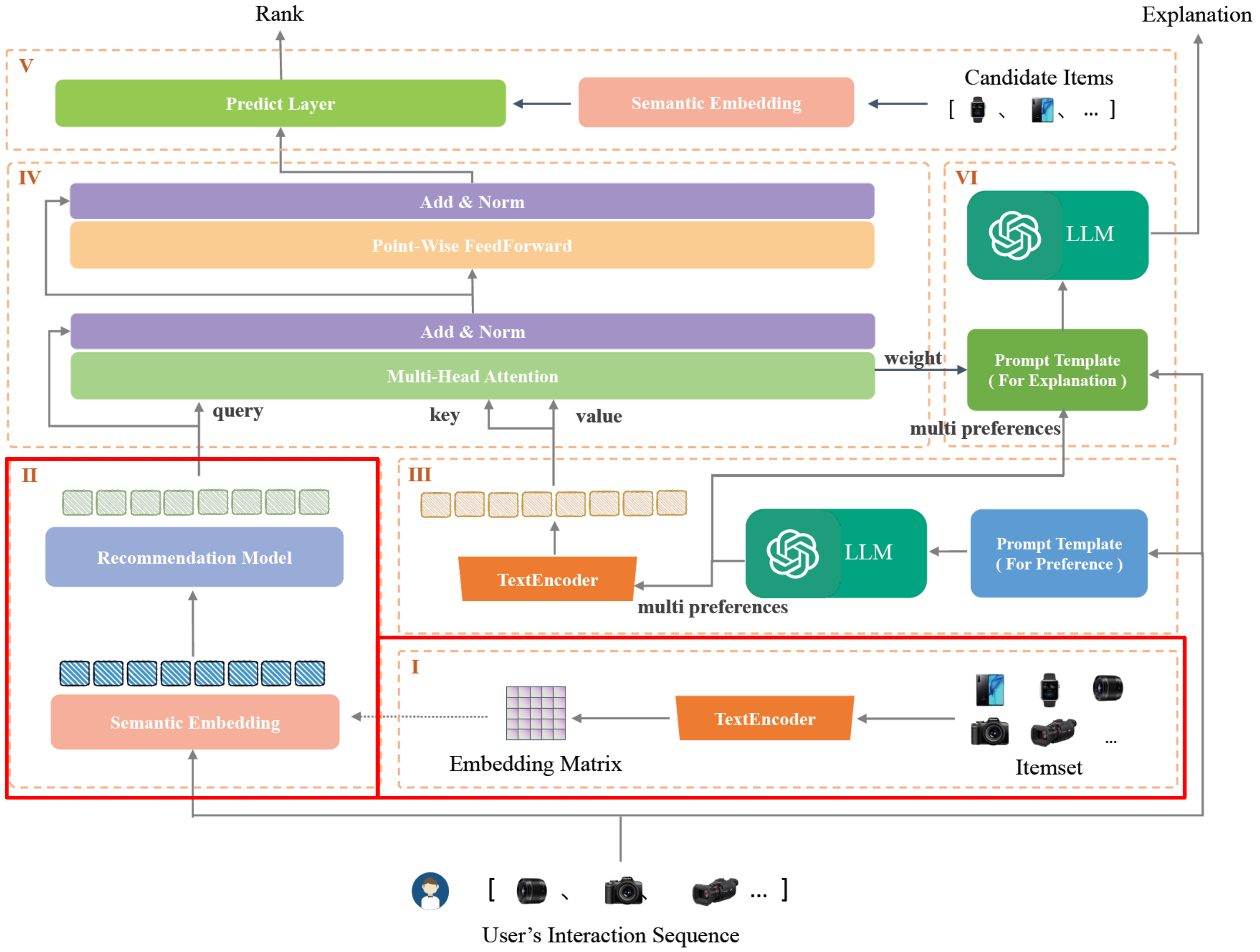
Zhao, Hongke, et al. "Lane: Logic alignment of non-tuning large language models and online recommendation systems for explainable reason generation." arXiv preprint arXiv:2407.02833 (2024).

User Only - LANE

- Semantic Embedding
  - Unlike traditional recommender systems that use item IDs, LANE encodes user history as semantic vectors using a text encoder

$$\mathbf{M} = \begin{bmatrix} \mathbf{m}_1 \\ \mathbf{m}_2 \\ \vdots \\ \mathbf{m}_n \end{bmatrix} = \mathrm{TextEncoder}\left(\begin{bmatrix} i_1 \\ i_2 \\ \vdots \\ i_n \end{bmatrix}\right)$$

- Integrated Model Module
  - LANE incorporates a standard sequential RS, but adapts its embedding layer to use the semantic vectors generated previously.



User Only - LANE

- Extracting Multi-Preferences using LLM Zero-Shot Prompting

$$\text{Preference}^u = \text{LLM}(\text{prompt}_p(S^u)),$$
$$\mathbf{P}^u = \text{TextEncoder}(\text{Preference}^u)$$

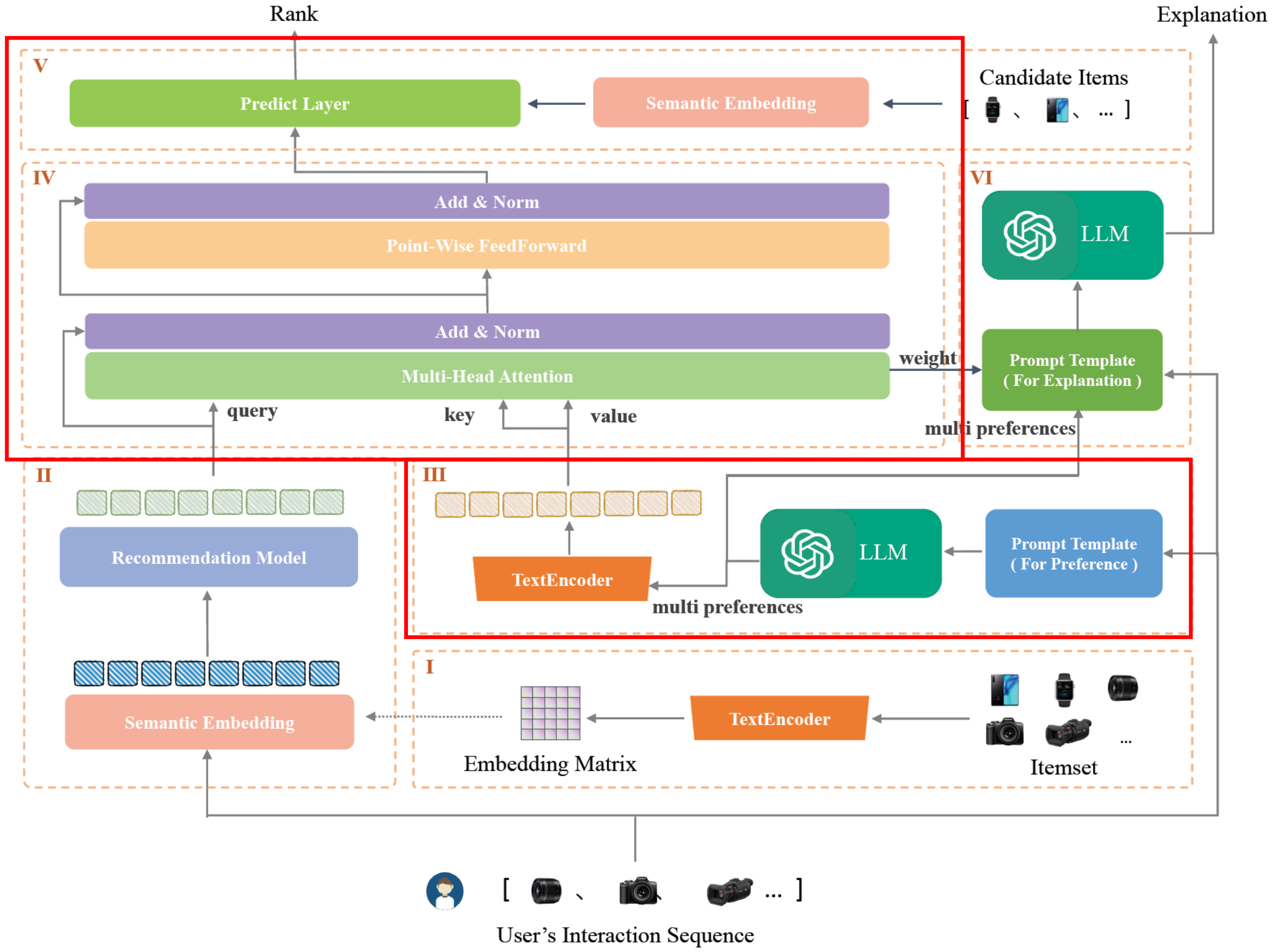
- Semantic Alignment via Multi-Head Attention

$$\text{att}^u = \text{LayerNorm}\left(\text{Multihead}(\mathbf{Q}^u, \mathbf{P}^u, \mathbf{P}^u)\right) + \mathbf{Q}^u,$$

$$\mathbf{F}^u = \begin{bmatrix} \mathbf{f}_1^u \\ \mathbf{f}_2^u \\ \vdots \\ \mathbf{f}_n^u \end{bmatrix} = \text{LayerNorm}\left(\text{FNN}(\text{att}^u)\right) + \text{att}^u$$

- Prediction

$$\mathbf{r}_{t,i}^u = \mathbf{f}_t^u \cdot \mathbf{m}_i,$$



User Only - LANE

- LANE consistently achieves significant improvements in both NDCG@10 and HR@10 across all datasets and baseline models.

Dataset	Beauty		Steam		ML-1M	
	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10	HR@10
GRU4Rec	0.2853	0.4422	0.5025	0.7492	0.5277	0.7614
LANE-GRU4Rec	0.3238	0.4825	0.5109	0.7598	0.5667	0.7844
Improv.	**13.49%**	**9.11%**	**1.67 %**	**1.41%**	**7.39%**	**3.02%**
BERT4Rec	0.224	0.3808	0.477	0.7261	0.4611	0.7151
LANE-ERT4Rec	0.2734	0.4526	0.4982	0.7495	0.5111	0.7646
Improv.	**22.05%**	**18.86%**	**4.44%**	**3.22%**	**10.84%**	**6.92%**
SASRec	0.2831	0.423	0.4789	0.728	0.5701	0.7983
LANE-SASRec	0.3511	0.5172	0.5649	0.803	0.5888	0.8106
Improv.	**24.02%**	**22.27%**	**17.96%**	**10.30%**	**3.28%**	**1.54%**

- **Practical Values:** LANE works without fine-tuning LLMs, making it easy to integrate into existing systems while also providing explainable recommendations.

Item Only - GPTAugNews

- Background
  - The explainability and informativeness of news metadata—especially category information—play a crucial role in personalized news recommendation.

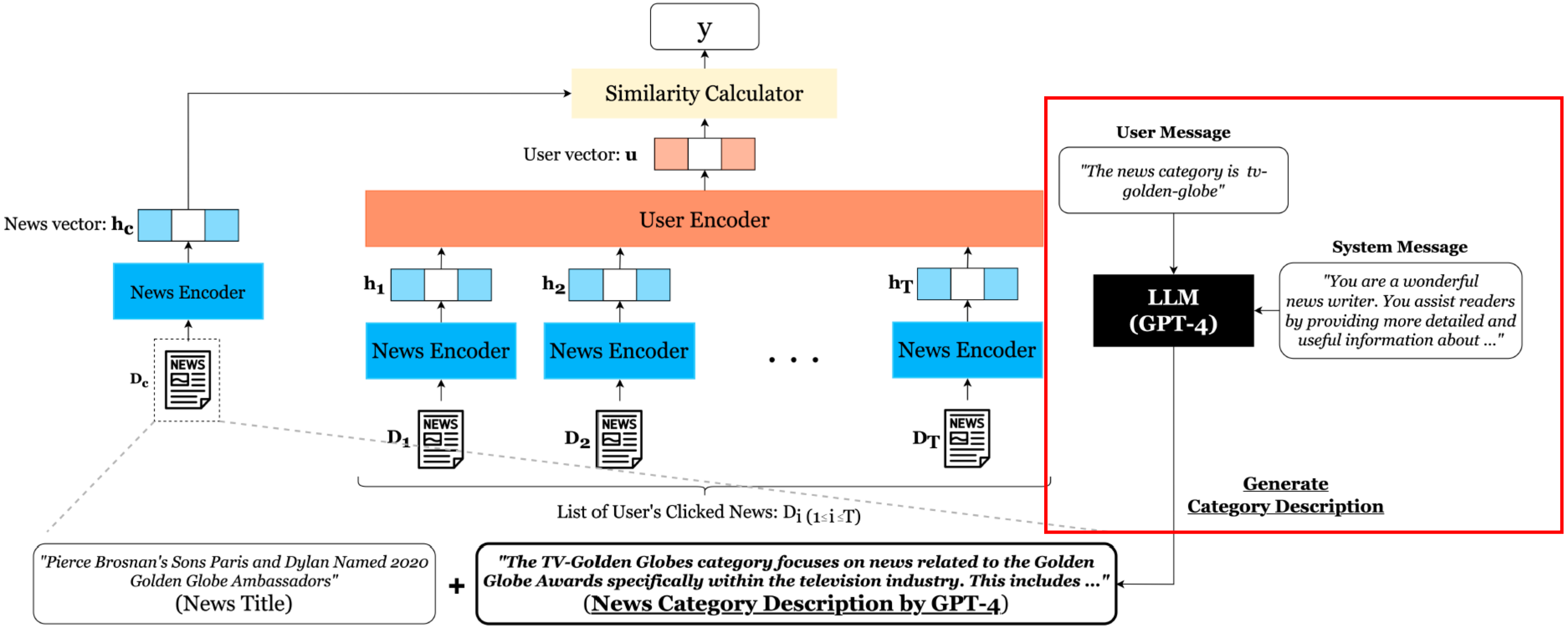
- Motivation
  - Manually creating detailed descriptions for news categories is labor-intensive.
  - Existing category templates are too generic and provide insufficient semantic guidance.
  - Pretrained language models often fail to interpret short category names effectively without richer context.

- Contribution
  - Generating descriptions with LLMs
  - Incorporating descriptions into recommendation models

Yada, Yuki, and Hayato Yamana. "News recommendation with category description by a large language model." arXiv preprint arXiv:2405.13007 (2024).

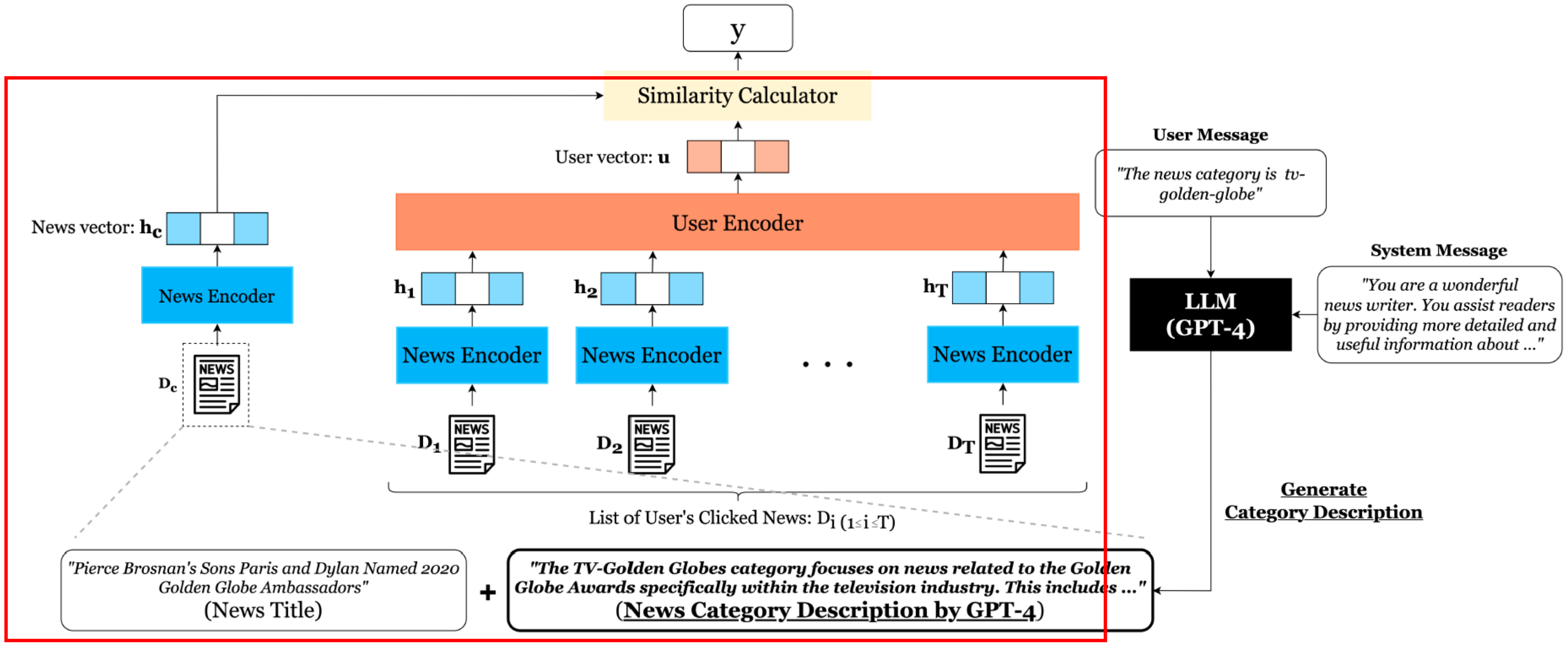
Item Only - GPTAugNews

- Category Description Generation
  - User Message + System Message -> LLM -> Generated Category Description



Item Only - GPTAugNews

- Integration into News Encoder
  - $D_{input} = D_{title}\ [sep] D_{desc}$, encoding with bert-style news encoder



Item Only - GPTAugNews

- The proposed method consistently outperforms both the title-only and template-based baselines across all models and settings, achieving up to 5.8% AUC improvement.

Recommendation Model	PLM	Method	AUC	MRR	nDCG@5	nDCG@10
NAML [19]	DistilBERT	*title only*	0.675	0.292	0.317	0.384
		title + template-based	0.690	0.295	0.327	0.393
		title + generate-description (ours)	**0.713**	**0.326**	**0.363**	**0.425**
	BERT	*title only*	0.700	0.318	0.350	0.414
		title + template-based	0.696	0.308	0.340	0.405
		title + generate-description (ours)	**0.707**	**0.322**	**0.357**	**0.420**
NRMS [21]	DistilBERT	*title only*	0.674	0.297	0.322	0.387
		title + template-based	0.675	0.311	0.341	0.400
		title + generate-description (ours)	**0.707**	**0.324**	**0.359**	**0.422**
	BERT	*title only*	0.689	0.306	0.336	0.400
		title + template-based	0.667	0.301	0.329	0.389
		title + generate-description (ours)	**0.706**	**0.320**	**0.355**	**0.418**
NPA [20]	DistilBERT	*title only*	0.700	0.311	0.344	0.408
		title + template-based	0.698	0.309	0.342	0.407
		title + generate-description (ours)	**0.707**	**0.319**	**0.354**	**0.417**
	BERT	*title only*	0.689	0.301	0.332	0.398
		title + template-based	0.694	0.314	0.345	0.410
		title + generate-description (ours)	**0.710**	**0.324**	**0.360**	**0.422**

- **Practical Values:** LLM-generated category descriptions offer a simple and generalizable way to enrich news representations, enabling better recommendations without additional training or domain-specific effort.

User & Item - KAR

- Background
  - Classical recommender systems are limited to closed-domain data and lack access to open-world knowledge, constraining their reasoning and generalization abilities.

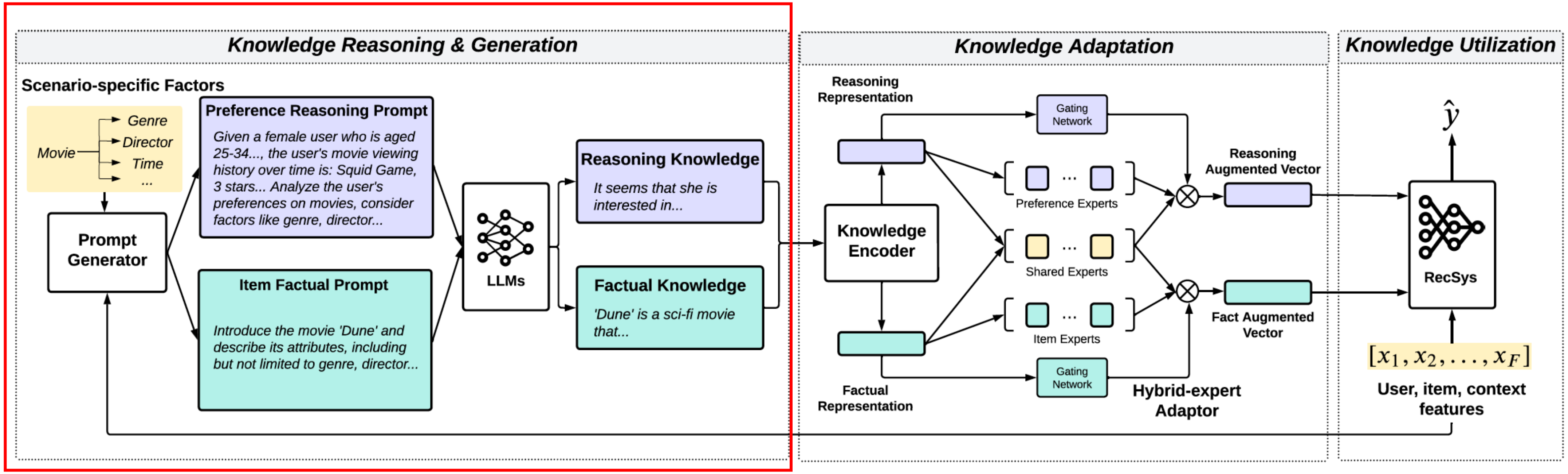
- Motivation
  - Directly using LLMs as recommenders leads to poor accuracy, high latency, and difficulty handling compositional reasoning.
  - It is challenging to extract useful, aligned, and reliable knowledge from LLMs that is compatible with recommendation models.

- Contribution
  - Propose KAR, a model-agnostic framework that extracts both reasoning knowledge about users and factual knowledge about items from LLMs.
  - Design a hybrid-expert adaptor to transform textual knowledge into dense augmented vectors aligned with recommendation space.

Xi, Yunjia, et al. "Towards open-world recommendation with knowledge augmentation from large language models." Proceedings of the 18th ACM Conference on Recommender Systems. 2024.

User & Item - KAR

- Knowledge Reasoning and Generation
  - Use LLMs with **factorized prompts** to extract two types of open-world knowledge:
    - Reasoning knowledge $k_i^{(p)}$ for user i
    - Factual knowledge $k_i^{(l)}$ for item i



User & Item - KAR

- Knowledge Adaptation and Utilization
  - Use a knowledge encoder (e.g., BERT) to encode LLM outputs
  - Apply hybrid-expert adaptor with gating networks and shared/dedicated experts to transform to

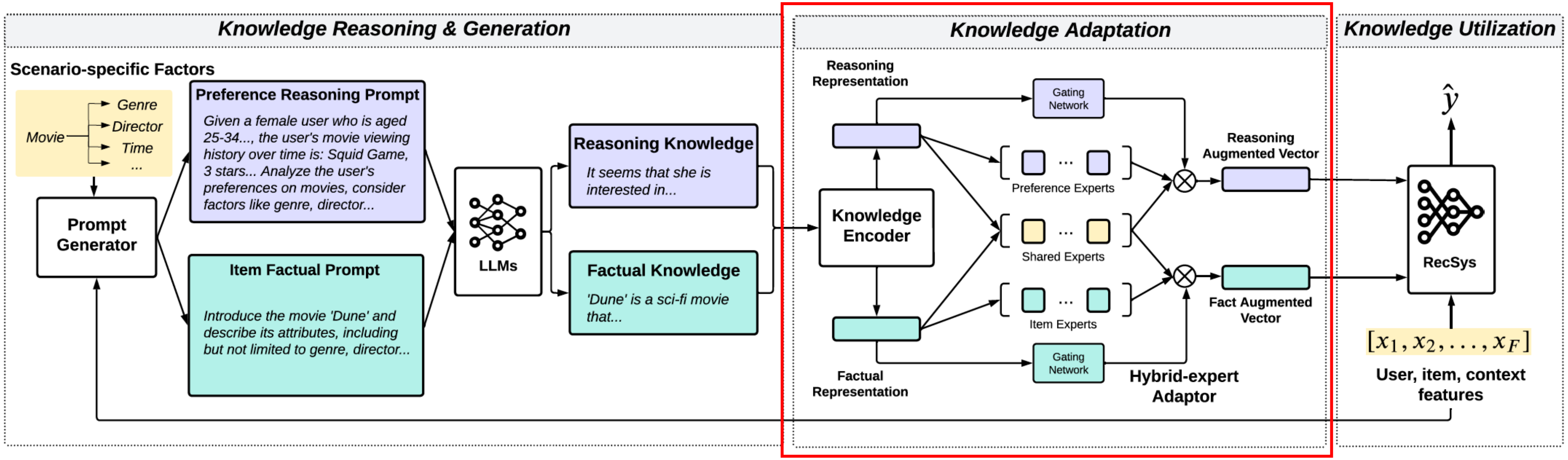
$$r_i^{(p)} = \mathrm{Aggr}(\mathrm{Encoder}(k_i^{(p)}))$$

$$r_i^{(\iota)} = \mathrm{Aggr}(\mathrm{Encoder}(k_i^{(\iota)}))$$

$$\alpha_i^p = \mathrm{Softmax}(g^p(r_i^p)), \quad \alpha_i^l = \mathrm{Softmax}(g^l(r_i^l)),$$

$$\hat{r}_i^p = \sum_{e \in S_s} \alpha_{i,e}^p \times e(r_i^p) + \sum_{e \in S_p} \alpha_{i,e}^p \times e(r_i^p),$$

$$\hat{r}_i^l = \sum_{e \in S_s} \alpha_{i,e}^l \times e(r_i^l) + \sum_{e \in S_\iota} \alpha_{i,e}^l \times e(r_i^l),$$



User & Item - KAR

- KAR consistently improves AUC and logloss across 9 CTR models, with up to **1.6% AUC and 3.1% logloss gain**

Backbone Model	AUC			Logloss		
	base	KAR	improv.	base	KAR	improv.
DCNv2	0.7924	**0.8049***	1.58 %	0.5451	**0.5315***	2.50 %
DCN	0.7929	**0.8043***	1.46 %	0.5457	**0.5319***	2.53 %
DeepFM	0.7928	**0.8041***	1.44 %	0.5462	**0.5321***	2.57 %
FiBiNet	0.7925	**0.8051***	1.59 %	0.5450	**0.5310***	2.56 %
AutoInt	0.7934	**0.8060***	1.59 %	0.5440	**0.5297***	2.65 %
FiGNN	0.7944	**0.8054***	1.39 %	0.5424	**0.5307***	2.16 %
xDeepFM	0.7942	**0.8041***	1.25 %	0.5457	**0.5317***	2.57 %
DIEN	0.7960	**0.8059***	1.25 %	0.5469	**0.5298***	3.13 %
DIN	0.7975	**0.8066***	1.15 %	0.5387	**0.5304***	1.55%

$*$ denotes statistically significant improvement (t-test with p-value < 0.05) over the backbone model.

- **Practical Values:** KAR offers a practical, modular, and efficient solution to enhance recommenders using LLM-generated knowledge, without modifying their internal architecture.

User & Item - SLIM

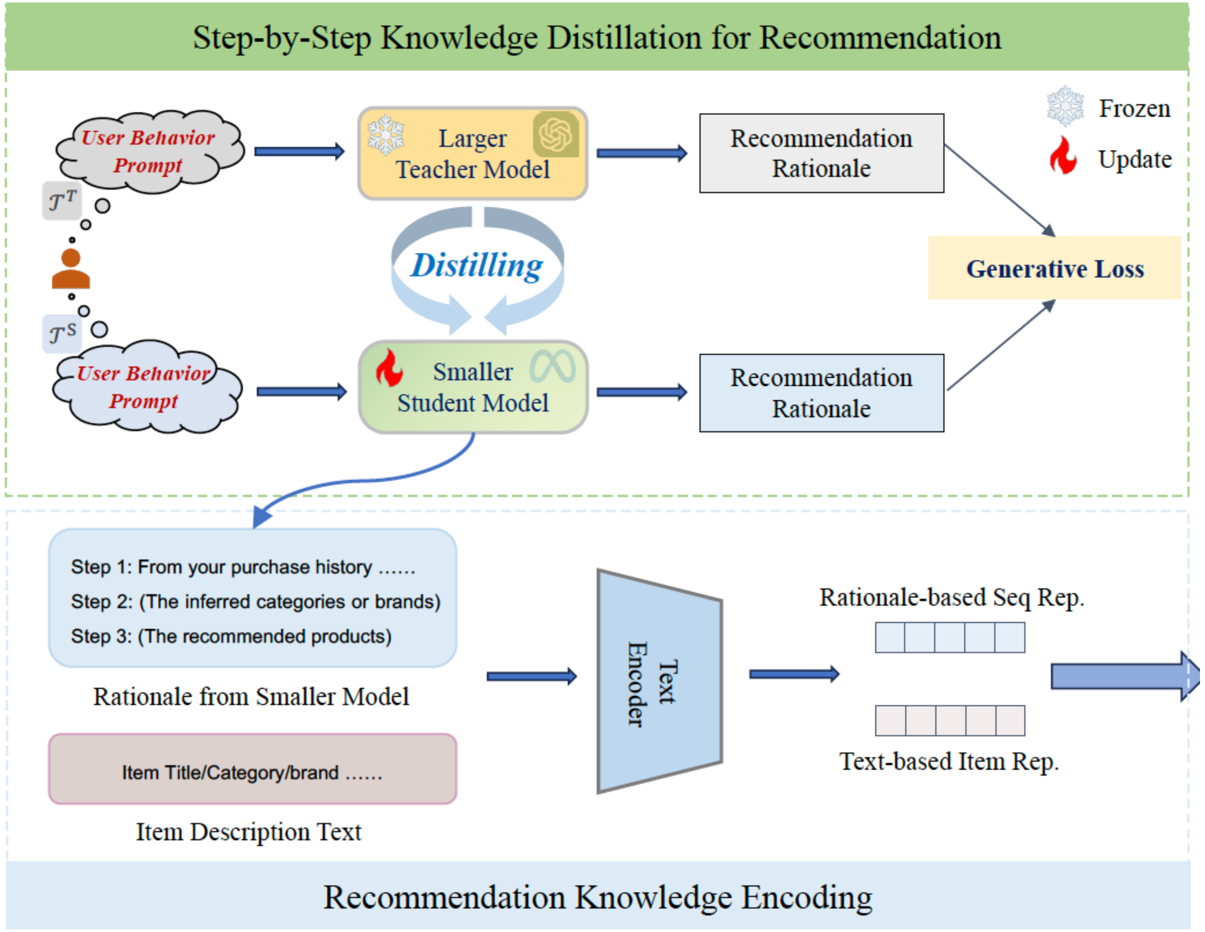
- Background
  - LLMs show strong reasoning ability and knowledge but are expensive and hard to deploy in real systems.
- Motivation
  - One-shot generation by LLMs often produces irrelevant or incorrect recommendations.
  - The scale and inference cost of LLMs are prohibitive for production use.
- Contribution
  - Propose SLIM, use CoT prompting with LLMs to generate rationales.
  - Distill these rationales into a small LLM for efficient deployment.
  - Enable both ID-based and ID-agnostic recommendation by encoding rationales and item descriptions via a text encoder.

Wang, Yuling, et al. "Can small language models be good reasoners for sequential recommendation?." Proceedings of the ACM Web Conference 2024. 2024.

User & Item - SLIM

- Step-by-Step Knowledge Distillation
  - Chain-of-Thought Prompting on Teacher Model with user prompt:
    - Summarize preferences
    - Infer categories or brands
    - Recommend specific products
  - Fine-tune Student Model with Rationales:
    - The objective is to minimize generative loss

$$\mathcal{L}_{\text{distill}} = \sum_{u \in U'} \sum_{t=1}^{|r'_u|} \log P_\theta(r'_{u,t} | p'_u, r'_{u,<t})$$



User & Item - SLIM

- Rationale Encoding and Enhanced Recommendation
  - Text Encoding:
    - Convert student-generated rationales r_u and item descriptions f_i into embeddings

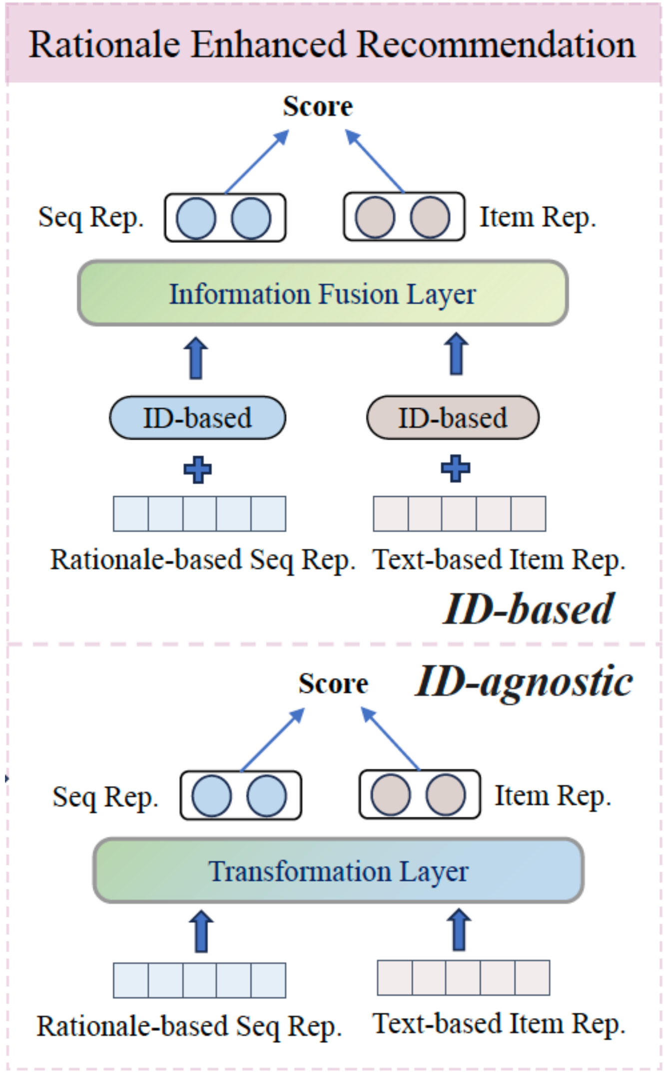
$$\mathbf{z}_{\text{text},i} = \text{TextEncoder}(f_i), \quad \mathbf{s}_{\text{text},u} = \text{TextEncoder}(r_u)$$

  - ID-Based Integration:
    - Use an Information Fusion Layer to combine text and ID embeddings

$$\mathbf{z}_i = g_f([g_l(\mathbf{z}_{\text{text},i}); \mathbf{z}_{\text{id},i}]), \quad \mathbf{s}_u = g_f([g_l(\mathbf{s}_{\text{text},u}); \mathbf{s}_{\text{SeqEnc},u}])$$

  - ID-Agnostic Matching:
    - Apply a transformation to match rationale and item vectors

$$\mathbf{z}_i = g_t(\mathbf{z}_{\text{text},i}), \quad \mathbf{s}_u = g_t(\mathbf{s}_{\text{text},u})$$



User & Item - SLIM

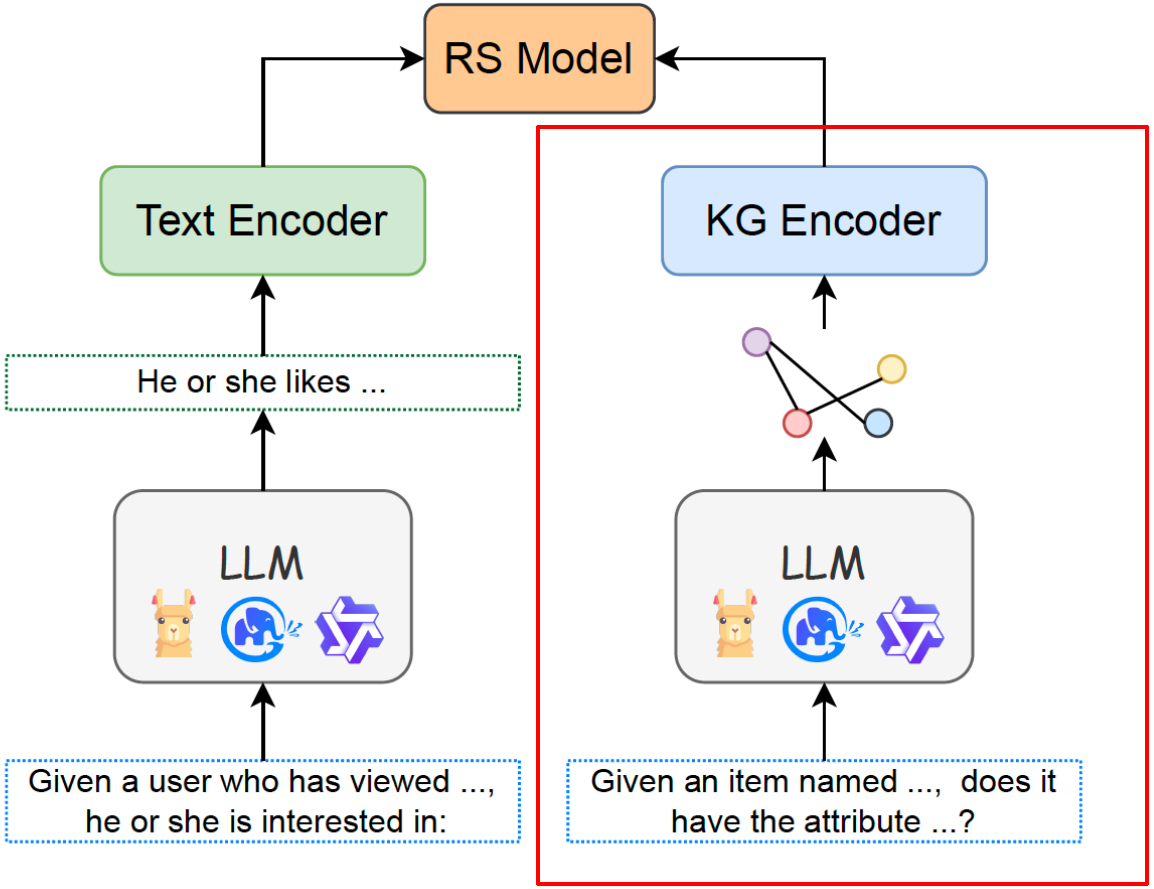
- SLIM consistently improves performance across three datasets and three backbone models (GRU4Rec, SASRec, SRGNN), outperforming both traditional and ChatGPT-augmented baselines.

Methods	Games			Food			Home		
	NDCG@10	Hit @10	Hit @20	NDCG@10	Hit @10	Hit @20	NDCG@10	Hit @10	Hit @20
GRU4Rec	17.61 ± 0.18	30.87 ± 0.56	42.39 ± 0.62	9.10 ± 0.30	15.27 ± 0.58	19.51± 0.24	2.19 ± 0.21	4.17 ± 0.39	7.53 ± 0.64
$GRU4Rec^+$	27.33 ± 0.53	44.06 ± 0.79	56.53 ± 1.24	17.75± 0.78	31.10 ± 1.09	45.01 ± 1.46	12.19 ± 1.02	26.76 ± 2.58	49.10 ± 3.82
$SLIM^-$	27.70 ± 0.47	45.13± 0.56	57.70 ± 0.37	17.97 ± 0.70	31.78 ± 1.47	46.88 ± 2.10	13.59 ± 1.05	30.30 ± 2.01	55.85 ± 3.55
SLIM	**28.37 ± 0.41**	**45.68 ± 0.53**	**58.09 ± 0.58**	**18.32 ± 0.53**	**32.56 ± 1.30**	**46.92 ± 1.82**	**15.64 ± 0.51**	**34.33 ± 1.53**	**62.93 ± 3.46**
Improv.	3.81%	3.68%	2.76%	3.21%	4.69%	4.24%	28.3%	28.29%	28.17%
SASRec	22.73 ± 0.28	37.77 ± 0.52	51.53 ± 0.39	26.78 ± 0.24	35.78 ± 0.36	43.32 ± 0.55	2.66 ± 0.22	5.56 ± 0.72	14.93 ± 1.53
$SASRec^+$	27.46 ± 0.19	44.88 ± 0.63	58.90 ± 0.38	30.95 ± 0.38	44.98 ± 0.53	55.61 ± 1.12	5.58 ± 0.10	11.09 ± 0.16	20.69 ± 0.77
$SLIM^-$	**31.58 ± 0.35**	50.83 ± 0.62	63.45 ± 0.71	32.65 ± 0.15	48.01 ± 0.48	59.25 ± 0.56	5.95 ± 0.32	11.83 ± 0.62	**22.43 ± 0.54**
SLIM	31.43 ± 0.39	**51.11 ± 0.82**	**64.10 ± 0.26**	**32.80 ± 0.40**	**48.27 ± 0.64**	**59.30 ± 0.89**	**6.01 ± 0.19**	**12.01 ± 0.38**	22.29 ± 0.85
Improv.	14.46%	13.88%	8.83%	5.98%	7.31%	6.64%	7.71%	8.3%	7.73%
SRGNN	16.45 ± 0.22	29.29 ± 0.14	40.99 ± 0.52	10.99 ± 2.07	20.32 ± 4.30	32.14 ± 6.55	5.04 ± 0.83	13.48 ± 2.23	37.22 ± 3.85
$SRGNN^+$	21.54 ± 0.64	36.77 ± 1.05	49.11 ± 1.54	11.91 ± 0.71	21.39 ± 1.91	33.63 ± 3.41	11.61 ± 1.14	25.22 ± 2.46	43.85 ± 3.58
$SLIM^-$	22.35 ± 1.48	37.69 ± 1.53	51.29 ± 0.59	**12.92 ± 0.78**	**23.80 ± 1.60**	**37.22 ± 2.75**	11.25 ± 1.42	24.28 ± 3.19	44.05 ± 5.97
SLIM	**23.77 ± 0.20**	**39.81 ± 0.52**	**52.34 ± 0.63**	12.38 ± 0.51	22.98 ± 1.30	36.44 ± 1.72	**12.29 ± 1.39**	**26.51 ± 2.71**	**47.01 ± 3.61**
Improv.	10.35%	8.27%	6.58%	3.95%	7.43%	8.36%	5.86%	5.11%	7.21%

- **Practical Values:** SLIM enables small language models to act as effective reasoners in sequential recommendation by distilling step-by-step rationales from LLMs at a fraction of the computational cost.

Knowledge Graph

- Summary text is a type of unstructured knowledge, while the **structured Knowledge Graph (KG)** may drive **better integration**.
- Many work explore how to **apply the LLM to generate a KG** or **augment an existing KG** for enhancing RS.
- Categories
  - Generation
  - Completion & Fusion



Generation - LLMRG

- Background
  - Recent works use knowledge graphs or behavior graphs to enhance reasoning, but most rely on pre-defined structures and cannot flexibly adapt to personalized needs.
  - LLMs exhibit powerful reasoning and abductive inference ability.

- Motivation
  - Can LLMs generate personalized multi-hop reasoning chains for sequential RS?
  - How can we verify, score, and incorporate these LLM-generated chains into real-world recommenders?
  - Can this be done without relying on explicit external knowledge graphs?

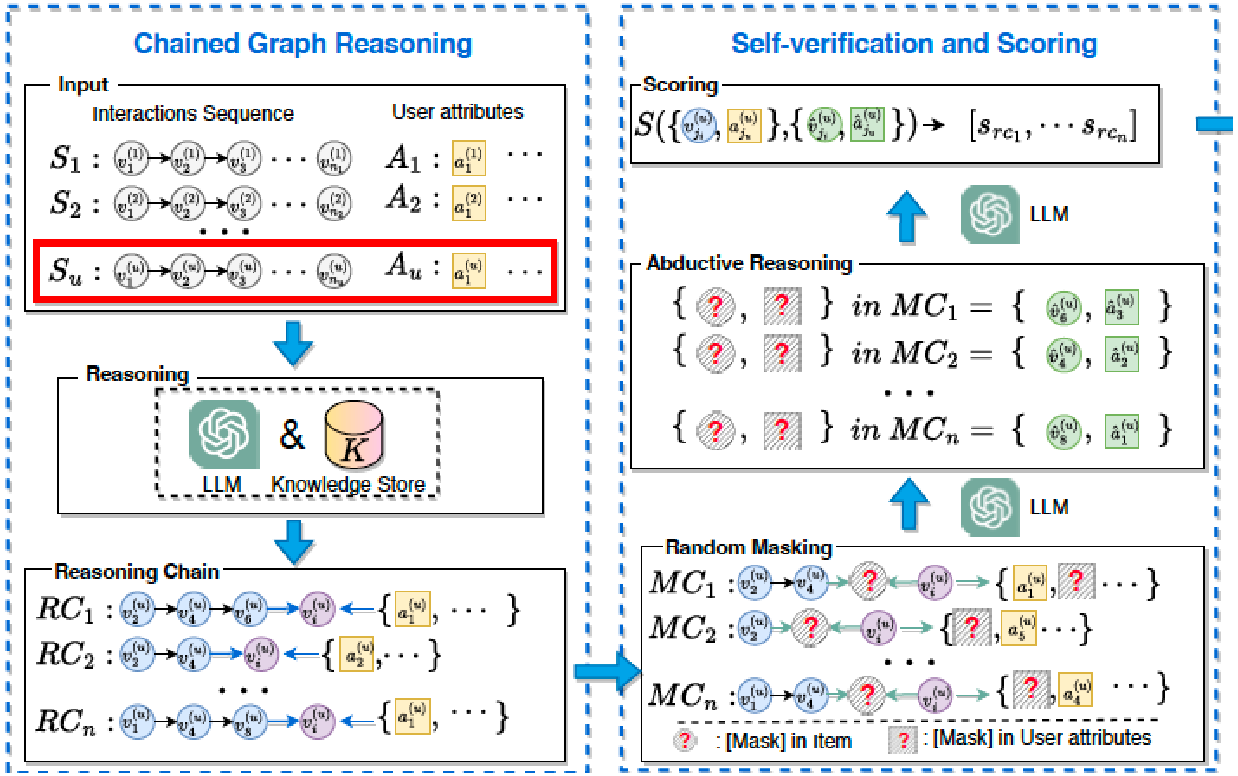
- Contributions
  - Adaptive Reasoning Module

Wang, Yan, et al. "Enhancing recommender systems with large language model reasoning graphs." arXiv preprint arXiv:2308.10835 (2023).

Generation - LLMRG

- Chain Construction and Verification
  - Chained Graph Reasoning
    - LLM builds reasoning chains RC from interaction sequence S_u and attribute A_u
  - Random Masking & Abductive Reasoning
    - Each chain is perturbed into masked candidates MC_i.
    - The LLM performs abductive reasoning to infer whether masked parts still lead to the correct reasoning path.
  - Scoring

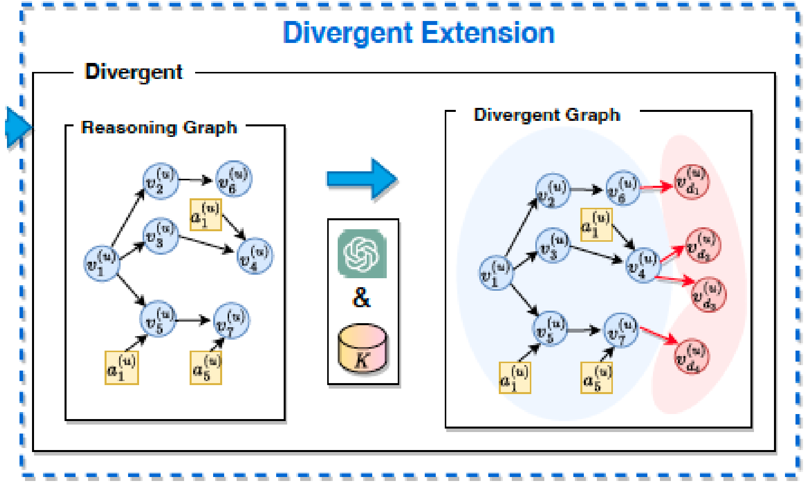
$$S(RC_i, A_u, S_u) \rightarrow [s_{RC_1}, \ldots, s_{RC_n}]$$

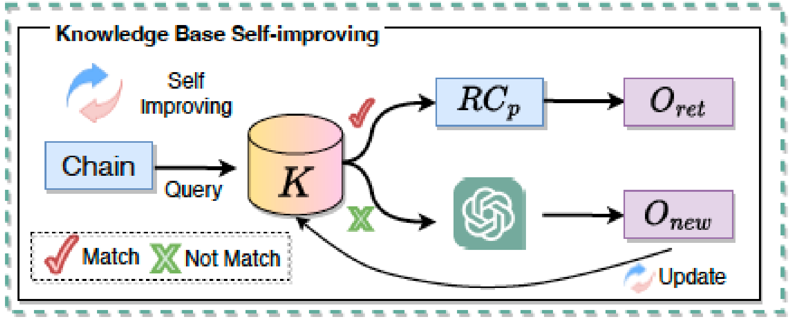


Generation - LLMRG

- Reasoning Enhancement and Model Fusion
  - Divergent Extension:
    - From the reasoning graph, LLM identifies alternative paths by querying unexplored nodes, forming divergent reasoning graphs.
  - Self-Improving Knowledge Base:
    - Mismatches between retrieved reasoning and newly inferred reasoning are used to update the knowledge base.
  - Final Fusion:

$$E_{\text{fusion}} = \text{Fusion}(E_{\text{base}}, E_{\text{ori}}, E_{\text{div}})$$





Generation - LLMRG

- LLMRG improves NDCG and HR by over 25–30% on ML-1M, Amazon Beauty, and Amazon Clothing, outperforming all baselines.

Dataset	Metric	FDSA			BERT4Rec			CL4SRec			DuoRec		
		Original	GPT3.5	GPT4	Original	GPT3.5	GPT4	Original	GPT3.5	GPT4	Original	GPT3.5	GPT4
ML-1M	HR@5	0.0909	+ 20.70%	+ 25.79%	0.1124	+ 26.67%	+ 32.56%	0.1141	+ 19.98%	+ 21.02%	0.2011	+ 12.87%	+ 14.76%
	HR@10	0.1631	+ 17.93 %	+ 22.87%	0.1910	+ 13.52 %	+ 16.49 %	0.1866	+ 17.30 %	+ 19.31 %	0.2837	+ 14.10 %	+ 15.53 %
	NDCG@5	0.0599	+ 21.33%	+ 30.27 %	0.0713	+ 25.74 %	+ 32.82 %	0.0721	+ 14.97 %	+ 16.78 %	0.1265	+ 23.55 %	+ 26.01 %
	NDCG@10	0.0878	+ 21.78%	+ 28.25%	0.0980	+ 23.34%	+ 28.06%	0.1013	+ 17.67%	+ 20.42%	0.1663	+ 12.86%	+ 13.77%
Amazon Beauty	HR@5	0.0237	+ 13.89 %	+ 17.53 %	0.0201	+ 19.17 %	+ 23.22 %	0.0398	+ 11.15 %	+ 14.15 %	0.0552	+ 9.31 %	+ 11.93 %
	HR@10	0.0418	+ 15.02 %	+ 17.78 %	0.0413	+ 17.79 %	+ 22.14 %	0.0664	+ 10.22 %	+ 11.32 %	0.0839	+ 5.14 %	+ 6.61 %
	NDCG@5	0.0195	+ 16.20 %	+ 18.64 %	0.0192	+ 14.21 %	+ 17.63 %	0.0221	+ 8.45 %	+ 10.18 %	0.0350	+ 7.42 %	+ 9.24 %
	NDCG@10	0.0275	+ 14.78 %	+ 17.64 %	0.0263	+ 11.53 %	+ 14.76 %	0.0322	+ 8.17 %	+ 9.68 %	0.0447	+ 6.67 %	+ 7.95 %
Amazon Clothing	HR@5	0.0119	+ 20.67 %	+ 23.92 %	0.0128	+ 16.09 %	+ 19.10 %	0.0166	+ 7.90 %	+ 10.92 %	0.0190	+ 9.98 %	+ 11.40 %
	HR@10	0.0197	+ 14.45 %	+ 17.88 %	0.0202	+ 10.52 %	+ 13.72 %	0.0273	+ 11.21 %	+ 14.99 %	0.0311	+ 7.65 %	+ 9.48 %
	NDCG@5	0.0073	+ 8.16 %	+ 10.86 %	0.0081	+ 7.39 %	+ 10.39 %	0.0093	+ 6.02 %	+ 9.09 %	0.0118	+ 6.74 %	+ 9.19 %
	NDCG@10	0.0109	+ 6.01 %	+ 8.13 %	0.0113	+ 5.21 %	+ 5.94 %	0.0125	+ 4.32 %	+ 8.07 %	0.0155	+ 7.89 %	+ 9.29 %

- **Practical Values:** LLMRG enables LLM-based, adaptive, multi-hop reasoning without external knowledge graphs, improving recommendation via personalized and verifiable chains.

Completion & Fusion - CoLaKG

- Background
  - Existing KG-based recommenders often convert textual information into IDs, ignore semantic connections, and struggle to capture high-order relations due to GNNs' limitations.
- Motivation
  - KGs are incomplete or too sparse
  - Converting entities into IDs discards rich semantic information, and GNNs fail to effectively propagate information over long distances in the KG.
- Contributions
  - Item-item semantic graph
  - Representation alignment and neighbor augmentation modules

Cui, Ziqiang, et al. "Comprehending Knowledge Graphs with Large Language Models for Recommender Systems." arXiv preprint arXiv:2410.12229 (2024).

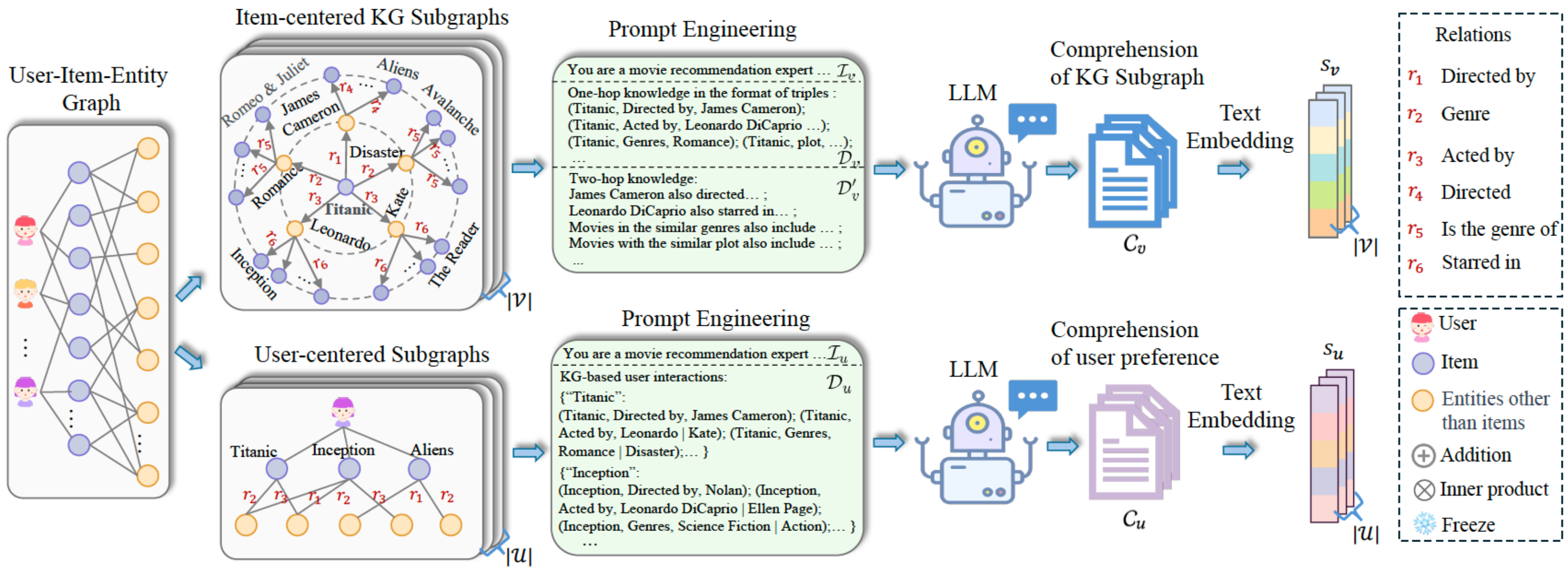
Completion & Fusion - CoLaKG

- KG Comprehension via LLM
  - Item-centered KG Subgraph Construction
  - Text Conversion & Prompting
  - Embedding Generation
    - Use pre-trained model P to obtain semantic embedding

$$T_v = \{(v, r, e)\}$$

$$C_v = \mathrm{LLMs}(I_v, D_v, D'_v)$$

$$s_v = P(C_v)$$



Completion & Fusion - CoLaKG

- Semantic Graph Construction & Integration
  - Semantic Item-Item Graph
    - Use cosine similarity to define edge weights
    - Construct graph
  - Representation Integration
    - Align semantic embeddings to ID space
    - Fuse with ID embeddings
  - Neighbor Augmentation

$$r(v_i, v_j) = \mathrm{sim}(s_{v_i}, s_{v_j})$$

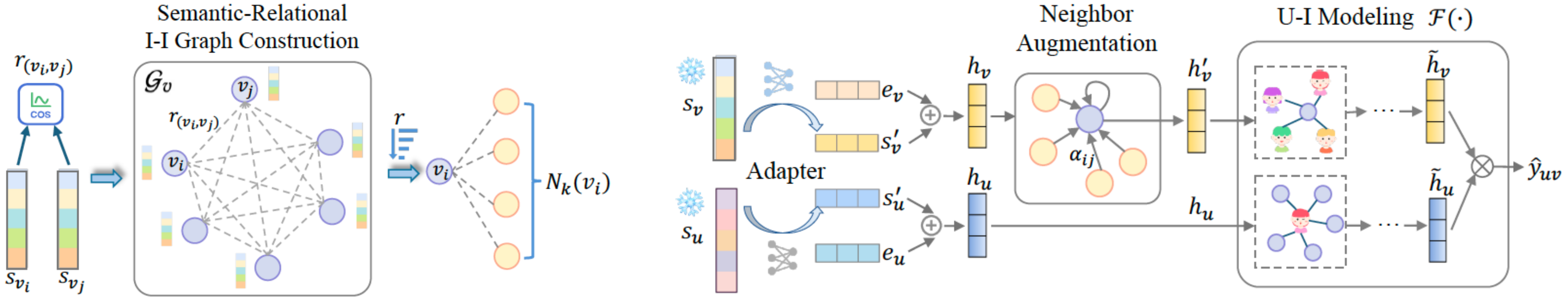
$$\mathcal{G}_v = \{(v_i, r(v_i, v_j), v_j)\}$$

$$s'_v = \sigma(W_1 s_v), \quad s'_u = \sigma(W_2 s_u)$$

$$h_v = \frac{1}{2}(e_v + s'_v), \quad h_u = \frac{1}{2}(e_u + s'_u)$$

$$w_{ij} = a(W s_{v_i} \| W s_{v_j}), \quad \alpha_{ij} = \mathrm{softmax}_j(w_{ij})$$

$$\mathbf{h}'_{v_i} = \sigma\left(\frac{1}{2}\left(\mathbf{h}_{v_i} + \sum\nolimits_{j \in N_k(v_i)} \alpha_{ij} \mathbf{h}_{v_j}\right)\right)$$



Completion & Fusion - CoLaKG

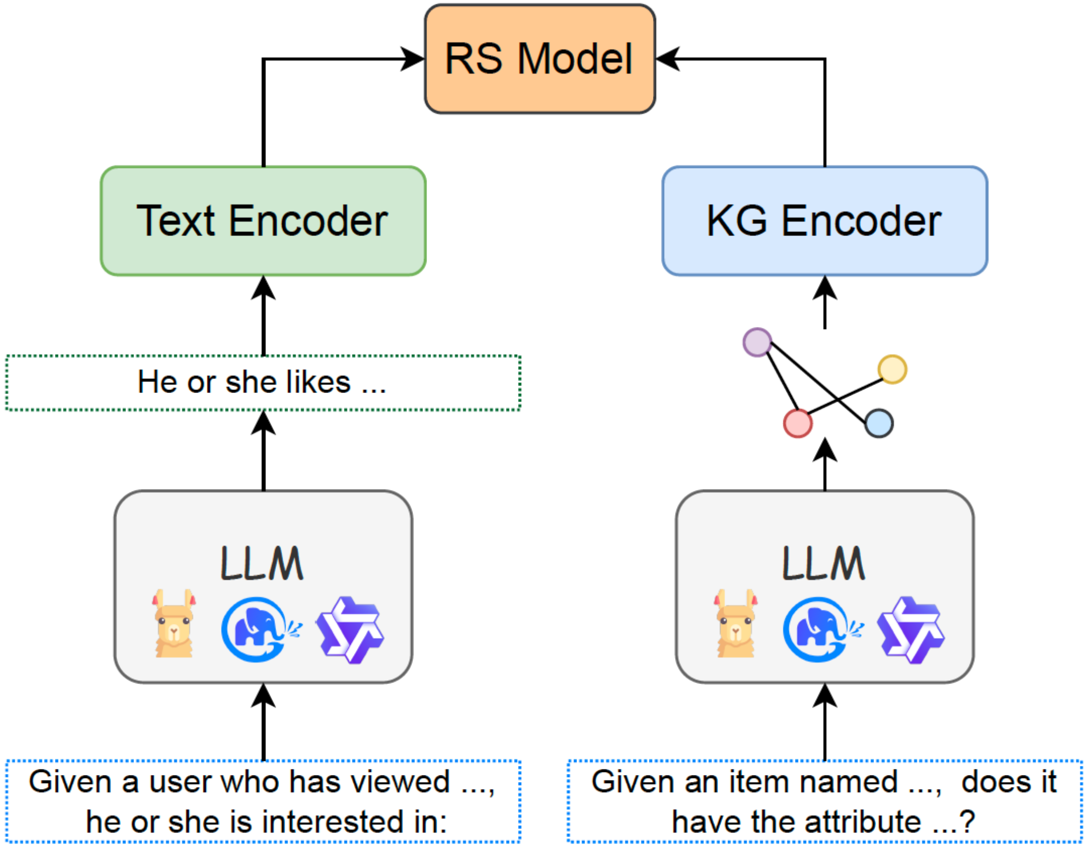
- CoLaKG achieves the best performance across all four datasets, demonstrating strong generalization and semantic reasoning ability.

Model	MovieLens				Last-FM				MIND				Funds			
	R@10	N@10	R@20	N@20	R@10	N@10	R@20	N@20	R@10	N@10	R@20	N@20	R@10	N@10	R@20	N@20
BPR-MF	0.1257	0.3100	0.2048	0.3062	0.1307	0.1352	0.1971	0.1685	0.0315	0.0238	0.0537	0.0310	0.4514	0.3402	0.5806	0.3809
NFM	0.1346	0.3558	0.2129	0.3379	0.2246	0.2327	0.3273	0.2830	0.0495	0.0356	0.0802	0.0458	0.4388	0.3187	0.5756	0.3651
LightGCN	0.1598	0.3901	0.2512	0.3769	0.2589	0.2799	0.3642	0.3321	0.0624	0.0492	0.0998	0.0609	0.4992	0.3778	0.6353	0.4204
CKE	0.1524	0.3783	0.2373	0.3609	0.2342	0.2545	0.3266	0.3001	0.0526	0.0417	0.0822	0.0510	0.4926	0.3702	0.6294	0.4130
RippleNet	0.1415	0.3669	0.2201	0.3423	0.2267	0.2341	0.3248	0.2861	0.0472	0.0364	0.0785	0.0451	0.4764	0.3591	0.6124	0.4003
KGAT	0.1536	0.3782	0.2451	0.3661	0.2470	0.2595	0.3433	0.3075	0.0594	0.0456	0.0955	0.0571	0.5037	0.3751	0.6418	0.4182
KGIN	0.1631	0.3959	0.2562	0.3831	0.2562	0.2742	0.3611	0.3215	0.0640	0.0518	0.1022	0.0639	0.5079	0.3857	0.6428	0.4259
KGCL	0.1554	0.3797	0.2465	0.3677	0.2599	0.2763	0.3652	0.3284	0.0671	0.0543	0.1059	0.0670	0.5071	0.3877	0.6355	0.4273
KGRec	0.1640	0.3968	0.2571	0.3842	0.2571	0.2748	0.3617	0.3251	0.0627	0.0506	0.1003	0.0625	0.5104	0.3913	0.6467	0.4304
RLMRec	0.1613	0.3920	0.2524	0.3787	0.2597	0.2812	0.3651	0.3335	0.0619	0.0486	0.0990	0.0602	0.4988	0.3784	0.6351	0.4210
CoLaKG	**0.1699**	**0.4130**	**0.2642**	**0.3974**	**0.2738**	**0.2948**	**0.3803**	**0.3471**	**0.0698**	**0.0562**	**0.1087**	**0.0684**	**0.5273**	**0.4012**	**0.6524**	**0.4392**

- **Practical Values:** By leveraging LLMs to comprehend and augment KG information, CoLaKG bridges the gap between structured knowledge and collaborative filtering, resulting in more robust and semantically aligned recommendation

Combination

- In consideration of the effectiveness of Summary text and Knowledge graph, some work also resort to combine both of them.



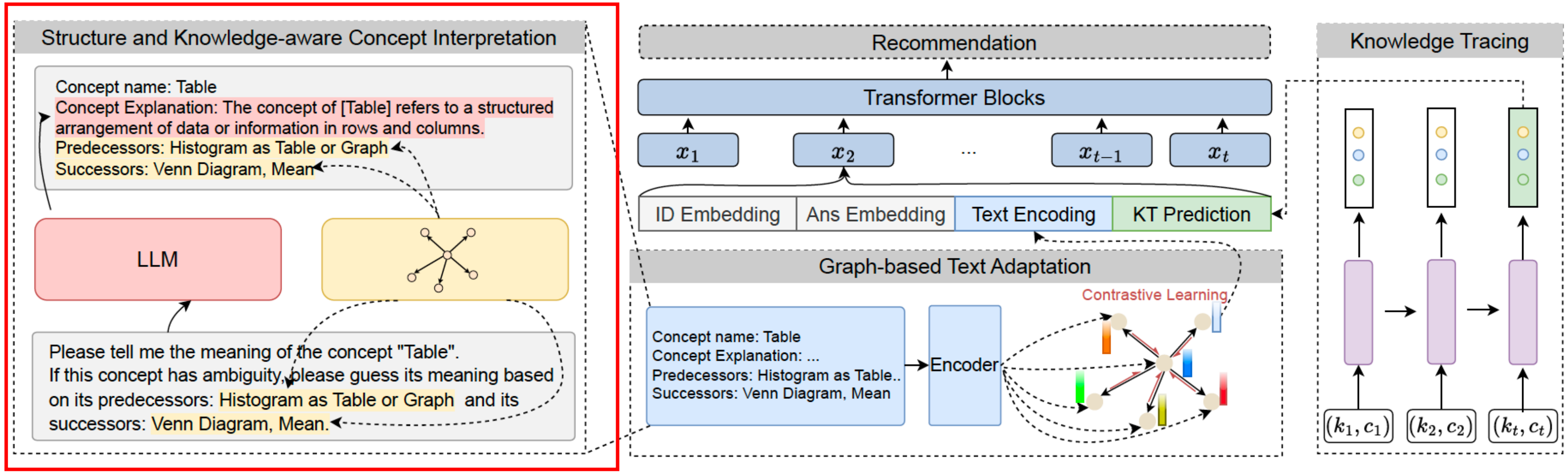
Combination - SKarRec

- Background
  - Existing KT and recommendation models often rely on ID-based concept representations, which fail to generalize and lack interpretability.
- Motivation
  - Prior methods overlook the textual and relational information between concepts, which leads to limited generalization, especially when dealing with unseen or ambiguous concepts.
- Contributions
  - Structure and Knowledge-aware Concept Interpretation
  - Graph-based Text Adaptation

Li, Qingyao, et al. "Learning Structure and Knowledge Aware Representation with Large Language Models for Concept Recommendation." arXiv preprint arXiv:2405.12442 (2024).

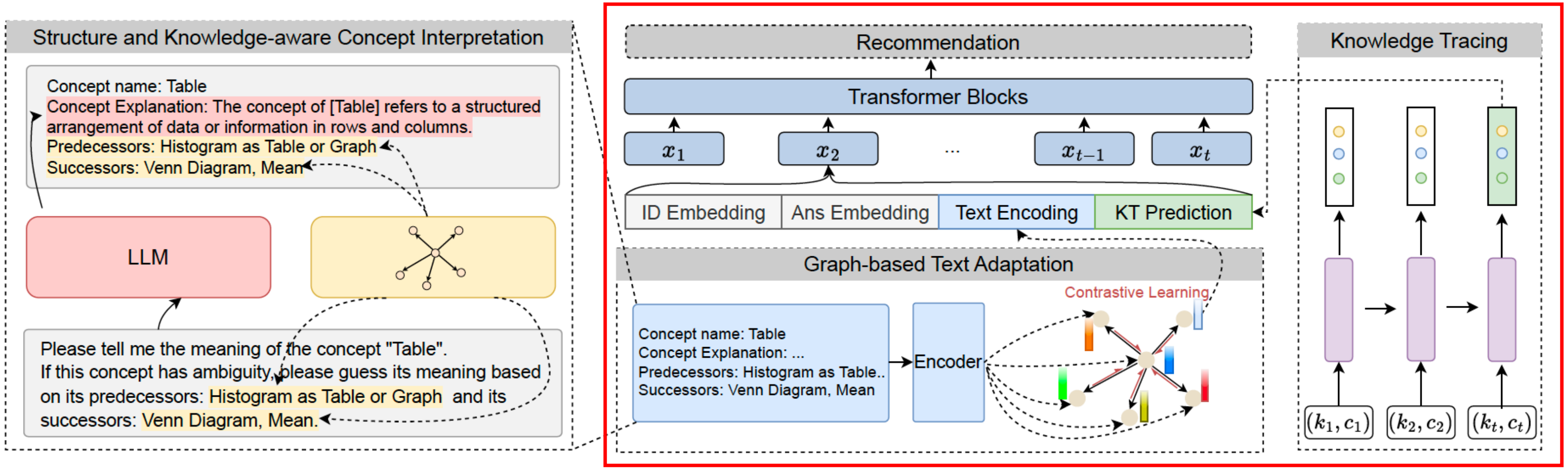
Combination - SKarRec

- Structure and Knowledge-aware Concept Interpretation
  - For a given concept c, SKarRec constructs prompt with (1) name and explanation (2) its predecessors and successors in the concept graph
  - Prompt is sent to LLM to generate rich interpretation
  - The interpreted concept text is used as semantic anchor for downstream tasks



Combination - SKarRec

- Graph-based Text Adaptation for Recommendation and Tracing
  - Interpreted concept texts are encoded into embeddings using a transformer encoder.
  - Graph G over concepts is built using their relations
  - Contrastive learning: positive pair (connected nodes), negative pair (unconnected)



Combination - SKarRec

- SKarRec achieves the best performance on ASSIST09, ASSIST12, and Junyi across all three metrics, significantly outperforming prior graph-based, text-only, and ID-only methods.

		Graph-based Methods		ID-Only Methods		Text-Only Methods		ID-Text Methods			
Dataset	**Metric**	**ACKRec**	**GCARec**	**SASRec**	**BERT4Rec**	**ZESRec**	**RECFORMER**	**FDSA**	**S3-Rec**	**UniSRec**	**SKarRec**
ASSIST09	HR@1	0.2408	0.2513	0.7649	0.1935	0.5809	0.6072	0.7627	<u>0.7661</u>	0.7597	**0.7922***
	NDCG@5	0.3787	0.3904	<u>0.8736</u>	0.2203	0.7269	0.6540	0.8722	0.8731	0.8647	**0.8838***
	MRR	0.3627	0.3756	0.8503	0.2342	0.6998	0.6711	0.8484	<u>0.8505</u>	0.8439	**0.8646***
ASSIST12	HR@1	0.1976	0.2119	0.2838	0.2540	0.2596	0.1990	0.2861	<u>0.2904</u>	0.2888	**0.2933***
	NDCG@5	0.3078	0.3169	0.4895	0.4247	0.4649	0.2055	0.4886	**0.4957**	0.4934	<u>0.4954</u>
	MRR	0.3064	0.3182	0.4574	0.4047	0.4343	0.2536	0.4579	<u>0.4633</u>	0.4617	**0.4645***
Junyi	HR@1	0.1447	0.1284	0.8469	0.3550	0.8546	0.8279	0.8492	<u>0.8608</u>	0.8407	**0.8730***
	NDCG@5	0.1872	0.1650	0.8907	0.4826	0.8930	0.7625	0.8917	<u>0.9022</u>	0.8890	**0.9076***
	MRR	0.1918	0.1740	0.8825	0.4661	0.8860	0.8497	0.8840	<u>0.8942</u>	0.8799	**0.9014***

- **Practical Values:** By integrating structural graph knowledge and language-model-based semantic reasoning, SKarRec offers a robust framework for educational recommendation and knowledge tracing with superior generalization and interpretability

Summary

- Knowledge §2
  - Summary Text §2.1
    - User Only
      - LANE [111]; LLM-BRec [28]; RecLM [31]
    - Item Only
      - ONCE [45]; LAMAR [57]; GPTAugNews [99]; SeRALM [67]; MMRec [75]; X-Reflect [62]
    - User & Item
      - KAR [94]; GaCLLM [11]; SLIM [86]; REKI [95]
  - Knowledge Graph §2.2
    - Generation
      - LLMRG [83]; SAGCN [43]; LLMHG [8]; HRGraph [91]; LLM-PKG [81]; AutoGraph [68]; CIKG [25]
    - Completion & Fusion
      - LLM-KERec [112]; CSRec [101]; CoLaKG [10]
  - Combination §2.3
    - KELLMRec [59]; SKarRec [36]

Agenda

August 3-7, 2025

1 Introduction

Zijian Zhang

2 Knowledge Enhancement

Pengyue Jia

3 Interaction Enhancement

Ziwei Liu

4.1 Model Enhancement 1

Maolin Wang

4.2 Model Enhancement 2

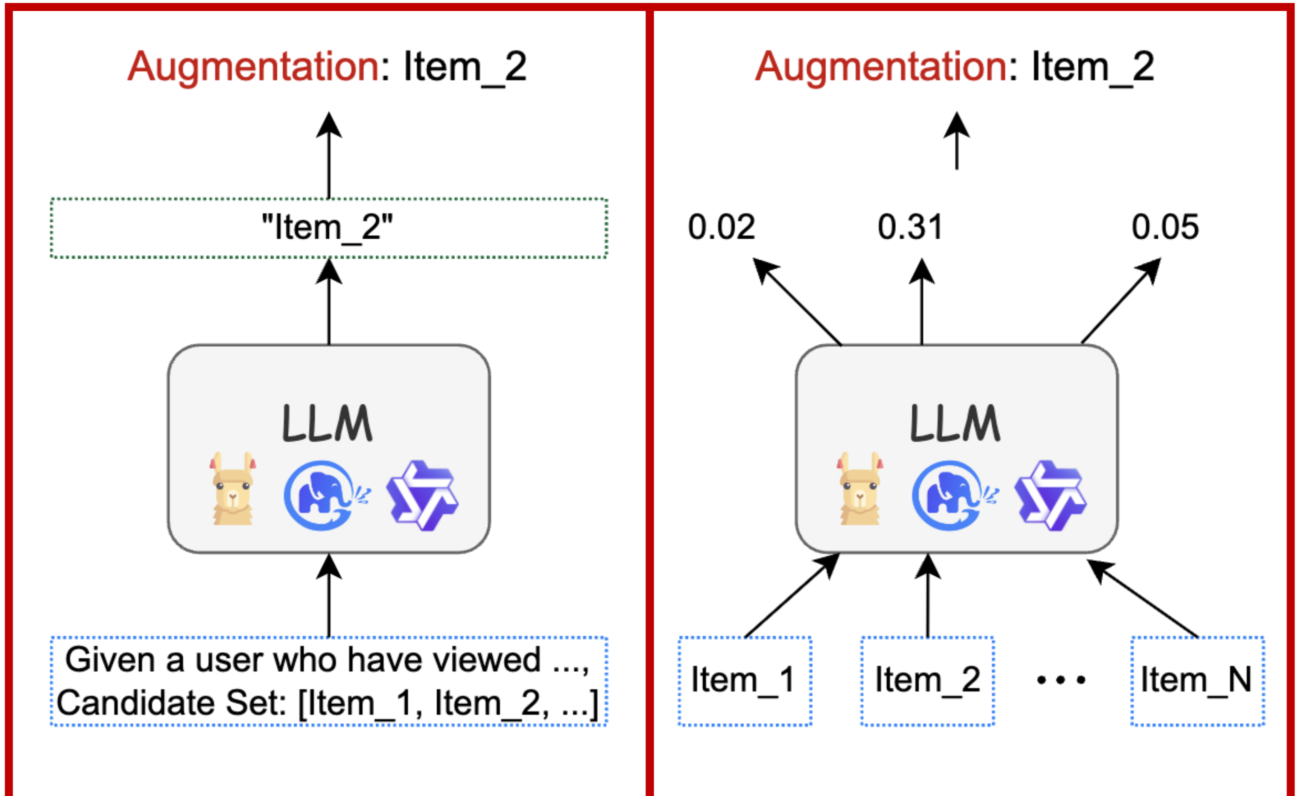
Yuhao Wang

5 Conclusion

Yejing Wang

Interaction Enhancement

We basically divide the categories of LLM for interaction enhancement in two aspects:



(a) Text-based (b) Score-based

- **Text-based:**
  As shown in Figure (a), test-based methods first conclude a candidate set and then utilize LLM to conduct the possible next interacted item to achieve augmentation.

- **Score-based:**
  For comparison (shown in Figure (b)), score-based methods utilize LLM to rank the importance of each item.

Text-based: LLMRec

Low-quality side information & **Data Sparsity** often contains bad affects, which severely affects the accuracy of recommendation.

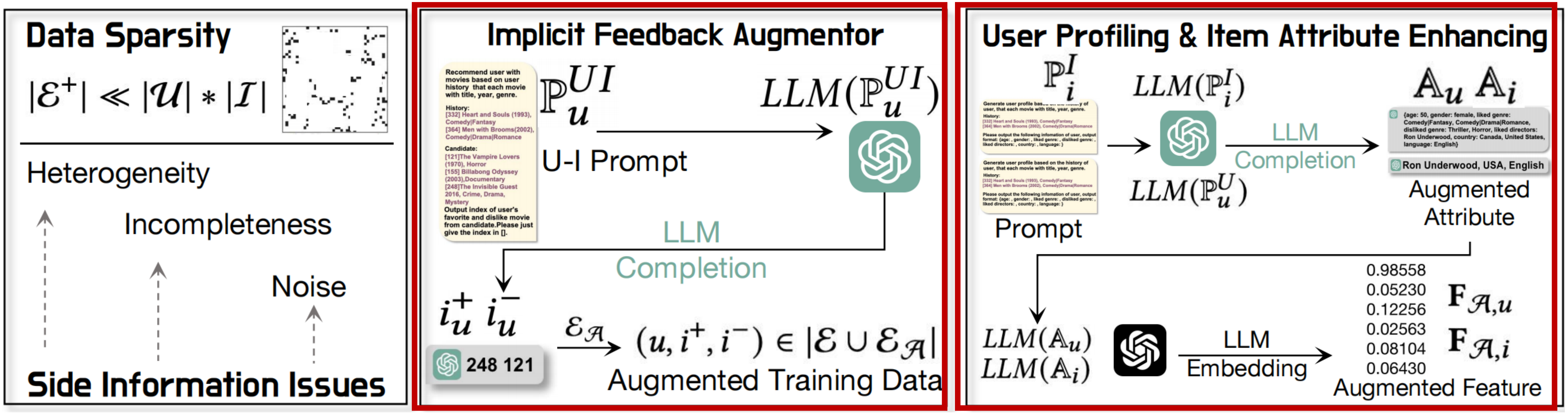
Large Language Model (LLM) is a promising technique to predict the user-item interaction and augment the side information of users/items. However, there are still two severe problems that hinder it work as a powerful data augmentor:

- **Incomplete input information**: It is hard to enable LLM to reason over full user-item interaction patterns due to the limited input token. (**P1**)
- **Low-quality of generated data**: LLM generators often introduce low-quality content that may compromise the results because of the noise and the hallucination of the model. (**P2**)

Wei, Wei, et al. "Llmrec: Large language models with graph augmentation for recommendation." Proceedings of the 17th ACM International Conference on Web Search and Data Mining. 2024.

Text-based: LLMRec

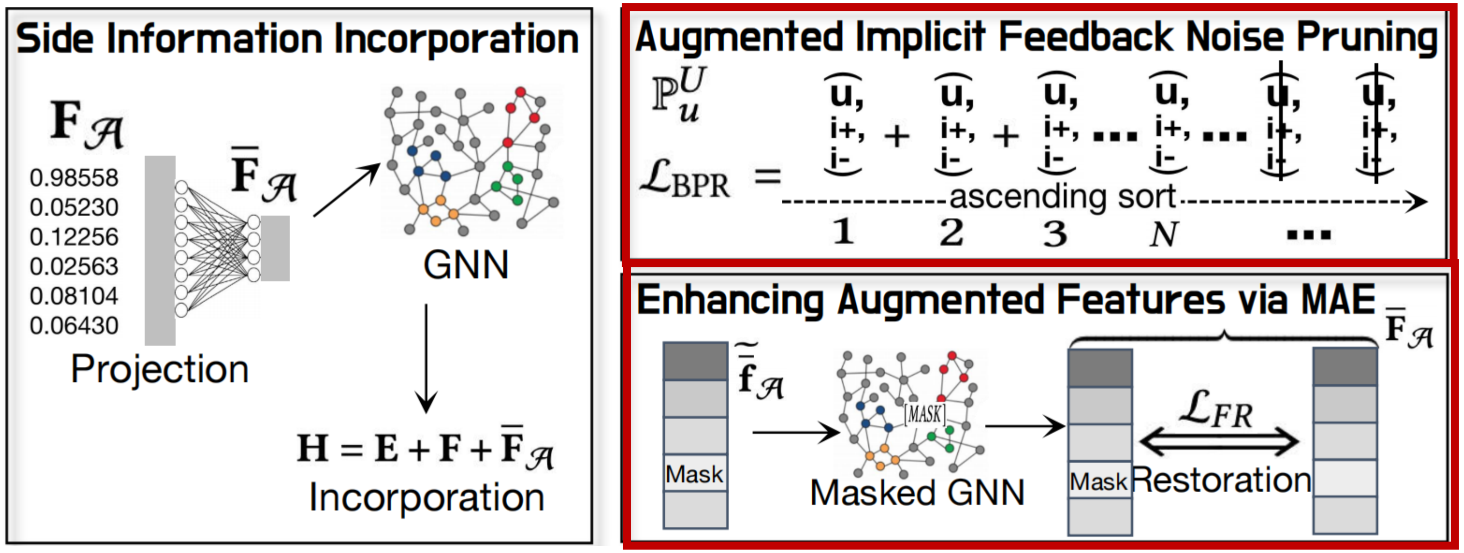
Specifically, for the first problem '**P1: Incomplete input information**', The authors design two individual modules to from a natural language perspective.



- **Implicit Feedback Augmentor**: Feeding the side feature of items and an item candidates pool (acquired by a basic Rec model) into LLM.
- **User Profiling & Item Attribute Enhancing**: Leveraging LLM's reasoning ability to summarize users' interactions and unify the item attributes.

Text-based: LLMRec

For the ‘**P2: Low-quality of generated data**’, The authors propose the Augmented Implicit Feedback Noise Pruning and Enhancing Augmented Features via MAE.



Augmented Implicit Feedback Noise Pruning:

- Leveraging ascending-sort mechanism with a BPR loss to filter values and selects the top-N score.

$$\mathcal{L}_{\text{BPR}} = \sum_{(u,i^+,i^-)}^{|\mathcal{E}\cup\mathcal{E}_{\mathcal{A}}|} -\log\left(\sigma(\hat{y}_{u,i^+} - \hat{y}_{u,i^-})\right) + \omega_2 \cdot \|\Theta\|^2$$

$$\mathcal{E}_{\mathcal{A}} \subseteq \{LLM(\mathbb{P}_u) | u \in \mathcal{U}\}, \quad |\mathcal{E}_{\mathcal{A}}| = \omega_3 * B$$

$$\sum_{(u,i^+,i^-)}^{(1-\omega_4)*|\mathcal{E}\cup\mathcal{E}_{\mathcal{A}}|} -SortAscend\left(\log(\sigma(\hat{y}_{u,i^+} - \hat{y}_{u,i^-}))\right)[0:N] + \omega_2 \cdot \|\Theta\|^2$$

- **Enhancing Augmented Semantic Features via MAE**: 1. Masking features of selected subset of the node; 2. Leveraging feature restoration loss to compare the masked matrix with the original one.

$$\widetilde{\bar{\mathbf{f}}}_{\mathcal{A}} = \begin{cases} \mathbf{f}_{[MASK]} & v \in \widetilde{\mathcal{V}} \\ \bar{\mathbf{f}}_{\mathcal{A}} & v \notin \widetilde{\mathcal{V}} \end{cases} \qquad \mathcal{L}_{FR} = \frac{1}{|\widetilde{\mathcal{V}}|} \sum_{v\in\widetilde{\mathcal{V}}} \left(1 - \frac{\widetilde{\bar{\mathbf{f}}}_{\mathcal{A}} \cdot \bar{\mathbf{f}}_{\mathcal{A}}}{\|\widetilde{\bar{\mathbf{f}}}_{\mathcal{A}}\| \cdot \|\bar{\mathbf{f}}_{\mathcal{A}}\|}\right)^{\gamma}$$

Text-based: LLMRec

For the experimental results, LLMRec achieves remarkable results under two public datasets and beats all the compared baselines with significant improvement.

Baseline	Netflix							MovieLens						
	R@10	N@10	R@20	N@20	R@50	N@50	P@20	R@10	N@10	R@20	N@20	R@50	N@50	P@20
General Collaborative Filtering Methods														
MF-BPR	0.0282	0.0140	0.0542	0.0205	0.0932	0.0281	0.0027	0.1890	0.0815	0.2564	0.0985	0.3442	0.1161	0.0128
NGCF	0.0347	0.0161	0.0699	0.0235	0.1092	0.0336	0.0032	0.2084	0.0886	0.2926	0.1100	0.4262	0.1362	0.0146
LightGCN	0.0352	0.0160	0.0701	0.0238	0.1125	0.0339	0.0032	0.1994	0.0837	0.2660	0.1005	0.3692	0.1209	0.0133
Recommenders with Side Information														
VBPR	0.0325	0.0142	0.0553	0.0199	0.1024	0.0291	0.0028	0.2144	0.0929	0.2980	0.1142	0.4076	0.1361	0.0149
MMGCN	0.0363	0.0174	0.0699	0.0249	0.1164	0.0342	0.0033	0.2314	0.1097	0.2856	0.1233	0.4282	0.1514	0.0147
GRCN	0.0379	0.0192	0.0706	0.0257	0.1148	0.0358	0.0035	0.2384	0.1040	0.3130	0.1236	0.4532	0.1516	0.0150
Data Augmentation Methods														
LATTICE	0.0433	0.0181	0.0737	0.0259	0.1301	0.0370	0.0036	0.2116	0.0955	0.3454	0.1268	0.4667	0.1479	0.0167
MICRO	0.0466	0.0196	0.0764	0.0271	0.1306	0.0378	0.0038	0.2150	0.1131	0.3461	0.1468	0.4898	0.1743	0.0175
Self-supervised Methods														
CLCRec	0.0428	0.0217	0.0607	0.0262	0.0981	0.0335	0.0030	0.2266	0.0971	0.3164	0.1198	0.4488	0.1459	0.0158
MMSSL	0.0455	0.0224	0.0743	0.0287	0.1257	0.0383	0.0037	0.2482	0.1113	0.3354	0.1310	0.4814	0.1616	0.0170
LLMRec	**0.0531**	**0.0272**	**0.0829**	**0.0347**	**0.1382**	**0.0456**	**0.0041**	**0.2603**	**0.1250**	**0.3643**	**0.1628**	**0.5281**	**0.1901**	**0.0186**
p-value	$2.9e^{-4}$	$3.0e^{-3}$	$9.4e^{-5}$	$1.5e^{-3}$	$2.8e^{-5}$	$2.2e^{-3}$	$3.4e^{-5}$	$2.8e^{-5}$	$1.6e^{-2}$	$3.1e^{-3}$	$4.1e^{-4}$	$1.9e^{-3}$	$1.3e^{-2}$	$1.8e^{-3}$
Improv.	13.95%	21.43%	8.51%	20.91%	5.82%	19.06%	7.89%	4.88%	10.52%	5.26%	10.90%	7.82%	9.06%	6.29%

Overall Performance Results

Text-based: LlamaRec

While **Conventional methods** excel at mining collaborative information and modeling sequential behavior, they struggle with data sparsity and long-tail problem.

Significant Gap

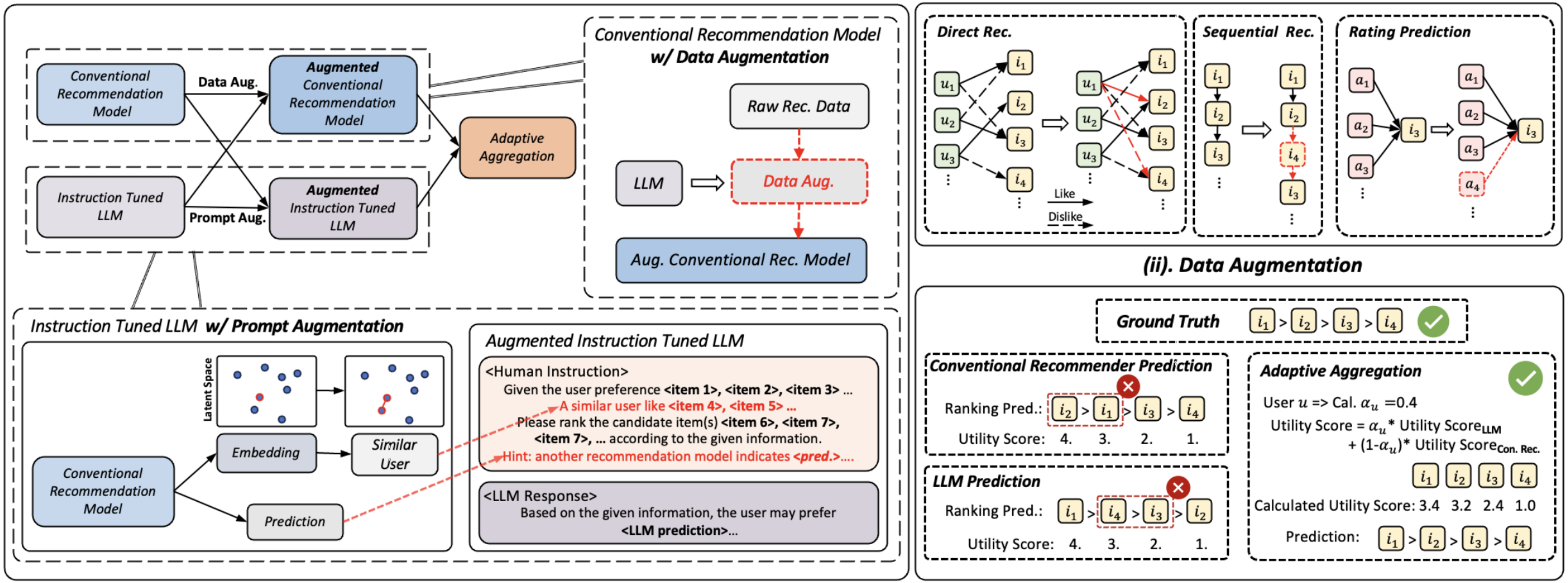
LLM-based methods are proficient at utilizing rich textual contexts, but face challenges in utilizing collaborative or sequential information.

- **Q1**: How to leverage LLM’s reasoning ability in conventional methods?
- **Q2**: How to introduce collaborative/sequential informtaion in LLM-based methods?
- **Q3**: How to aggregate the LLM-based method and conventional methods properly?

Luo, Sichun, et al. "Integrating large language models into recommendation via mutual augmentation and adaptive aggregation." arXiv preprint arXiv:2401.13870 (2024).

Text-based: LlamaRec

To overcome the above problems, the authors propose LlamaRec, which allows the conventional & LLM-based augmentation to mutually augment each other and adaptively aggregates the results.



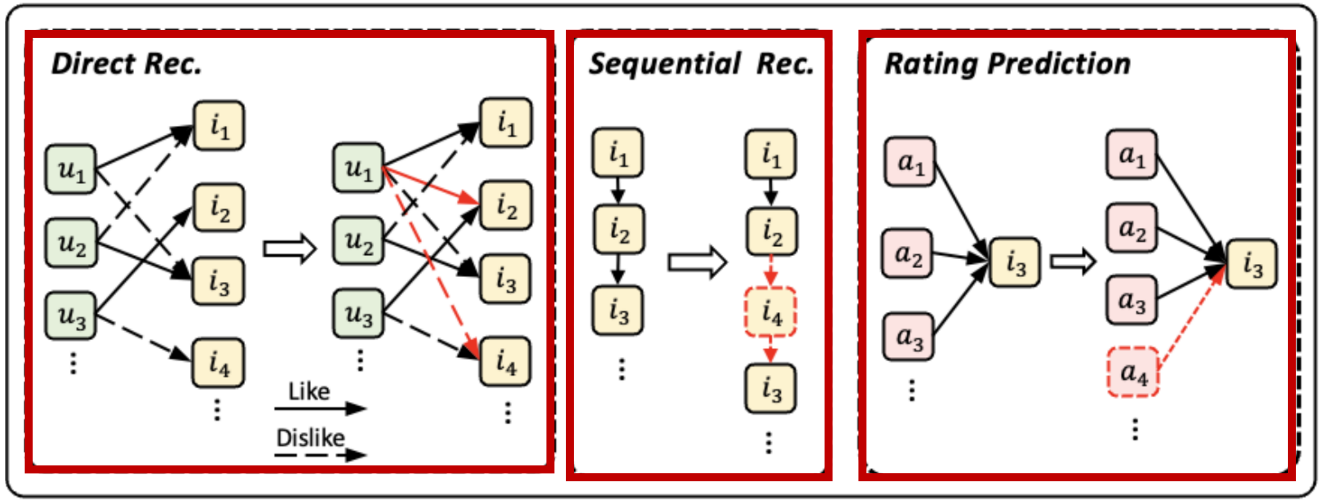
(i). Overall Framework

(ii). Data Augmentation

(iii). Adaptive Aggregation

Text-based: LlamaRec

Specifically, for the first problem ‘**Q1**: How to leverage LLM’s reasoning ability in conventional methods?’, The authors designed the Data Augmentation module.



- **Data Augmentation**:
  - Direct Recommendation:
    Leveraging LLM to rank each item pair and form the corresponding BPR loss:

$$\mathcal{L}'_{BPR} = -\sum_{(u,i,j)\in\mathcal{D}'} \log\sigma(\hat{y}_{ui} - \hat{y}_{uj})$$

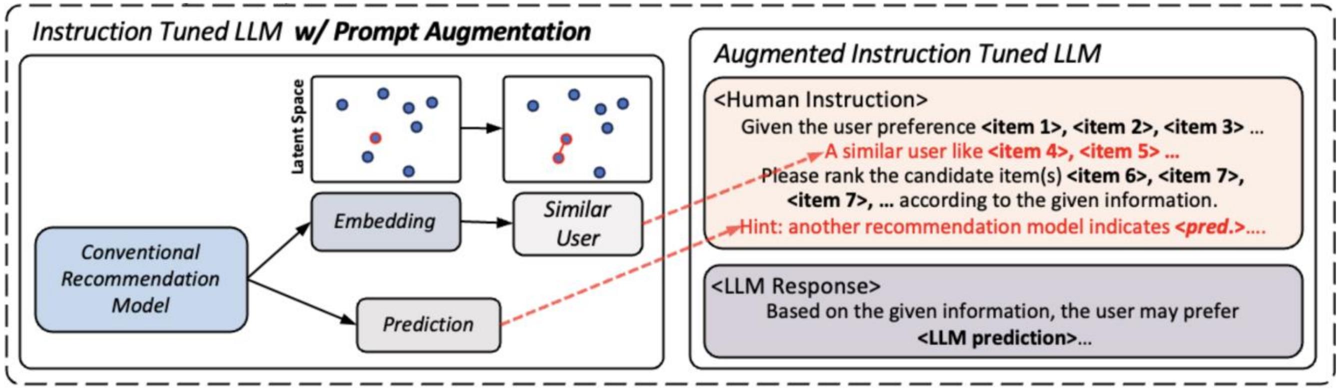
  - Sequential Recommendation:
    Leveraging LLM to predict the user prefered items in the un-interacted list, and then inserting them in the interaction sequence.
  - Rating Prediction:
    Leveraging LLM’s world knowledge to provide side information of item features.

Text-based: LlamaRec

For the second problem ‘**Q2**: How to introduce collaborative/sequential informtaion in LLM-based methods?’, The authors design the two Prompt Augmentation strategies.

Top-*k* Recommendation Prompt Example:
Instruction: Rank the candidate movies based on user historical interactions and make the top *k* recommendations.
Interaction History: Beyond Rangoon (1995); Alien (1979); Hollow Reed (1996); Primary Colors (1998); ...; Birds, The (1963)
Candidate Items: Last Dance (1996); Remains of the Day, The (1993); Assassins (1995); ...; Fatal Instinct (1993)
Similar User Interaction History: L.A. Confidential (1997); Apt Pupil (1998); Kolya (1996); ...; Star Wars (1977)
Conventional Model Prediction: Remains of the Day, The (1993); Addiction, The (1995); ...; Fugitive, The (1993)
Output: Fugitive, The (1993); Angel Baby (1995); ...; Remains of the Day, The (1993)

Rating Prediction Prompt Example:
Instruction: Predict the rating of a target movie based on the user’s historical movie ratings.
Rating History: Independence Day (1996): 3; Grosse Fatigue (1994): 3; Face/Off (1997): 4; ...; Shall We Dance? (1996): 3
Candidate Item: Pink Floyd - The Wall (1982)
Similar User Rating History: L.A. Confidential (1997): 3; Apt Pupil (1998): 4; ...; English Patient, The (1996): 3
Conventional Model Prediction: 3.2
Output: 3



- **Information from Similar User:**

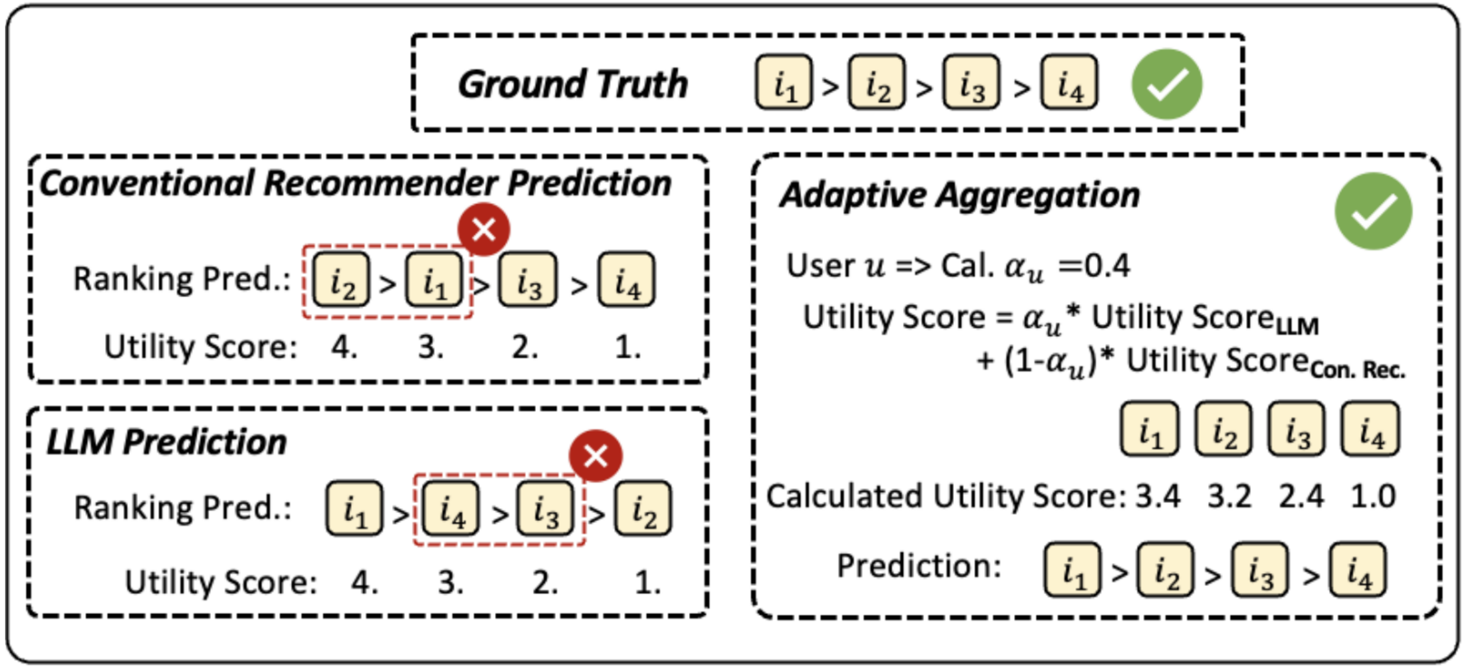
Regarding the interacted items of similar users as collaborative information to enrich the prompt for LLM Instruction-tuning.

- **Knowledge from Conventional Rec Model:**

Combining the prediction results from conventional models with the prompt.

Text-based: LlamaRec

Moreover, for ‘**Q3**: How to aggregate the LLM-based method and conventional methods properly?’, LlamaRec introduce the Adaptive Aggregation.



Long-tail coefficient:

$$\ell_u = \log\left(N(u) + 1\right)$$

The Utility Score of Rating Prediction task can be formulize as:

$$\alpha_u = \max\left(\frac{\ell_{max} - \ell_u}{\ell_{max} - \ell_{min}}, \alpha_2\right) \cdot \alpha_1$$

$$U_u = \alpha_u U_{LLM} + (1 - \alpha_u) U_{Rec}$$

For the top-k recommendation task, the LLM is employed to rerank the item list and the Utility Score can be calculated as the same.

Text-based: LlamaRec

For the experimental results, LlamaRec achieves optimal results under three public datasets and three recommendation tasks.

Backbone	Method	ML-100K				ML-1M				BookCrossing			
		H@3 ↑	N@3 ↑	H@5 ↑	N@5 ↑	H@3 ↑	N@3 ↑	H@5 ↑	N@5 ↑	H@3 ↑	N@3 ↑	H@5 ↑	N@5 ↑
MF	Base	0.0455	0.0325	0.0690	0.0420	0.0255	0.0187	0.0403	0.0248	0.0294	0.0227	0.0394	0.0269
	IFT	0.0546	0.0388	0.0790	0.0488	0.0242	0.0175	0.0410	0.0244	0.0247	0.0177	0.0377	0.0230
	Llama4Rec	**0.0645***	**0.0474***	**0.0919***	**0.0588***	**0.0281***	**0.0203***	**0.0433***	**0.0265***	**0.0365***	**0.0284***	**0.0462***	**0.0324***
	Impro.	18.13%	22.16%	16.33%	20.49%	10.20%	8.56%	5.61%	6.85%	24.15%	25.55%	17.26%	20.45%
LightGCN	Base	0.0492	0.0343	0.0744	0.0447	0.0283	0.0203	0.0432	0.0264	0.0358	0.0272	0.0480	0.0322
	IFT	0.0537	0.0381	0.0846	0.0507	0.0268	0.0193	0.0441	0.0263	0.0287	0.0202	0.0448	0.0268
	Llama4Rec	**0.0647***	**0.0476***	**0.0967***	**0.0608***	**0.0304***	**0.0222***	**0.0461***	**0.0286***	**0.0434***	**0.0338***	**0.057***	**0.0394***
	Impro.	20.48%	24.93%	14.30%	19.92%	7.42%	9.36%	4.54%	8.33%	21.23%	24.26%	18.75%	22.36%
MixGCF	Base	0.0526	0.0401	0.0757	0.0496	0.0159	0.0115	0.0238	0.0147	0.0426	0.0330	0.0556	0.0384
	IFT	0.0617	0.0452	0.0906	0.0570	0.0162	0.0114	**0.0259**	0.0154	0.0337	0.0243	0.0506	0.0312
	Llama4Rec	**0.0690***	**0.0515***	**0.0949***	**0.0621***	**0.0174***	**0.0128***	**0.0259**	**0.0162***	**0.0495***	**0.0384***	**0.0635***	**0.0441***
	Impro.	11.83%	13.94%	4.75%	8.95%	7.41%	11.30%	0.00%	5.19%	16.20%	16.36%	14.21%	14.84%
SGL	Base	0.0505	0.0380	0.0729	0.0472	0.0284	0.0206	0.0434	0.0267	0.0419	0.0319	0.0566	0.0380
	IFT	0.0520	0.0392	0.0792	0.0503	0.0275	0.0202	0.0438	0.0269	0.0326	0.0237	0.0499	0.0307
	Llama4Rec	**0.0632***	**0.0479***	**0.0917***	**0.0596***	**0.0308***	**0.0224***	**0.0480***	**0.0294***	**0.0501***	**0.0393***	**0.0634***	**0.0448***
	Impro.	21.54%	22.19%	15.78%	18.49%	8.45%	8.74%	9.59%	9.29%	19.57%	23.20%	12.01%	17.89%

Overall Performance achieved by Direct Recommendation Models

Text-based: LlamaRec

For the experimental results, LlamaRec achieves optimal results under three public datasets and three recommendation tasks.

Backbone	Method	ML-100K				ML-1M				BookCrossing			
		H@3 ↑	N@3 ↑	H@5 ↑	N@5 ↑	H@3 ↑	N@3 ↑	H@5 ↑	N@5 ↑	H@3 ↑	N@3 ↑	H@5 ↑	N@5 ↑
SASRec	Base	0.0187	0.0125	0.0385	0.0205	0.0277	0.0165	0.0502	0.0257	0.0086	0.0049	0.0163	0.0081
	IFT	0.0204	0.0136	0.0379	0.0207	0.0241	0.0159	0.0473	0.0254	0.0124	0.0086	0.0185	0.0111
	Llama4Rec	**0.0238***	**0.0155***	**0.0449***	**0.0240***	**0.0293***	**0.0201***	**0.0504**	**0.0287***	**0.0142***	**0.0098***	**0.0227***	**0.0131***
	Impro.	16.67%	13.97%	16.62%	15.94%	5.78%	21.82%	0.40%	11.67%	14.52%	13.95%	22.70%	18.02%
BERT4Rec	Base	0.0153	0.0104	0.0294	0.0161	0.0107	0.0069	**0.0211**	0.0112	0.0088	0.0058	0.0161	0.0088
	IFT	0.0174	0.0119	0.0326	0.0100	0.0106	0.0071	0.0188	0.0104	0.0127	0.0092	0.0180	0.0113
	Llama4Rec	**0.0198***	**0.0134***	**0.0332**	**0.0189***	**0.0115***	**0.0078***	0.0206	**0.0115***	**0.0154***	**0.0108***	**0.023***	**0.0139***
	Impro.	13.79%	12.61%	1.84%	17.39%	7.48%	9.86%	-2.37%	2.68%	21.26%	17.39%	27.78%	23.01%
CL4SRec	Base	0.0243	0.0143	0.0436	0.0222	0.0259	0.0153	**0.0492**	0.0248	0.0083	0.0048	0.0165	0.0082
	IFT	0.0230	0.0149	0.0428	0.0230	0.0234	0.0155	0.0447	0.0241	0.0102	0.0071	0.0177	0.0102
	Llama4Rec	**0.0255***	**0.0182***	**0.0440**	**0.0255***	**0.0278***	**0.0185***	0.0482	**0.0268***	**0.0138***	**0.0093***	**0.0220***	**0.0127***
	Impro.	4.94%	22.15%	0.92%	10.87%	7.34%	19.35%	-2.03%	8.06%	35.29%	30.99%	24.29%	24.51%

Overall Performance achieved by Sequential Recommendation Models

Text-based: LlamaRec

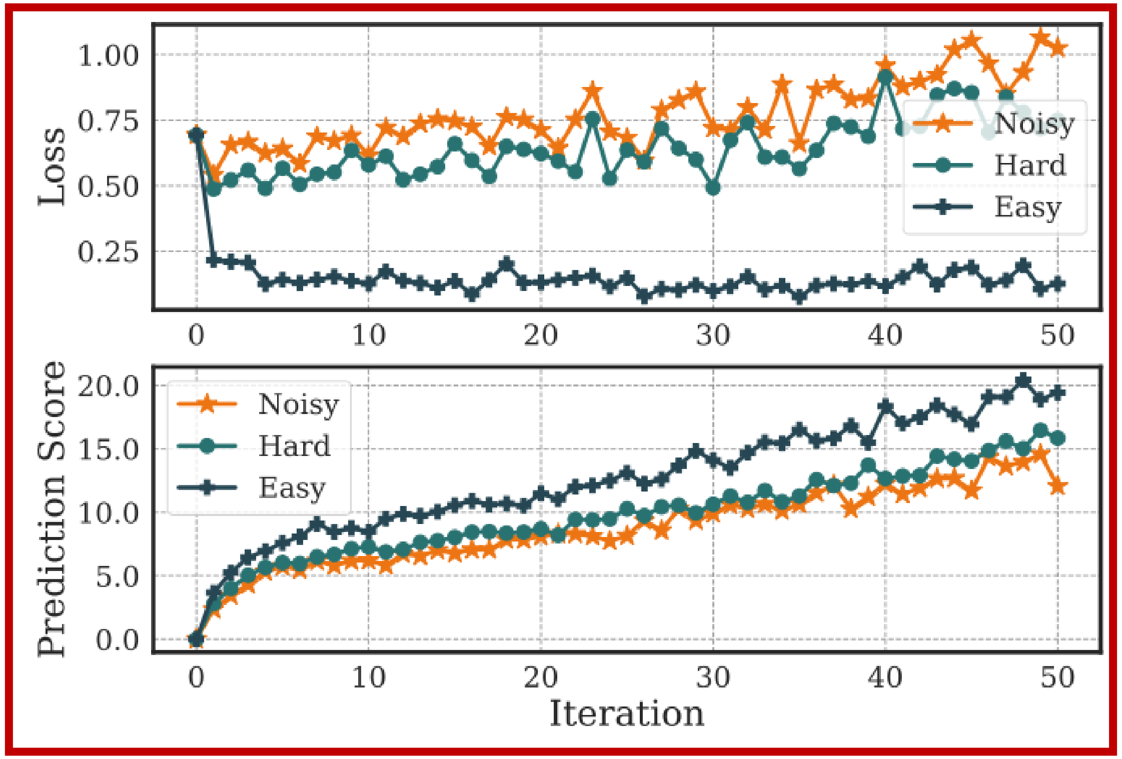
For the experimental results, LlamaRec achieves optimal results under three public datasets and three recommendation tasks.

Backbone	Method	ML-100K		ML-1M		BookCrossing	
		RMSE ↓	MAE ↓	RMSE ↓	MAE ↓	RMSE ↓	MAE ↓
LLaMA	IFT	1.2792	0.8940	1.2302	0.8770	2.0152	1.3782
DeepFM	Base	1.0487	0.8082	0.9455	0.7409	1.7738	1.3554
	Llama4Rec	**1.0306***	**0.7987***	**0.9360***	**0.7321***	**1.6958***	**1.2843***
	Impro.	1.73%	1.18%	1.00%	1.19%	4.40%	5.25%
NFM	Base	1.0284	0.8005	0.9438	0.7364	2.121	1.5984
	Llama4Rec	**1.0189***	**0.7961**	**0.9369***	**0.7303***	**1.9253***	**1.4473***
	Impro.	0.92%	0.55%	0.73%	0.83%	9.23%	9.45%
DCN	Base	1.0478	0.8063	0.9426	0.7342	2.0216	1.4622
	Llama4Rec	**1.0367***	**0.8033**	**0.9345***	**0.7272***	**1.8518***	**1.3566***
	Impro.	1.06%	0.37%	0.86%	0.95%	8.40%	7.22%
AFM	Base	1.0471	0.8035	0.9508	0.7464	1.6516	1.2614
	Llama4Rec	**1.0340***	**0.7996**	**0.9426***	**0.7394***	**1.6244***	**1.2259***
	Impro.	1.25%	0.49%	0.86%	0.94%	1.65%	2.81%
xDeepFM	Base	1.1472	0.8836	0.9519	0.7428	2.1756	1.6461
	Llama4Rec	**1.0947***	**0.8483***	**0.9401***	**0.7336***	**1.9610***	**1.4833***
	Impro.	4.58%	4.00%	1.24%	1.24%	9.86%	9.89%
AutoInt	Base	1.0500	0.8120	0.9471	0.7404	1.9148	1.4501
	Llama4Rec	**1.0369***	**0.8059***	**0.9382***	**0.7326***	**1.7917***	**1.3492***
	Impro.	1.25%	0.75%	0.94%	1.05%	6.43%	6.96%

Overall Performance achieved by Rating Prediction Models

Text-based: LLMHD

Although **Large Language Models (LLM)** show potential in denoising recommendation, there are still some challenges that hinders the directly application of LLM:



- **C1: High Cost of LLM Summarization:**

Assessing the preferences of all users across all items is computationally intensive.

- **C2: Misidentifying of Hard Sample & Noise:**

It is challenging to distinguish the difference between the Noisy and Hard Example.

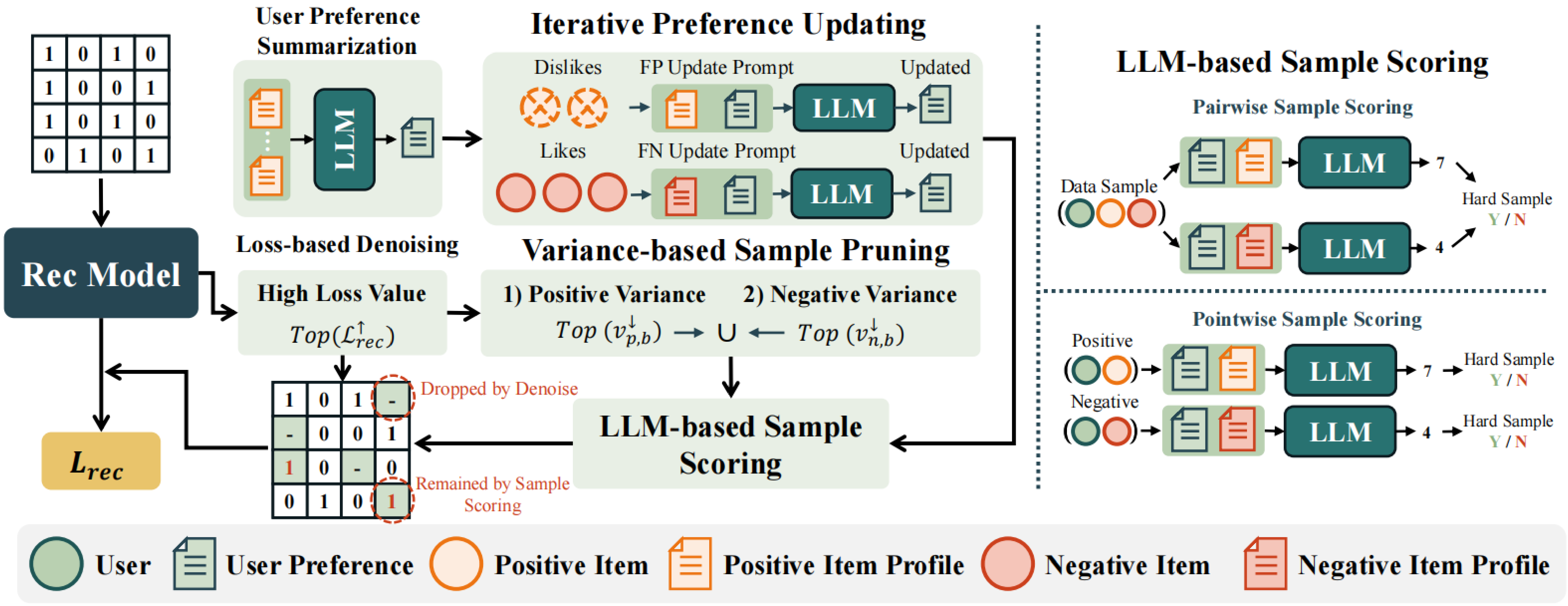
- **C3: Biased User Preference:**

False-positive items can lead to biased user preference summarization.

Song, Tianrui, Wenshuo Chao, and Hao Liu. "Large Language Model Enhanced Hard Sample Identification for Denoising Recommendation." arXiv preprint arXiv:2409.10343 (2024).

Text-based: LLMHD

To address the above challenges, the authors proposed LLMHD, featuring with Variance-based Sample Pruning, LLM-based Sample Scoring, and Iterative Preference Updating to fully explore the capabilities of LLM in denosing recommendation.



Text-based: LLMHD

To address **C1: High Cost of LLM Summarization**, LLMHD introduce the Variance-based Sample Pruning module.

1) Positive Variance **2) Negative Variance**

$$Top\,(v^{\downarrow}_{p,b}) \rightarrow \cup \leftarrow Top\,(v^{\downarrow}_{n,b})$$

Following the observation of previous work: hard samples exhibit relatively higher prediction score variance compared to noisy samples, LLMHD calculate and sort the prediction score of positive/negative items as follows:

$$v^{\downarrow}_{p,1} > v^{\downarrow}_{p,2} > \cdots > v^{\downarrow}_{p,b} > \cdots > v^{\downarrow}_{p,|\mathcal{B}^p_N|}, b \in \mathcal{B}_N,$$

$$v^{\downarrow}_{n,1} > v^{\downarrow}_{n,2} > \cdots > v^{\downarrow}_{n,b} > \cdots > v^{\downarrow}_{n,|\mathcal{B}^n_N|}, b \in \mathcal{B}_N,$$

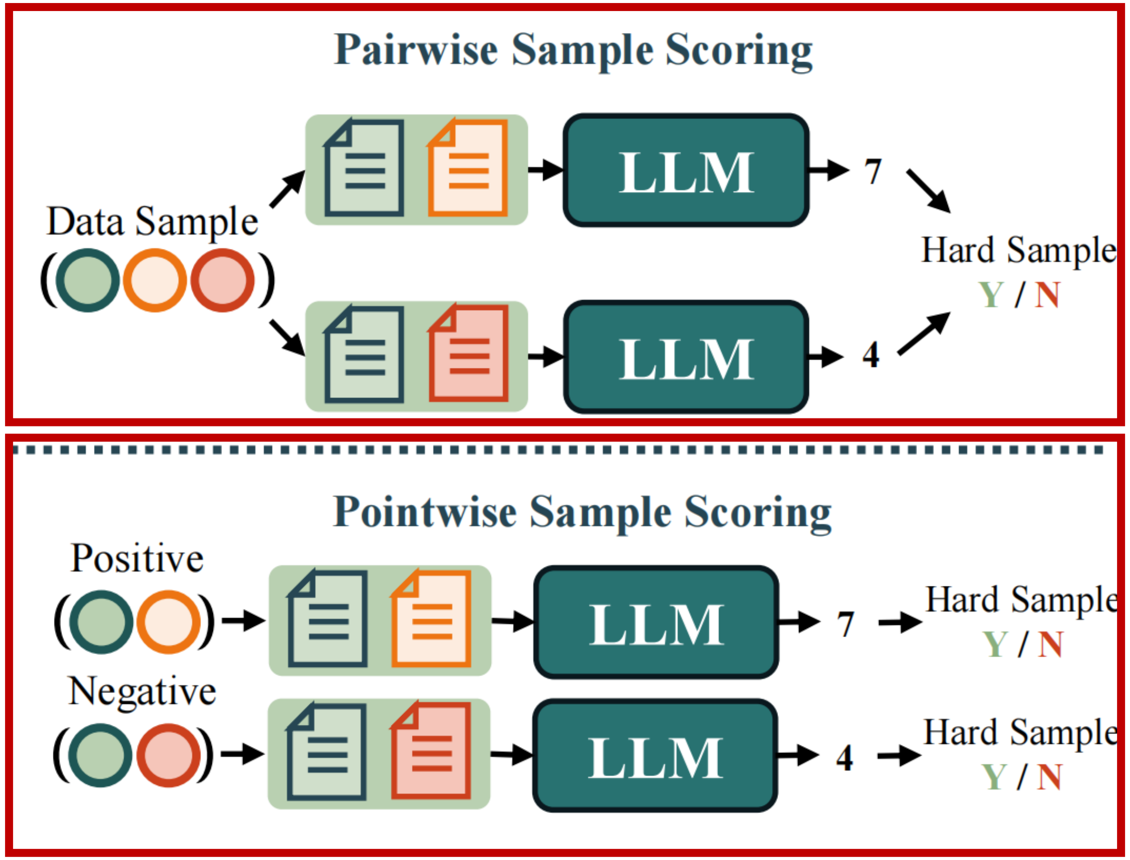
So the hard samples can then be defined as:

$$\mathcal{B}_{HC} = \{b_j | v^{\downarrow}_{p,j} \geq v^{\downarrow}_{p,\varepsilon_v|\mathcal{B}^p_N|}\} \cup \{b_j | v^{\downarrow}_{n,b_j} \geq v^{\downarrow}_{n,\varepsilon_v|\mathcal{B}^n_N|}\}$$

Preventing to feed all identified noisy samples to the LLMs for scoring, significantly reduce the computation cost of LLM.

Text-based: LLMHD

For **C2: Misidentifying of Hard Sample & Noise**, LLMHD introduce a novel Sample Scoring module that leverage LLM to provide auxiliary information for evaluating the sample hardness.



- **Pairwise Sample Scoring**:

Considering the user's preference for the positive does not significantly surpass the negative, hard samples are identified through the indicator function:

$$\mathbb{I}_{pair}(s_{u,i_p} - s_{u,i_n} > \varepsilon_{pair})$$

Also, for achieving the generalization ability, the threshold gradually decreases to increase the hardness by the number of iteration T

$$\varepsilon_{pair} = max(\varepsilon_{pair}^{max} - \frac{1}{\alpha}T, \varepsilon_{pair}^{min})$$

- **Pointwise Sample Scoring**:

For both positive and negative pairs, setting different thresholds to control the hardness

$$\mathbb{I}_{point}(u,i) = \begin{cases} 1, & \text{if } s_{u,i} < \varepsilon_{pos} \text{ and } y_{u,i} = 1 \\ 1, & \text{if } s_{u,i} > \varepsilon_{neg} \text{ and } y_{u,i} = 0 \\ 0, & \text{otherwise,} \end{cases}$$

And smoothly changing each threshold during each training iteration T to achieve generalization ability.

$$\varepsilon_{pos} = max(\varepsilon_{pos}^{max} - \frac{1}{\alpha}T, \varepsilon_{pos}^{min})$$

$$\varepsilon_{neg} = min(\varepsilon_{neg}^{min} + \frac{1}{\alpha}T, \varepsilon_{neg}^{max})$$

Text-based: LLMHD

To address **C3: Biased User Preference Summarization**, LLMHD introduce the Iterative Preference Updating module.

Specifically, the authors refine user preferences iteratively by excluding dislikes (false-positive) and incorporating likes (false-negative).

The confident items' selection can then be formularize as:

$$v_d = \frac{1}{m} \sum_{j=t-m+1}^{t} \left(\hat{y}_d^j - \frac{\sum_{j=t-m+1}^{t} \hat{y}_d^j}{m} \right)^2$$

$$\xrightarrow{\text{Positive}} v^{\uparrow}_{d^{\mathrm{p}}_1} < v^{\uparrow}_{d^{\mathrm{p}}_2} < \cdots < v^{\uparrow}_{d^{\mathrm{p}}_{|\mathcal{D}_{pos}|}}, d^{\mathrm{p}}_k \in \mathcal{D}_{pos} \longrightarrow \mathbb{I}_{\mathrm{FP}}\left(\sum_{j=0}^{t} \mathbb{I}(d^{\mathrm{p}}_k) \geq \varepsilon_\gamma\right)$$

$$\xrightarrow{\text{Negative}} v^{\uparrow}_{d^{\mathrm{n}}_1} < v^{\uparrow}_{d^{\mathrm{n}}_2} < \cdots < v^{\uparrow}_{d^{\mathrm{n}}_{|\mathcal{D}_{neg}|}}, d^{\mathrm{n}}_k \in \mathcal{D}_{neg} \longrightarrow \mathbb{I}_{\mathrm{FN}}\left(\sum_{j=0}^{t} \mathbb{I}(d^{\mathrm{n}}_k) \geq \varepsilon_\gamma\right)$$

At last, LLMHD can refine the original preference based on identified false-positives and false-negatives to construct the unbiased user profiles.

$$\mathcal{P}^*_u = LLM(T_{\mathrm{FP}}(\mathcal{P}_u, \mathcal{P}_{i_{\mathrm{fp}}}))$$

$$\mathcal{P}^*_u = LLM(T_{\mathrm{FN}}(\mathcal{P}_u, \mathcal{P}_{i_{\mathrm{fn}}}))$$

Text-based: LLMHD

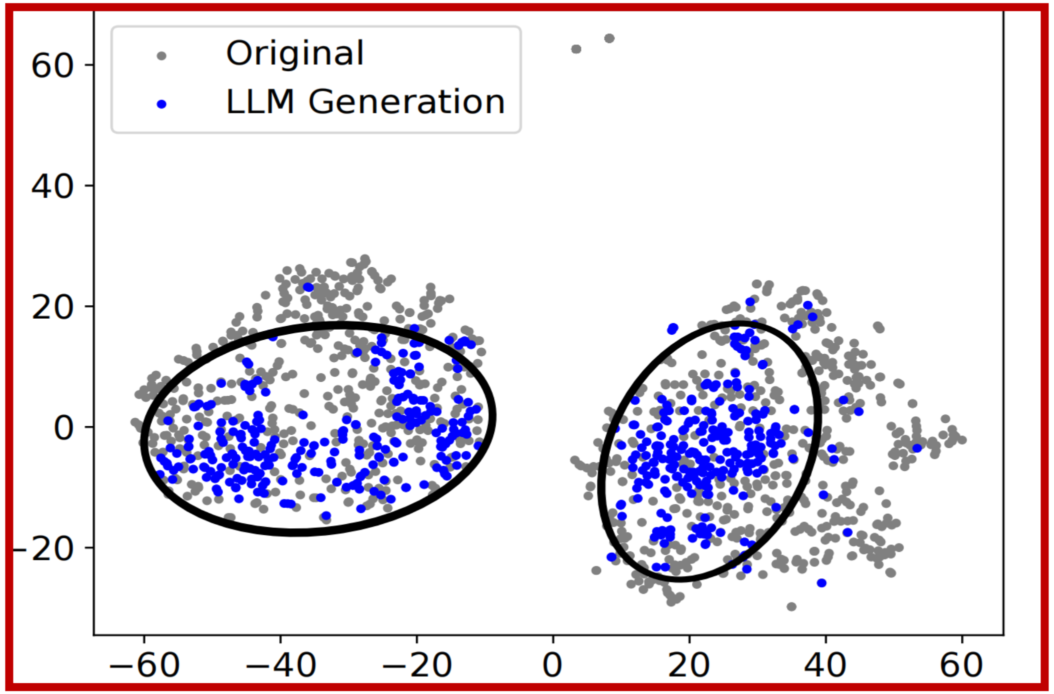
Dataset		Amazon-book				Yelp				Steam			
Backbone	**Method**	**R@5**	**R@10**	**N@5**	**N@10**	**R@5**	**R@10**	**N@5**	**N@10**	**R@5**	**R@10**	**N@5**	**N@10**
NGCF	BCE	0.0353	0.0570	0.0365	0.0438	0.0236	0.0431	0.0283	0.0350	0.0223	0.0405	0.0236	0.0305
	BPR	0.0389	0.0651	0.0406	0.0494	0.0280	0.0495	0.0338	0.0405	0.0381	0.0629	0.0453	0.0525
	T-CE	0.0393	0.0650	0.0402	0.0489	0.0259	0.0450	0.0313	0.0373	0.0257	0.0448	0.0288	0.0354
	R-CE	0.0366	0.0587	0.0369	0.0444	0.0254	0.0438	0.0302	0.0360	0.0236	0.0435	0.0254	0.0328
	RGCF	0.0415	0.0658	0.0422	0.0502	0.0287	0.0485	0.0344	0.0406	0.0401	0.0644	0.0472	0.0543
	DCF	0.0398	0.0617	0.0399	0.0472	0.0281	0.0488	0.0353	0.0414	0.0264	0.0446	0.0308	0.0365
	LLMHD$_{\text{BCE}}$	0.0406	0.0668	0.0413	0.0503	0.0276	0.0477	0.0329	0.0386	0.0267	0.0459	0.0297	0.0364
	LLMHD$_{\text{BPR}}$	**0.0455**	**0.0743**	**0.0455**	**0.0552**	**0.0338**	**0.0579**	**0.0398**	**0.0474**	**0.0418**	**0.0696**	**0.0496**	**0.0579**
LightGCN	BCE	0.0558	0.0849	0.0565	0.0665	0.0390	0.0660	0.0481	0.0557	0.0448	0.0732	0.0529	0.0612
	BPR	0.0587	0.0904	0.0598	0.0704	0.0359	0.0609	0.0446	0.0516	0.0510	0.0828	0.0597	0.0693
	T-CE	0.0590	0.0895	0.0592	0.0697	0.0401	0.0677	0.0504	0.0580	0.0463	0.0758	0.0555	0.0640
	R-CE	0.0557	0.0834	0.0566	0.0658	0.0389	0.0650	0.0474	0.0550	0.0461	0.0757	0.0543	0.0630
	RGCF	0.0619	0.0956	0.0644	0.0753	0.0420	0.0693	0.0501	0.0579	0.0519	0.0849	0.0599	0.0702
	DCF	0.0590	0.0898	0.0596	0.0701	0.0403	0.0680	0.0503	0.0579	0.0477	0.0778	0.0562	0.0650
	LLMHD$_{\text{BCE}}$	0.0607	0.0921	0.0607	0.0711	0.0408	0.0689	0.0514	0.0589	0.0469	0.0767	0.0563	0.0647
	LLMHD$_{\text{BPR}}$	**0.0652**	**0.0999**	**0.0655**	**0.0767**	**0.0427**	**0.0731**	**0.0518**	**0.0611**	**0.0536**	**0.0867**	**0.0624**	**0.0722**
SGL	BCE	0.0589	0.0902	0.0604	0;.0707	0.0377	0.0655	0.0470	0.0548	0.0433	0.0682	0.0505	0.0676
	BPR	0.0608	0.0956	0.0621	0.0736	0.0373	0.0629	0.0465	0.0538	0.0529	0.0838	0.0613	0.0704
	T-CE	0.0602	0.0909	0.0622	0.0720	0.0408	0.0697	0.0502	0.0587	0.0449	0.0720	0.0532	0.0609
	R-CE	0.0591	0.0901	0.0601	0.0702	0.0386	0.0645	0.0476	0.0550	0.0456	0.0732	0.0538	0.0618
	RGCF	0.0675	0.1049	0.0681	0.0808	0.0416	0.0715	0.0512	0.0606	**0.0552**	0.0881	0.0639	0.0736
	DCF	0.0626	0.0933	0.0641	0.0740	0.0413	0.0683	0.0506	0.0583	0.0455	0.0727	0.0536	0.0615
	LLMHD$_{\text{BCE}}$	0.0615	0.0931	0.0640	0.0739	0.0414	0.0708	0.0509	0.0596	0.0462	0.0742	0.0543	0.0619
	LLMHD$_{\text{BPR}}$	**0.0693**	**0.1051**	**0.0717**	**0.0837**	**0.0426**	**0.0718**	**0.0523**	**0.0619**	0.0546	**0.0887**	**0.0641**	**0.0739**
NCL	BCE	0.0574	0.0871	0.0598	0.0694	0.0391	0.0647	0.0477	0.0548	0.0450	0.0731	0.0529	0.0612
	BPR	0.0605	0.0942	0.0628	0.0740	0.0369	0.0609	0.0451	0.0515	0.0511	0.0835	0.0602	0.0698
	T-CE	0.0599	0.0898	0.0619	0.0719	0.0411	0.0679	0.0507	0.0582	0.0461	0.0751	0.0543	0.0627
	R-CE	0.0585	0.0874	0.0604	0.0696	0.0399	0.0655	0.0487	0.0558	0.0459	0.0750	0.0540	0.0625
	RGCF	0.0694	0.1045	0.0706	0.0819	0.0396	0.0660	0.0480	0.0560	0.0534	**0.0863**	0.0621	**0.0718**
	DCF	0.0619	0.0929	0.0624	0.0727	0.0424	0.0696	0.0513	0.0589	0.0465	0.0759	0.0550	0.0635
	LLMHD$_{\text{BCE}}$	0.0609	0.0915	0.0629	0.0730	0.0417	0.0690	0.0517	0.0591	0.0468	0.0762	0.0551	0.0633
	LLMHD$_{\text{BPR}}$	**0.0719**	**0.1053**	**0.0741**	**0.0846**	**0.0432**	**0.0716**	**0.0542**	**0.0620**	**0.0540**	0.0861	**0.0624**	0.0717

For the experimental results, LLMHD achieves optimal results on three public datasets, validating the effectiveness of proposed methods.

Text-based: SampleLLM

Existing methods in **Tabular data synthesis** often fall short in RS because of the difficulty in understanding the complicated semantic feature relations.

Large Language Models (LLMs) have shown potential in generating synthetic data. However, directly applying LLM to data synthesis still face severe problems:



- **P1: Inconsistent Distribution & Diversity**:

The distribution mismatch between LLM's inherent knowledge and the target datasets can lead to reduced output diversity.

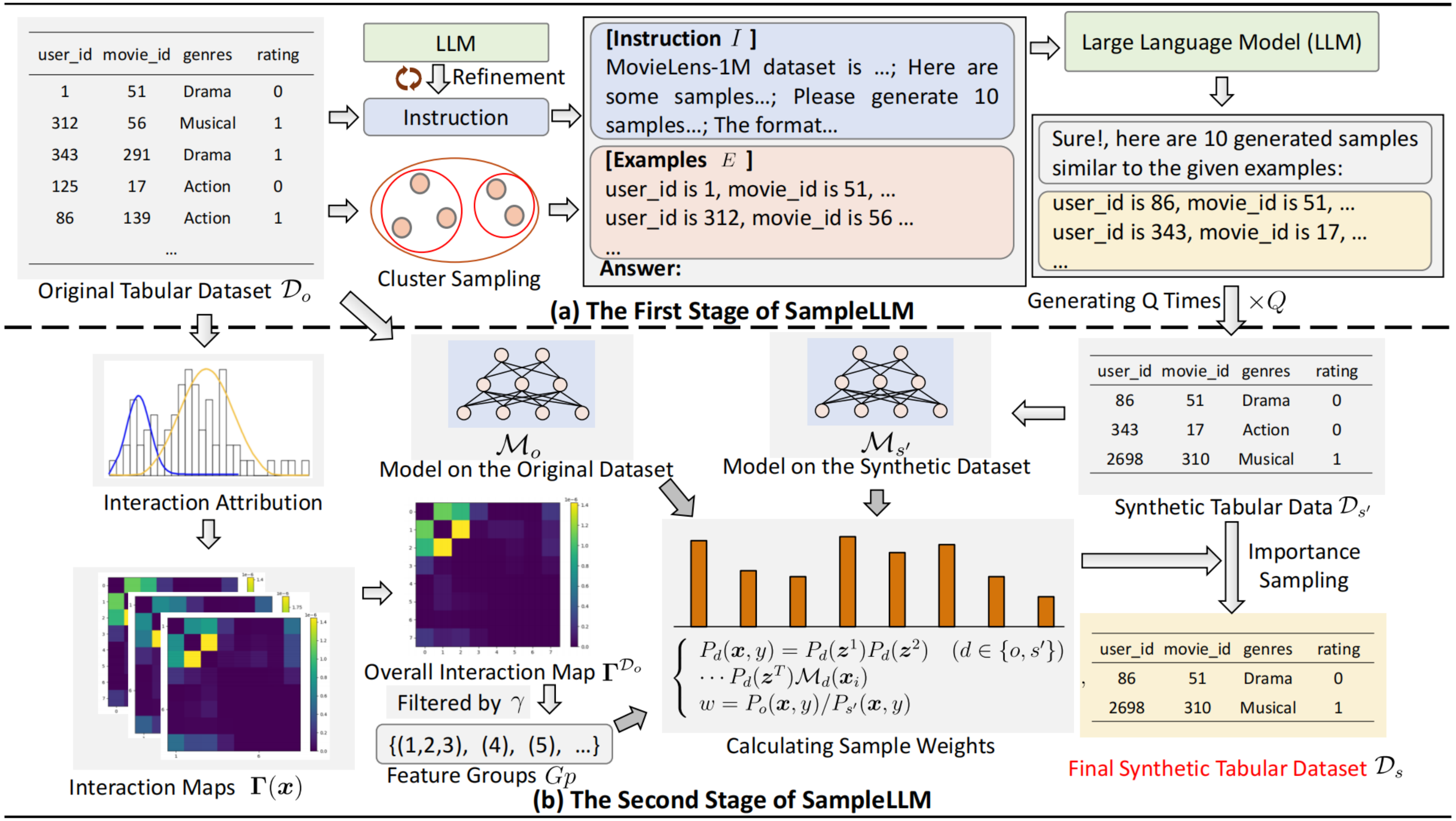
- **P2: Distribution Difference Caused by LLM**:

Inherent difference caused by LLM's input-output processing still result in a distribution gap.

Gao, Jingtong, et al. "SampleLLM: Optimizing Tabular Data Synthesis in Recommendations." Companion Proceedings of the ACM on Web Conference 2025. 2025.

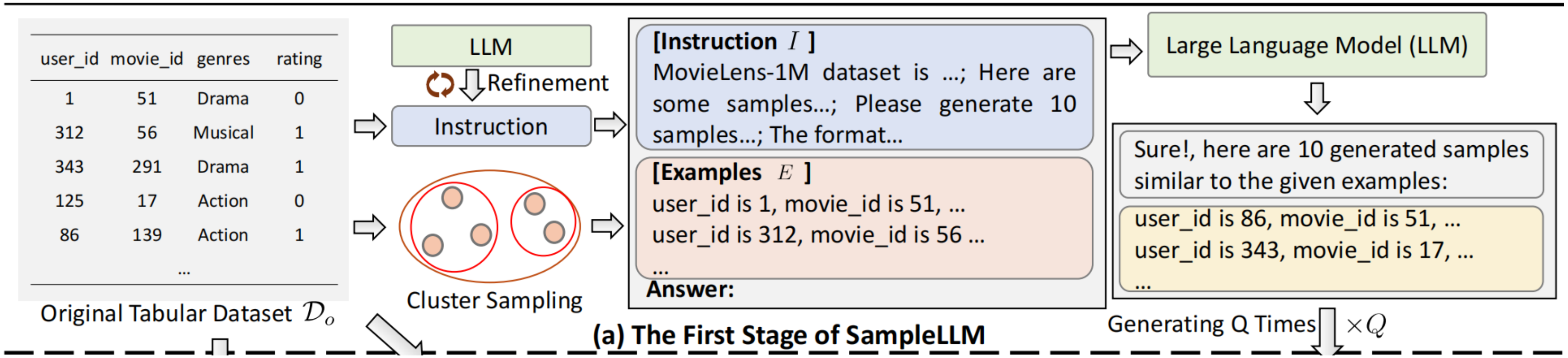
Text-based: SampleLLM

To address the above problems, the authors propose SampleLLM, a two-stage framework designed to enhance the quality of LLM-based tabular data synthesis in recommendation tasks.



Text-based: SampleLLM

To address **P1: Inconsistent Distribution & Diversity**, in the First Stage, the designed instruction and sampled exemplars from the original dataset are selected to serve as the input for the LLM, producing the initial synthetic tabular data.



(a) The First Stage of SampleLLM

- **Example Selection**

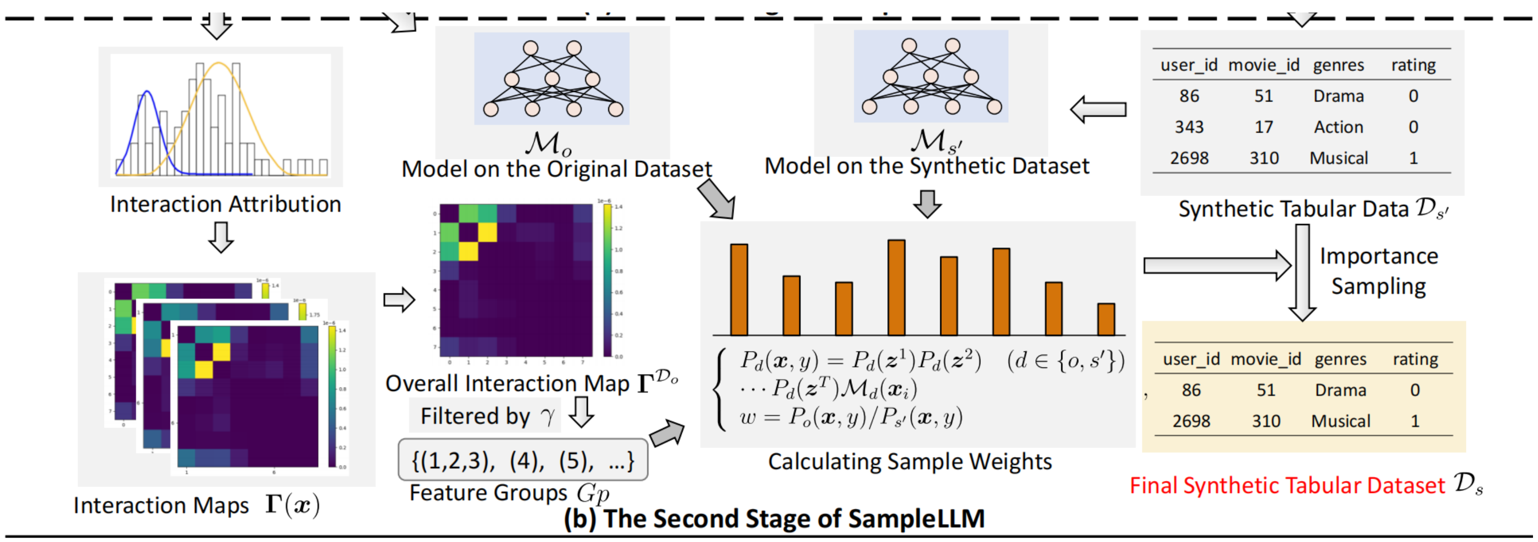
Leveraging K-Means method, SampleLLM can ensures that each group of exemplars contains samples from diverse regions of the original dataset's distribution by only using clusters for LLM's few-shot learning.

The whole approach can be formulized as follows:

$$\begin{cases} Clu = h(\mathcal{D}_o, a) = \{clu_1, clu_2, \ldots, clu_a\} \\ E = \{\boldsymbol{x}_i | \boldsymbol{x}_i = g(clu_i); i \in \{1, 2, \ldots, a\}\} \end{cases}$$

Text-based: SampleLLM

In the second stage, a feature attribution-based importance sampling operation is performed to address '**P2: Distribution Difference Caused by LLM**'.



(b) The Second Stage of SampleLLM

- **Feature Attribution-based Importance Sampling**:

- Current Assumption:

$$P(\boldsymbol{x}_i) = P(x_i^1)P(x_i^2)\ldots P(x_i^K)P(y_i \mid x_i^1, x_i^2, \ldots, x_i^K)$$

- Semi-independence assumption:

$$P(\boldsymbol{x}_i) = P(x_i^1, x_i^8)P(x_i^2)\ldots P(x_i^K)P(y_i \mid x_i^1, x_i^2, \ldots, x_i^K)$$

Still challenging to identify the significance. **-> Feature Interaction Extraction**

Approximating performance change after setting an informative feature to the non-informative one:

$$\mathrm{EG}_i(\boldsymbol{x}) := \mathop{\mathbb{E}}_{\boldsymbol{x}' \sim \mathcal{D}^b}\left[\left(x_i - x_i^b\right)\frac{\delta f(\boldsymbol{x})}{\delta x_i}\right] \longrightarrow \Gamma_{i,j}(x) = \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mathcal{D}^b}\left[\left(x_i - x_i^b\right)\left(x_j - x_j^b\right)\frac{\partial^2 f\left(x^b\right)}{\partial x_i \partial x_j}\right]$$

Text-based: SampleLLM

In the second stage, a feature attribution-based importance sampling operation is performed to address '**P2: Distribution Difference Caused by LLM**'.

- **Feature Interaction Extraction**
- **Calculating Sample Weights**

The import ... e deriv ... a:

$$\mathrm{EG}_i(\boldsymbol{x}) := \mathop{\mathbb{E}}_{\boldsymbol{x}' \sim \mathcal{D}^b}\left[\left(x_i - x_i^b\right)\frac{\delta f(\boldsymbol{x})}{\delta x_i}\right] \longrightarrow \Gamma_{i,j}(x) = \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mathcal{D}^b}\left[\left(x_i - x_i^b\right)\left(x_j - x_j^b\right)\frac{\partial^2 f\left(\boldsymbol{x}^b\right)}{\partial x_i \partial x_j}\right]$$

The su ... tion can highlight the significance in the whole dataset and ca ... ment refine

$$\begin{cases} F_\gamma = \gamma \cdot max(\Gamma^{\mathcal{D}_o}) \\ Pairs = \mathrm{argwhere}_{\Gamma^{\mathcal{D}_o}}(\Gamma^{\mathcal{D}_o} > F_\gamma) \\ Gp = \mathrm{Merge}(Pairs) \end{cases}$$

$$\mathcal{M}(\boldsymbol{x}_i) = P(y_i \mid x_i^1, x_i^2, \ldots, x_i^K)$$

$$\longrightarrow \begin{cases} F_\gamma = \ldots \\ Pairs \\ Gp = \mathrm{Merge}(Pairs) \end{cases}$$

$$\begin{cases} P_d(\boldsymbol{x}, y) = P_d(\boldsymbol{z}^1)P_d(\boldsymbol{z}^2) \quad (d \in \{o, s'\}) \\ \cdots P_d(\boldsymbol{z}^T)\mathcal{M}_d(\boldsymbol{x}_i) \\ w = P_o(\boldsymbol{x}, y)/P_{s'}(\boldsymbol{x}, y) \end{cases}$$

- **Obtaining Discriminant Probability**

Then, the label of each tabular sample can be obtained by training a task-specific model:

$$\mathcal{M}(\boldsymbol{x}_i) = P(y_i \mid x_i^1, x_i^2, \ldots, x_i^K)$$

Text-based: SampleLLM

The experimental results have shown that SampleLLM achieves remarkable results on five public datasets, validating the effectiveness of proposed methods.

The authors first test the utility by supplanting the real data with synthetic data in training of predictive models.

They also validate the utility of synthetic data by comparing the prediction performance of different methods after injecting synthetic data into the original data (Augmentation Utility).

Approach	ML-1M		Amazon		Douban		HELOC		CoverType		
	AUC ↑	Logloss ↓	AUC ↑	Logloss ↓	AUC ↑	Logloss ↓	AUC ↑	Logloss ↓	Precision ↑	Recall ↑	F1 ↑
Original	0.8173	0.5163	0.7075	0.4602	0.8016	0.5158	0.7638	0.6445	0.7395	0.7339	0.7250
PATE-GAN	0.8154	<u>0.5153</u>	0.7022	0.4620	0.8010	0.5162	0.7648	0.6461	0.7388	0.7350	0.7258
ADS-GAN	0.8147	0.5157	0.7026	0.4619	0.8016	0.5151	0.7672	0.6451	0.7414	0.7413	0.7312
CTGAN	0.8148	0.5156	0.7027	0.4631	0.8016	0.5150	0.7677	0.6419	0.7417	0.7366	0.7295
TVAE	0.8143	0.5162	0.7031	0.4616	0.8019	0.5146	0.7701	0.6455	0.7422	0.7378	0.7305
TabDDPM	0.8141	0.5159	0.7036	0.4619	0.8020	0.5150	0.7704	0.6424	0.7418	0.7388	0.7299
GReaT	0.8153	0.5198	0.7040	0.4616	0.8021	0.5147	0.7703	0.6444	0.7423	**0.7440**	<u>0.7317</u>
REaLTabFormer	0.8156	0.5182	0.7041	0.4611	0.8022	0.5145	0.7707	0.6417	0.7428	0.7351	0.7264
SampleLLM	**0.8180***	**0.5140***	**0.7082***	**0.4601***	**0.8027***	**0.5139***	**0.7732***	**0.6403***	**0.7445***	<u>0.7437</u>	**0.7364***

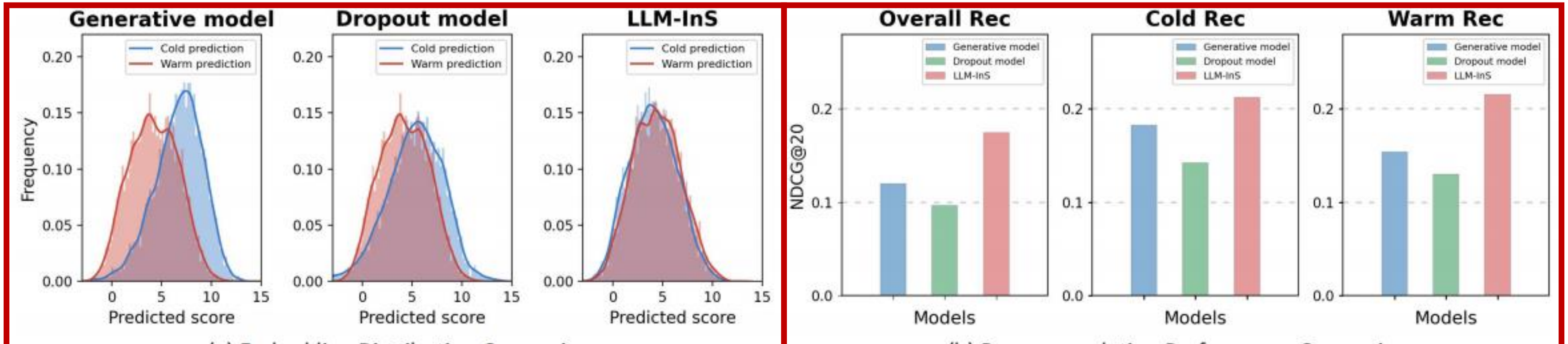
MLE Utility Performance (10% synthetic data)

Augmentation Utility Performance (10% synthetic data)

Score-based: LLM-Ins

Cold-Start Items are common in pratice and severely affect the performance.

Current Methods adopt **mapping function** to generate fake embedding from cold-start items' content feature, but still face some challenges:



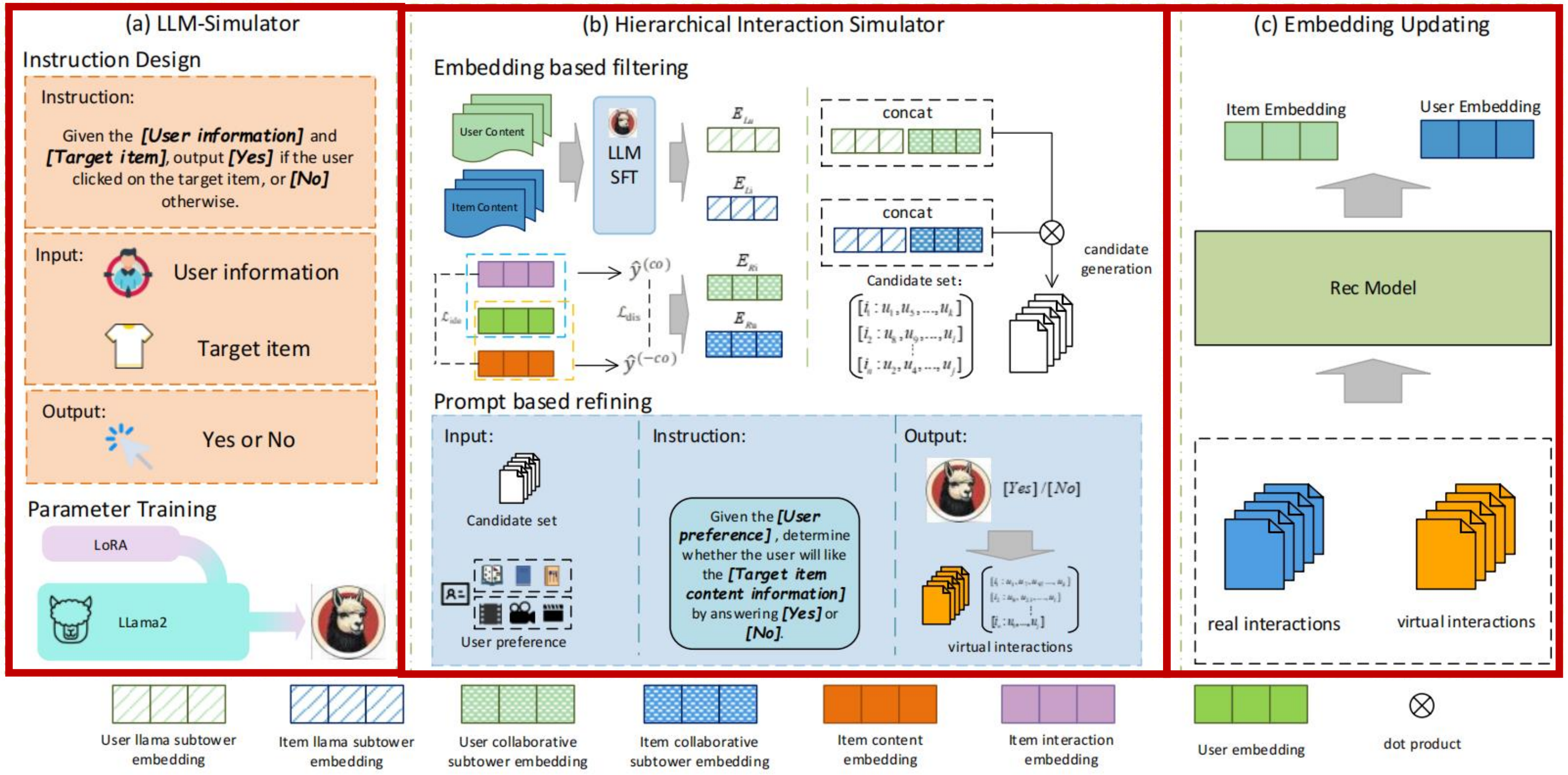
(a) Embedding Distribution Comparison
(b) Recommendation Performance Comparison

- **C1: Distribution Gap**. The generated embeddings are created from content features, while the behavioral embeddings are trained from sequences.
- **C2: LLM's Inadaptability**. Applying LLM to interaction generation need further alignment & unify.

Huang, Feiran, et al. "Large language model interaction simulator for cold-start item recommendation." arXiv e-prints (2024): arXiv-2402.

Score-based: LLM-Ins

To address the challenges, the authors propose LLM-Ins to model interactions based on content for each cold item (**C1**) and transform them from cold to warm items (**C2**).



(a) LLM-Simulator:

Generating possible user set for cold items.

(b) Interaction Simulator:

Mimicing user interactions for each cold item and transform them to warm.

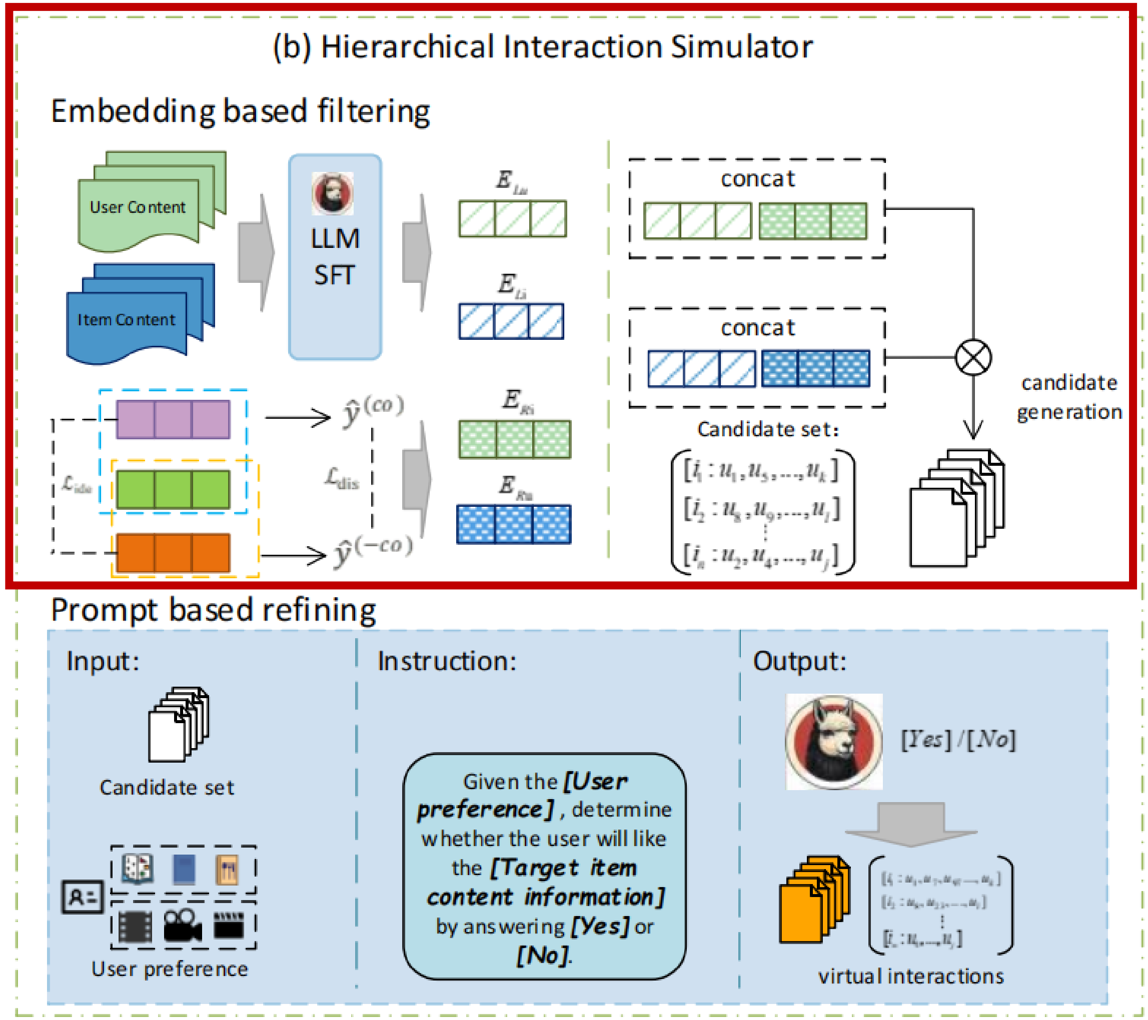
(c) Embedding Updating :

Adaptively combining the cold items and warm items.

We'll then detail the procedure of **Hierarchical Interaction Simulator** (main procedure).

Score-based: LLM-Ins

We'll detail the procedure of **Hierarchical Interaction Simulator** (main procedure).



❑ Embedding based filtering :

Ensuring that the candidate selection aligns closely with real-world scenarios.

- Collaborative Subtower:

Main loss (BPR): Narrowing the gap between **(two sides: semantic & collaborative)**

Auxillary Subtower

Auxillary Loss (dis): Collaborative/non-collaborative side:

Featuring with a user tower and a item tower, it ... an...

$$\mathcal{L}_{dis} = \frac{1}{|\mathcal{B}|}\sum_{(u,i,j)\in\mathcal{B}}\left(\left|\hat{y}_{ui}^{(co)} - \hat{y}_{ui}^{(-co)}\right|^2 + \left|\hat{y}_{uj}^{(co)} - \hat{y}_{uj}^{(-co)}\right|^2\right),$$

Auxillary Loss (ide): Embedding distance side.

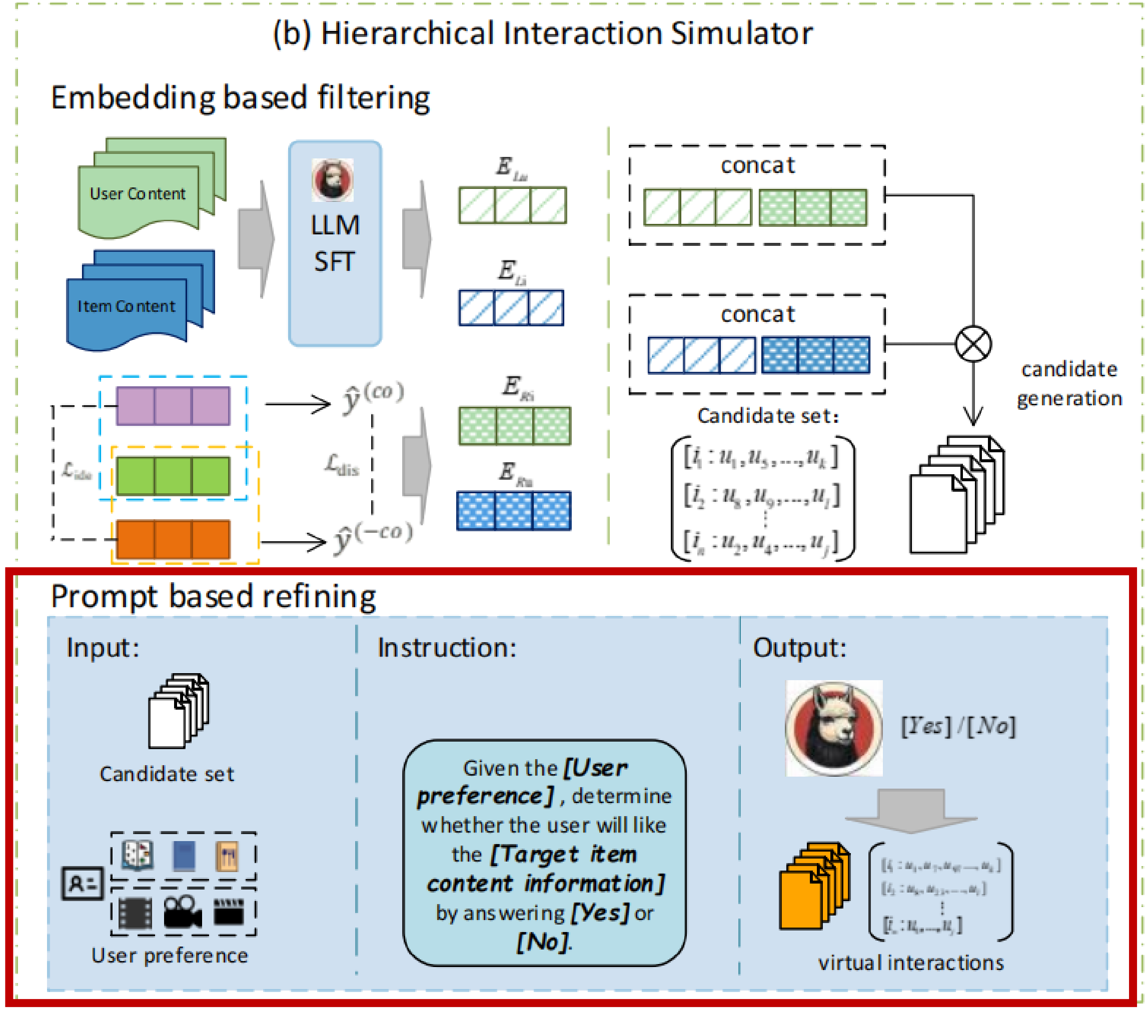
$$\mathcal{L}_{bce} = -\frac{1}{\ldots}\sum \ldots y_{u,i}\log(\hat{y}_{u,i}) + (1-y_{u,i})\log(1-\hat{y}_{u,i})$$

$$\mathcal{L}_{ide} = -\sum_{i\in\mathcal{B}_I}\left(\bar{d}_i^{(co)}\ln\bar{d}_i^{(-co)} + (1-\bar{d}_i^{(co)})\ln(1-\bar{d}_i^{(-co)})\right),$$

$$\bar{d}_i^{(co)} = \sigma\left(\boldsymbol{e}_i^\top\cdot\left(\boldsymbol{e}_i - \frac{1}{|\mathcal{B}|}\sum_{j\in\mathcal{B}_J}\boldsymbol{e}_j\right)\right),$$

Score-based: LLM-Ins

We'll detail the procedure of **Hierarchical Interaction Simulator** (main procedure).



☐ Prompt based refining :

The authors concat these two embeddings from **Llama Tower** and **Collaborative Tower**:

$$E_{LTi} = E_{Li} \| E_{Ci}, \quad E_{LTu} = E_{Lu} \| E_{Cu}.$$

More LLM to further refine the generated interactions to filter the low-quality interactions:

$$prompt = \mathcal{G}_{u,i}(\mathcal{H}_u, \mathcal{I}_t | \mathcal{I}_t \in \mathcal{I}_c),$$

$$C_f = L_t(prompt | A_{ns} = yes),$$

Score-based: LLM-Ins

For the experimental results, LLM-Ins achieves optimal results under three backbone MF, NGCF, LightGCN and presents remarkable performance in comparison with other LLM-based Recommendation model:

	Method	Overall Recommendation				Cold Recommendation				Warm Recommendation			
		CiteULike		MovieLens		CiteULike		MovieLens		CiteULike		MovieLens	
		Recall	NDCG	Recall	NDCG	Recall	NDCG	Recall	NDCG	Recall	NDCG	Recall	NDCG
LightGCN	Backbone	0.0812	0.0622	0.1418	0.2330	0.0041	0.0019	0.0177	0.0190	0.2528	0.1541	0.3130	0.3008
	DropoutNet	0.0883	0.0639	0.1165	0.1978	0.2309	0.1312	0.0340	0.0373	0.1175	0.0692	0.2560	0.2518
	MTPR	0.1001	0.0753	0.1011	.0.1551	0.2585	0.1454	0.0779	0.0802	0.1753	0.0697	0.2247	0.2009
	CLCRec	0.1293	0.0965	0.1253	0.2037	0.2435	0.1425	0.0677	0.0816	0.2149	0.1302	0.2764	0.2612
	DeepMusic	0.0985	0.0745	0.1418	0.2330	0.2239	0.1259	0.0635	0.0719	0.2528	0.1541	0.3130	0.3008
	MetaEmb	0.0924	0.0714	0.1418	0.2330	0.2252	0.1295	0.0248	0.0244	0.2528	0.1541	0.3130	0.3008
	GPatch	0.1609	0.1197	0.1293	0.2106	0.2606	0.1532	0.0771	0.0760	0.2528	0.1541	0.3130	0.3008
	GAR	0.1357	0.1062	0.0106	0.0195	0.2539	0.1489	0.0110	0.0130	0.2339	0.1455	0.2873	0.2794
	UCC	0.1374	0.1260	0.1277	0.2020	0.0020	0.0011	0.0063	0.0073	0.3002	0.2010	0.2830	0.2641
	ALDI	0.1626	0.1201	0.1428	0.2316	0.2692	0.1539	0.1229	0.1295	0.2528	0.1541	0.3130	0.3008
	LLM-InS	**0.2285**	**0.1747**	**0.1506**	**0.2468**	**0.3601**	**0.2126**	**0.1759**	**0.1762**	**0.3252**	**0.2156**	**0.3314**	**0.3186**
	%Improv.	40.52%	38.65%	5.46%	5.92%	33.76%	38.14%	43.12%	36.06%	8.33%	7.26%	5.97%	5.92%

Experiment results when backbone = LightGCN

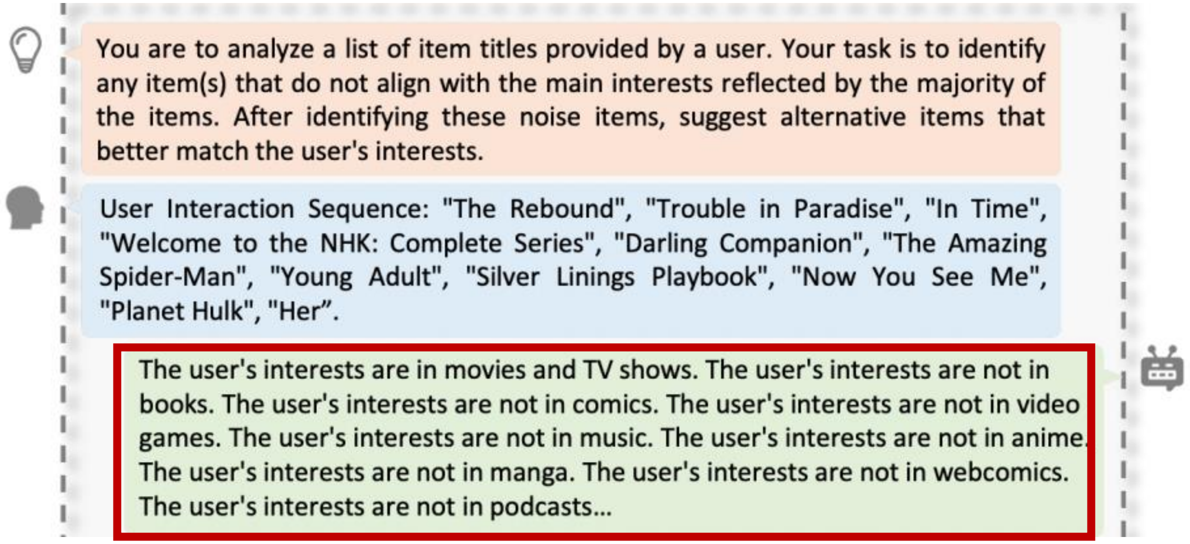
Task	LLM	CiteULike		MovieLens	
		Recall	NDCG	Recall	NDCG
Overall	ChatGPT	0.2054	0.1641	0.1396	0.2267
	TALLRec	0.2141	0.1661	0.1428	0.2324
	LLMRec	0.1983	0.1535	0.1372	0.2257
	LLM-InS	**0.2285**	**0.1747**	**0.1461**	**0.2368**
	%Improv.	6.73%	5.18%	2.31%	1.90%
Cold	ChatGPT	0.3477	0.2079	0.1480	0.1556
	TALLRec	0.3352	0.1990	0.1374	0.1379
	LLMRec	0.3453	0.2076	0.1425	0.1508
	LLM-InS	**0.3601**	**0.2126**	**0.1563**	**0.1566**
	%Improv.	3.57%	2.27%	5.61%	0.64%
Warm	ChatGPT	0.3001	0.1933	0.3074	0.2921
	TALLRec	0.3102	0.2037	0.3172	0.3016
	LLMRec	0.2613	0.1645	0.3022	0.2902
	LLM-InS	**0.3252**	**0.2156**	**0.3217**	**0.3060**
	%Improv.	4.84%	5.84%	1.42%	1.46%

Comparison Experiment

Score-based: LLM4DSR

Sequential recommendation systems (SRS) are often contaminated by noisy interactions.

Large Language Models (LLMs) has merged as a potential avenue to alleviate the noisy data. However, directly apply LLM to denoising task faces some challenges:



- **C1: Inadaptability of LLM:**

Direct application of pretrained LLMs may not be competent for the denoising task.

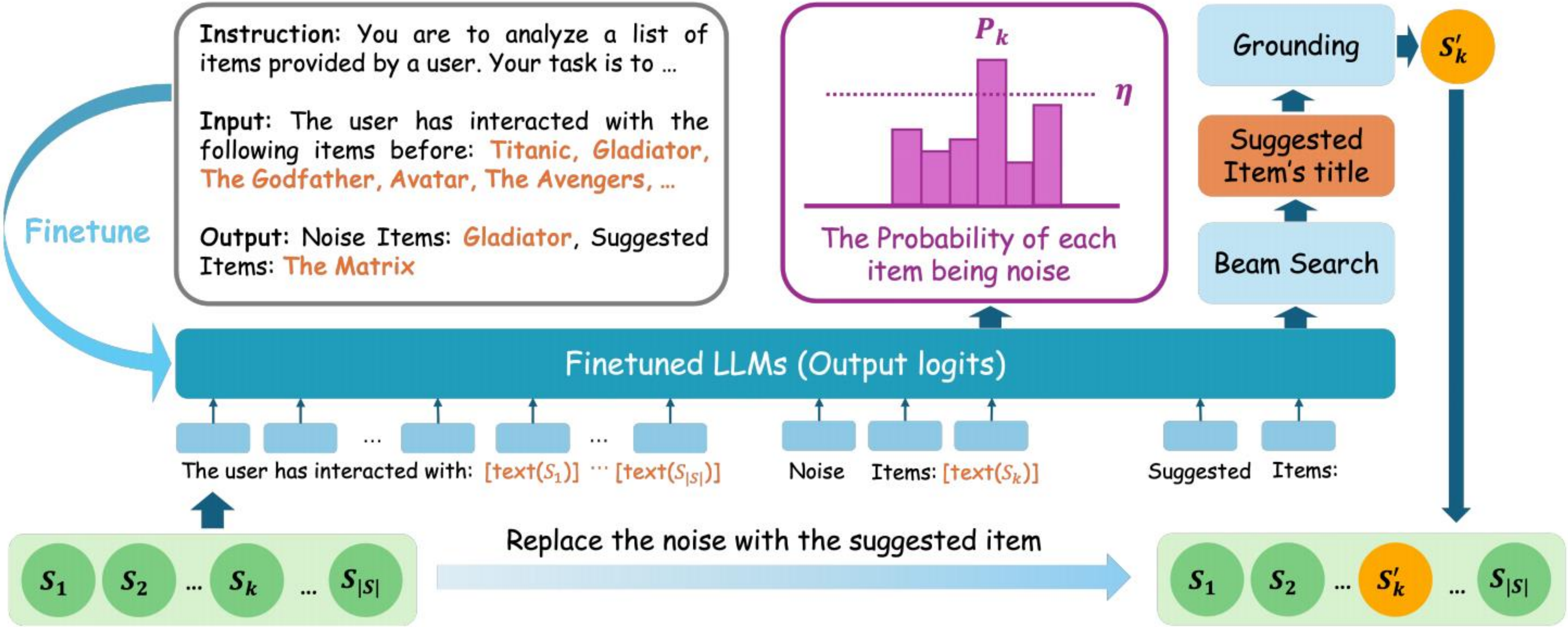
- **C2: Hallucination of LLM:**

The reliability of LLM's outputs remains questionable.

Wang, Bohao, et al. "Llm4dsr: Leveraing large language model for denoising sequential recommendation." arXiv preprint arXiv:2408.08208 (2024).

Score-based: LLM4DSR

To address the above challenges, the authors proposed LLM4DSR, featuring with self-supervised fine-tuning method and an uncertainty estimation mechanism to fully exploit the capabilities of LLMs.



Score-based: LLM4DSR

Specifically, for **C1: Inadaptability of LLM**, LLM4DSR constructed a new dataset and leveraged SFT to empower LLMs with denoising ability by using a "find & replace" task.

Prompt for Self-Supervised Task

Instruction: You are to analyze a list of items provided by a user. Your task is to identify an item that do not align with the main interests reflected by the majority of the items. After identifying these noise items, suggest alternative items that better match the user's interests.

Input: The user has interacted with the following items before: $text(s_1), \cdots, text(\hat{s}_t), \cdots, text(\hat{s}_{|S|})$

Output: Noise Items: $text(\hat{s}_t)$, Suggested Items: $text(s_t)$

- **Training Objective**:

By prompting the LLM using left instruction example, the authors construct a new binary prediction dataset and formulize the training objective as follows:

$$\max_{\Theta} \sum_{(x,z)} \sum_{i=1}^{|z|} \log \left(P_{\Phi}\left(z_i \mid x, z_{<i}\right)\right)$$

However, there is still one limitation that current LLM-based denoiser can only address a single noise item per sequence. (**L1: Inadaptability for Multi-noise Scenario**)

Score-based: LLM4DSR

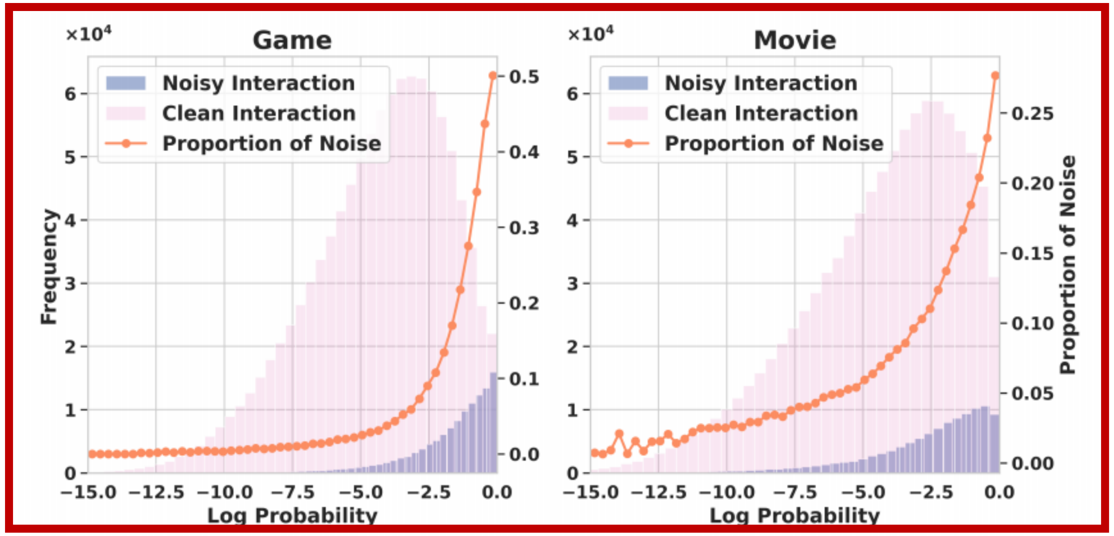
To address **L1: Inadaptability for Multi-noise Scenario & C2: Hallucination of LLM**, the Uncertainty Estimation module is introduced.

Prompt for Self-Supervised Task

Instruction: You are to analyze a list of items provided by a user. Your task is to identify an item that do not align with the main interests reflected by the majority of the items. After identifying these noise items, suggest alternative items that better match the user's interests.

Input: The user has interacted with the following items before: $text(s_1), \cdots, text(\hat{s}_t), \cdots, text(\hat{s}_{|S|})$

Output: Noise Items: $text(\hat{s_t})$, Suggested Items: $text(s_t)$



☐ Uncertainty Estimation:

Regarding the LLM provided probability after text "Noise Items:" as LLM's confidence, the probability values of all noisy items in the sequence can be obtained by the formula below:

$$
\begin{aligned}
P_{\text{noise}}(s_i|S) &= \prod_{j=1}^{|T_i|} P(T_{i,j}|x, z_{\text{noise}}, T_{i,<j}) \\
&= P(T_{i,1}|x, z_{\text{noise}}) P(T_{i,2}, \cdots, T_{i,|T_i|}|x, z_{\text{noise}}, T_{i,1}) \\
&= P(T_{i,1}|x, z_{\text{noise}})
\end{aligned}
$$

Score-based: LLM4DSR

For the experimental results, LLM4DSR achieves optimal results on three raw public datasets and noise-added datasets, validating the effectiveness of proposed methods.

Backbone	Method	Games				Toy				Movie			
		NDCG@20	NDCG@50	HR@20	HR@50	NDCG@20	NDCG@50	HR@20	HR@50	NDCG@20	NDCG@50	HR@20	HR@50
SASRec	None	0.0252	0.0314	0.0526	0.0838	0.0139	0.0178	0.0348	0.0542	0.0390	0.0455	0.0750	0.1084
	FMLP	0.0253	0.0315	0.0508	0.0824	0.0138	0.0173	0.0365	0.0547	0.0415	0.0484	0.0772	0.1120
	CL4SRec	0.0278	0.0336	0.0556	0.0864	0.0151	0.0183	0.0394	0.0557	0.0405	0.0464	0.0770	0.1070
	HSD	0.0198	0.0250	0.0412	0.0678	0.0090	0.0127	0.0254	0.0444	0.0340	0.0407	0.0700	0.1042
	STEAM	0.0261	0.0333	0.0532	0.0900	0.0123	0.0161	0.0321	0.0513	0.0401	0.0476	0.0740	0.1118
	LLM4DSR	**0.0292**	**0.0357**	**0.0624**	**0.0956**	**0.0152**	**0.0200**	**0.0401**	**0.0642**	**0.0431**	**0.0517**	**0.0876**	**0.1314**
BERT4Rec	None	0.0266	0.0335	0.0534	0.0880	0.0150	0.0195	0.0346	0.0574	0.0485	0.0569	0.0826	0.1258
	FMLP	0.0281	0.0353	0.0598	0.0962	0.0147	0.0188	0.0330	0.0536	0.0478	0.0561	0.0818	0.1240
	CL4SRec	0.0268	0.0341	0.0538	0.0906	0.0153	0.0196	0.0369	0.0586	0.0498	0.0579	0.0864	0.1276
	HSD	0.0288	0.0361	0.0600	0.0970	0.0152	0.0189	0.0340	0.0530	0.0390	0.0459	0.0682	0.1032
	STEAM	0.0294	0.0367	0.0562	0.0936	0.0182	0.0218	0.0388	0.0572	0.0472	0.0544	0.0836	0.1202
	LLM4DSR	**0.0316**	**0.0397**	**0.0644**	**0.1056**	**0.0183**	**0.0231**	**0.0401**	**0.0645**	**0.0509**	**0.0591**	**0.0890**	**0.1306**
LLaRA	None	0.0128	0.0172	0.0260	0.0484	0.0085	0.0108	0.0188	0.0305	0.0117	0.0136	0.0198	0.0294
	HSD	0.0141	0.0184	0.0274	0.0488	0.0109	0.0141	0.0240	0.0403	0.0111	0.0132	0.0186	0.0294
	STEAM	0.0108	0.0146	0.0224	0.0416	0.0092	0.0120	0.0202	0.0348	0.0107	0.0127	0.0182	0.0286
	LLM4DSR	**0.0150**	**0.0193**	**0.0286**	**0.0504**	**0.0121**	**0.0162**	**0.0280**	**0.0482**	**0.0159**	**0.0186**	**0.0250**	**0.0390**

Performance Comparison on Raw Datasets

Performance Comparison on Noise-added Datasets

Score-based: EIMF

Recent Studies have shown the rich semantic information can benefit the multi-interest modeling in Sequential Recommendation Systems (SRS).

However, there are still some problems when leveraging the semantic information to model user's behavioral information.

- **P1: Demand Conflict**

Traditional SRS has high real-time requirements while the fine-tuning and inference of LLM requires a lot of time and computing resources, leading to the demand conflict.

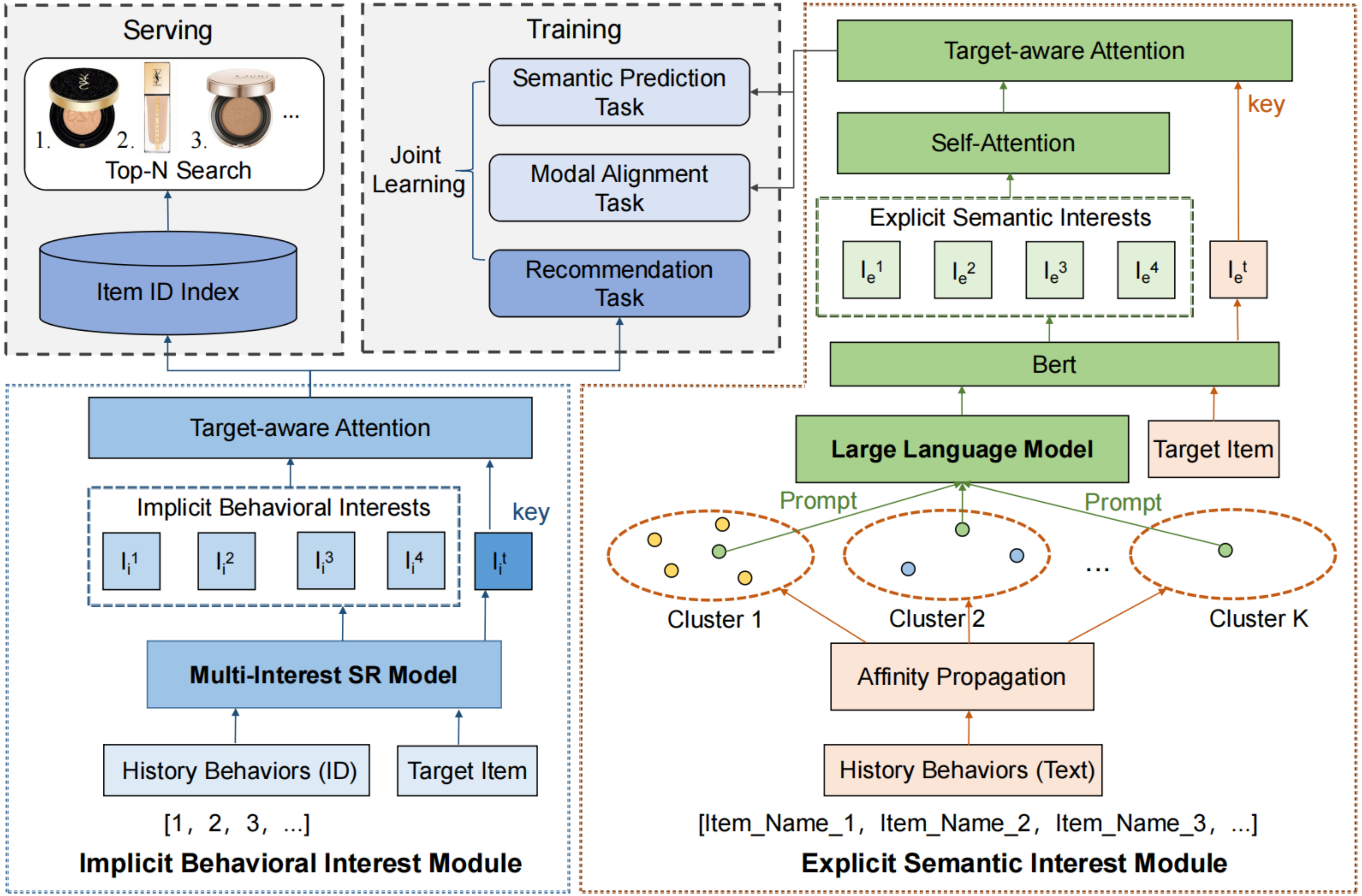
- **P2: Misunderstanding of User's Real Interests**

The semantic information from user/item content has significant gap with the behavioral information from well-trained recommendation model, leading to misunderstanding of user's real interests.

Qiao, Shutong, et al. "LLM-assisted Explicit and Implicit Multi-interest Learning Framework for Sequential Recommendation." arXiv preprint arXiv:2411.09410 (2024).

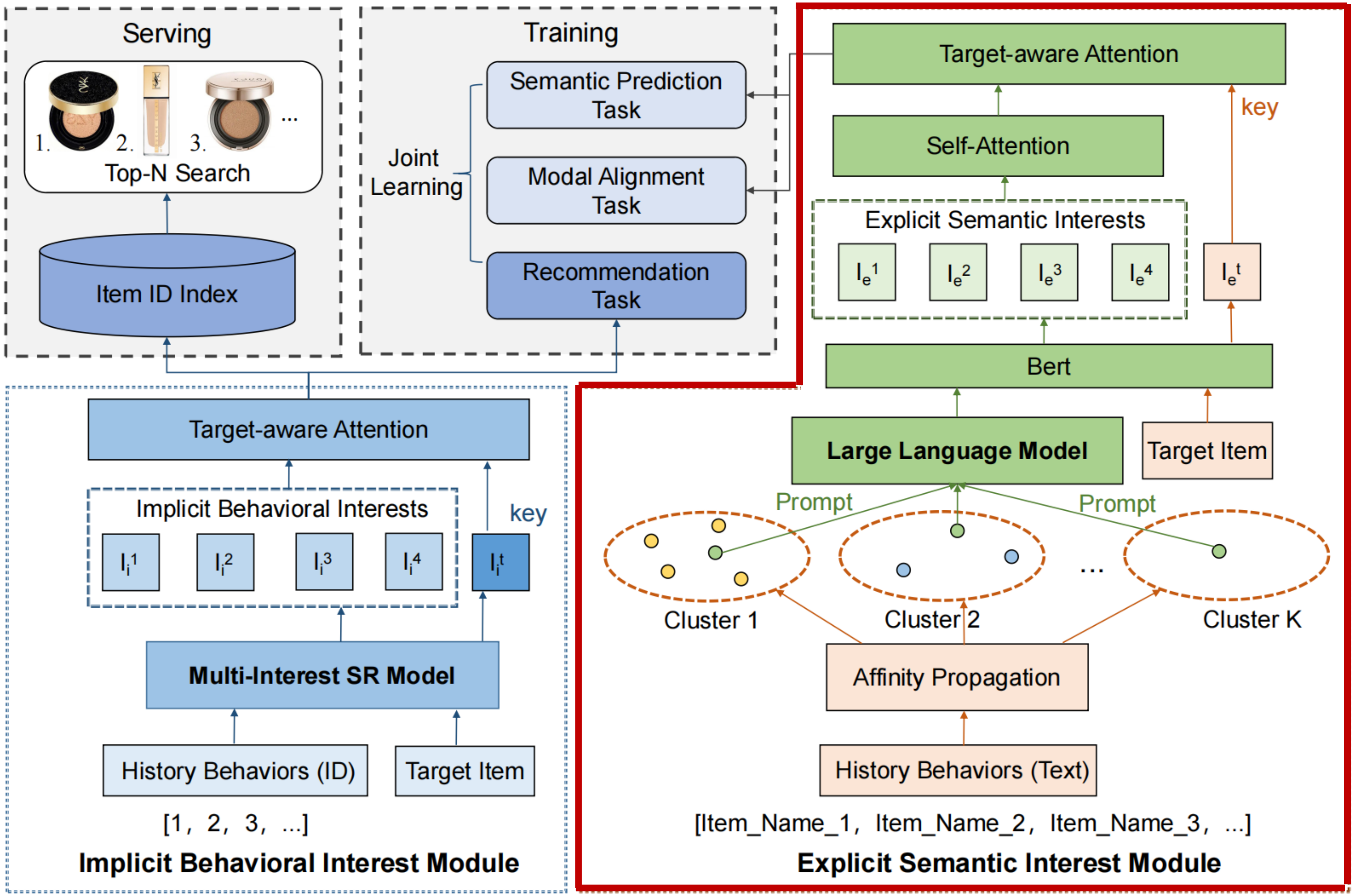
Score-based: EIMF

To address the above problems, the authors propose EIMF to adaptively combine traditional model with LLM.



Score-based: EIMF

Specifically, to address the '**P1: Demand Conflict**', EIMF introduces Explicit Behavioral Interest to fully explore the LLM's capability while reducing the inference cost.



- **Affinity Propagation & LLM Summarization:**

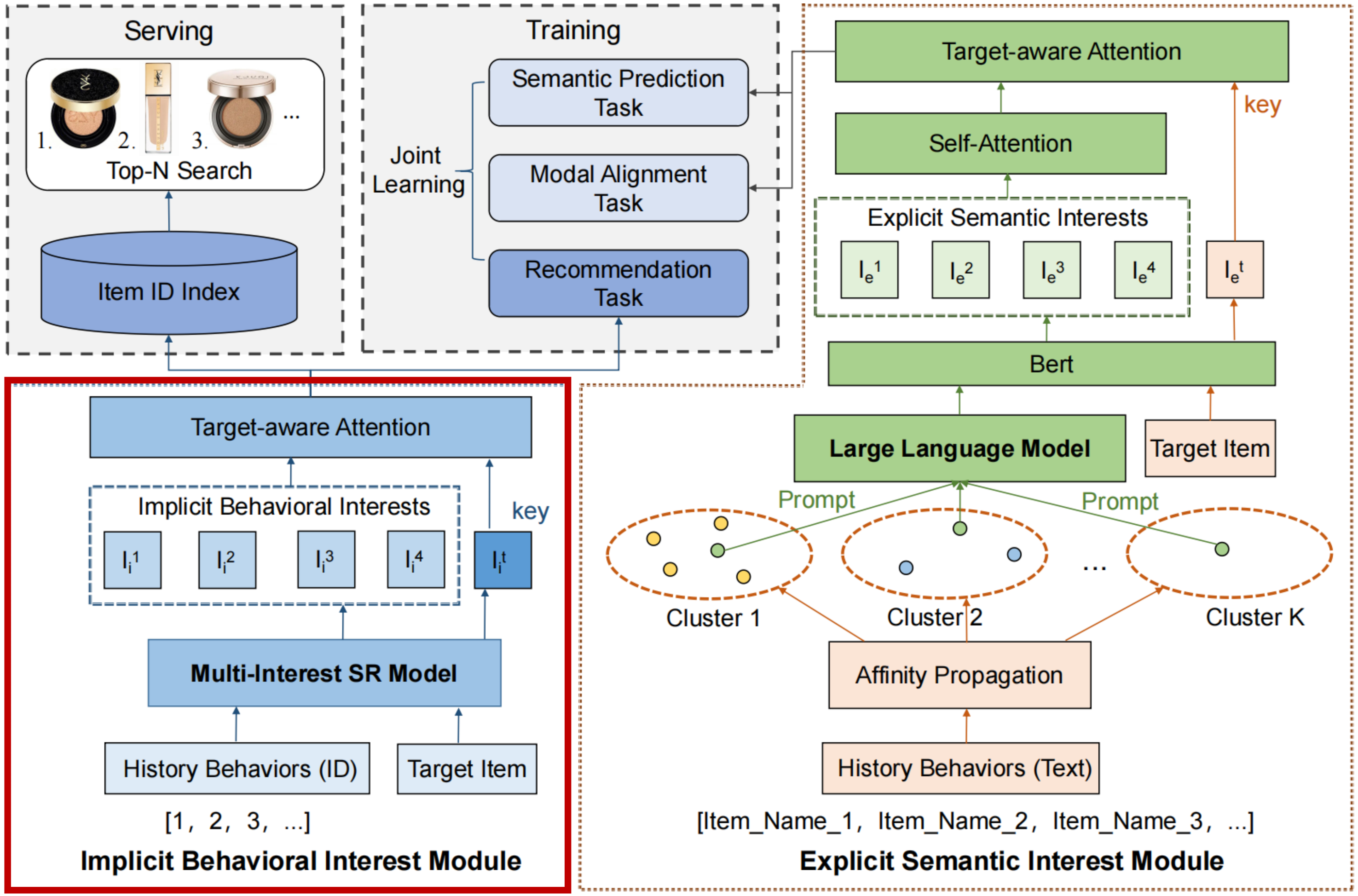
EIMF dividse users into groups and utilize LLM as summarizer to reduce the inference cost.

- **Interest Embedding**

EIMF queries the semantic interest embedding by utilizing the target item as key, which can be used for alignment and semantic prediction.

Score-based: EIMF

To address the '**P2: Misunderstanding of User's Real Interests**', EIMF combines the Behavioral Interest with the Semantic Interest by jointly learning three kinds of tasks.



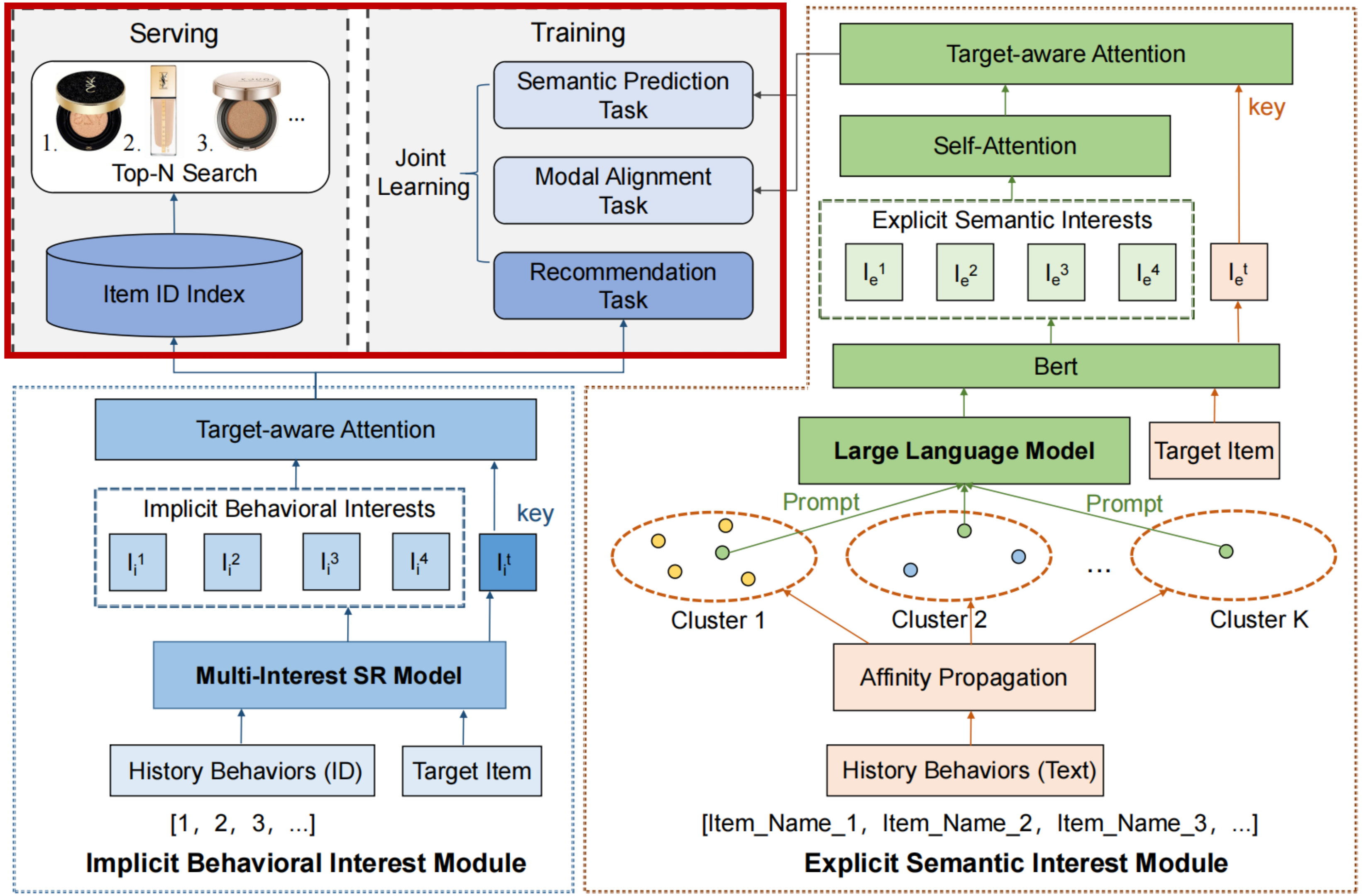
☐ Behavioral Interest:

Firstly, EIMF obtains the implicit behavioral interest by a SR model.

Then using similar Target-aware Attention network to acquire the behavioral embeddings to construct the loss.

Score-based: EIMF

To address the '**P2: Misunderstanding of User's Real Interests**', EIMF combines the Behavioral Interest with the Semantic Interest by jointly learning three kinds of tasks.



☐ Joint Learning:

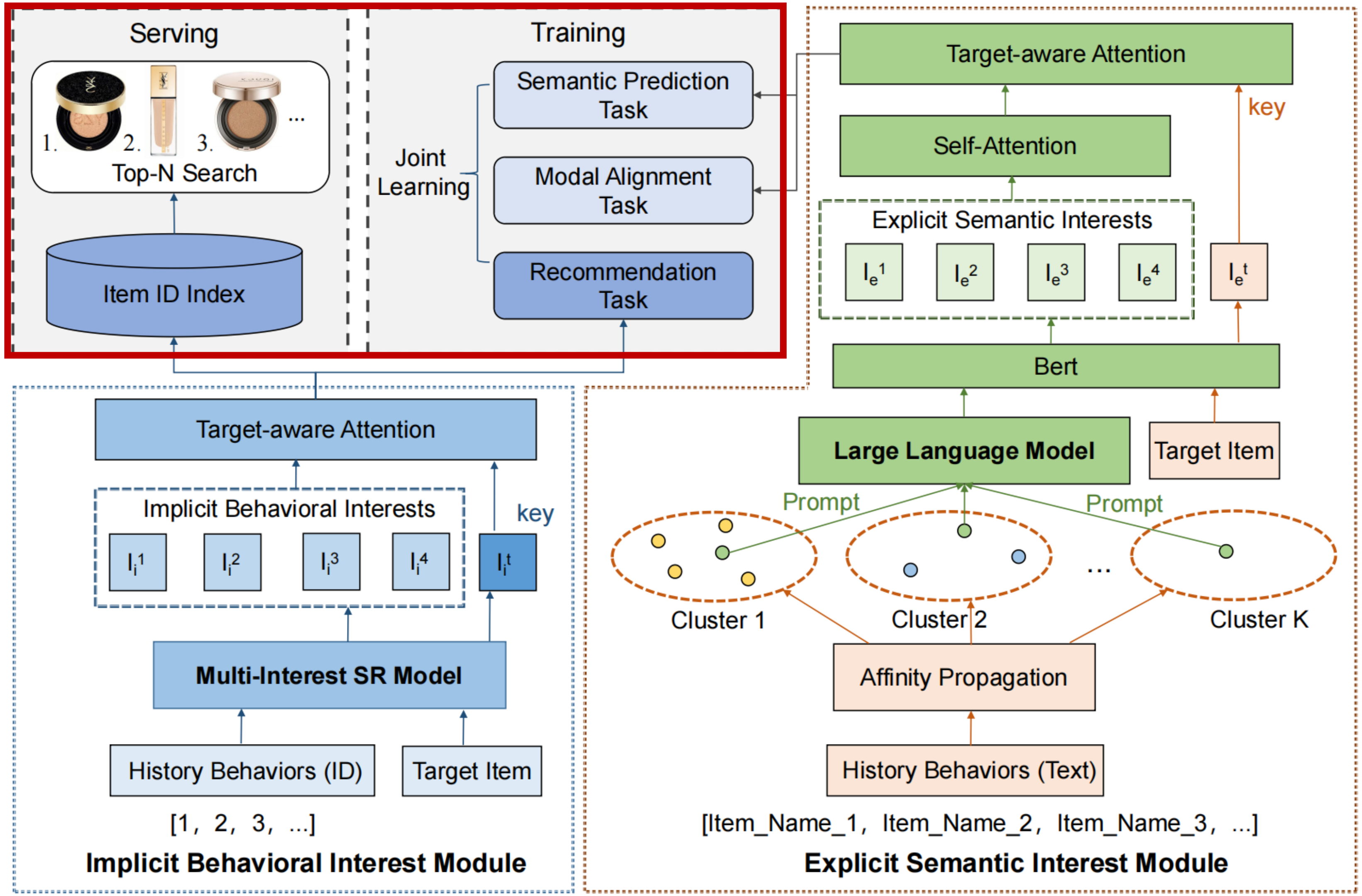
- Semantic Prediction Task

Calculating the score between user's semantic interest and item text:

$$\mathcal{L}_S = \sum_{k=1}^{n} y_t^k \log(\hat{y}_t^k)$$

Score-based: EIMF

To address the '**P2: Misunderstanding of User's Real Interests**', EIMF combines the Behavioral Interest with the Semantic Interest by jointly learning three kinds of tasks.



☐ Joint Learning:

- Modal Alignment Task

Building the contrastive loss to align the semantic embedding and behavioral embedding as:

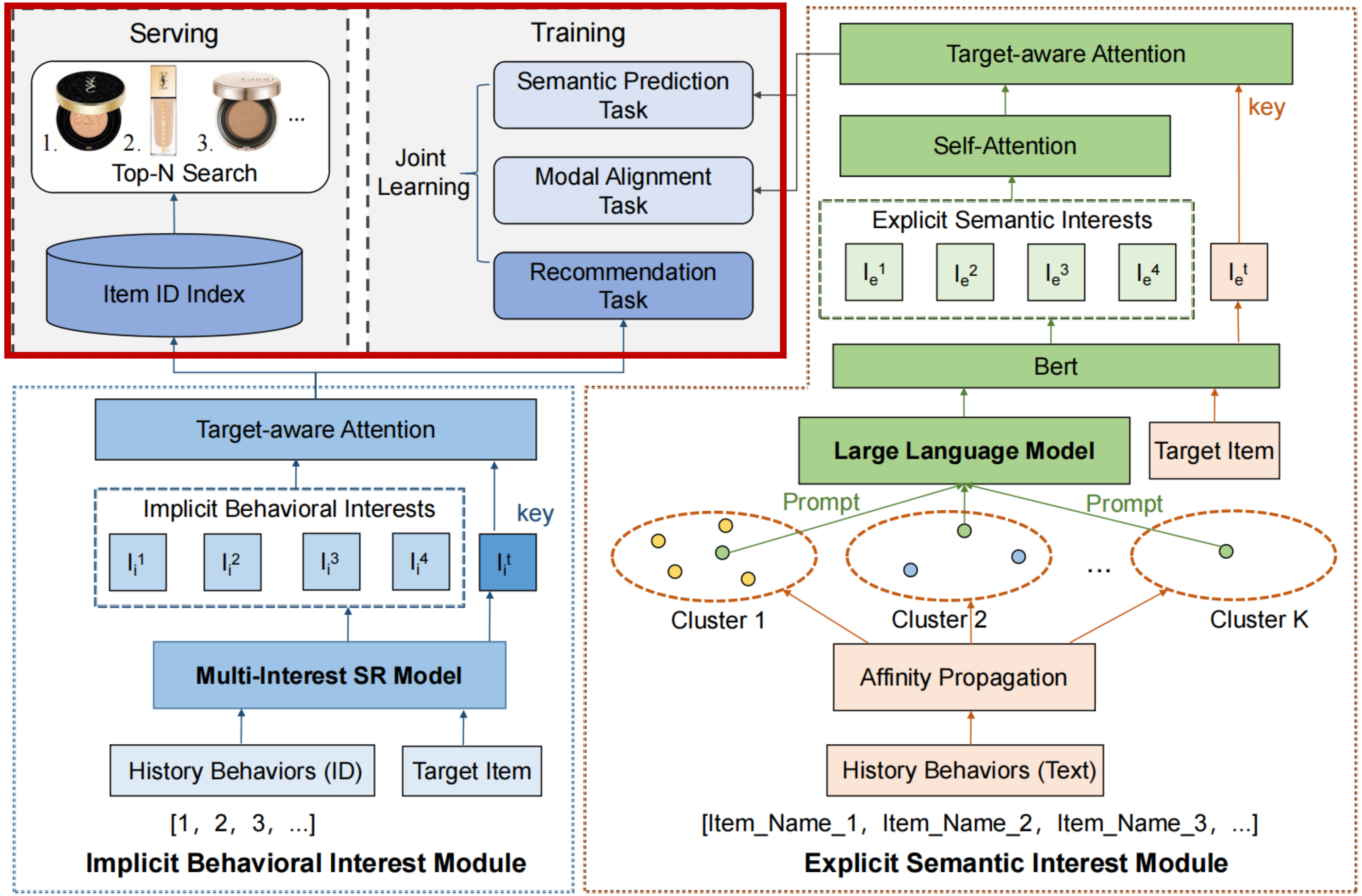
$$\mathrm{CL}(e_a, e_b) = -\frac{1}{N}\sum_{k=1}^{N} log\left(\frac{\exp(\mathrm{Sim}(e_a^k, e_b^k)/\tau)}{\sum_{j=1}^{N}\exp(\mathrm{Sim}(e_a^k, e_b^j)/\tau)}\right),$$

$$\mathrm{Cos}(e_a, e_b) = \frac{1}{N}\sum_{k=1}^{N}\left(1 - \mathrm{Sim}(e_a^k, e_b^k)\right),$$

$$\mathcal{L}_A = \alpha(\mathrm{CL}(h_{ex}, h_{im}) + \mathrm{CL}(e_t^T, e_i^T)) + \beta(\mathrm{Cos}(h_{ex}, h_{im}) + \mathrm{Cos}(e_t^T, e_i^T))$$

Score-based: EIMF

To address the '**P2: Misunderstanding of User's Real Interests**', EIMF combines the Behavioral Interest with the Semantic Interest by jointly learning three kinds of tasks.



☐ Joint Learning:

- Recommendation Task

$$\hat{y}_i^k = \text{softmax}(h_{\text{ex}}^\top e_i^k)$$

$$\mathcal{L}_R = \sum_{k=1}^{n} y_i^k log(\hat{y}_i^k)$$

Finally, the whole loss can be formulized as:

$$\mathcal{L} = \mathcal{L}_R + \gamma\left(\mathcal{L}_S + \mathcal{L}_A\right)$$

Score-based: EIMF

The experimental results show that EIMF achieves remarkable results on three public datasets (two large and one small), presenting that EIMF's superiority.

Dataset	Model	Pop	DNN	GRU4Rec	SASRec	Bert4Rec	LLM2Bert4Rec	MIND	ComiRec-SA	REMI	EIMF(REMI)	Improv.(%)
Grocery	Recall@20	0.0729	0.1252	0.1387	0.1536	0.1544	0.1259	0.1445	0.1122	0.1617	**0.1758**	+8.71
	Recall@50	0.1305	0.2044	0.2300	0.2508	0.2372	0.2080	0.2162	0.2076	0.2574	**0.2704**	+5.05
	NDCG@20	0.0448	0.0804	0.0905	0.1041	**0.1074**	0.0793	0.0844	0.0706	0.0953	0.1025	-4.56
	NDCG@50	0.0624	0.0969	0.1097	0.1139	0.1130	0.0954	0.0967	0.0918	0.1108	**0.1181**	+3.69
	HR@20	0.1252	0.2076	0.2321	0.2586	0.2614	0.2164	0.2328	0.1851	0.2539	**0.2750**	+6.34
	HR@50	0.2137	0.3227	0.3649	0.3750	0.3608	0.3294	0.3322	0.3220	0.3832	**0.3955**	+3.21
Beauty	Recall@20	0.0452	0.1613	0.1388	0.1495	0.1430	0.1383	0.1563	0.1582	0.2189	**0.2323**	+6.12
	Recall@50	0.0660	0.2361	0.2109	0.2220	0.2050	0.2053	0.2384	0.2594	0.3420	**0.3636**	+6.31
	NDCG@20	0.0213	0.0886	0.0798	0.0854	0.0846	0.0788	0.0787	0.0854	0.1139	**0.1204**	+5.71
	NDCG@50	0.0270	0.0932	0.0844	0.0879	0.0852	0.0823	0.0900	0.1004	0.1304	**0.1348**	+3.37
	HR@20	0.0675	0.2401	0.2141	0.2320	0.2221	0.2181	0.2203	0.2325	0.3111	**0.3268**	+5.04
	HR@50	0.0966	0.3299	0.2982	0.3129	0.2986	0.2919	0.3272	0.3545	0.4532	**0.4802**	+5.95

Performance comparison on two large datasets

Model	Recall@20	NDCG@20	HR@20	Recall@50	NDCG@50	HR@50
GRU4Rec	0.0800	0.0537	0.1384	0.1770	0.0809	0.2811
SASRec	0.1152	0.0643	0.1792	0.1963	0.0805	0.2871
Bert4Rec	0.1037	0.0635	0.1812	0.1920	0.0834	0.3116
MIND	0.0915	0.0513	0.1466	0.1593	0.0706	0.2505
ComiRec-SA	0.0788	0.0456	0.1202	0.1589	0.0653	0.2321
REMI	0.1072	0.0637	0.1751	0.2018	0.0825	0.2973
LLM2Bert4Rec	0.1085	**0.0660**	**0.1812**	0.2117	**0.0939**	0.3360
EIMF(REMI)	**0.1222**	0.0649	0.1772	**0.2296**	0.0930	**0.3503**
Improv.(%)	+6.08	-1.66	-2.20	+8.45	-0.95	+4.25

Performance comparison on one small dataset

Score-based: EIMF

Moreover, the experiment performing on different backbones shows a significant improvement, further validating the effectiveness of proposed modules in EIMF.

Dataset	Beauty			Office		
Model	Recall@50	NDCG@50	HR@50	Recall@50	NDCG@50	HR@50
Bert4Rec	0.2050	0.0852	0.2986	0.1920	0.0834	0.3116
EIMF(Bert4Rec)	0.2155	0.0854	0.3071	0.2101	0.0909	0.3380
Improv.(%)	+5.12	+0.23	+2.84	+9.42	+8.99	+8.47
SASRec	0.2220	0.0879	0.3129	0.1964	0.0806	0.2872
EIMF(SASRec)	0.2499	0.0999	0.3469	0.2088	0.0878	0.3198
Improv.(%)	+12.56	+13.65	+10.86	+6.31	+8.93	+11.35
MIND	0.2384	0.0900	0.3272	0.1594	0.0707	0.2505
EIMF(MIND)	0.2518	0.0968	0.3424	0.2023	0.0793	0.2953
Improv.(%)	+5.62	+7.55	+4.64	+26.91	+12.16	+17.88

Performance comparison on different backbones

Summary

Here, we give an overview of the works on interaction enhancement for LLM Enhanced Sequential Recommendation (LLMESR).

- Interaction §3
  - Text-based §3.1
    - ONCE [45]; LLMRec [92]; LlamaRec [58]; ColdAug [80]; LLMHD [71]; LLM4IDRec [7]; SampleLLM [17]; LLMSeR [72]
  - Score-based §3.2
    - LLM-Ins [27]; LLM4DSR [78]; EIMF [64]

- **Text-based**: This kind of methods leverage LLM to give out the names of pseudo-interacted items in the user-item sequence as the augmentation.

- **Score-based**: The score-based methods utilize LLM to derive the logits of the possible interactions and generate the augmented items by ranking.

Agenda

1 Introduction

Zijian Zhang

2 Knowledge Enhancement

Pengyue Jia

3 Interaction Enhancement

Ziwei Liu

4.1 Model Enhancement 1

Maolin Wang

4.2 Model Enhancement 2

Yuhao Wang

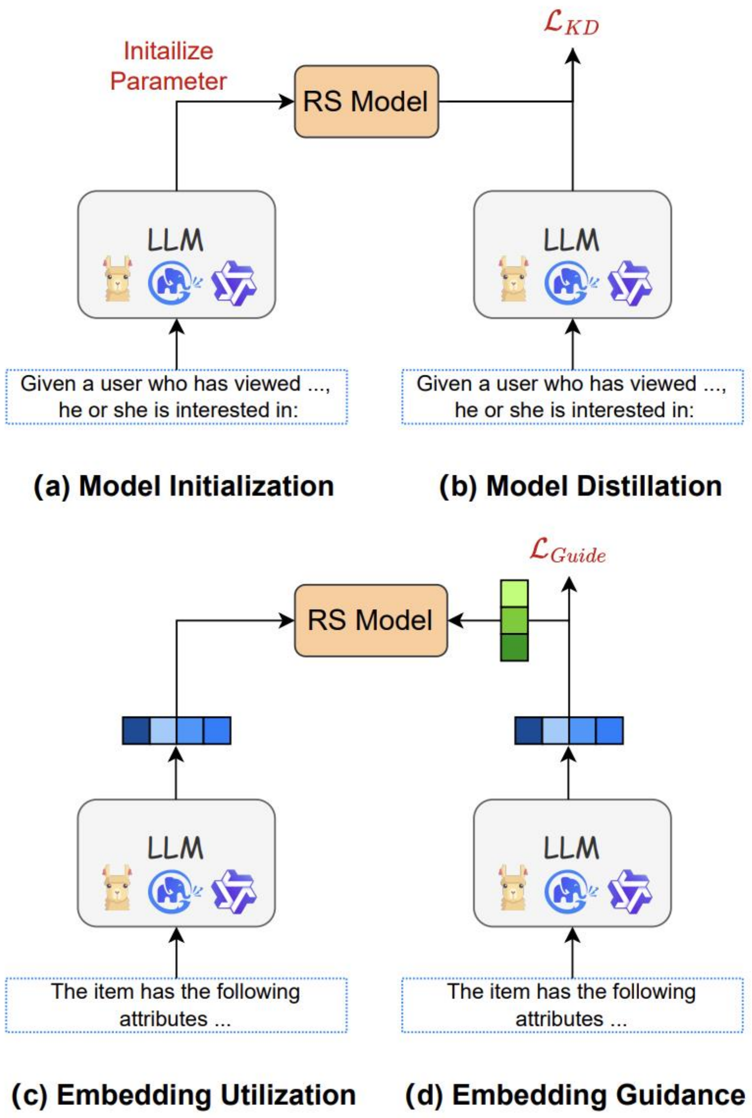
5 Conclusion

Yejing Wang

Model Enhancement

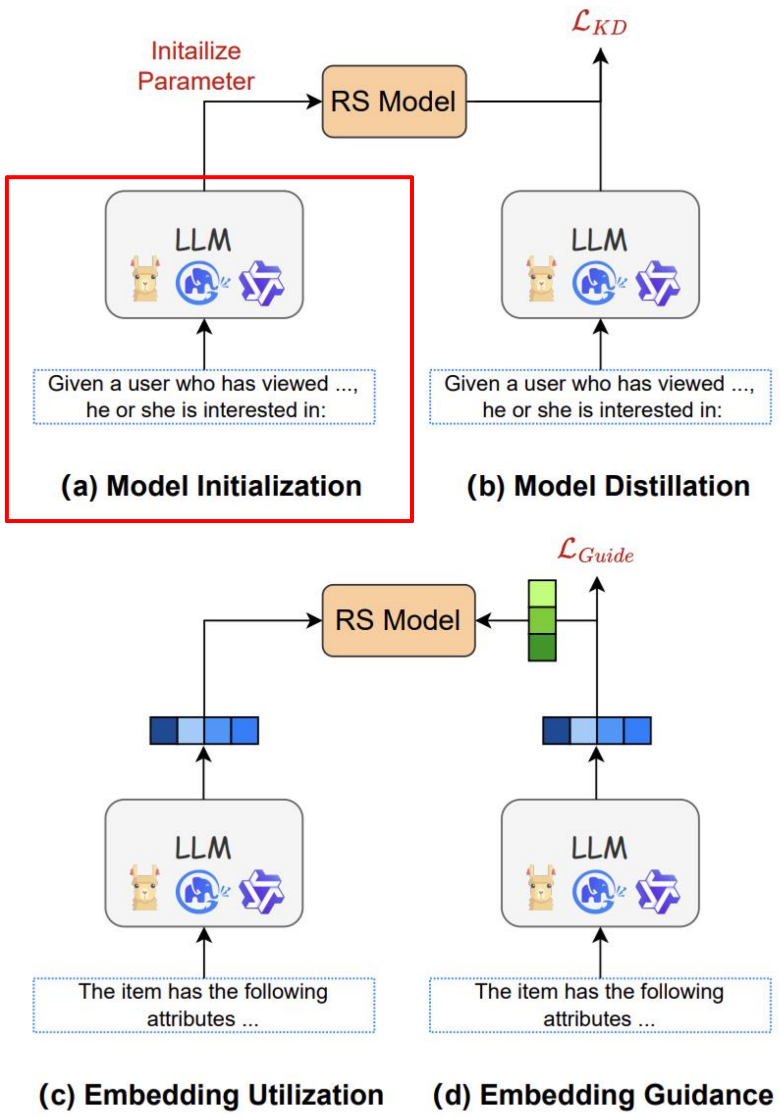
- LLMs possess powerful semantic capabilities that can be directly integrated into recommendation systems to enhance models.

- Categories
  - Model Initialization
  - Model Distillation
  - Embedding Utilization
  - Embedding Guidance



Model Initialization

- This subcategory refers to utilizing LLM-derived semantics to initialize the weights of recommendation models before training begins.
- This approach accelerates model convergence while preserving semantic knowledge from LLM pretraining for downstream recommendation tasks.
- Categories
  - Whole
  - Embedding



Whole - CTRL

- Background
  - CTR prediction is a core task in recommender systems, where traditional collaborative models capture user preferences through feature co-occurrence.

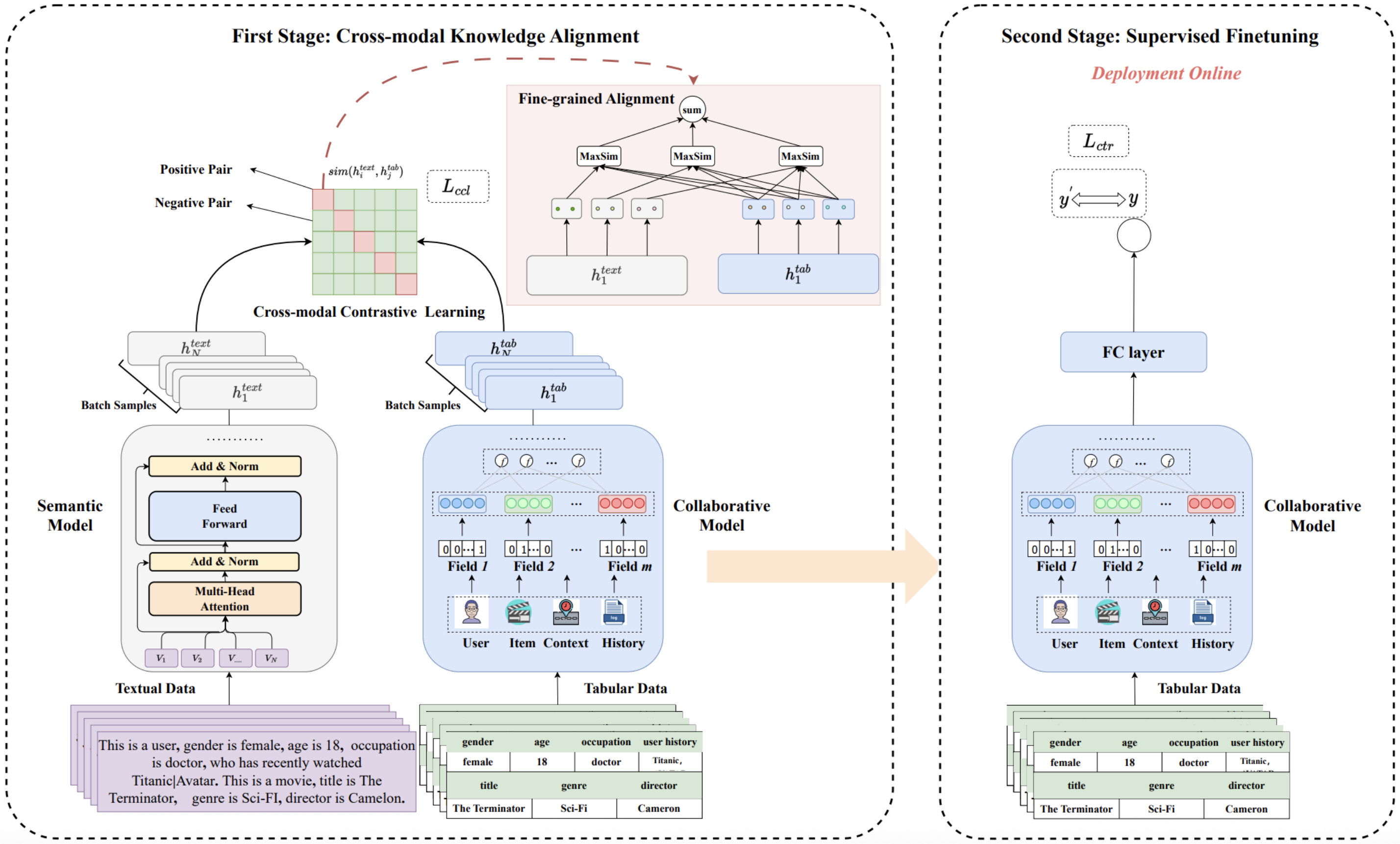
- Motivation
  - One-hot encoding discards semantic relationships, leading to poor performance in cold-start scenarios.
  - Direct use of Pre-trained Language Models is computationally expensive and fails to meet low-latency requirements for online inference.

- Key Stages
  - Cross-modal Knowledge Alignment
  - Supervised Fine-tuning

Li, Xiangyang, et al. "Ctrl: Connect collaborative and language model for ctr prediction." ACM Transactions on Recommender Systems (2023).

Whole - CTRL

- **CTRL** proposes a two-stage framework to effectively integrate semantic knowledge into collaborative.
- **Cross-modal Knowledge Alignment**
  - Tabular data feeds collaborative model, text prompts feed PLM. Contrastive learning aligns representations to distill semantic knowledge.
- **Supervised Fine-tuning**
  - Enhanced collaborative model fine-tunes on CTR task. Only lightweight collaborative model deploys online, no PLM needed.



Whole - CTRL

- CTRL significantly outperforms SOTA collaborative and semantic models on multiple public datasets and an industrial system, while maintaining high inference efficiency.

Category	Model	MovieLens			Amazon			Alibaba		
		AUC	Logloss	RelaImpr	AUC	Logloss	RelaImpr	AUC	Logloss	RelaImpr
Collaborative Models	Wide&Deep	0.8261	0.4248	3.52%	0.6968	0.4645	5.30%	0.6272	0.1943	5.19%
	DeepFM	0.8268	0.4219	3.30%	0.6969	0.4645	5.33%	0.6280	0.1951	4.53%
	DCN	<u>0.8313</u>	<u>0.4165</u>	1.90%	0.6999	0.4642	3.75%	<u>0.6281</u>	0.1949	4.45%
	PNN	0.8269	0.4220	3.27%	0.6979	0.4657	4.80%	0.6271	0.1956	5.27%
	AutoInt	0.8290	0.4178	2.61%	<u>0.7012</u>	<u>0.4632</u>	3.08%	0.6279	<u>0.1948</u>	4.61%
	FiBiNet	0.8196	0.4188	5.63%	0.7003	0.4704	3.54%	0.6270	0.1951	5.35%
	xDeepFM	0.8296	0.4178	2.43%	0.7009	0.4642	3.23%	0.6272	0.1959	5.19%
Semantic Models	P5	0.7583	0.4912	30.70%	0.6923	0.4608	7.85%	0.6034	0.3592	29.40%
	CTR-BERT	0.7650	0.4944	27.40%	0.6934	0.4629	7.24%	0.6005	0.3620	33.13%
	P-Tab	0.8031	0.4612	11.38%	0.6942	0.4625	6.80%	0.6112	0.3584	20.32%
CTRL		**0.8376**⋆	**0.4025**⋆	-	**0.7074**⋆	**0.4577**⋆	-	**0.6338**⋆	**0.1890**⋆	-

- **Practical Values:** CTRL achieves superior performance by fusing collaborative and semantic signals, maintains efficient inference, and is industrial-friendly with flexible model compatibility.

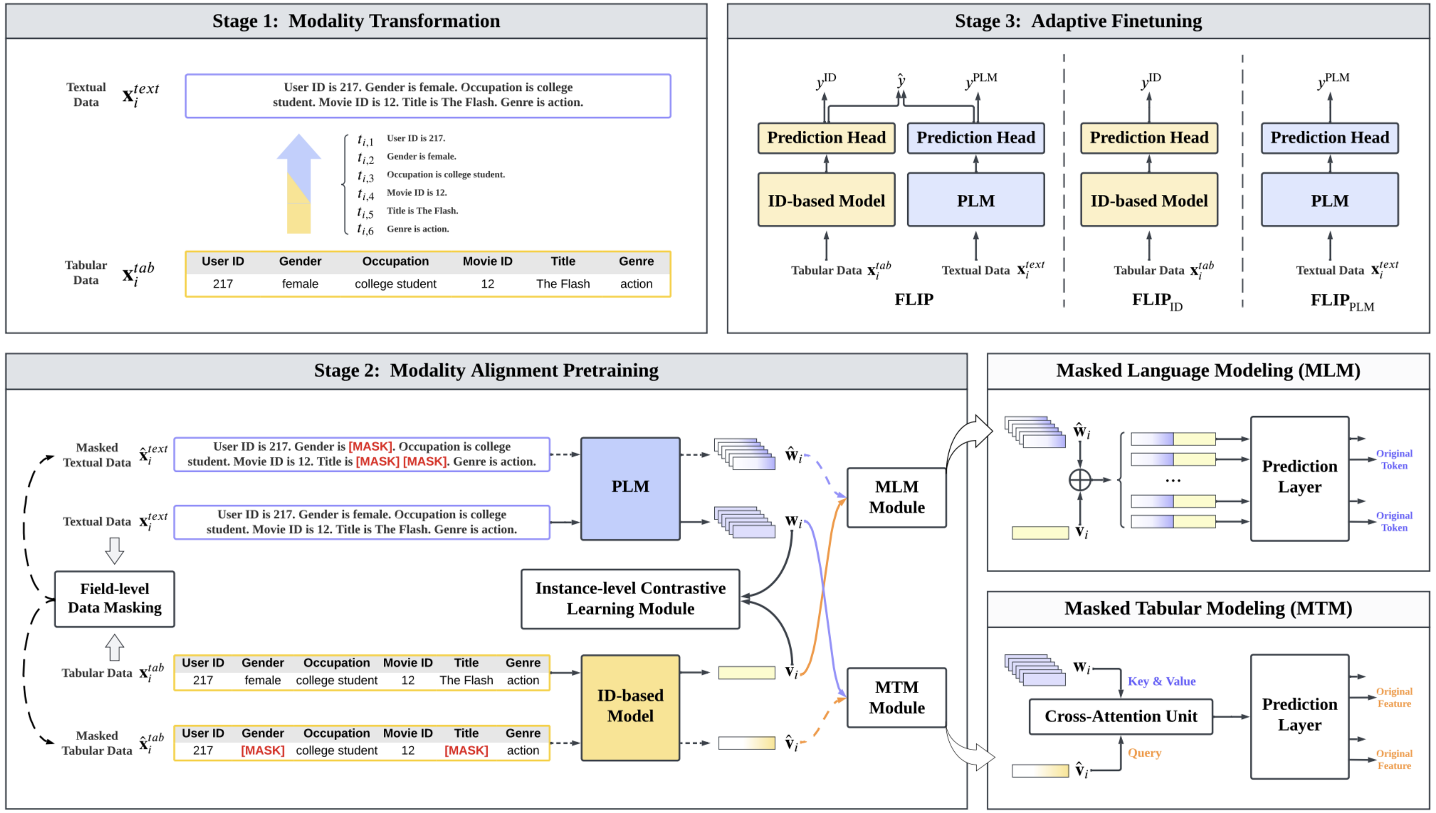
Whole - FLIP

- Background
  - CTR prediction is a core task in recommender systems, where traditional ID-based models use one-hot encoding and PLMs use textual modality.
- Motivation
  - One-hot encoding discards semantic information, leading to poor performance in sparse scenarios.
    PLMs struggle with field-wise collaborative signals and have high computational costs for online inference.
- Key Components
  - Modality Transformation
  - Modality Alignment Pretraining
  - Adaptive Finetuning

Wang, Hangyu, et al. "FLIP: Fine-grained Alignment between ID-based Models and Pretrained Language Models for CTR Prediction." Proceedings of the 18th ACM Conference on Recommender Systems. 2024.

Whole - FLIP

- **Modality Transformation**
  - Convert tabular features to textual sentences using simple templates, preserving semantic information in natural language format.
- **Modality Alignment Pretraining**
  - Field-level masking with MLM, MTM, and ICL for fine-grained cross-modal alignment.
- **Adaptive Finetuning**
  - Joint training with learnable weight α combining both model outputs for CTR prediction.



Whole - FLIP

- **FLIP** significantly outperforms SOTA on three public datasets. It also shows strong compatibility with various ID models and PLMs.

Model		MovieLens-1M			BookCrossing			GoodReads		
		AUC	Logloss	Rel.Impr	AUC	Logloss	Rel.Impr	AUC	Logloss	Rel.Impr
ID-based	AFM	0.8449	0.3950	1.79%	0.7946	0.5116	1.06%	0.7630	0.5160	1.96%
	PNN	0.8546	0.3946	0.63%	0.7956	0.5131	0.93%	0.7725	0.5055	0.70%
	Wide&Deep	0.8509	0.3957	1.07%	0.7951	0.5116	1.00%	0.7684	0.5090	1.24%
	DCN	0.8509	0.4056	1.07%	0.7957	0.5108	0.92%	0.7693	0.5086	1.12%
	DeepFM	0.8539	0.3905	0.71%	0.7947	0.5122	1.04%	0.7671	0.5138	1.41%
	xDeepFM	0.8454	0.3934	1.72%	0.7953	0.5108	0.97%	0.7720	0.5079	0.77%
	AFN	0.8525	0.3868	0.88%	0.7932	0.5139	1.24%	0.7654	0.5118	1.64%
	AutoInt	0.8509	0.4013	1.07%	0.7953	0.5118	0.97%	0.7716	0.5071	0.82%
	DCNv2	0.8548	0.3893	0.61%	0.7956	0.5103	0.93%	0.7724	0.5057	0.72%
	FLIP$_{ID}$ (Ours)	**0.8600***	**0.3802***	-	**0.8030***	**0.5043***	-	**0.7779***	**0.5014***	-
PLM-based	CTR-BERT	0.8304	0.4131	1.88%	0.7795	0.5300	1.65%	0.7385	0.5316	1.23%
	P5	0.8304	0.4173	1.88%	0.7801	0.5261	1.58%	0.7365	0.5336	1.51%
	PTab	0.8426	0.4195	0.41%	0.7880	0.5384	0.56%	0.7456	0.5268	0.27%
	FLIP$_{PLM}$ (Ours)	**0.8460***	**0.4127***	-	**0.7924***	**0.5304***	-	**0.7476***	**0.5255***	-
ID+PLM	CTRL	0.8572	0.3838	0.57%	0.7985	0.5101	0.95%	0.7741	0.5045	0.59%
	MoRec	0.8561	0.3896	0.70%	0.7990	0.5087	0.89%	0.7731	0.5085	0.72%
	FLIP (Ours)	**0.8621***	**0.3788***	-	**0.8061***	**0.5004***	-	**0.7787***	**0.5001***	-

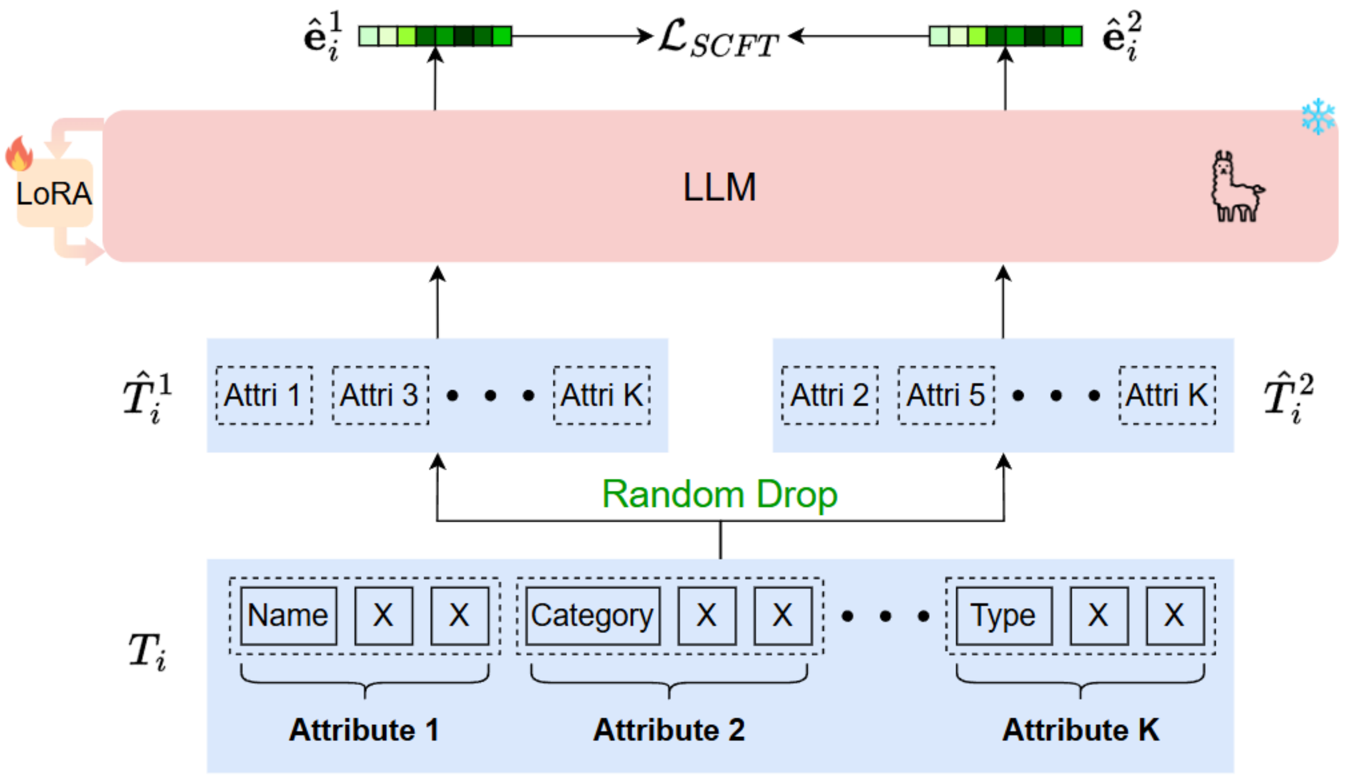
- **Practical Values:** FLIP's fine-grained alignment (joint MLM+MTM) outperforms instance-level methods, enables meaningful feature-level interactions, and enhances diverse ID-based models (DeepFM, AutoInt, DCNv2) and PLMs of varying sizes.

Embedding - LLMEmb

- Background
  - Sequential recommender systems (SRS) predict user preferences through interaction history using learned item embeddings
- Motivation
  - SRS models lack semantic understanding of textual item attributes.
  - LLMs have strong semantics but poor item-level distinction for recommendations.
- Key Stages
  - Supervised Contrastive Fine-Tuning (SCFT)
  - Recommendation Adaptation Training (RAT)

Liu, Qidong, et al. "LLMEmb: Large Language Model Can Be a Good Embedding Generator for Sequential Recommendation." Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 39. No. 11. 2025.

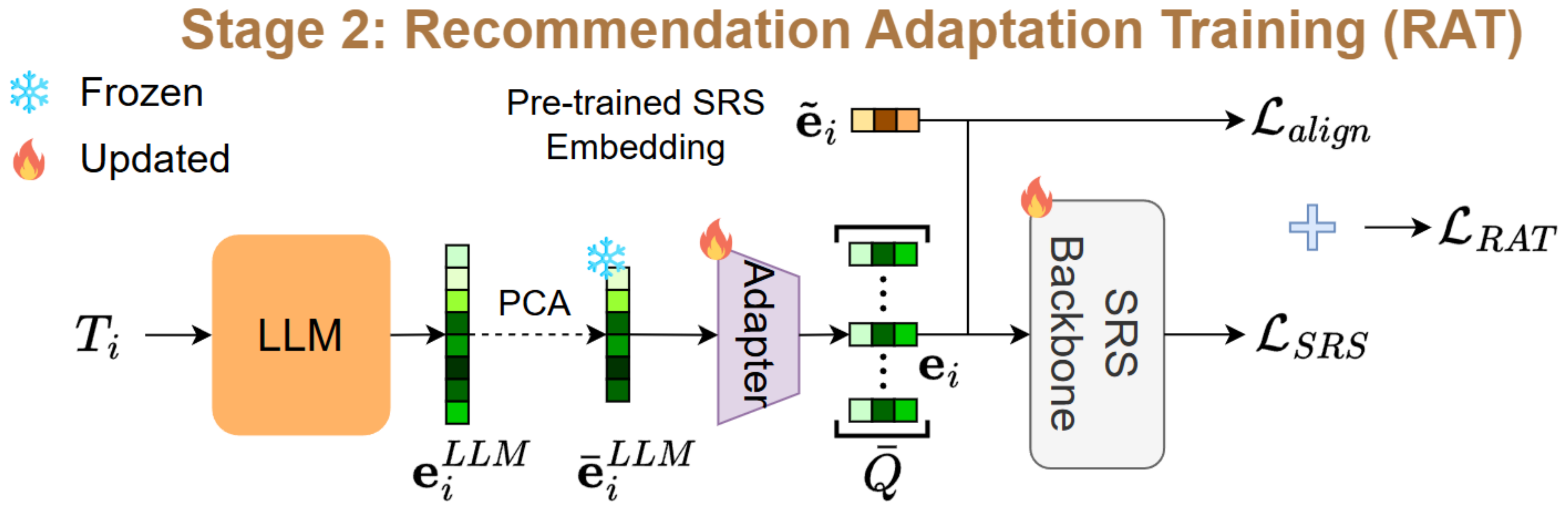
Embedding - LLMEmb



Stage 1: Supervised Contrastive Fine-Tuning (SCFT)

- **Build prompts for each item**
  - Includes instruction + attribute-value pairs.
- **Data Augmentation**
  - Randomly drop attributes to create two prompt views per item (as positive pairs).
- **Contrastive Learning**
  - Use contrastive loss to bring same-item embeddings closer, push different-item embeddings apart.
- **Outcome**
  - LLM learns to encode subtle attribute differences, providing stronger item representations.

Embedding - LLMEmb



- **Embedding Transformation:** Use PCA to reduce dimensionality, then an adapter (two-layer MLP) to match the SRS size.
- **Adaptation:** Freeze LLM, train the adapter and SRS backbone together using SRS loss.
- **Collaborative Alignment**: Align adapter output with embeddings from a trained SRS, using contrastive loss to prevent overfitting.

Embedding - LLMEmb

- **LLMEmb** outperforms all baselines on Yelp, Amazon Beauty, and Fashion, especially improving recommendations for less popular (long-tail) items.

Backbone	Model	Yelp				Fashion				Beauty			
		Overall		Tail		Overall		Tail		Overall		Tail	
		H@10	N@10	H@10	N@10	H@10	N@10	H@10	N@10	H@10	N@10	H@10	N@10
GRU4Rec	- None	0.4879	0.2751	0.0171	0.0059	0.4798	0.3809	0.0257	0.0101	0.3683	0.2276	0.0796	0.0567
	- MELT	0.4985	0.2825	0.0201	0.0079	0.4884	0.3975	0.0291	0.0112	0.3702	0.2161	0.0009	0.0003
	- LLM2X	0.4872	0.2749	0.0201	0.0072	0.4881	0.4100	0.0264	0.0109	0.4151	0.2713	0.0896	0.0637
	- SAID	0.4891	0.2764	0.0180	0.0062	0.4920	0.4168	0.0347	0.0151	0.4193	0.2621	0.0936	0.0661
	- TSLRec	0.4528	0.2509	0.0255	0.0095	0.4814	0.4042	0.0149	0.0071	0.3119	0.1865	0.0750	0.0474
	- LLMEmb	**0.5270***	**0.2980***	**0.1116***	**0.0471***	**0.5062***	**0.4329***	**0.1046***	**0.0477***	**0.4445***	**0.2726**	**0.3183***	**0.1793***
Bert4Rec	- None	0.5307	0.3035	0.0115	0.0044	0.4668	0.3613	0.0142	0.0067	0.3984	0.2367	0.0101	0.0038
	- MELT	0.6206	0.3770	0.0429	0.0149	0.4897	0.3810	0.0059	0.0019	0.4716	0.2965	0.0709	0.0291
	- LLM2X	0.6199	0.3781	0.0874	0.0330	0.5109	0.4159	0.0377	0.0169	0.5029	0.3209	0.0927	0.0451
	- SAID	0.6156	0.3732	0.0973	0.0382	0.5135	0.4124	0.0694	0.0433	0.5127	0.3360	0.1124	0.0664
	- TSLRec	0.6069	0.3680	0.0969	0.0388	0.5078	0.4143	0.0418	0.0182	0.4936	0.3178	0.1013	0.0589
	- LLMEmb	**0.6294***	**0.3881***	**0.1876***	**0.1094***	**0.5244***	**0.4238***	**0.1485***	**0.0764***	**0.5247***	**0.3485***	**0.2430***	**0.1224***
SASRec	- None	0.5940	0.3597	0.1142	0.0495	0.4956	0.4429	0.0454	0.0235	0.4388	0.3030	0.0870	0.0649
	- MELT	0.6257	0.3791	0.1015	0.0371	0.4875	0.4150	0.0368	0.0144	0.4334	0.2775	0.0460	0.0172
	- LLM2X	0.6415	0.3997	0.1760	0.0789	0.5210	0.4486	0.0768	0.0473	0.5043	0.3319	0.1608	0.0940
	- SAID	0.6277	0.3841	0.1548	0.0669	0.5316	0.4619	0.0901	0.0540	0.5097	0.3343	0.1549	0.0906
	- TSLRec	0.6152	0.3795	0.1383	0.0620	0.5125	0.4594	0.0652	0.0382	0.4977	0.3366	0.1211	0.0789
	- LLMEmb	**0.6647***	**0.4113***	**0.2951***	**0.1456***	**0.5521***	**0.4730***	**0.1513***	**0.0826***	**0.5277***	**0.3460***	**0.4194***	**0.2595***

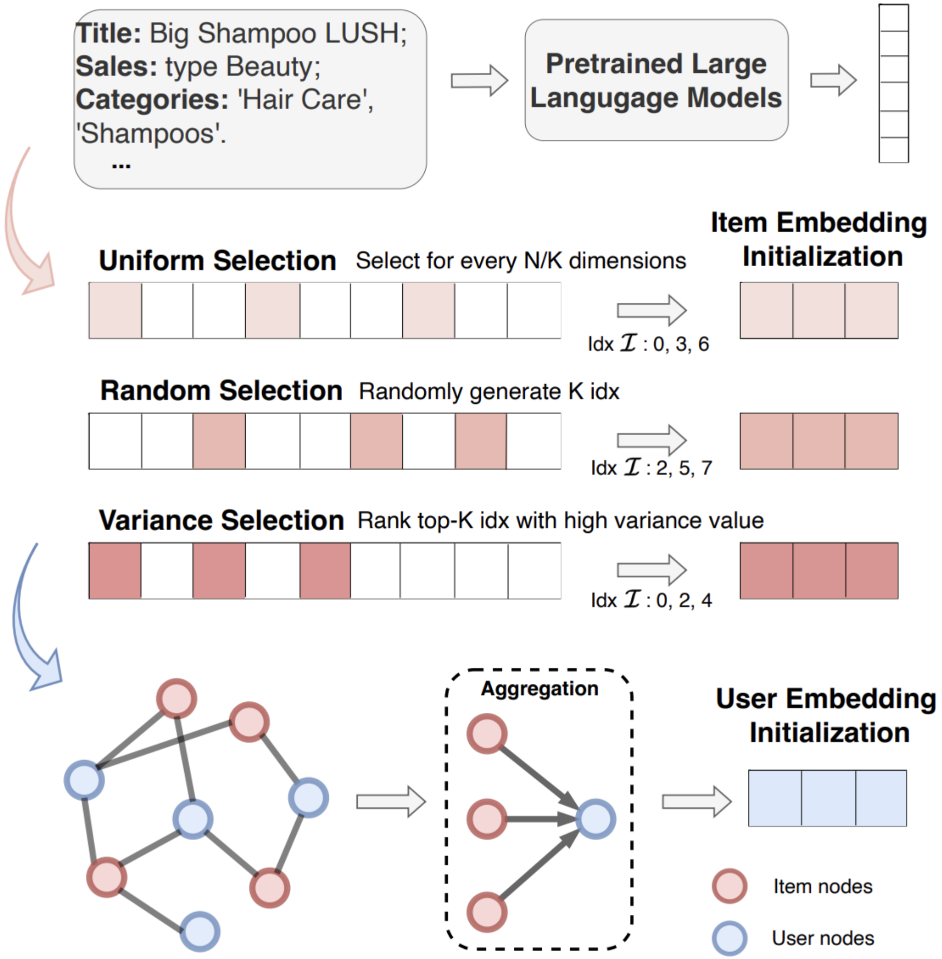
- **Practical Values:** LLMEmb's SCFT+RAT framework bridges semantic understanding with collaborative filtering, enables efficient precomputed embeddings, and enhances sequential recommenders without runtime overhead.

Embedding - LLMInit

- Background
  - Collaborative filtering models predict user preferences through interaction patterns but face cold-start and embedding collapse challenges.
- Motivation
  - CF models capture interaction patterns but lack semantic understanding.
  - LLMs provide rich semantics but struggle with item-level distinctions.
- Key Stages
  - Selective Sampling
  - Efficient Initialization

Zhang, Weizhi, et al. "Llminit: A free lunch from large language models for selective initialization of recommendation." arXiv preprint arXiv:2503.01814 (2025).

Embedding - LLMInit



- **LLMInit** bridges the LLM-CF embedding gap using selective initialization with sampling strategies to transfer knowledge from large LLM embeddings to lightweight CF models.
- **Selective Sampling**
  - Extract K-dimensional embeddings from LLM representations using variance-based and random selection strategies.
- **Efficient Initialization**
  - Initialize CF models with LLM embeddings to inherit semantic knowledge while maintaining scalability.

Embedding - LLMInit

- **LLMInit** significantly outperforms baseline CF models across all datasets (Beauty, Toys-Games, Tools-Home, Office-Products).

Method	Beauty		Toys-Games		Tools-Home		Office-Products	
	Recall@10	NDCG@10	Recall@10	NDCG@10	Recall@10	NDCG@10	Recall@10	NDCG@10
LightGCN [7]	0.0910	0.0432	0.0775	0.0360	0.0574	0.0283	0.0745	0.0365
+LLMInit-Rand	0.0960 (+5.5%)	0.0467 (+8.1%)	0.0805 (+3.9%)	0.0387 (+7.50%)	0.0612 (+6.6%)	0.0313 (+10.6%)	0.0773 (+3.8%)	0.0387 (+6.0%)
+LLMInit-Uni	0.1006 (+10.6%)	0.0469 (+8.6%)	0.0806 (+4.0%)	0.0388 (+7.8%)	0.0633 (+10.3%)	**0.0319** (+12.7%)	0.0791 (+6.2%)	0.0395 (+8.2%)
+LLMInit-Var	**0.1019** (+12.0%)	**0.0485** (+12.3%)	**0.0808** (+4.3%)	**0.0389** (+8.1%)	**0.0633** (+10.3%)	0.0317 (+12.0%)	**0.0816** (+9.5%)	**0.0414** (+13.4%)
SGL [23]	0.1017	0.0474	0.0832	0.0380	0.0580	0.0284	0.0669	0.0297
+LLMInit-Rand	0.1069 (+5.1%)	0.0520 (+9.7%)	0.0885 (+6.4%)	0.0418 (+10.0%)	**0.0692** (+19.3%)	0.0337 (+18.7%)	**0.0810** (+21.1%)	**0.0426** (+43.4%)
+LLMInit-Uni	0.1101 (+8.3%)	0.0513 (+8.2%)	0.0920 (+10.6%)	0.0424 (+11.6%)	0.0676 (+16.6%)	0.0333 (+17.3%)	0.0773 (+15.6%)	0.0350 (+17.9%)
+LLMInit-Var	**0.1106** (+8.8%)	**0.0530** (+11.8%)	**0.0927** (+11.4%)	**0.0427** (+12.4%)	0.0686 (+18.3%)	**0.0339** (+19.4%)	0.0794 (+18.7%)	0.0421 (+41.8%)
SGCL [32]	0.1027	0.0499	0.0828	0.0382	0.0585	0.0294	0.0647	0.0298
+LLMInit-Rand	0.1094 (+6.5%)	0.0512 (+2.6%)	0.0929 (+12.2%)	0.0418 (+9.4%)	**0.0651** (+11.3%)	0.0326 (+10.9%)	0.0770 (+19.0%)	0.0365 (+22.5%)
+LLMInit-Uni	**0.1115** (+8.6%)	0.0513 (+2.8%)	0.0923 (+11.5%)	**0.0422** (+10.5%)	0.0650 (+11.1%)	0.0327 (+11.2%)	0.0742 (+14.7%)	0.0354 (+18.8%)
+LLMInit-Var	0.1104 (+7.5%)	**0.0522** (+4.6%)	**0.0941** (+13.6%)	0.0421 (+10.2%)	0.0646 (+10.4%)	**0.0327** (+11.2%)	**0.0776** (+19.9%)	**0.0366** (+22.8%)

- **Practical Values:** LLMInit provides a "free lunch" approach for enhancing CF models with LLM semantic knowledge while maintaining high efficiency and scalability for real-world deployment.

Embedding - LEARN

- Background
  - Real-world LLM deployment in recommendation is hindered by the Domain Gap and Catastrophic Forgetting.

- Motivation
  - Domain Gap: Mismatch between LLM's open-world knowledge and recommendations' collaborative knowledge.
  - Catastrophic Forgetting: Fine-tuning LLMs to their full extent causes them to forget pre-trained knowledge.

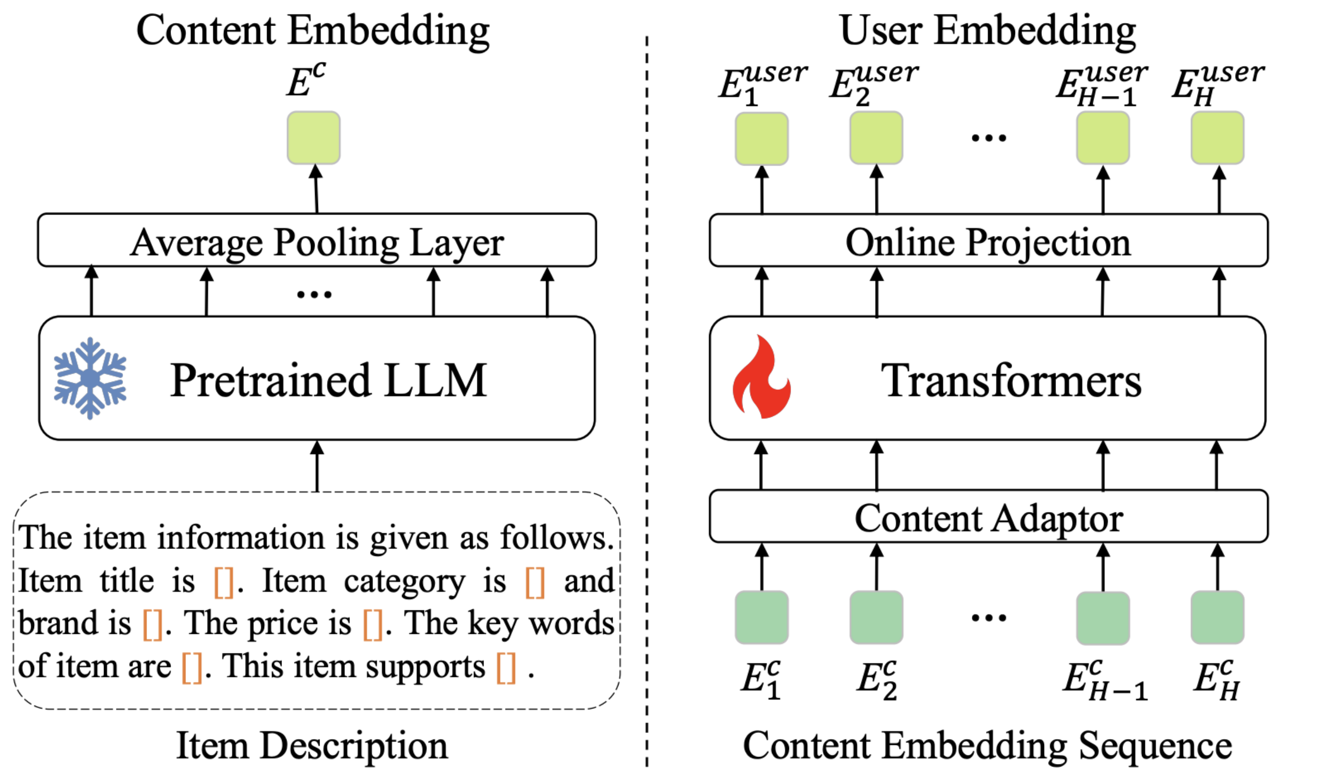
- Key Components
  - Content-Embedding Generation (CEG)
  - Preference Comprehension (PCH)

Jia, Jian, et al. "LEARN: Knowledge Adaptation from Large Language Model to Recommendation for Practical Industrial Application." Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 39. No. 11. 2025.

Embedding - LEARN

LEARN is a twin-tower framework that adapts LLM knowledge by freezing the LLM and using self-supervised training.

- **Content-Embedding Generation**
  - Uses a frozen LLM as an item encoder to convert text into rich content embeddings. Freezing is key to preventing catastrophic forgetting.
- **Preference Comprehension**
  - A lightweight Transformer learns user preferences from item embeddings via contrastive learning.

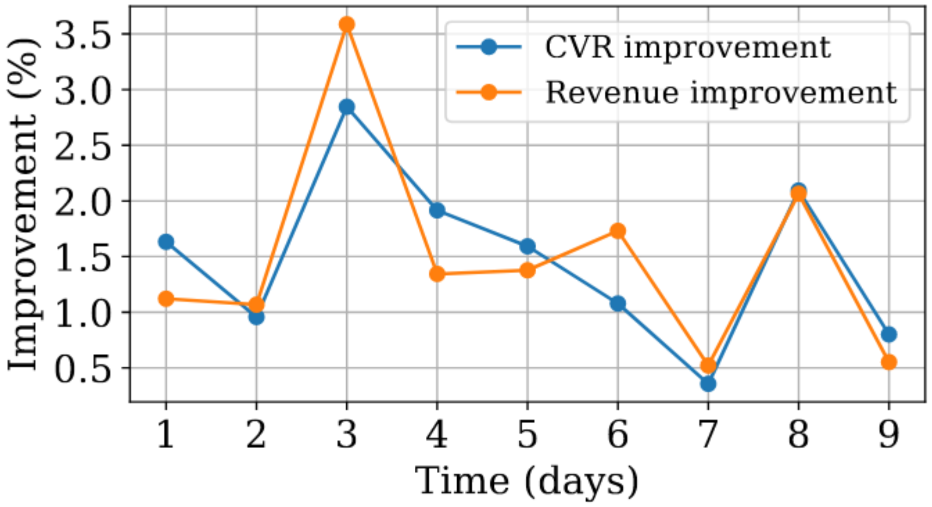


(a) Content Extraction Module (b) Preference Alignment Module

Embedding - LEARN

- **LEARN** achieves SOTA performance on large-scale offline datasets and delivers significant business impact in real-world online A/B tests on Kuaishou.

Method	H@10	H@50	H@200
SASRec [20]	0.0306	0.0754	0.1431
HSTU [43]	**0.0416** (+35.95%)	0.0957 (+26.92%)	0.1735 (+21.24%)
LEARN (Ours)	0.0407 (+33.01%)	**0.0979**(+29.84%)	**0.1874** (+30.96%)
–	N@10	N@50	N@200
SASRec [20]	0.0164	0.0260	0.0362
HSTU [43]	**0.0227** (+38.41%)	0.0344 (+32.31%)	0.0461 (+27.35%)
LEARN (Ours)	0.0224 (+36.59%)	**0.0371** (+42.69%)	**0.0483** (+33.43%)



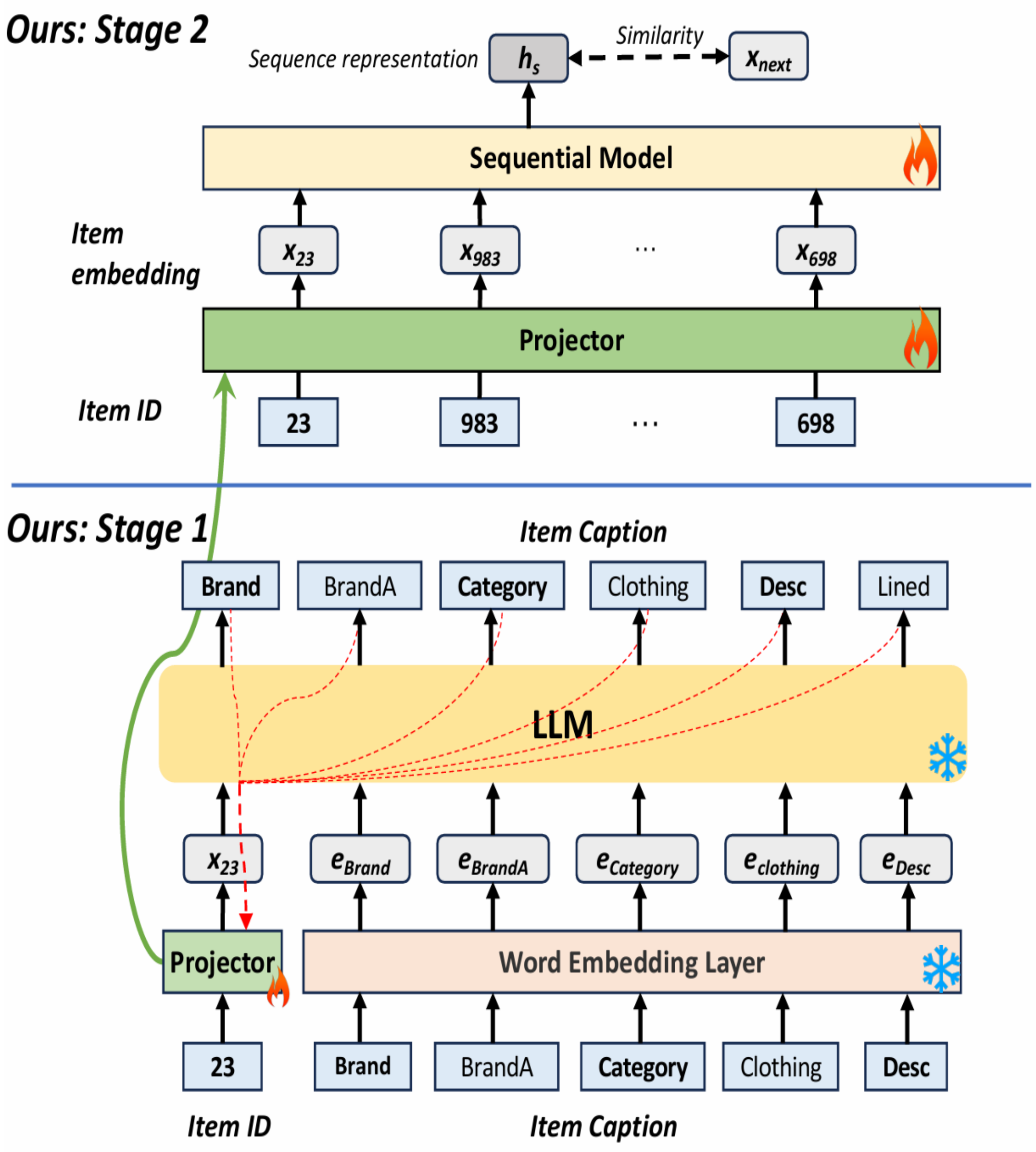
- **Practical Values:** LEARN is the first LLM-enhanced solution successfully deployed and monetized in a large-scale industrial recommender system, setting a new standard for practical application.

Embedding - SAID

- Background
  - LLMs enhance sequential recommendation models with their generalization ability and knowledge base.
- Motivation
  - LLM embeddings may lose fine-grained item information
  - Long token sequences cause efficiency issues
- Key Stages
  - Semantically Aligned Embedding Learning
  - Model-agnostic Sequential Recommender Training

Hu, Jun, et al. "Enhancing Sequential Rec ommendation via LLM-based Semantic Embedding Learning." Proceedings of the ACM Web Conference 2024 (WWW'24 Companion), 2024.

Embedding - SAID



SAID learns item embeddings aligned with LLM text descriptions, usable with lightweight sequential models.

- **Semantically Aligned Embedding Learning**
  - SAID learns to generate an embedding for each item by leveraging the projector module and an on-the-shelf LLM.
- **Model-agnostic Sequential Recommender Training**
  - The embeddings acquired during the first stage are utilized as initial features of the items, which are then inputted into a downstream model for sequential recommendation.

Embedding - SAID

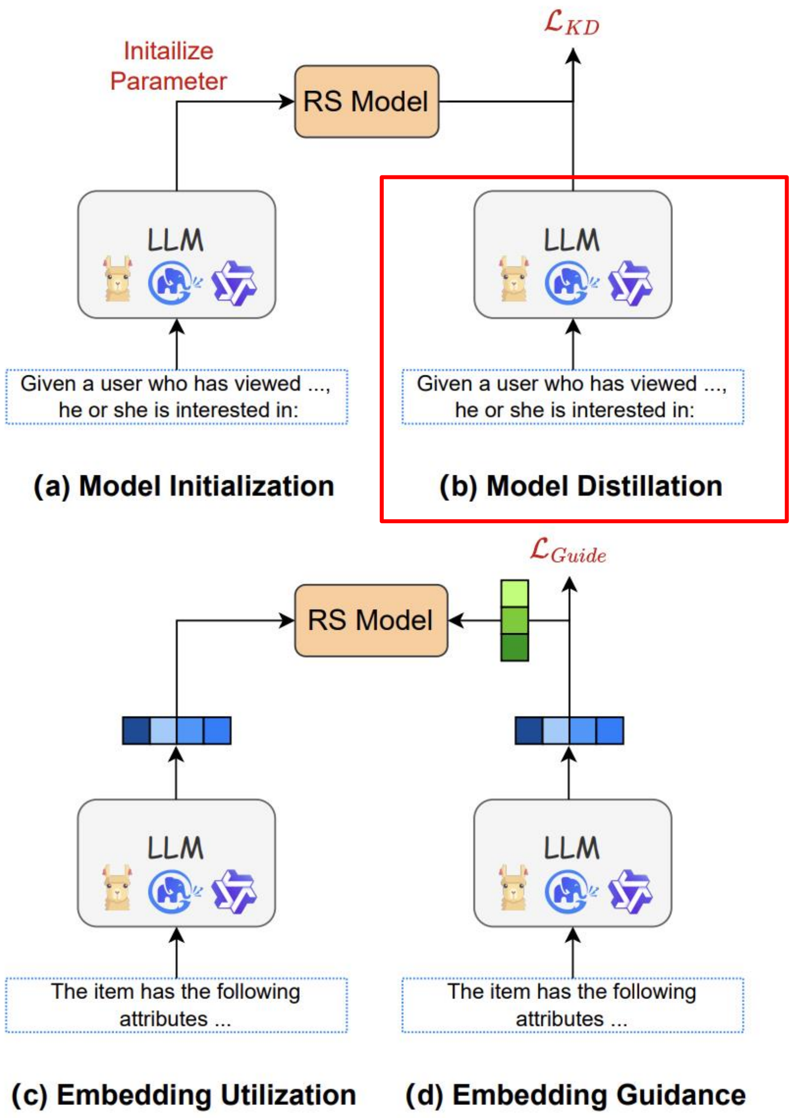
- Experiments conducted on public datasets above and Alipay's online advertising deployment justify the efficiency and efficacy of SAID.

Dataset	Metrics	GRU4Rec				SASRec			
		Random	LH	SAID	Improv.	Random	LH	SAID	Improv.
Scientific	NDCG@10	0.0896	0.0958	**0.1021**	6.6%	0.0869	0.0992	**0.1050**	5.8%
	Recall@10	0.1037	0.1317	**0.1321**	0.3%	0.1088	0.1313	**0.1353**	3.0%
	MRR	0.0877	0.0913	**0.0977**	7.0%	0.0825	0.0945	**0.1005**	6.3%
Instruments	NDCG@10	0.0760	0.0774	**0.0862**	11.4%	0.0768	0.0837	**0.0928**	10.9%
	Recall@10	0.0910	0.1085	**0.1122**	3.4%	0.0920	0.1158	**0.1211**	4.6%
	MRR	0.0741	0.0741	**0.0832**	12.3%	0.0750	0.0802	**0.0896**	11.7%
Arts	NDCG@10	0.0878	0.1057	**0.1180**	11.6%	0.1017	0.0995	**0.1090**	7.2%
	Recall@10	0.0978	**0.1441**	0.1413	-	0.1302	0.1433	**0.1487**	3.8%
	MRR	0.0866	0.0993	**0.1145**	15.3%	0.0951	0.0910	**0.1017**	6.9%
Office	NDCG@10	0.1127	0.1136	**0.1202**	5.8%	0.1084	0.1134	**0.1208**	6.5%
	Recall@10	0.1285	0.1374	**0.1440**	4.8%	0.1265	0.1377	**0.1450**	5.3%
	MRR	0.1097	0.1098	**0.1160**	5.6%	0.1047	0.1092	**0.1165**	6.7%
Games	NDCG@10	0.0641	0.0748	**0.0785**	4.9%	0.0673	0.0752	**0.0812**	8.0%
	Recall@10	0.0877	**0.1221**	0.1173	-	0.0936	**0.1221**	0.1204	-
	MRR	0.0617	0.0694	**0.0741**	6.8%	0.0647	0.0696	**0.0770**	10.6%
Pet	NDCG@10	0.0854	0.0925	**0.0960**	3.8%	0.0878	0.0881	**0.0951**	7.9%
	Recall@10	0.0945	0.1120	**0.1152**	2.9%	0.0978	0.1062	**0.1129**	6.3%
	MRR	0.0843	0.0902	**0.0934**	3.5%	0.0866	0.0859	**0.0929**	7.3%

- **Practical Values:** SAID significantly enhances inference efficiency and recommendation accuracy, making large language models practically applicable in industrial recommendation scenarios.

Model Initialization

- This subcategory refers to compressing LLM knowledge into smaller RS models through distillation techniques.
- This approach transfers LLM capabilities to lightweight models while maintaining recommendation quality for efficient deployment.
- Categories
  - Feature-based
  - Response-based



Feature-based - LEADER

- Background
  - Medication recommendation provides prescription suggestions to doctors. LLMs have two advantages: medical semantic understanding and cold-start capabilities.

- Motivation
  - Complex medication names cause LLMs to output recommendations not in drug databases, leading to failures.
  - Resource-intensive LLM inference challenges resource-constrained medical institutions in using LLMs for drug recommendations.

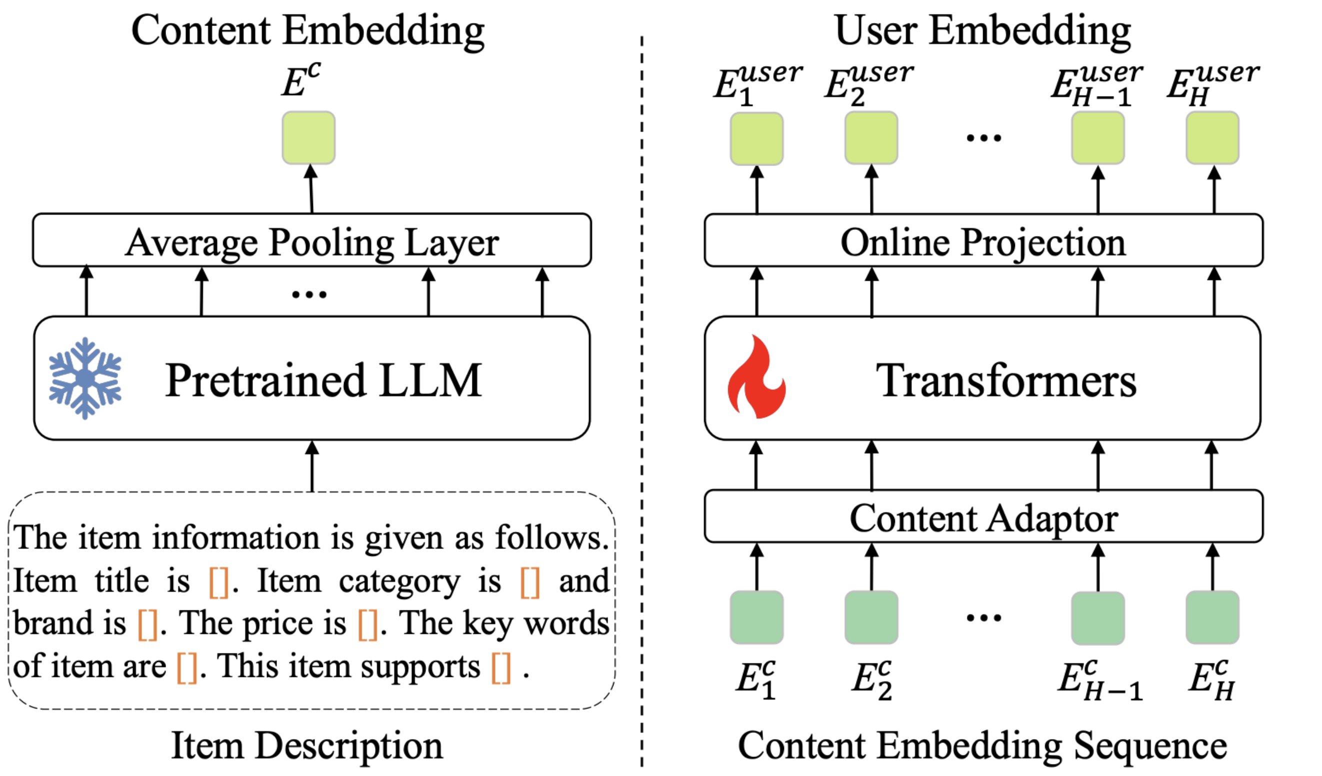
- Key Stages
  - Improve for LLMs
  - Distilling LLMs

Liu, Qidong, et al. "Large language model distilling medication recommendation model." arXiv preprint arXiv:2402.02803 (2024).

Feature-based - LEADER

LEADER improved the recommendation performance of LLMs by enhancing the output layer.

- **Improve for LLMs**
  - Replace the original text generation head layer of the large model with a new classification layer, and fine-tune the model using BCE loss.
- **Distilling LLMs**
  - Design the distillation process where the hidden states from the final layer of the LLMs are used as guidance to distill semantic understanding into the designed smaller model.



(a) Content Extraction Module (b) Preference Alignment Module

Feature-based - LEADER

- **LEADER** significantly outperforms baseline. The distilled smaller model demonstrates excellent performance in both overall and cold-start scenarios.

Model	Overall			Multi-visit			Single-visit		
	PRAUC	Jaccard	F1	PRAUC	Jaccard	F1	PRAUC	Jaccard	F1
RETAIN	0.7513 ± 0.0025	0.4943 ± 0.0023	0.6516 ± 0.0022	0.7580 ± 0.0020	0.5106 ± 0.0023	0.6674 ± 0.0022	0.7337 ± 0.0067	0.4811 ± 0.0053	0.6403 ± 0.0049
G-Bert	-	-	-	0.6904 ± 0.0017	0.4578 ± 0.0019	0.6186 ± 0.0018	-	-	-
GAMENet	0.7605 ± 0.0011	0.5024 ± 0.0010	0.6595 ± 0.0008	0.7638 ± 0.0023	0.5070 ± 0.0028	0.6635 ± 0.0025	0.7451 ± 0.0053	0.4840 ± 0.0038	0.6442 ± 0.0036
SafeDrug	0.7582 ± 0.0020	0.5054 ± 0.0024	0.6621 ± 0.0021	0.7623 ± 0.0029	0.5095 ± 0.0027	0.6658 ± 0.0024	0.7416 ± 0.0044	0.4900 ± 0.0043	0.6481 ± 0.0042
MICRON	-	-	-	0.7651 ± 0.0027	0.5110 ± 0.0025	0.6741 ± 0.0023	-	-	-
COGNet	-	-	-	0.7771 ± 0.0019	0.5275 ± 0.0021	0.6805 ± 0.0019	-	-	-
REFINE	-	-	-	0.7791 ± 0.0017	0.5235 ± 0.0018	0.6794 ± 0.0017	-	-	-
LEADER(T)	**0.7816 ± 0.0015***	**0.5391 ± 0.0015***	**0.6921 ± 0.0014***	**0.7854 ± 0.0015***	**0.5450 ± 0.0021***	**0.6971 ± 0.0018***	0.7590 ± 0.0046*	**0.5090 ± 0.0044***	**0.6668 ± 0.0041***
LEADER(S)	0.7795 ± 0.0025*	0.5175 ± 0.0022*	0.6737 ± 0.0019*	0.7830 ± 0.0019	0.5208 ± 0.0020	0.6768 ± 0.0017	**0.7631 ± 0.0056***	0.5038 ± 0.0062*	0.6614 ± 0.0057*

- **Practical Values:** LEADER can be used for recommending prescriptions for patients who first visit the hospital (cold-start scenarios).

Feature-based - SLMRec

- Background
  - Large Language Models (LLMs) have demonstrated exceptional performance in Sequential Recommendation (SR).

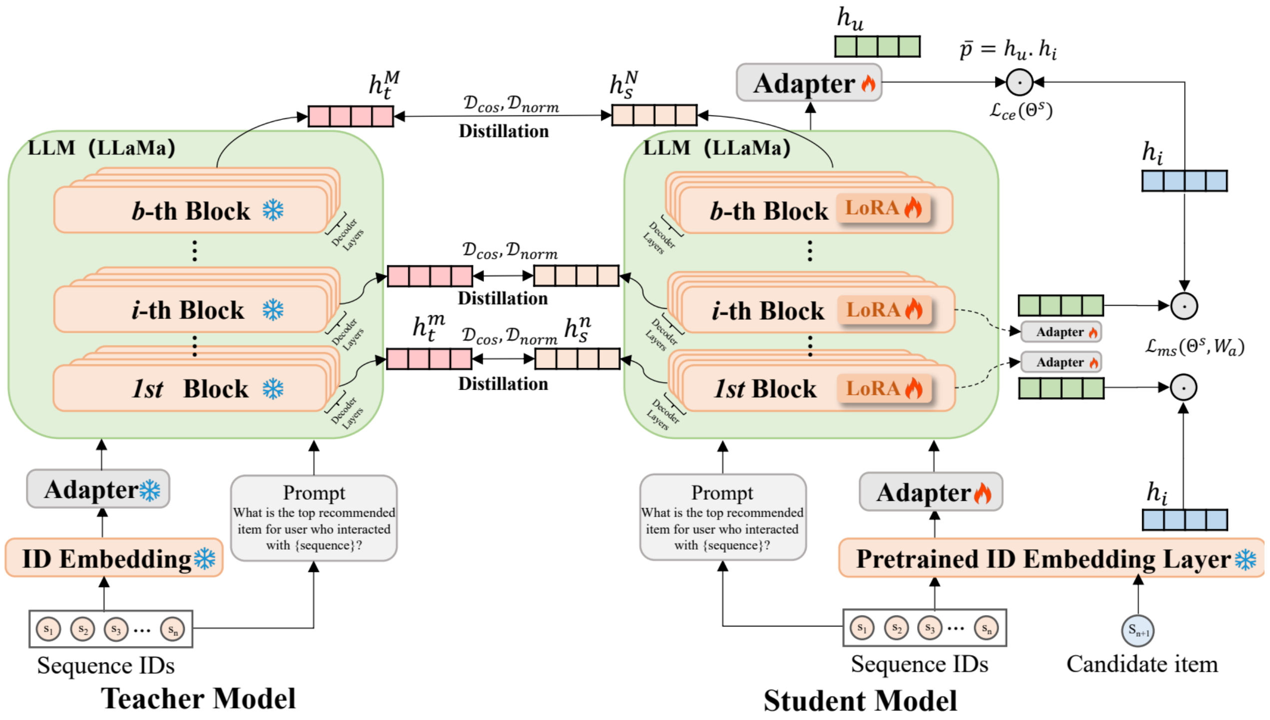
- Motivation
  - Intermediate layers of LLMs are largely redundant for SR tasks.
  - The massive computational and deployment costs make LLMs impractical for industrial applications.

- Key Stages
  - Layer-wise Feature Distillation
  - Dual-Loss Feature Alignment
  - Auxiliary Supervision

Wenfang Lin, et al. "SLMRec: Empowering Small Language Models for Sequential Recommendation." arXiv preprint arXiv:2405.17890 (2024).

Feature-based - SLMRec

- **Layer-wise Feature Distillation**
  - Aligns intermediate hidden states to transfer rich feature-level knowledge, not just mimic final outputs.
- **Dual-Loss Feature Alignment**
  - Employs a dual-loss mechanism to match features in both direction and magnitude.
- **Auxiliary Supervision**
  - Applies early, task-specific supervision to shallow layers, enabling efficient learning in a compact model.



Feature-based - SLMRec

- **SLMRec** achieves state-of-the-art performance while reducing computational costs.

Model	Cloth HR@1	HR@5	NDCG@5	MRR	Movie HR@1	HR@5	NDCG@5	MRR	Rank
Caser	9.66	15.18	12.66	13.03	4.27	14.96	9.57	10.36	13.50
GRU4Rec	13.79	15.46	14.64	15.15	10.56	19.47	15.11	15.46	9.25
BERT4Rec	13.60	14.66	14.14	14.59	9.68	14.91	12.40	12.74	11.63
SASRec	13.08	16.94	15.01	15.76	5.57	16.80	11.17	12.08	11.63
HGN	15.96	18.70	17.30	18.27	7.54	19.20	13.42	14.73	6.50
LightSANs	14.12	20.32	17.30	16.86	6.08	17.54	11.81	12.82	8.00
S^3-Rec	14.10	18.67	16.10	16.95	7.75	20.39	15.69	14.34	7.50
DuoRec	13.06	18.29	15.79	15.42	10.07	20.37	17.96	16.61	7.88
MAERec	13.29	18.35	15.68	16.13	8.89	20.24	16.03	15.28	8.38
Open-P5	14.13	17.68	17.02	-	12.66	21.98	17.13	-	5.67
E4SRec	16.71	19.45	18.09	18.77	14.74	23.79	19.45	19.74	1.75
E4SRec_8	15.30	18.54	16.91	17.60	13.32	22.49	17.99	18.46	4.00
E4SRec_4	14.58	18.05	16.32	17.01	11.80	21.54	16.73	17.20	5.75
$\text{SLMRec}_{4\leftarrow 8}$	**16.69**	**19.47**	**18.07**	**18.74**	**15.29**	**24.25**	**19.90**	**20.36**	**1.50**

- **Practical Values:** SLMRec offers the industry a path to harness the power of LLMs in recommendation systems in an economical, efficient, and high-performance manner.

Response-based - DLLM2Rec

- Background
  - LLM-based recommenders face serious inference inefficiency issues, posing substantial challenges to their practical applications.

- Motivation
  - Teacher Knowledge Reliability: The teacher's knowledge isn't always reliable.
  - Model Capacity Gap: The student struggles to assimilate teacher knowledge.
  - Semantic Space Divergence: Different "languages" make direct alignment counterproductive.

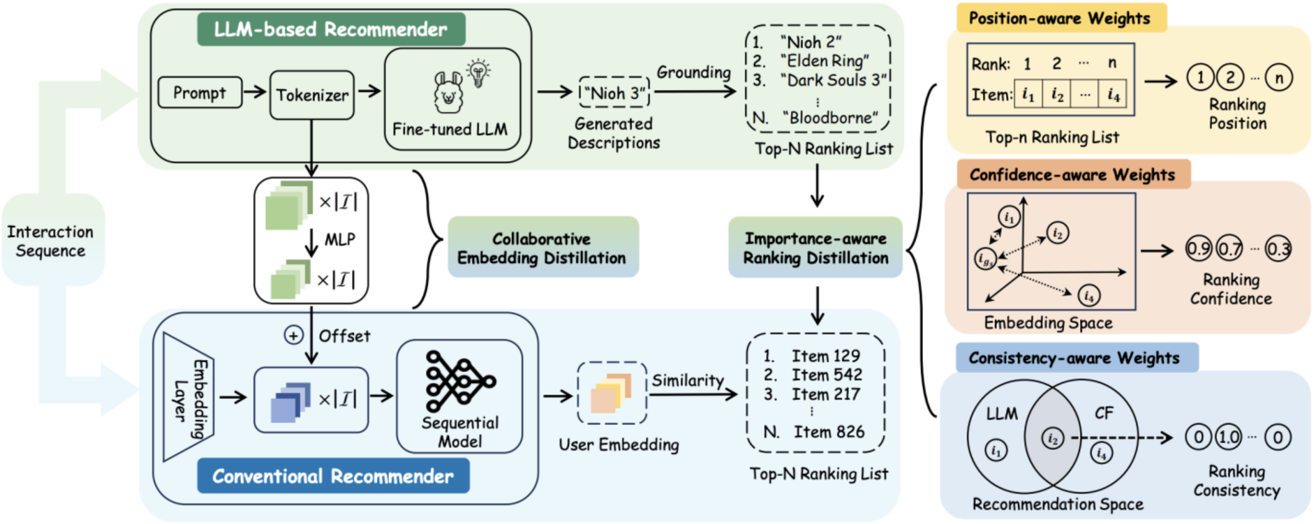
- Key Stages
  - Importance-aware Ranking Distillation
  - Collaborative Embedding Distillation

Cui, Yu, et al. "Distillation matters: empowering sequential recommenders to match the performance of large language models." Proceedings of the 18th ACM Conference on Recommender Systems. 2024.

Response-based - DLLM2Rec

DLLM2Rec is designed to effectively distill knowledge from LLM-based recommenders to conventional recommenders.

- **Importance-aware Ranking Distillation**
  - DLLM2Rec introduces importance weights to selectively learn reliable knowledge rather than blindly mimicking teacher rankings.
- **Collaborative Embedding Distillation**
  - DLLM2Rec performs collaborative fusion to bridge semantic space divergence between teacher and student.



Response-based - DLLM2Rec

- **DLLM2Rec's** lightweight student model not only achieves a million-fold speedup in inference but also outperforms the massive LLM teacher on several datasets.

Backbone	Model	Games		MovieLens		Toys	
		HR@20	NDCG@20	HR@20	NDCG@20	HR@20	NDCG@20
Teacher	BIGRec	0.0532	0.0341	0.0541	0.0370	0.0420	0.0207
GRU4Rec	+None	0.0305	0.0150	0.0608	0.0236	0.0172	0.0081
	+Hint	0.0284	0.0120	0.0646	0.0240	0.0128	0.0058
	+HTD	0.0299	0.0128	0.0578	0.0229	0.0155	0.0062
	+RD	0.0398	0.0177	0.0667	0.0254	0.0157	0.0076
	+CD	0.0306	0.0149	<u>0.0699</u>	0.0256	0.0126	0.0052
	+RRD	0.0359	0.0163	0.0657	0.0243	0.0215	0.0097
	+DCD	<u>0.0427</u>	<u>0.0190</u>	0.0666	<u>0.0263</u>	<u>0.0262</u>	<u>0.0114</u>
	+UnKD	0.0370	0.0170	0.0607	0.0226	0.0235	0.0114
	KAR	0.0307	0.0149	0.0603	0.0229	0.0184	0.0079
	LLM-CF	0.0393	0.0174	0.0677	0.0246	0.0132	0.0058
	+DLLM2Rec	**0.0446**	**0.0205**	**0.0815**	**0.0308**	**0.0281**	**0.0118**
	Gain.S	+46.17%	+36.94%	+34.05%	+30.43%	+63.88%	+42.18%
	Gain.B	+4.56%	+7.64%	+16.60%	+16.80%	+7.40%	+1.27%

Dataset	Model	HR@20	NDCG@20	Inference time
Games	BIGRec	0.0532	0.0341	2.3×10^4s
	DLLM2Rec	0.0751	0.0331	1.8s
	Gain	+37.41%	-2.99%	$+1.3\times10^6$%
MovieLens	BIGRec	0.0541	0.0370	1.8×10^4s
	DLLM2Rec	0.1063	0.0437	1.7s
	Gain	+96.49%	+18.18%	$+1.1\times10^6$%
Toys	BIGRec	0.0420	0.0207	1.1×10^4s
	DLLM2Rec	0.0463	0.0225	1.6s
	Gain	+10.24%	+8.70%	$+6.8\times10^5$%

- **Practical Values:** DLLM2Rec provides the industry with a path to integrate the powerful capabilities of large language models into existing recommendation systems in an economical, efficient, and feasible manner.

Agenda

1 Introduction

Zijian Zhang

2 Knowledge Enhancement

Pengyue Jia

3 Interaction Enhancement

Ziwei Liu

4.1 Model Enhancement 1

Maolin Wang

4.2 Model Enhancement 2

Yuhao Wang

5 Conclusion

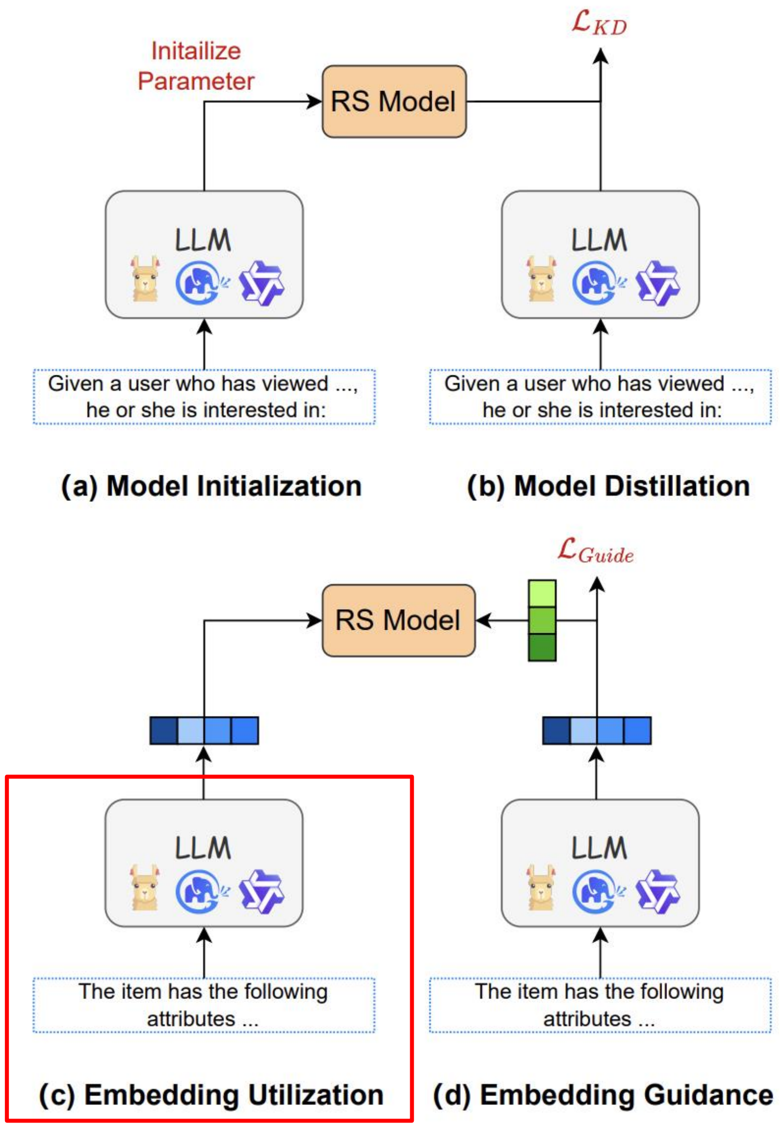
Yejing Wang

Embedding Utilization

- This subcategory refers to utilizing the embeddings derived from LLM to enhance traditional RS as a semantic supplement.

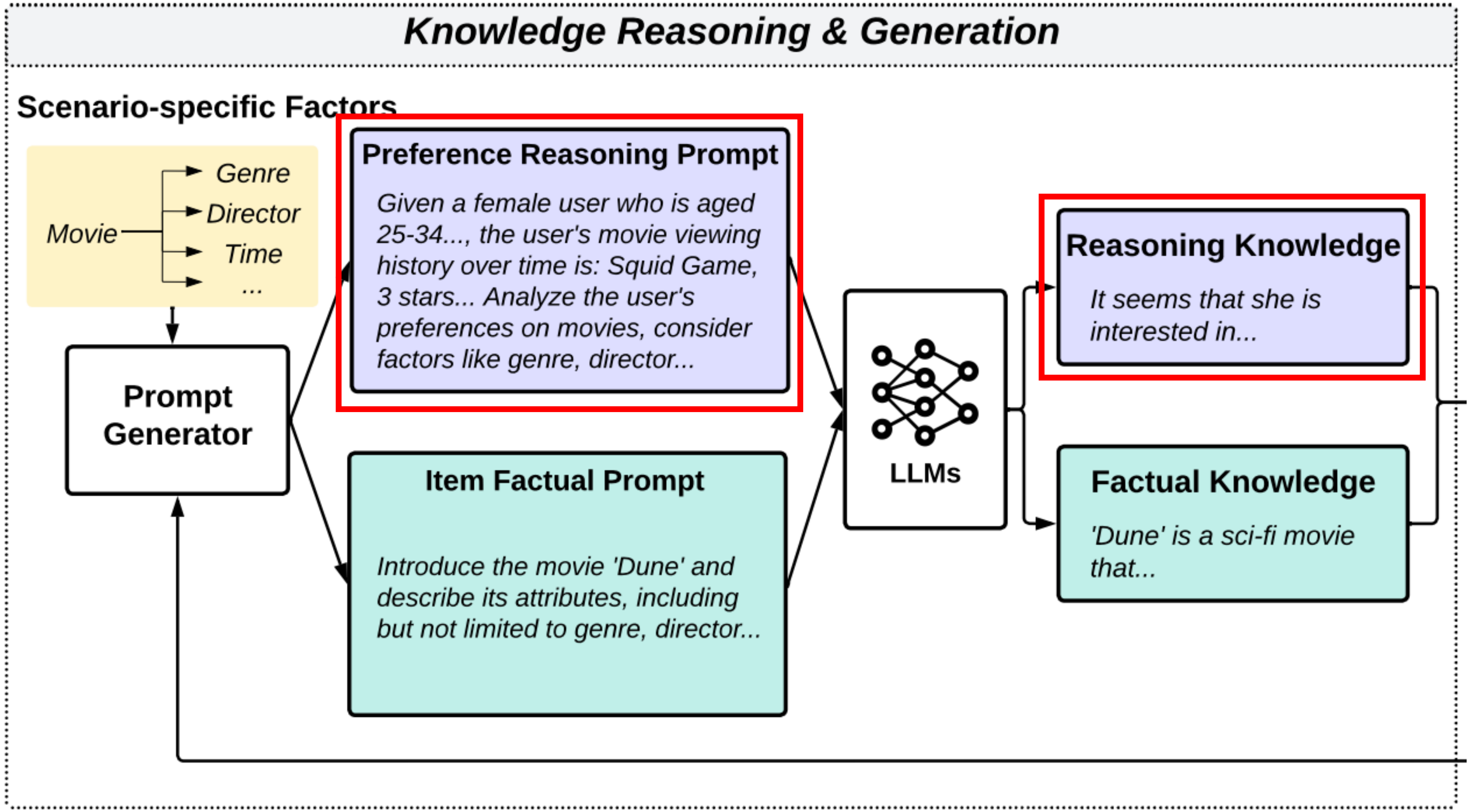
- This approach tackles the deficiency that textual outputs are often difficult and inefficient to be integrated into RS directly.

- Categories
  - User Only
  - Item Only
  - User & Item



KAR

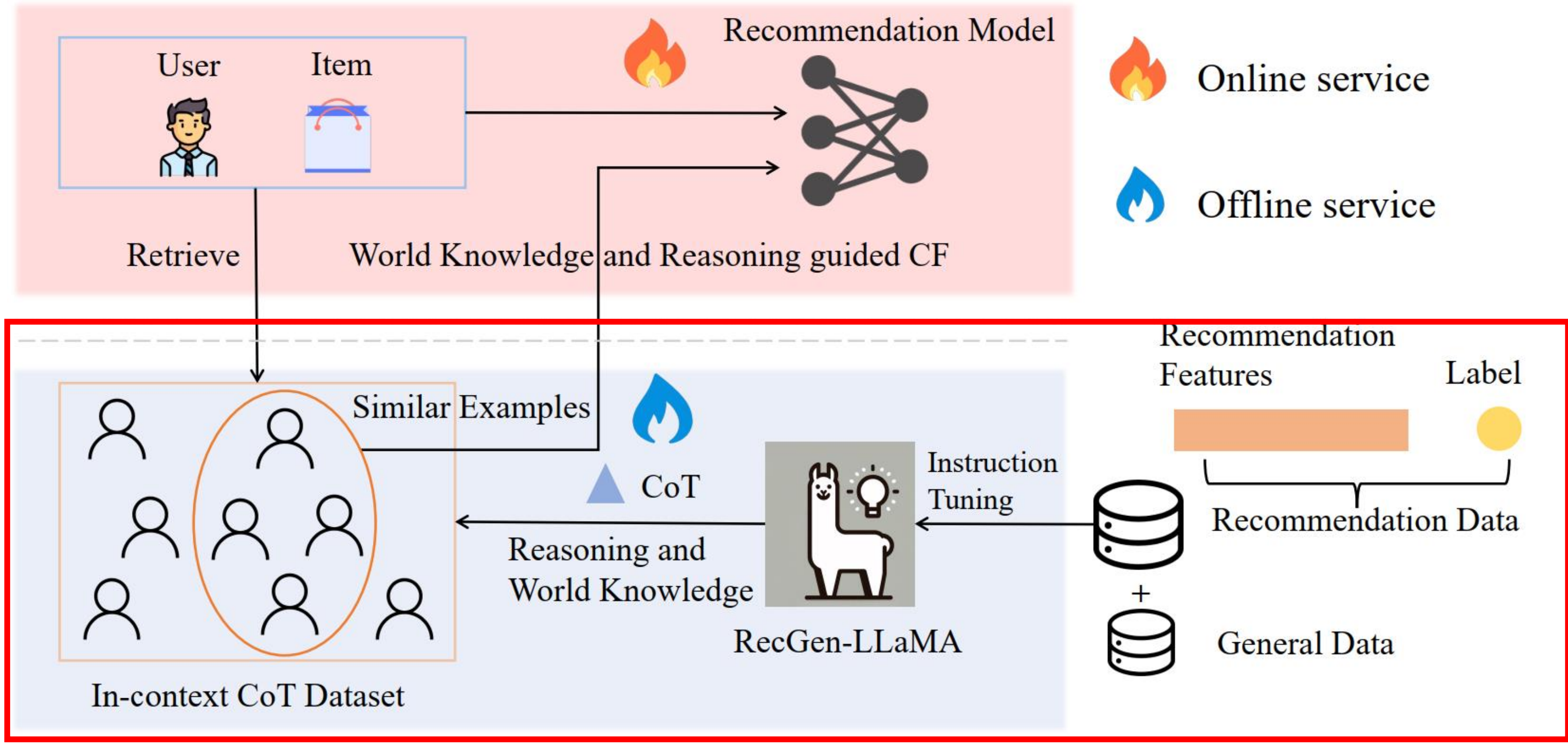
- Reasoning about user preferences given user profile and interactions



Xi, Yunjia, et al. "Towards Open-World Recommendation with Knowledge Augmentation from Large Language Models." RecSys 2024.

LLM-CF

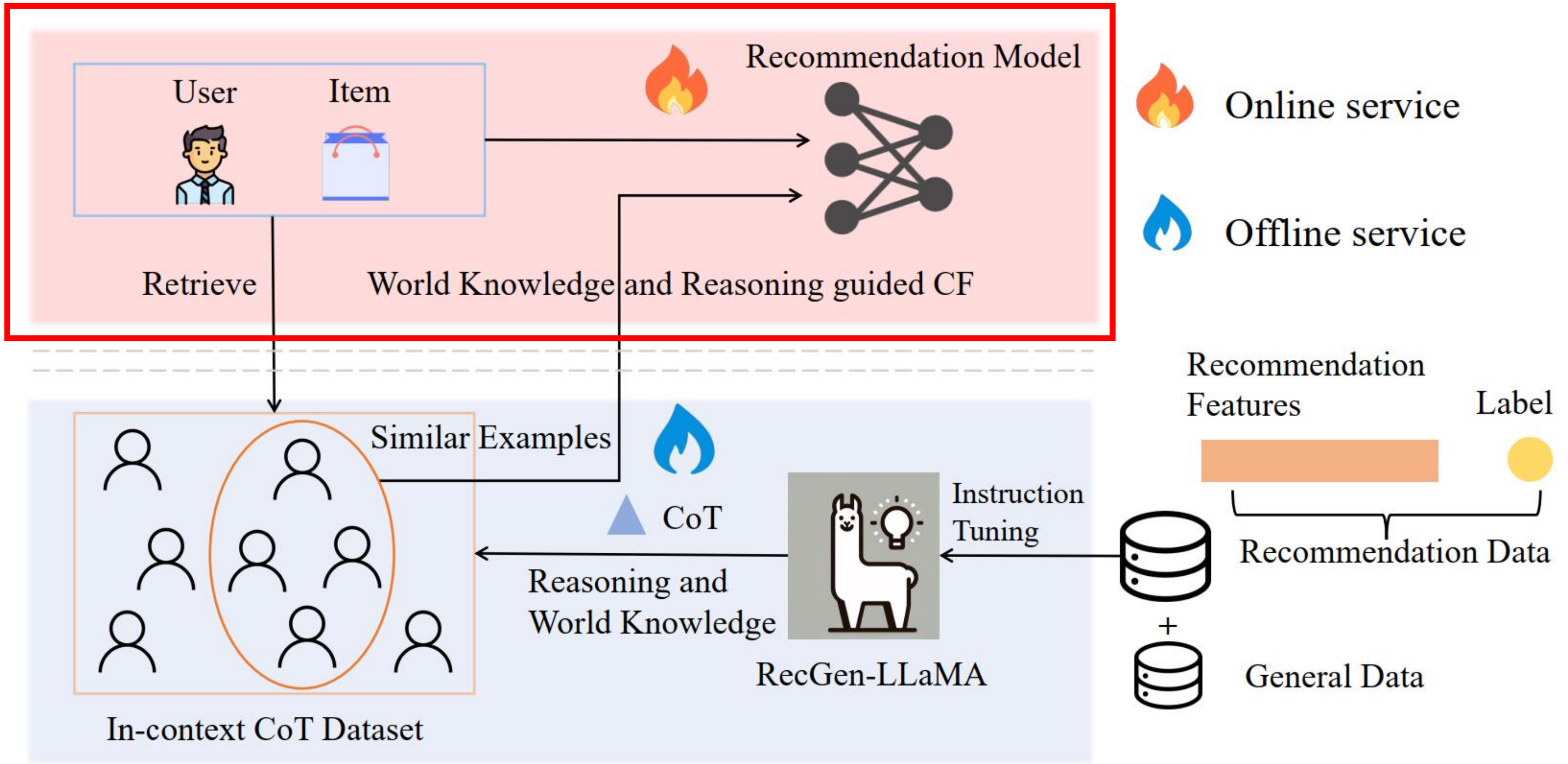
- **Offline service**
  - Chain of Thought (CoT) reasoning
  - In-context CoT dataset construction



Sun, Zhongxiang, et al. "Large Language Models Enhanced Collaborative Filtering." CIKM 2024.

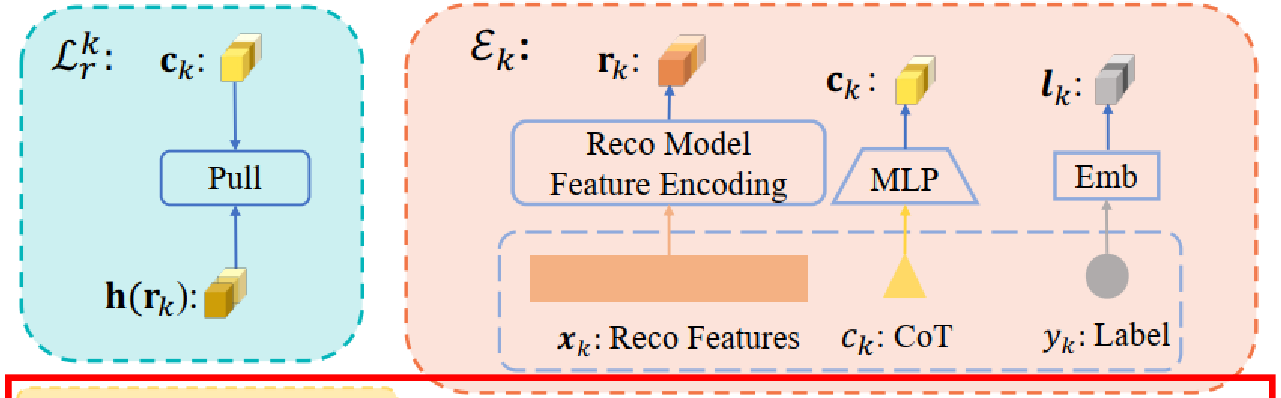
LLM-CF

- **Online service**
  - In-context CoT examples Retrieval
  - World knowledge & reasoning guided CF
  - Feature enhanced RS



LLM-CF

- **CoT Prompt**
  - Generating CoT of the user’s decision making on the target item

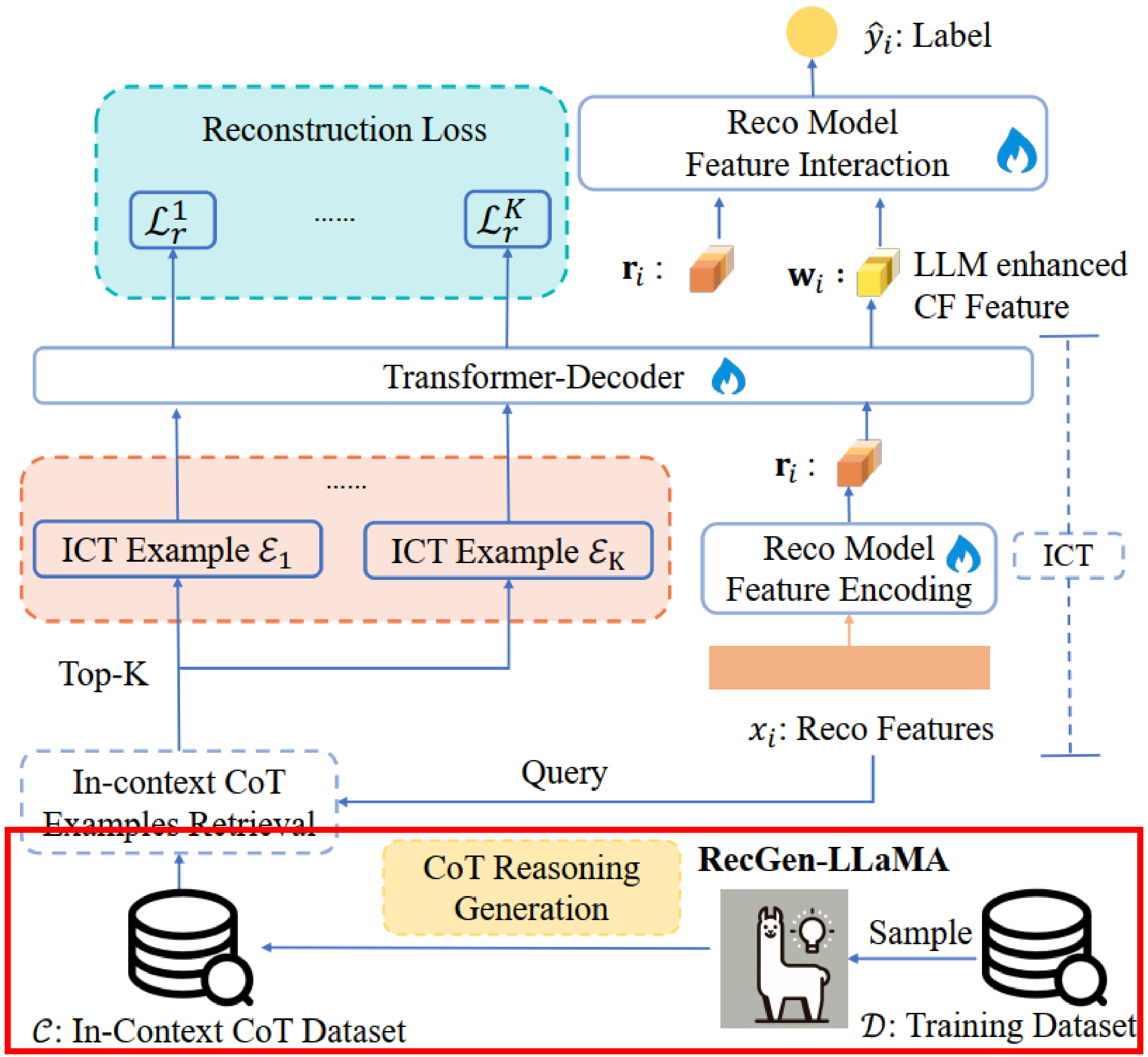


Chain of Thought Prompt

<<SYS>> As an AI model developed for analyzing consumer behavior, your task is to generate a chain of thought that considers the following points:
1. Utilize the user's interaction history and review comments to summarize their profiles.
2. Introduce the target new item and detail its features precisely. In addition, integrate information about items related to the current target new item ...
3. Contemplate the alignment between the user's profile and the features of the target item.
4. Reflect on the user's potential desire for diversity in their purchases.
Your output should be a clear and logical chain of thought... Ensure your analysis is impartial... Focus should be on understanding factors that influence the user‘s decision-making regarding the target item. <</SYS>>
Please generate a chain of thought based on the user‘s … considering how these might relate to their interest in the target new item.
{Recommendation Features}
User's decision-making: The user {Ground-Truth Label} the target new item.
Let's think step by step and develop the chain of thought for the above considerations. Commence with the chain of thought immediately:

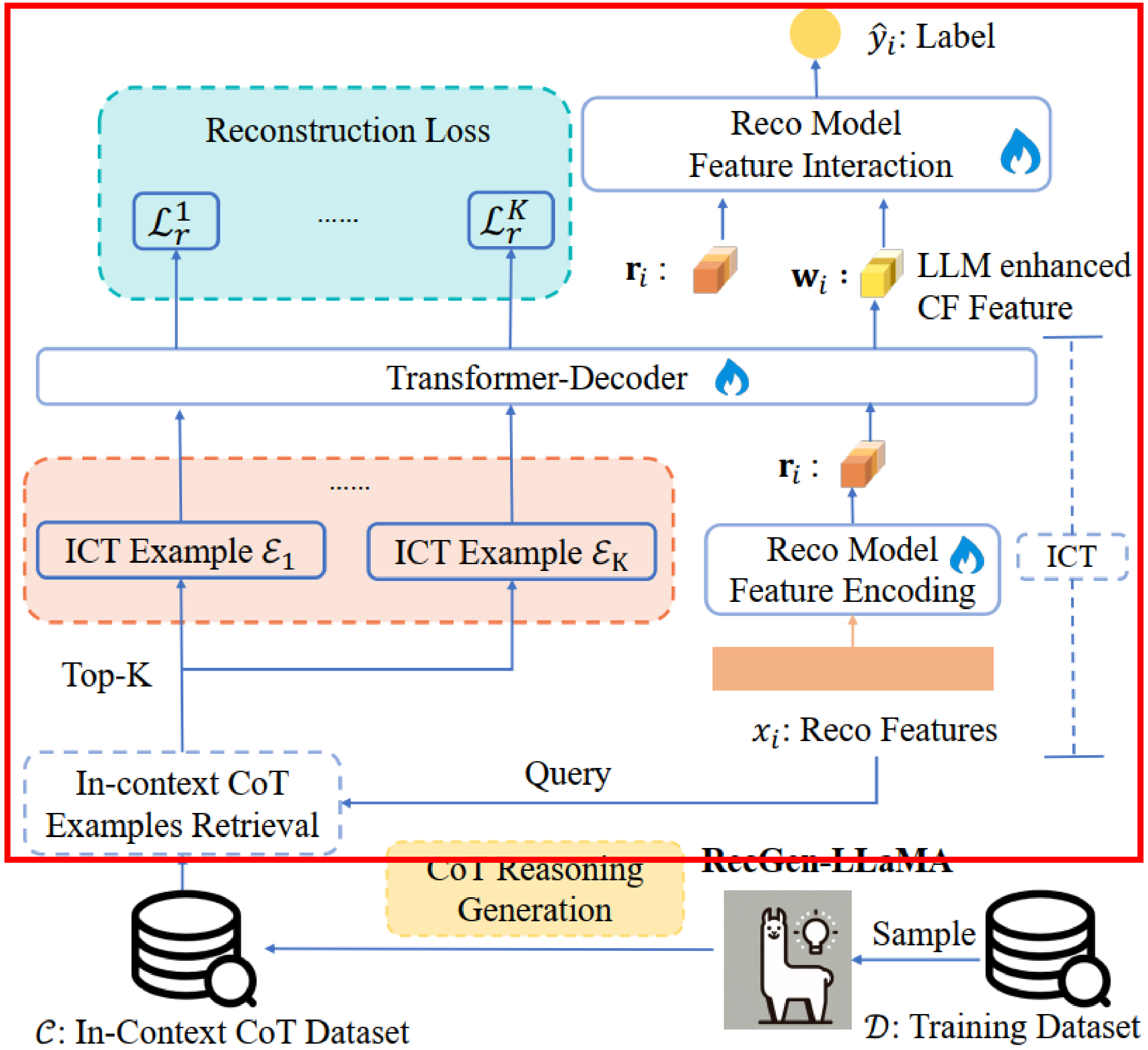
LLM-CF

- **Offline Service**
  - Instruction tuning on LLaMA2
  - RecGen-LLaMA generating CoT forming ICT dataset



LLM-CF

- **Online Service**
  - Embedding-based revrieval forming ICT examples
  - Learning world-knowledge and reasoning guided CF feature

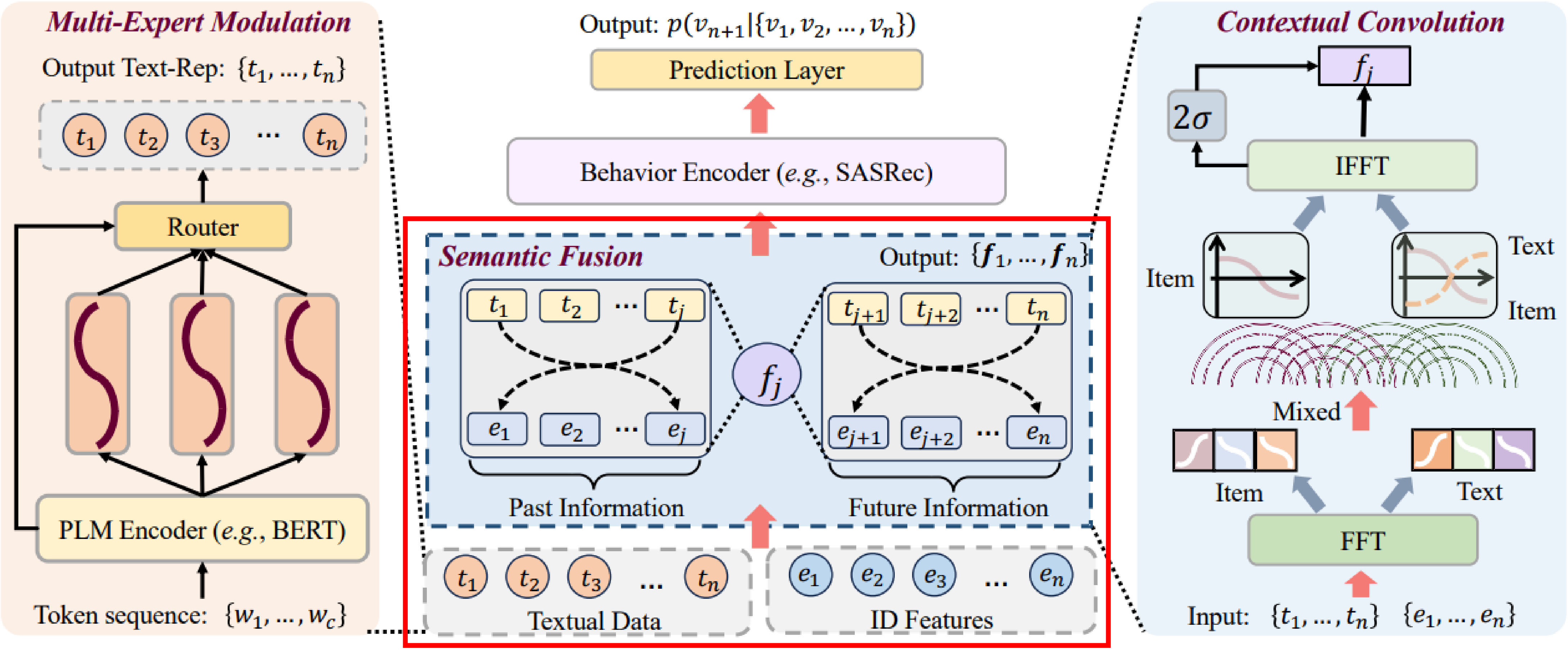


Overall performance

Backbone	Framework	Sports			Beauty			Toys		
		AUC↑	Logloss↓	RelaImpr↑	AUC↑	Logloss↓	RelaImpr↑	AUC↑	Logloss↓	RelaImpr↑
DeepFM	None	0.7990	0.5471	0.000%	0.7853	0.5545	0.000%	0.7681	0.5770	0.000%
	KD	0.8043	0.5404	1.773%	0.7959	0.5442	3.715%	0.7713	0.5716	1.194%
	KAR	0.7991	0.5469	0.033%	0.7870	0.5546	0.596%	0.7698	0.5718	0.634%
	LLM-CF	**0.8137***	**0.5306***	**4.916%**	**0.8044***	**0.5366***	**6.695%**	**0.7881***	**0.5581***	**7.460%**
xDeepFM	None	0.8158	0.5318	0.000%	0.8065	0.5359	0.000%	0.7836	0.5589	0.000%
	KD	0.8169	0.5298	0.348%	0.8104	0.5345	1.272%	0.7865	0.5553	1.023%
	KAR	0.8161	0.5279	0.094%	0.8101	0.5315	1.175%	0.7898	0.5529	2.186%
	LLM-CF	**0.8196***	**0.5248***	**1.203%**	**0.8113**	**0.5311**	**1.566%**	**0.7947***	**0.5473***	**3.914%**
AutoInt	None	0.8003	0.5444	0.000%	0.7949	0.5469	0.00%	0.7630	0.5770	0.000%
	KD	0.8012	0.5439	0.300%	0.7961	0.5444	0.407%	0.7635	0.5770	0.190%
	KAR	0.8039	**0.5390**	1.199%	0.7939	0.5476	-0.339%	0.7683	0.5741	2.015%
	LLM-CF	**0.8088***	0.5391	**2.831%**	**0.8090***	**0.5321***	**4.781%**	**0.7754***	**0.5685***	**4.714%**
DCNv1	None	0.8023	0.5442	0.000%	0.8146	0.5255	0.000%	0.7621	0.5831	0.000%
	KD	0.8040	0.5441	0.562%	0.8147	0.5286	0.031%	0.7652	0.5847	1.183%
	KAR	0.8024	0.5469	0.033%	0.8165	0.5229	0.604%	0.7651	0.5821	1.144%
	LLM-CF	**0.8092***	**0.5368***	**2.282%**	**0.8182***	**0.5216***	**1.144%**	**0.7702***	**0.5745***	**3.090%**
DCNv2	None	0.8110	0.5331	0.000%	0.8028	0.5378	0.00%	0.7774	0.5650	0.000%
	KD	0.8112	0.5320	0.064%	**0.8057**	**0.5343**	**0.958%**	**0.7827**	**0.5609**	**1.911%**
	KAR	0.8087	0.5363	-0.739%	0.8003	0.5404	-0.825%	0.7759	0.5662	-0.541%
	LLM-CF	**0.8131***	**0.5307***	**0.675%**	0.8033	0.5372	0.165%	0.7812	0.5619	1.370%
DIN	None	0.7986	0.5519	0.000%	0.7861	0.5613	0.000%	0.7586	0.5885	0.000%
	KD	0.8023	0.5422	1.239%	0.7934	0.5518	2.551%	0.7652	0.5847	2.552%
	KAR	0.7971	0.5525	-0.502%	0.7861	0.5604	0.000%	0.7620	0.5874	1.315%
	LLM-CF	**0.8089***	**0.5374***	**3.449%**	**0.7967***	**0.5492***	**3.705%**	**0.7783***	**0.5699***	**7.618%**

- LLM-CF achieves improvement on different CTR backbones.

TedRec

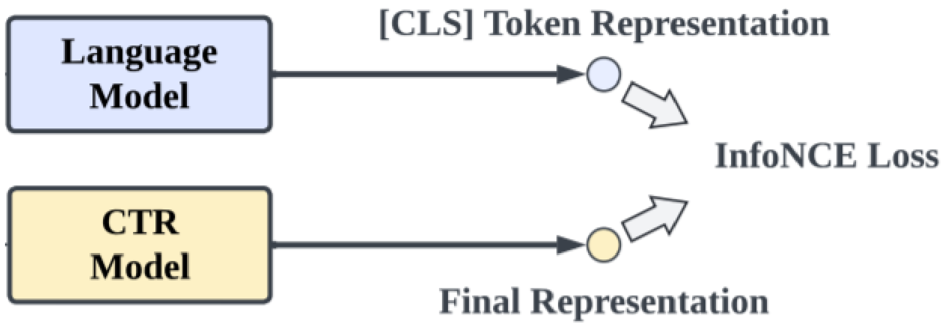


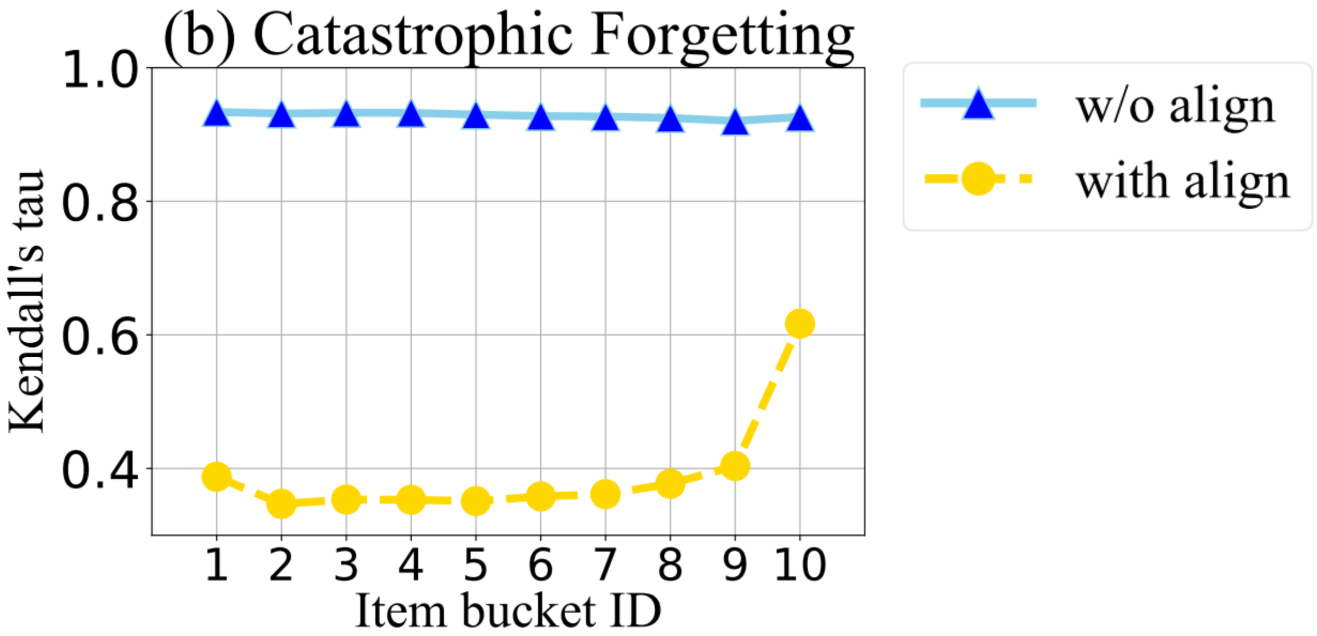
- TedRec applies Fourier Transform to conduct semantic fusion in the frequency domain.

Xu, Lanling, et al. "Sequence-level Semantic Representation Fusion for Recommender Systems." CIKM 2024.

PAD

- **Challenges for Alignment + SFT**
  - **Inability to Capture All Statistics of Data Distribution**: Contrastive learning with common cosine kernel is not optimal
  - **Catastrophic Forgetting in Alignment**: With only one set of collaborative embeddings, the alignment leads to catastrophic forgetting on the collaborative embeddings

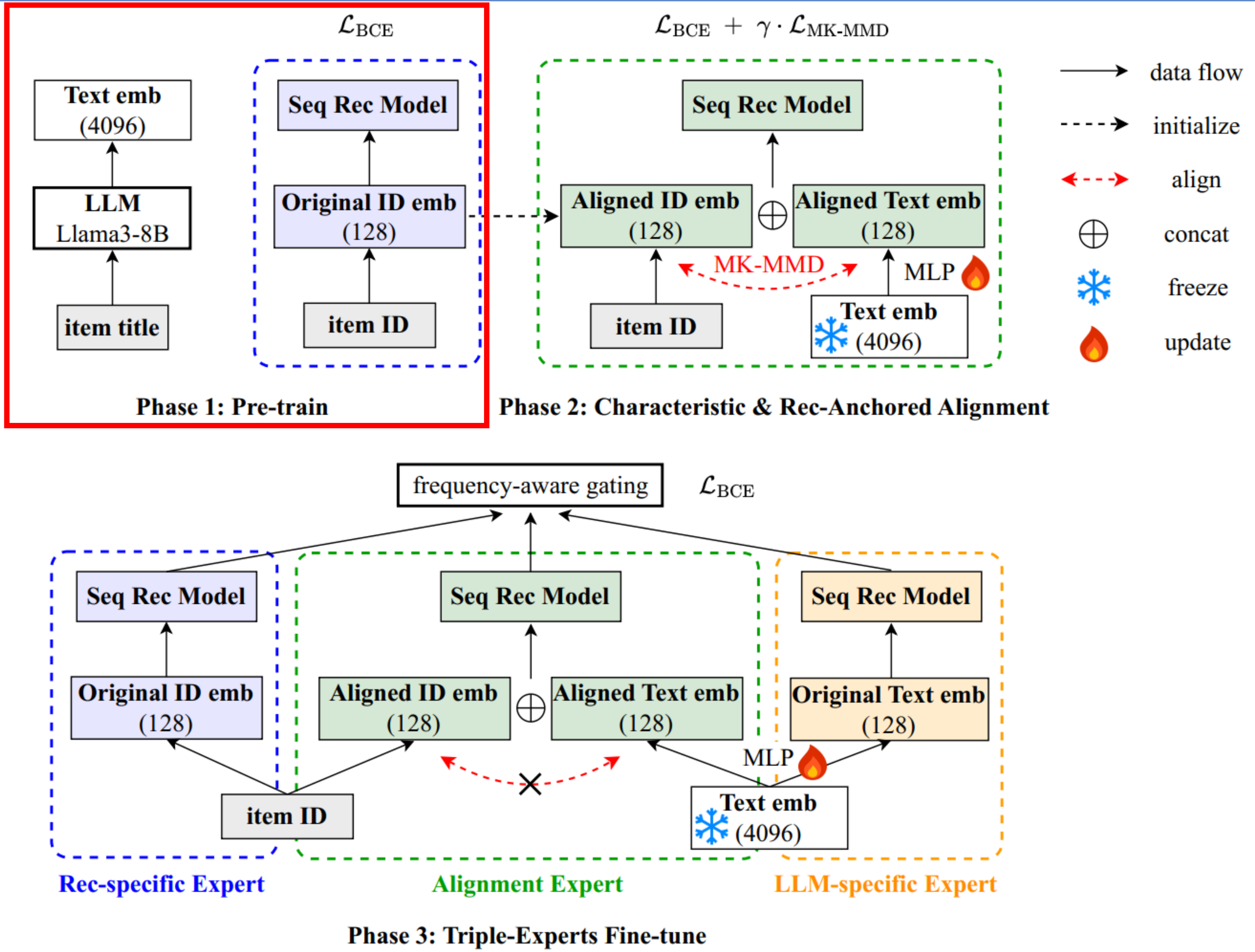




Wang, Yuhao, et al. "Pre-train, Align, and Disentangle: Empowering Sequential Recommendation with Large Language Models." SIGIR 2025.

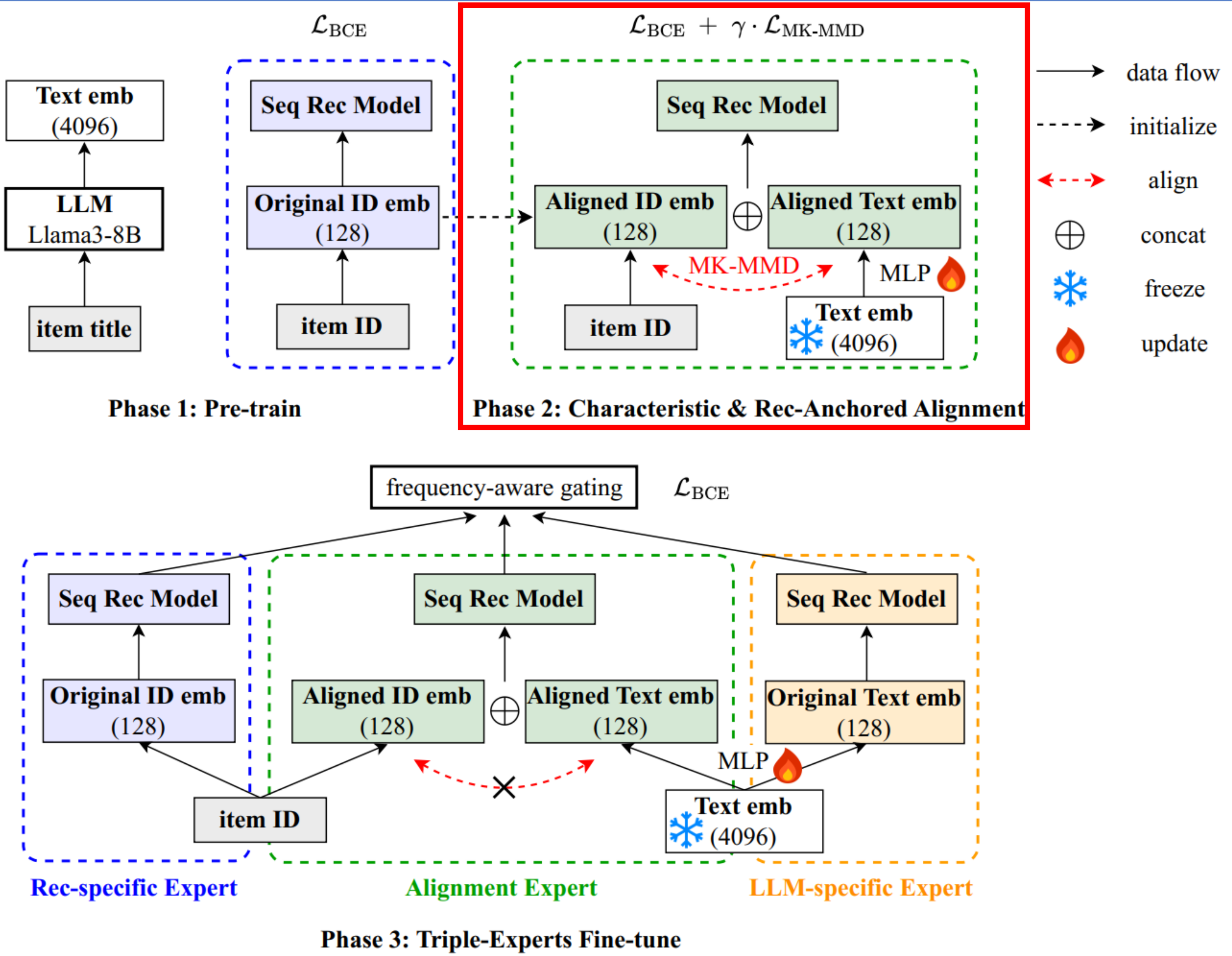
PAD

- **Pre-train**
  - Text & Collaborative emb



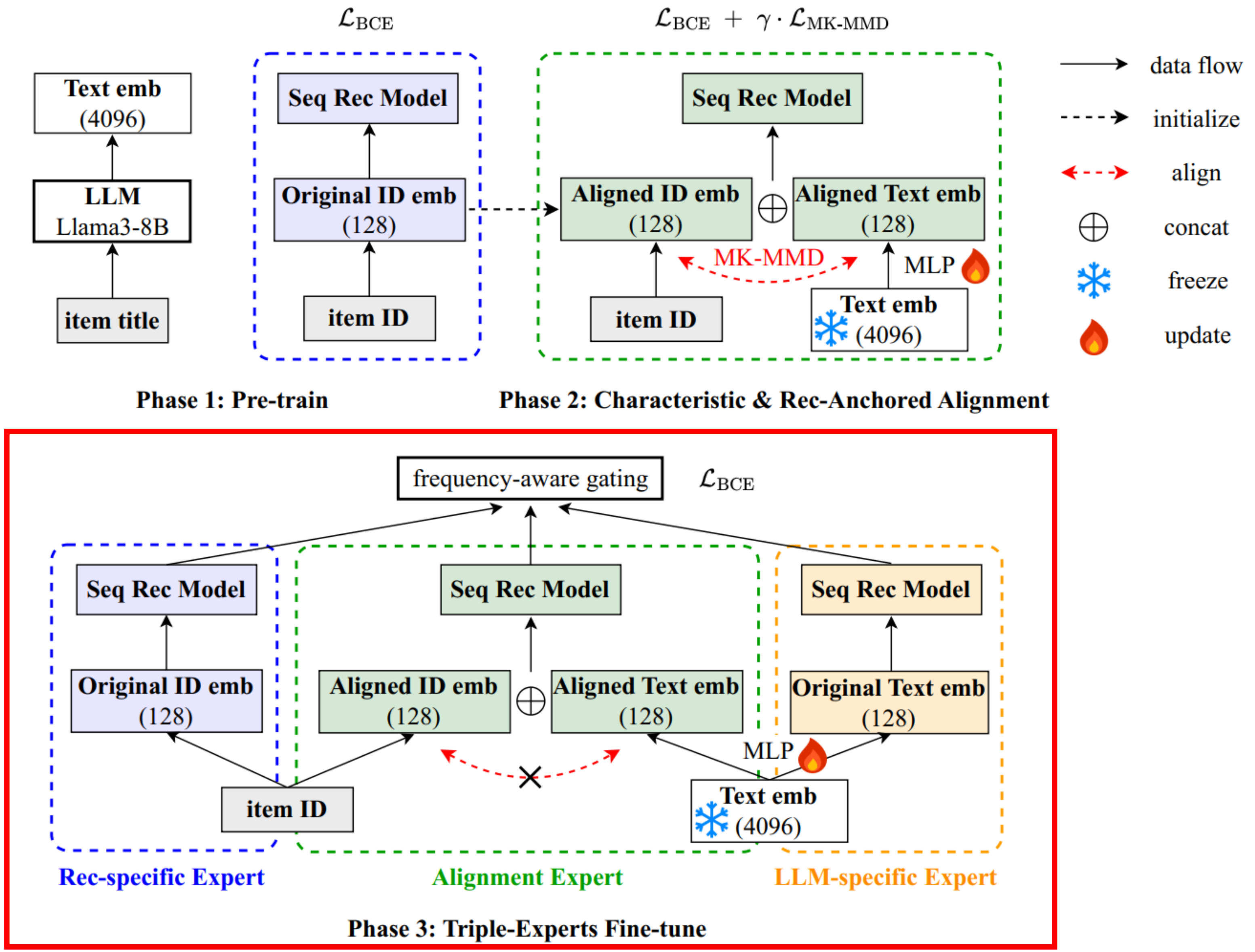
PAD

- **Align**
  - Recommendation as anchor
  - Multi-kernel maximum mean discrepancy as alignment loss based on characteristic kernel



PAD

- **Disentangle**
  - Triple-Experts & Multi-Emb

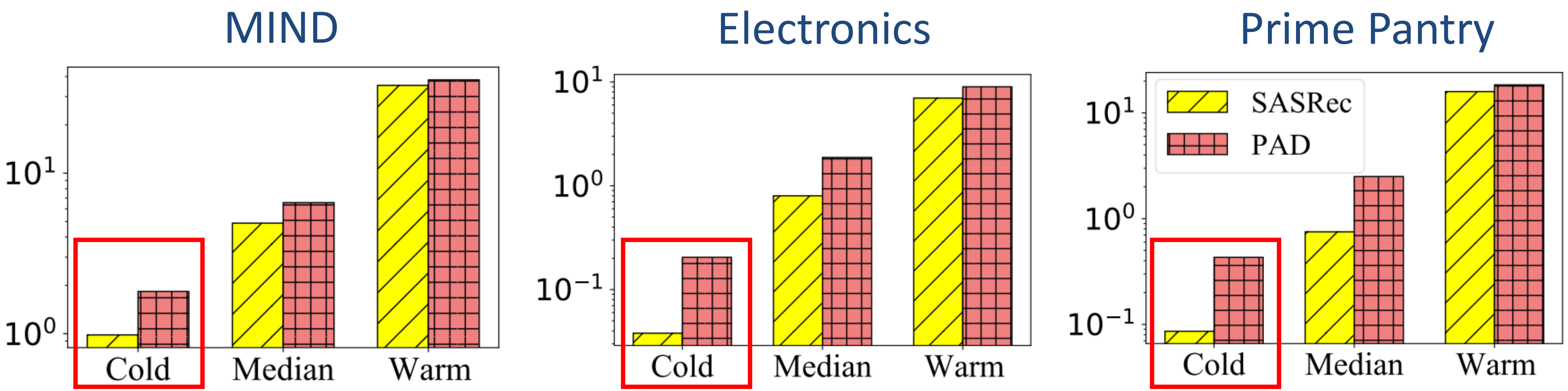


PAD

Datasets	Metric	SASRec	Hybrid	MoRec	CTRL	MAKE	DisCo	SMEM	Ours	Impr.
MIND	HR@10	16.8437	17.3385	14.1235	18.1318	16.4317	17.3367	<u>18.4319</u>	**18.6703**★	10.84%
	nDCG@10	9.0520	9.3321	7.7090	9.7996	8.9684	9.4131	<u>10.0003</u>	**10.1515**★	12.15%
Electronics	HR@10	1.6754	1.5673	0.9562	1.9468	0.9495	1.8148	<u>2.3492</u>	**2.4804**★	48.05%
	nDCG@10	0.7938	0.8385	0.4721	0.9370	0.4611	0.9025	<u>1.4287</u>	**1.5373**★	93.67%
Prime Pantry	HR@10	2.6338	3.1000	3.1699	2.8048	3.3408	2.8358	<u>3.4341</u>	**3.8303**★	45.43%
	nDCG@10	1.2926	1.5606	1.6013	1.3700	1.6379	1.3451	<u>1.7440</u>	**1.9104**★	47.80%

- PAD surpasses all baselines and achieves significant improvement.

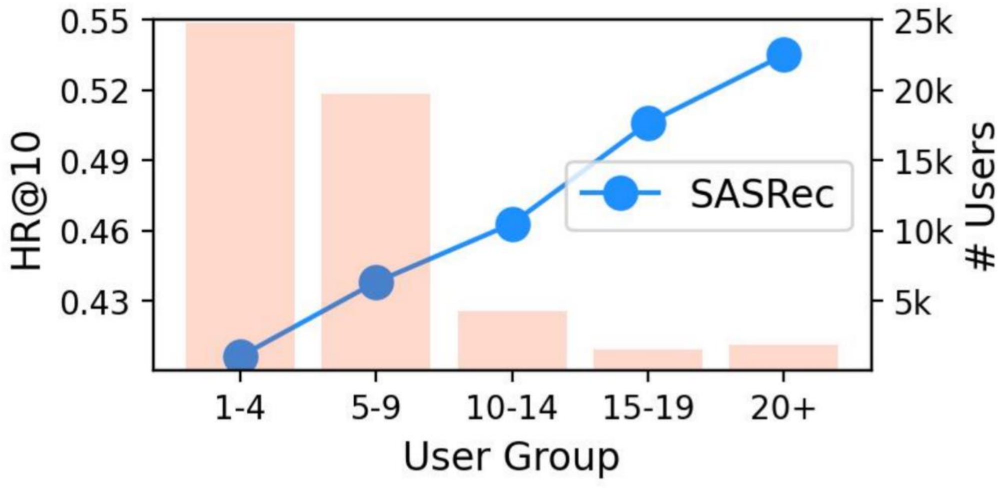
PAD



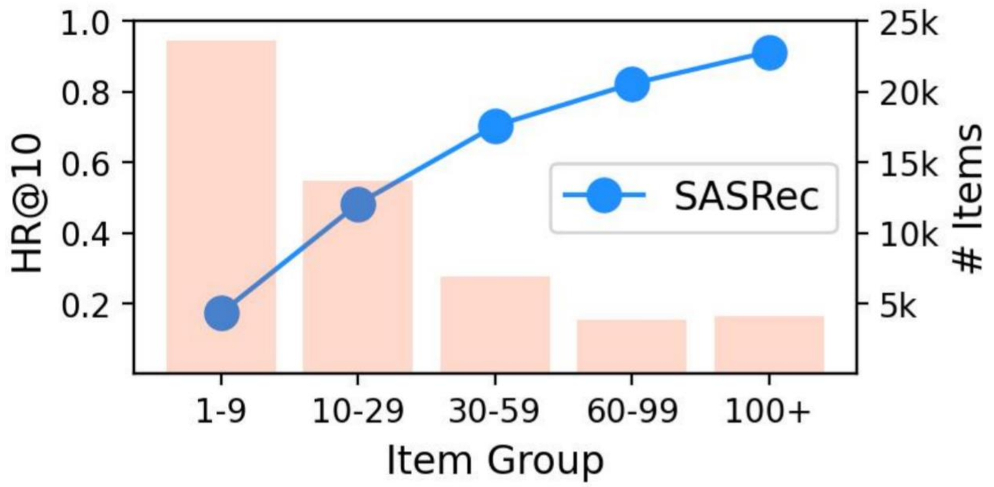
- PAD can mitigate cold-start problem with LLM knowledge.

LLM-ESR

- **Long-tail Challenges for SR**
  - **Long-tail User Challenge**: The majority of users receive less than optimal recommendation services
    → Poor experience for new users
  - **Long-tail Item Challenge**: Current SRS models often underperform those unpopular items
    → Less profits for small sellers
- **LLMs are promising to address the long-tail challenges from semantic view!**



(a) Long-tail User Challenge

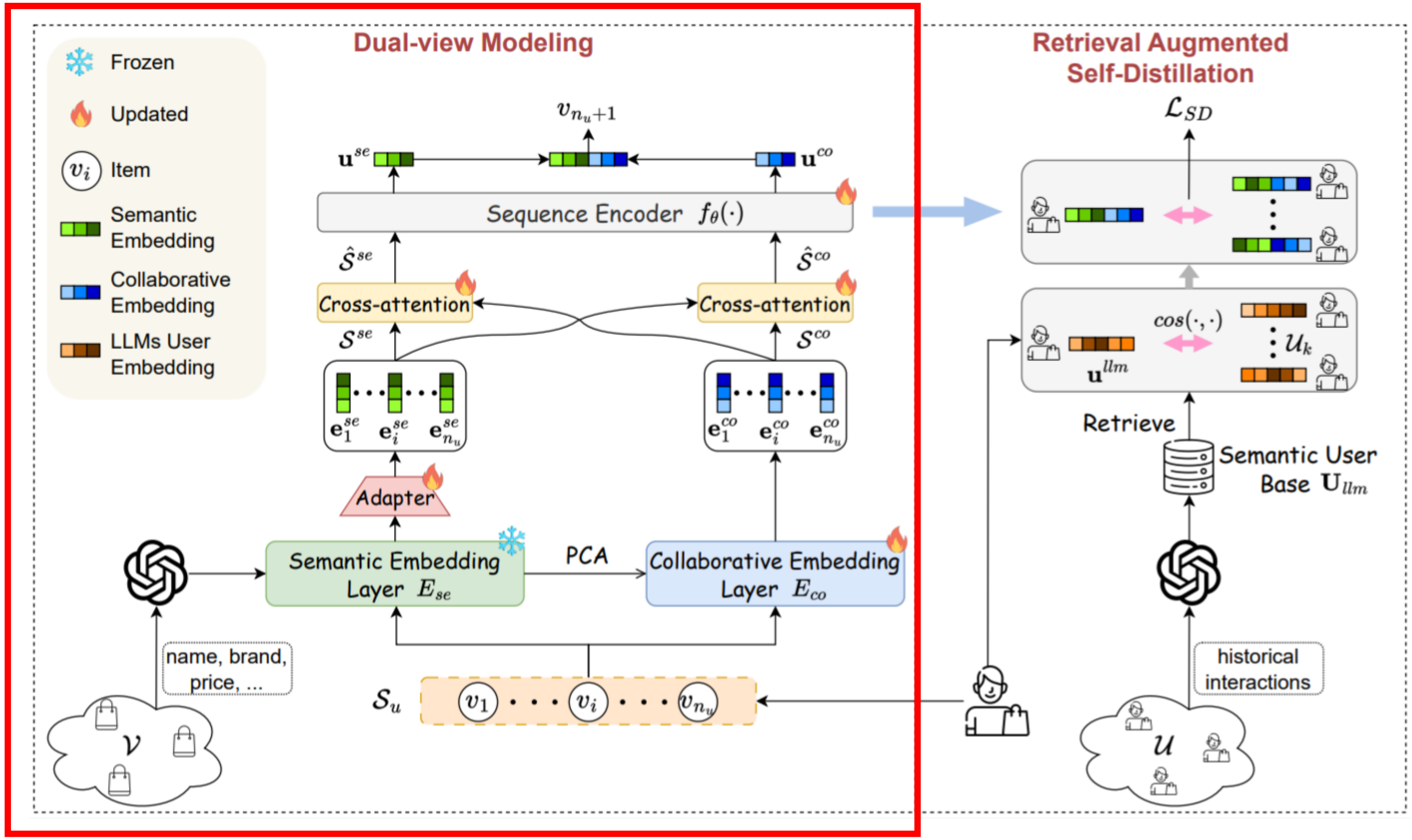


(b) Long-tail Item Challenge

Liu, Qidong, et al. "LLM-ESR: Large Language Models Enhancement for Long-tailed Sequential Recommendation." NeurIPS 2024.

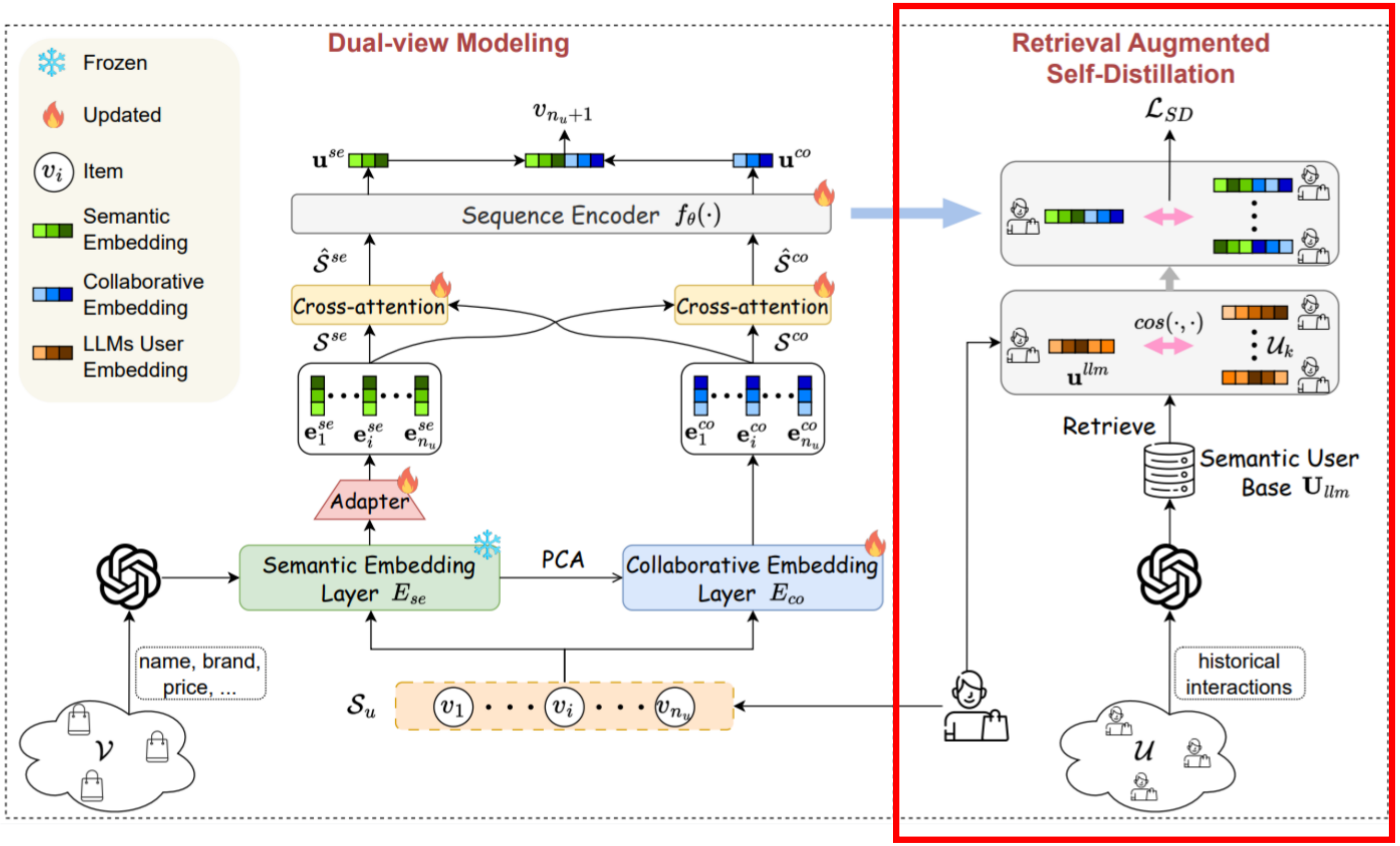
LLM-ESR

- **Dual-view Modeling:** Consisting of **semantic-view modeling** and **collaborative-view modeling** to address the long-tail item issue



LLM-ESR

- **Retrieval Augmented Self-Distillation**: Enhancing the SRS model to address the long-tail user problem



LLM-ESR

Dataset	Model	Overall		Tail Item		Head Item		Tail User		Head User	
		H@10	N@10	H@10	N@10	H@10	N@10	H@10	N@10	H@10	N@10
Yelp	GRU4Rec	0.4879	0.2751	0.0171	0.0059	0.6265	0.3544	0.4919	0.2777	0.4726	0.2653
	- CITIES	0.4898	0.2749	0.0134	0.0051	0.6301	0.3543	0.4936	0.2783	0.4756	0.2618
	- MELT	0.4985	0.2825	0.0201	0.0079	0.6393	0.3633	0.5046	0.2865	0.4750	0.2671
	- RLMRec	0.4886	0.2777	0.0188	0.0067	0.6269	0.3574	0.4920	0.2804	0.4756	0.2671
	- LLMInit	0.4872	0.2749	0.0201	0.0072	0.6246	0.3537	0.4908	0.2775	0.4732	0.2647
	- LLM-ESR	**0.5724***	**0.3413***	**0.0763***	**0.0318***	**0.7184***	**0.4324***	**0.5782***	**0.3456***	**0.5501***	**0.3247***
	Bert4Rec	0.5307	0.3035	0.0115	0.0044	0.6836	0.3916	0.5325	0.3047	0.5241	0.2988
	- CITIES	0.5249	0.3015	0.0041	0.0014	0.6783	0.3899	0.5274	0.3032	0.5155	0.2954
	- MELT	0.6206	0.3770	0.0429	0.0149	0.7907	0.4836	0.6210	0.3780	0.6191	0.3733
	- RLMRec	0.5306	0.3039	0.0104	0.0040	0.6938	0.3922	0.5351	0.3065	0.5137	0.2936
	- LLMInit	0.6199	0.3781	0.0874	0.0330	0.7766	0.4797	0.6204	0.3796	0.6178	0.3723
	- LLM-ESR	**0.6623***	**0.4222***	**0.1227***	**0.0500***	**0.8212***	**0.5318***	**0.6637***	**0.4247***	**0.6571***	**0.4127***
	SASRec	0.5940	0.3597	0.1142	0.0495	0.7353	0.4511	0.5893	0.3578	0.6122	0.3672
	- CITIES	0.5828	0.3540	0.1532	0.0700	0.7093	0.4376	0.5785	0.3511	0.5994	0.3649
	- MELT	0.6257	0.3791	0.1015	0.0371	0.7801	0.4799	0.6246	0.3804	0.6299	0.3744
	- RLMRec	0.5990	0.3623	0.0953	0.0412	0.7474	0.4568	0.5966	0.3613	0.6084	0.3658
	- LLMInit	0.6415	0.3997	0.1760	0.0789	0.7785	0.4941	0.6403	0.4010	0.6462	0.3948
	- LLM-ESR	**0.6673***	**0.4208***	**0.1893***	**0.0845***	**0.8080***	**0.5199***	**0.6685***	**0.4229***	**0.6627***	**0.4128***

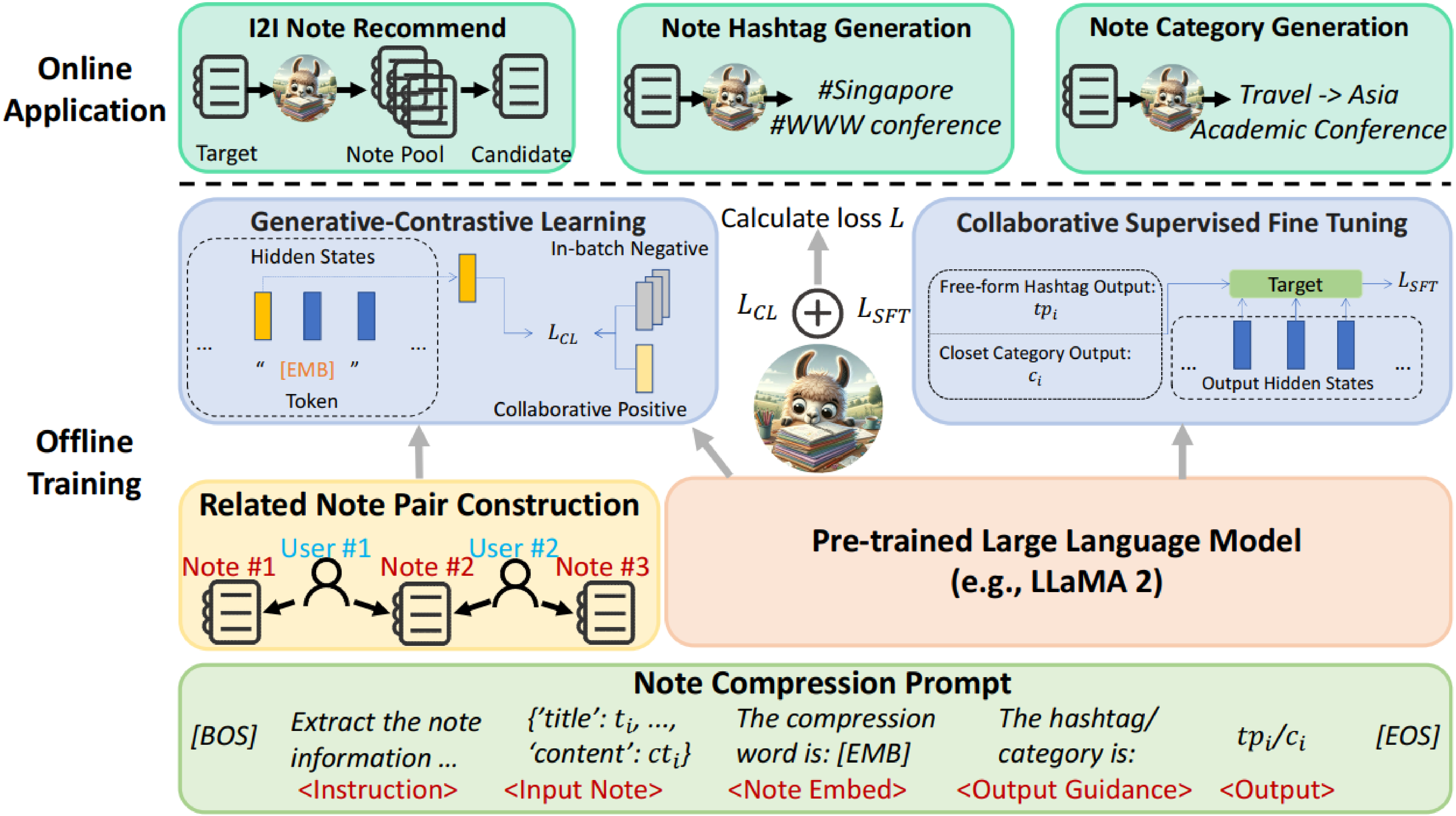
- LLM-ESR leads the overall performance, which indicates better enhancing effects.

LLM-ESR

Dataset	Model	Overall		Tail Item		Head Item		Tail User		Head User	
		H@10	N@10	H@10	N@10	H@10	N@10	H@10	N@10	H@10	N@10
Yelp	GRU4Rec	0.4879	0.2751	0.0171	0.0059	0.6265	0.3544	0.4919	0.2777	0.4726	0.2653
	- CITIES	0.4898	0.2749	0.0134	0.0051	0.6301	0.3543	0.4936	0.2783	0.4756	0.2618
	- MELT	0.4985	0.2825	0.0201	0.0079	0.6393	0.3633	0.5046	0.2865	0.4750	0.2671
	- RLMRec	0.4886	0.2777	0.0188	0.0067	0.6269	0.3574	0.4920	0.2804	0.4756	0.2671
	- LLMInit	0.4872	0.2749	0.0201	0.0072	0.6246	0.3537	0.4908	0.2775	0.4732	0.2647
	- LLM-ESR	**0.5724***	**0.3413***	**0.0763***	**0.0318***	**0.7184***	**0.4324***	**0.5782***	**0.3456***	**0.5501***	**0.3247***
	Bert4Rec	0.5307	0.3035	0.0115	0.0044	0.6836	0.3916	0.5325	0.3047	0.5241	0.2988
	- CITIES	0.5249	0.3015	0.0041	0.0014	0.6783	0.3899	0.5274	0.3032	0.5155	0.2954
	- MELT	0.6206	0.3770	0.0429	0.0149	0.7907	0.4836	0.6210	0.3780	0.6191	0.3733
	- RLMRec	0.5306	0.3039	0.0104	0.0040	0.6938	0.3922	0.5351	0.3065	0.5137	0.2936
	- LLMInit	0.6199	0.3781	0.0874	0.0330	0.7766	0.4797	0.6204	0.3796	0.6178	0.3723
	- LLM-ESR	**0.6623***	**0.4222***	**0.1227***	**0.0500***	**0.8212***	**0.5318***	**0.6637***	**0.4247***	**0.6571***	**0.4127***
	SASRec	0.5940	0.3597	0.1142	0.0495	0.7353	0.4511	0.5893	0.3578	0.6122	0.3672
	- CITIES	0.5828	0.3540	0.1532	0.0700	0.7093	0.4376	0.5785	0.3511	0.5994	0.3649
	- MELT	0.6257	0.3791	0.1015	0.0371	0.7801	0.4799	0.6246	0.3804	0.6299	0.3744
	- RLMRec	0.5990	0.3623	0.0953	0.0412	0.7474	0.4568	0.5966	0.3613	0.6084	0.3658
	- LLMInit	0.6415	0.3997	0.1760	0.0789	0.7785	0.4941	0.6403	0.4010	0.6462	0.3948
	- LLM-ESR	**0.6673***	**0.4208***	**0.1893***	**0.0845***	**0.8080***	**0.5199***	**0.6685***	**0.4229***	**0.6627***	**0.4128***

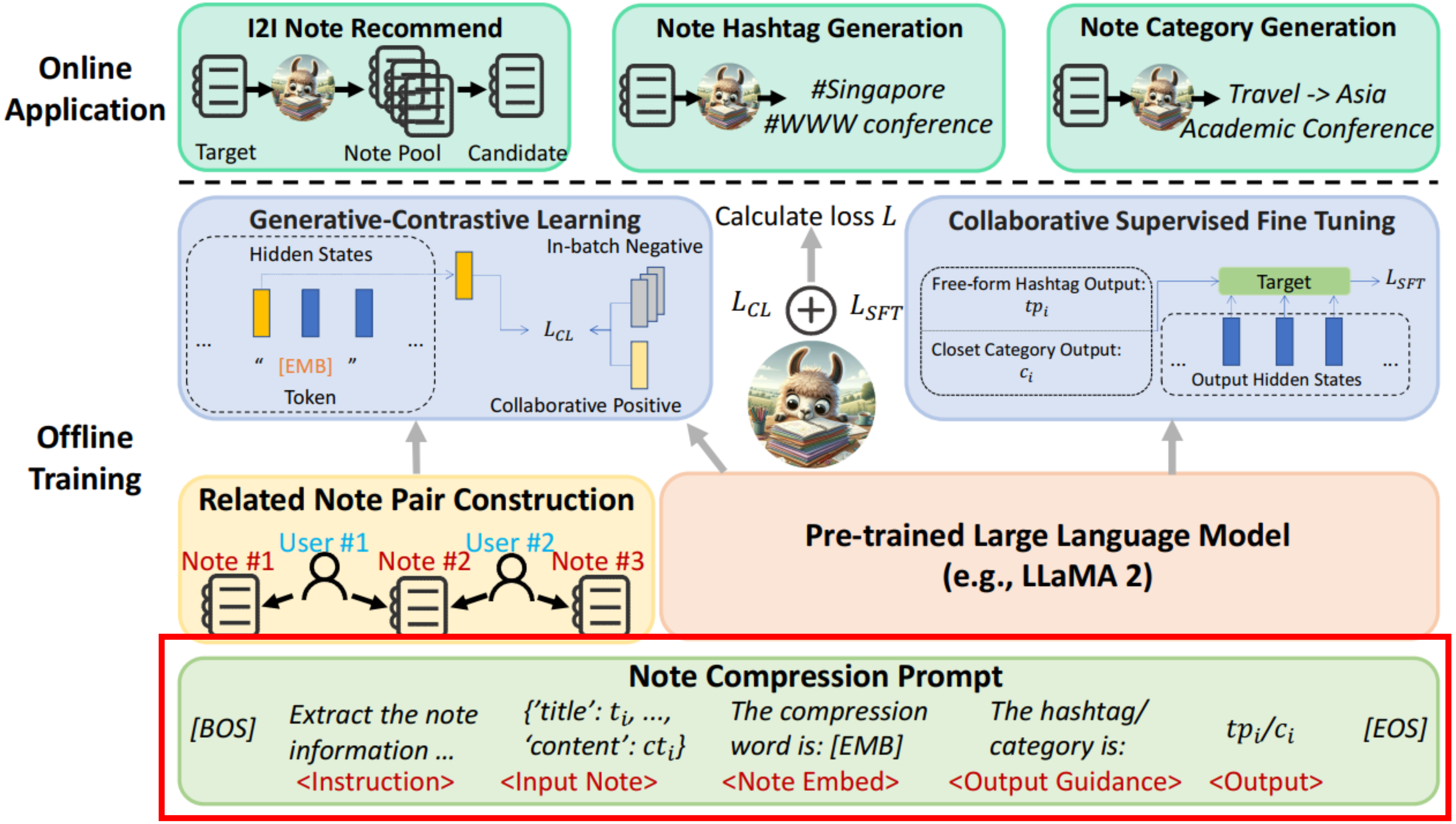
- LLM-ESR achieves the best on tail and popular item group.
- LLM-ESR can augment the tail user group better.

NoteLLM



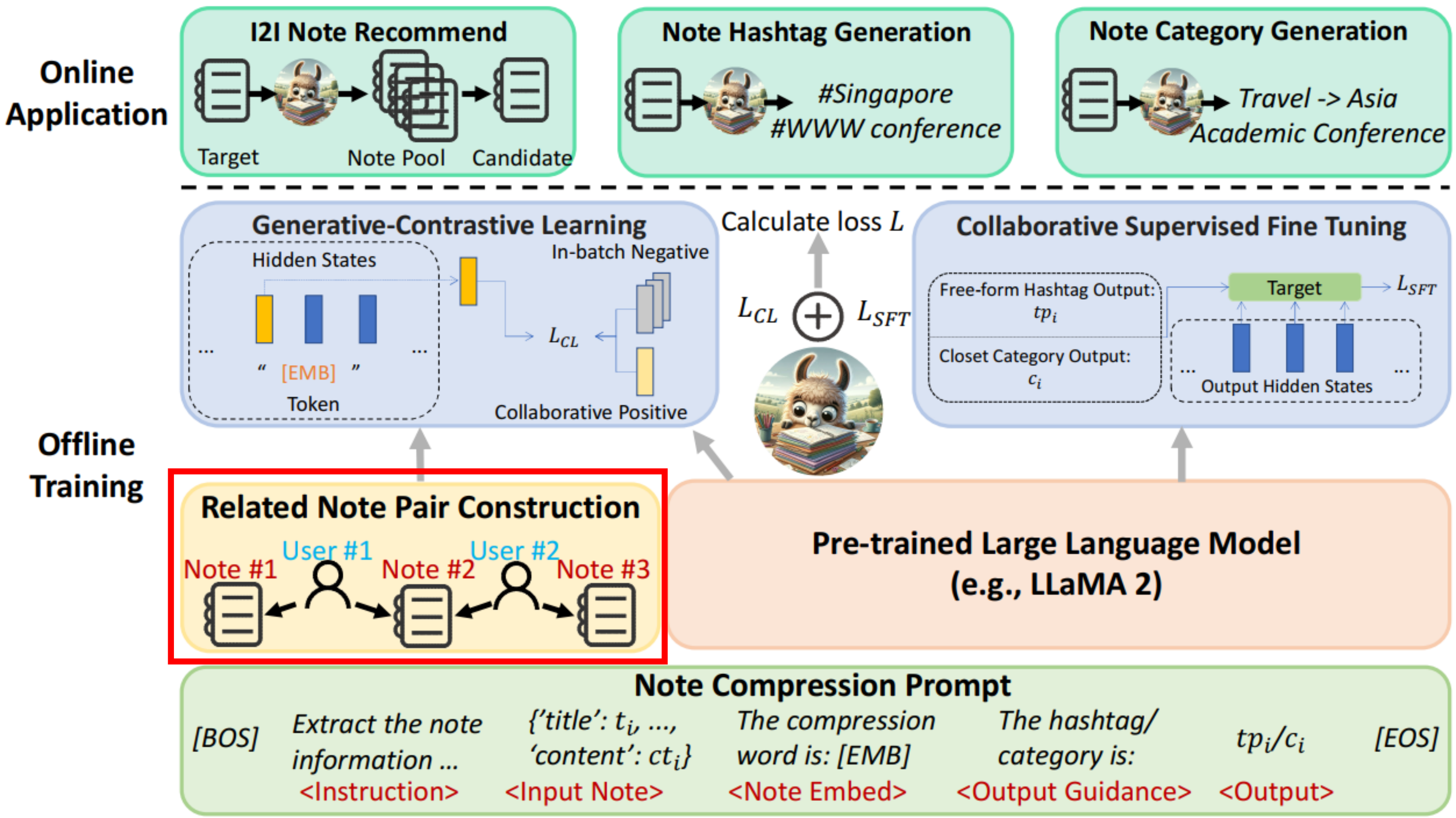
Zhang, Chao, et al. "NoteLLM: A Retrievable Large Language Model for Note Recommendation." WWW 2024.

NoteLLM



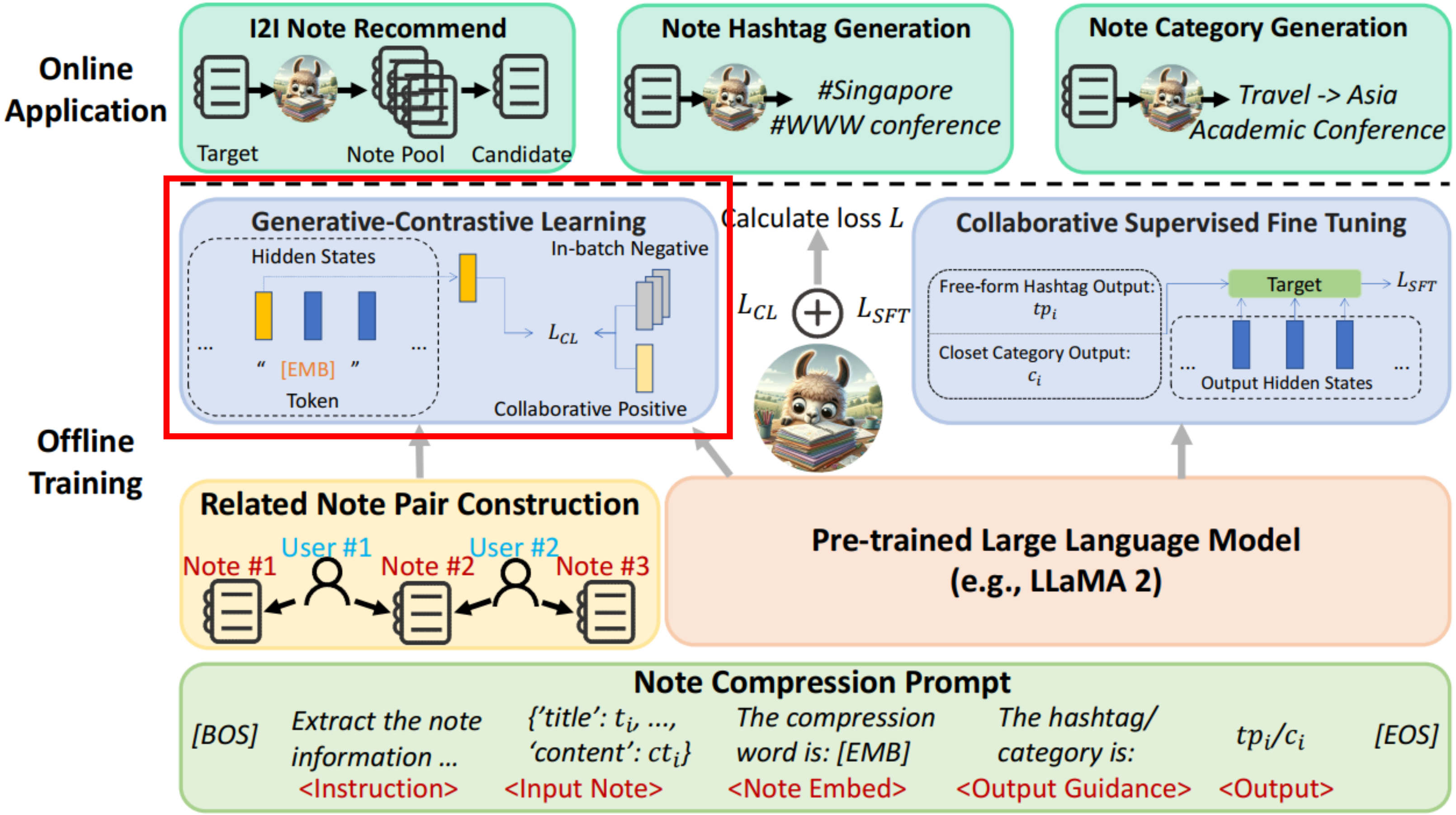
- **Note Compression Prompt**
  - Compressing the note content
  - Generating hashtags/categories

NoteLLM



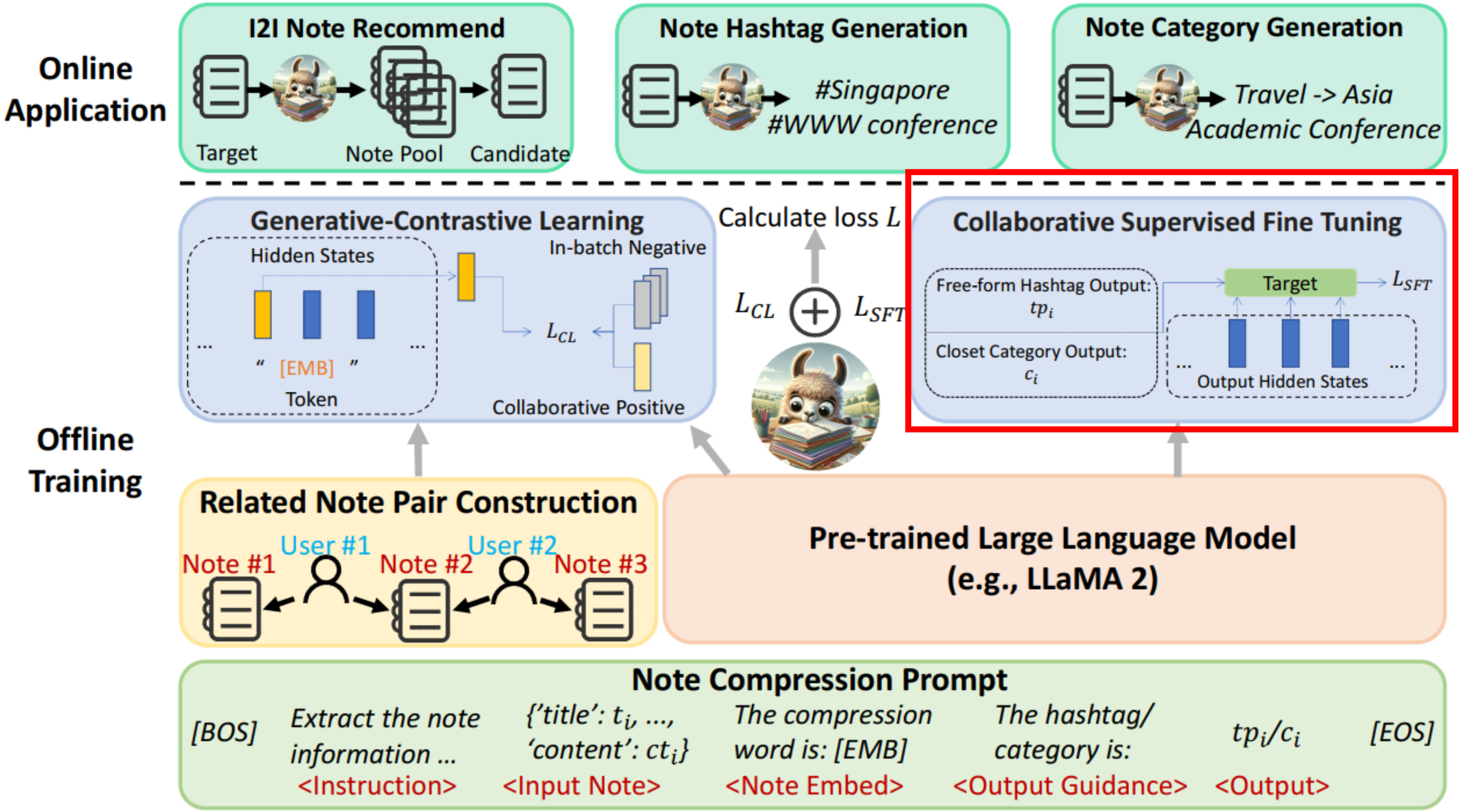
- **Constructing related note pairs**
  - Counting co-occurrence scores
  - Selecting notes with highest scores as related notes

NoteLLM



- **Generative-Contrastive Learning**
  - Identifying related notes from in-batch negatives

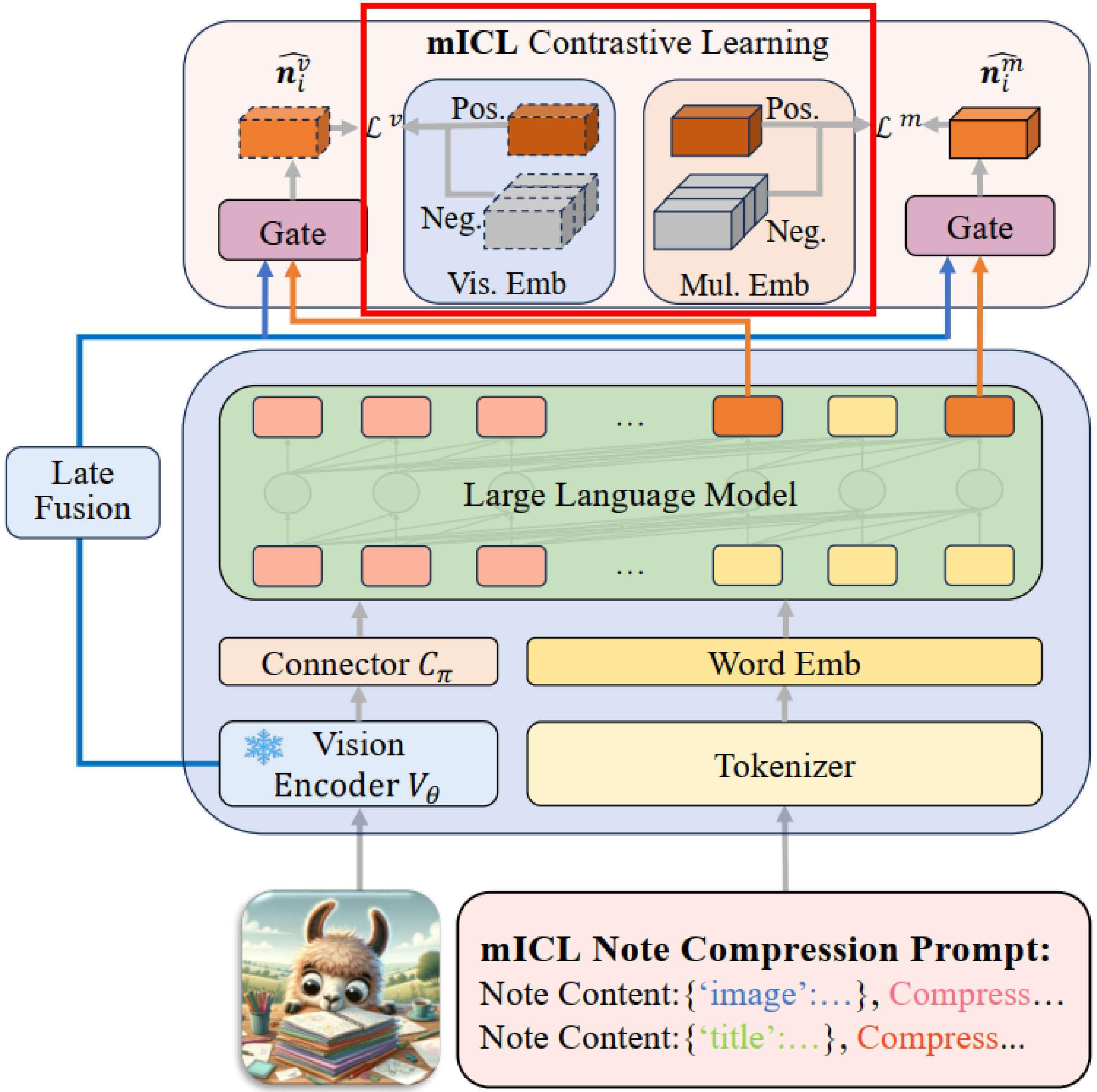
NoteLLM



- **Collaborative Supervised Fine-tuning**
  - Generating hashtags/categories for each note

NoteLLM-2

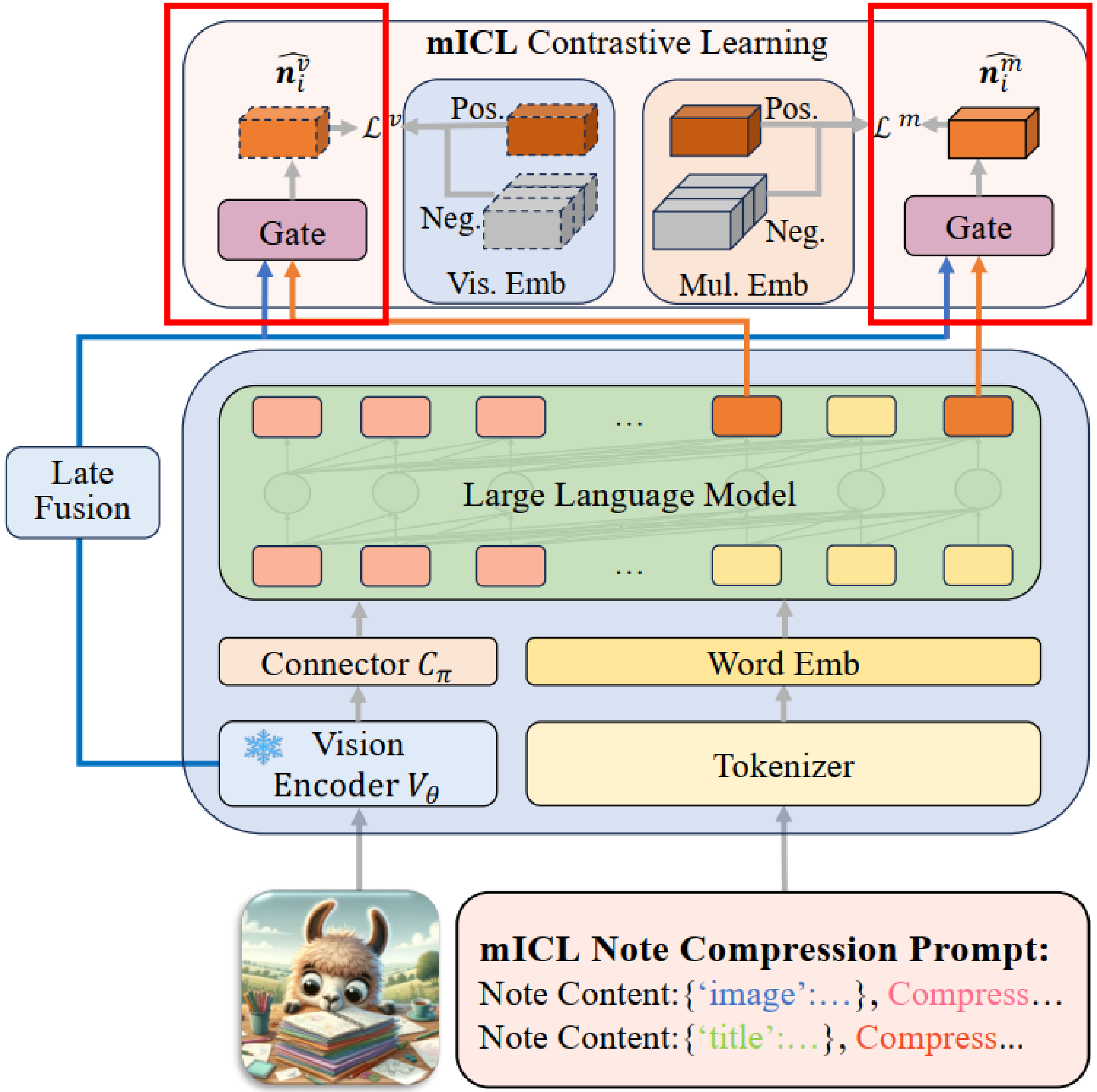
- **Multimodal In-Context Learning**
  - Separating multimodal content into visual & textual components
  - Compressing into modality-compressed words



Zhang, Chao, et al. "NoteLLM-2: Multimodal Large Representation Models for Recommendation." KDD 2025.

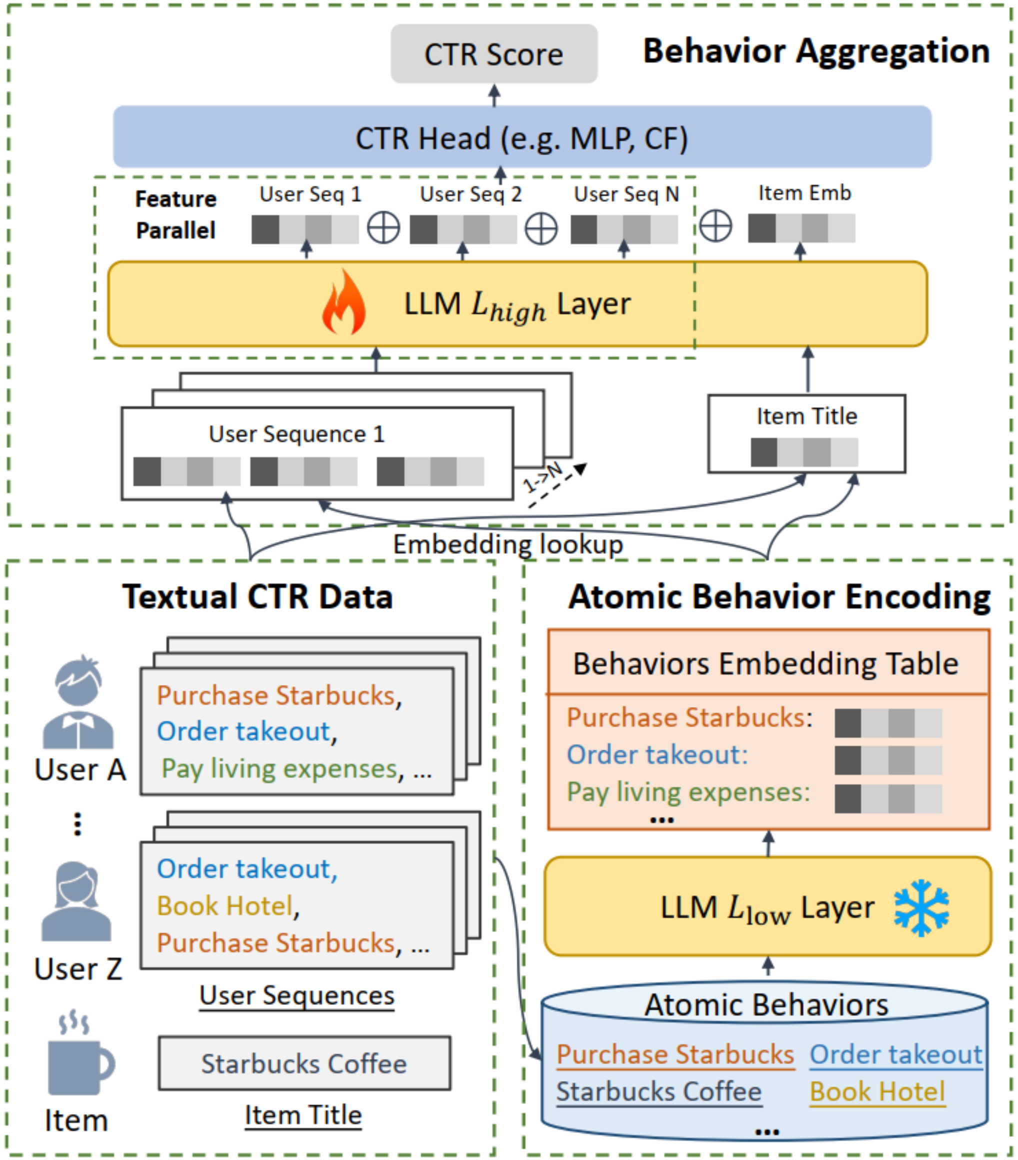
NoteLLM-2

- **Late Fusion**
  - Multimodal Gating mechanism



BAHE

- **Hierarchical Encoding**
  - Lower layers encoding item into atomic representation
  - Higer layers generating high-level user representations

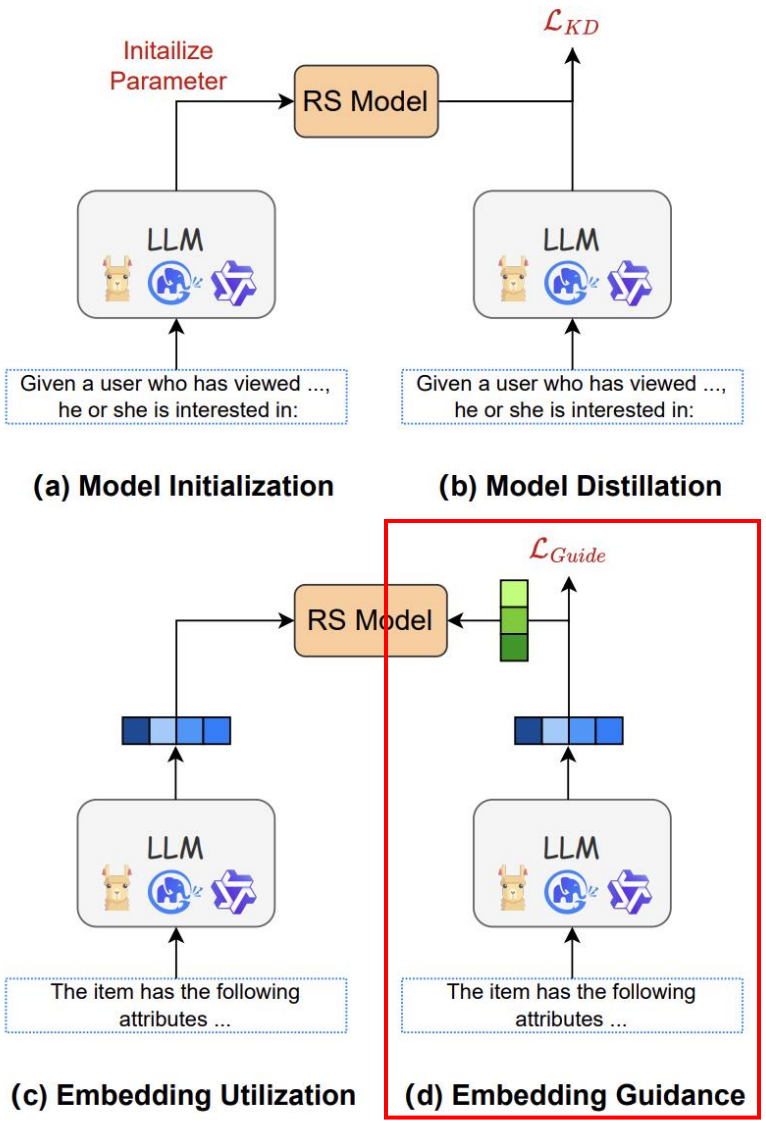


Geng, Binzong, et al. "Breaking the Length Barrier: LLM-Enhanced CTR Prediction in Long Textual User Behaviors." SIGIR 2024.

Embedding Guidance

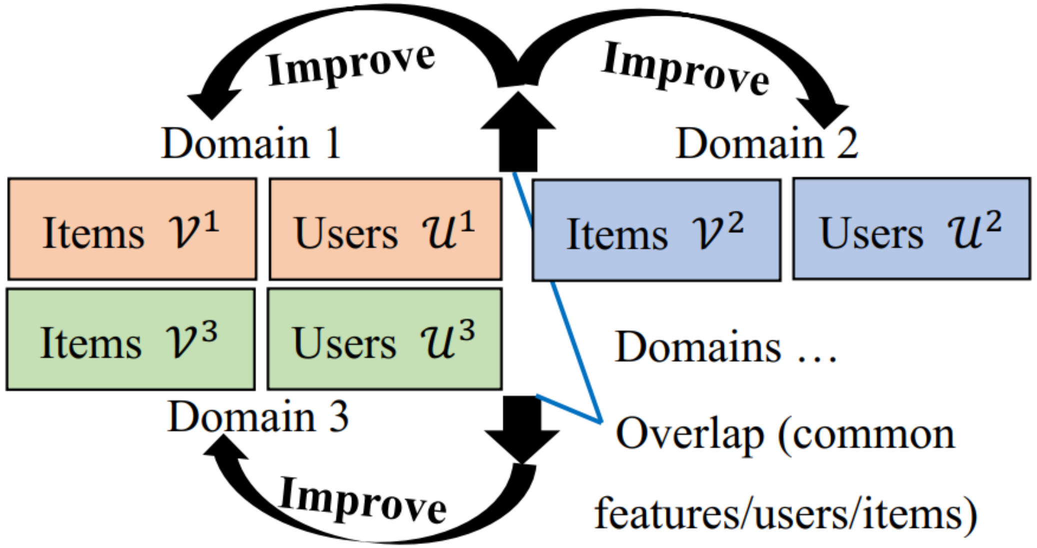
- This subcategory refers to only using the LLM embeddings as the guidance for training or parameter synthesis.

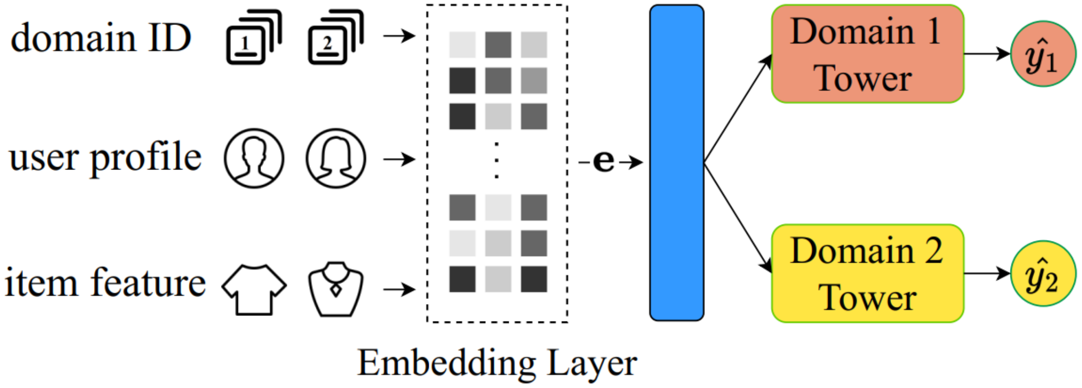
- Categories
  - User Only
  - User & Item



LLM4MSR

- **Multi-Scenario Modeling**
  - Tackling data sparsity
  - Reducing computation cost
  - High efficiency
- **Challenges**
  - Insufficient scenario knowledge
  - Ignoring cross-scenario preferences
- **Motivation**
  - Incorporating LLM to improve conventional recommender system
  - LLM as reasoner + encoder

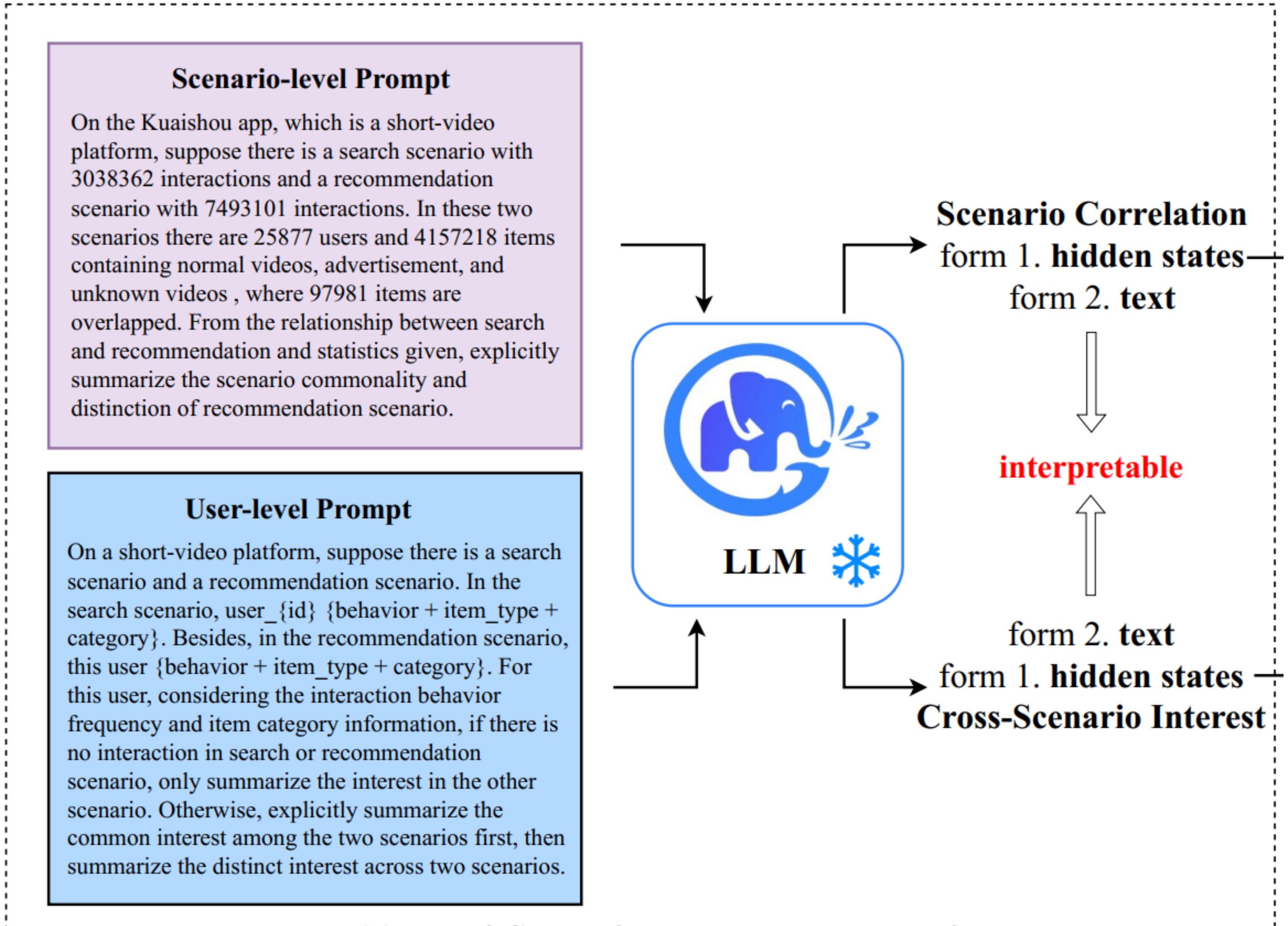




Wang, Yuhao, et al. "LLM4MSR: An LLM-Enhanced Paradigm for Multi-Scenario Recommendation." CIKM 2024.

LLM4MSR

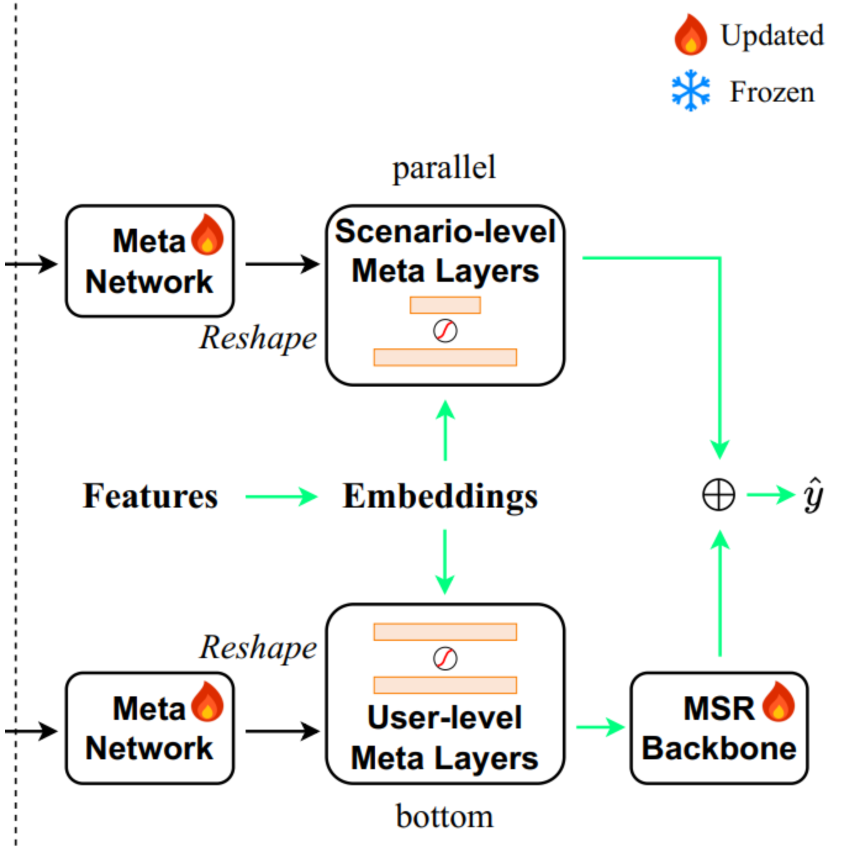
- **Multi-scenario reasoning**
  - Scenario-level prompt
  - User-level prompt



(a) Multi-Scenario Knowledge Reasoning

LLM4MSR

- **Multi-level Fusion**
  - Meta-networks generating meta layers
  - Hierarchical bottom + parallel structure



(b) Multi-Level Knowledge Fusion

LLM4MSR

- LLM4MSR achieves an increase of 1.5%, 1%, and 40% in AUC on three datasets
  - Domain correlation & Personalized interest + Adaptive meta network

Backbones	KuaiSAR-small		KuaiSAR		Amazon		
	Rec	Search	Rec	Search	Rec#1	Rec#2	Rec#3
STAR	0.7225	0.6089	0.7387	0.6268	0.6156	0.6320	0.6225
STAR_DN	0.7241	0.6116	0.7404	0.6270	0.6306	0.6350	0.6355
STAR_EP	0.7230	0.6082	0.7388	0.6266	0.6211	0.6302	0.6378
STAR_ours	**0.7276**★	**0.6181**★	**0.7408**★	**0.6332**★	**0.8720**★	**0.6364**★	**0.7543**★
OMoE	0.7241	0.6163	0.7394	0.6310	0.5995	0.6043	0.6252
OMoE_DN	0.7249	0.6183	0.7402	0.6319	0.6027	0.6171	0.6289
OMoE_EP	0.7246	0.6179	0.7392	0.6330	0.6143	0.6176	0.6312
OMoE_ours	**0.7265**★	**0.6186**★	**0.7413**★	**0.6332**★	**0.8182**★	**0.6180**★	**0.6823**★
MMoE	0.7235	0.6150	0.7392	0.6313	0.5907	0.5999	0.6032
MMoE_DN	0.7246	0.6164	0.7394	0.6330	0.6310	0.6201	0.6371
MMoE_EP	0.7245	**0.6178**	0.7397	0.6326	0.6218	0.6221	0.6315
MMoE_ours	**0.7264**★	0.6166★	**0.7410**★	**0.6341**★	**0.7860**★	**0.6222**★	**0.6854**★
PLE	0.7249	0.6149	0.7396	0.6313	0.6059	0.6127	0.5995
PLE_DN	0.7258	0.6165	0.7400	0.6333	0.6066	0.6195	0.6162
PLE_EP	0.7252	**0.6184**	0.7402	0.6339	0.6173	0.6229	0.6263
PLE_ours	**0.7269**★	**0.6184**★	**0.7408**★	**0.6343**★	**0.8265**★	**0.6250**★	**0.7074**★
AITM	0.7236	0.6154	0.7398	0.6329	0.6039	0.6126	0.6046
AITM_DN	0.7243	0.6172	0.7398	0.6318	0.6057	0.6151	0.6166
AITM_EP	0.7237	0.6177	0.7389	0.6337	0.6064	0.6163	0.6238
AITM_ours	**0.7273**★	**0.6178**★	**0.7407**★	**0.6349**★	**0.8196**★	**0.6178**★	**0.7089**★
Shared Bottom	0.7228	0.6169	0.7389	0.6321	0.6073	0.6077	0.6268
Shared Bottom_DN	0.7250	0.6176	0.7396	0.6323	0.6081	0.6253	0.6294
Shared Bottom_EP	0.7243	**0.6182**	0.7392	0.6331	0.6268	0.6174	0.6233
Shared Bottom_ours	**0.7269**★	**0.6182**★	**0.7400**★	**0.6341**★	**0.8323**★	**0.6262**★	**0.7182**★

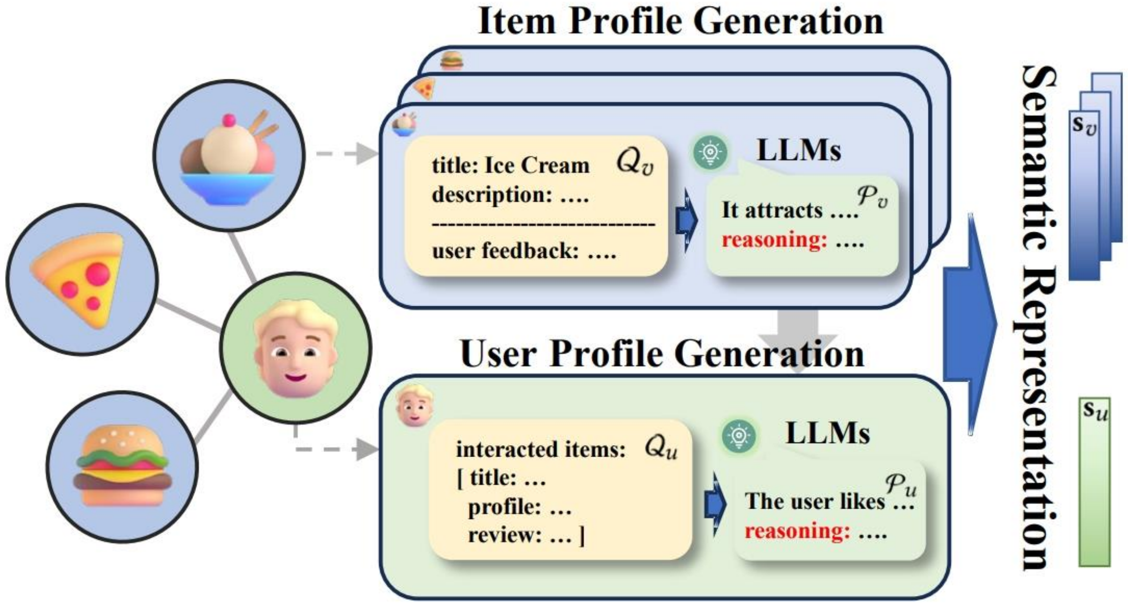
LLM4MSR

- LLM4MSR enhances various MSR backbone models, showing great compatibility and deployability

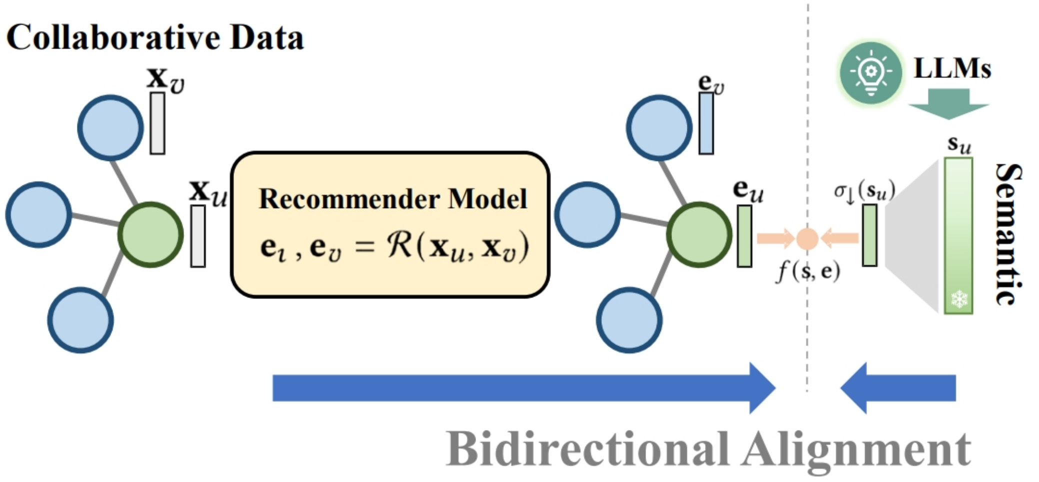
Backbones	KuaiSAR-small		KuaiSAR		Amazon		
	Rec	Search	Rec	Search	Rec#1	Rec#2	Rec#3
STAR	0.7225	0.6089	0.7387	0.6268	0.6156	0.6320	0.6225
STAR_DN	0.7241	0.6116	0.7404	0.6270	0.6306	0.6350	0.6355
STAR_EP	0.7230	0.6082	0.7388	0.6266	0.6211	0.6302	0.6378
STAR_ours	**0.7276***	**0.6181***	**0.7408***	**0.6332***	**0.8720***	**0.6364***	**0.7543***
OMoE	0.7241	0.6163	0.7394	0.6310	0.5995	0.6043	0.6252
OMoE_DN	0.7249	0.6183	0.7402	0.6319	0.6027	0.6171	0.6289
OMoE_EP	0.7246	0.6179	0.7392	0.6330	0.6143	0.6176	0.6312
OMoE_ours	**0.7265***	**0.6186***	**0.7413***	**0.6332***	**0.8182***	**0.6180***	**0.6823***
MMoE	0.7235	0.6150	0.7392	0.6313	0.5907	0.5999	0.6032
MMoE_DN	0.7246	0.6164	0.7394	0.6330	0.6310	0.6201	0.6371
MMoE_EP	0.7245	**0.6178**	0.7397	0.6326	0.6218	0.6221	0.6315
MMoE_ours	**0.7264***	0.6166*	**0.7410***	**0.6341***	**0.7860***	**0.6222***	**0.6854***
PLE	0.7249	0.6149	0.7396	0.6313	0.6059	0.6127	0.5995
PLE_DN	0.7258	0.6165	0.7400	0.6333	0.6066	0.6195	0.6162
PLE_EP	0.7252	**0.6184**	0.7402	0.6339	0.6173	0.6229	0.6263
PLE_ours	**0.7269***	**0.6184***	**0.7408***	**0.6343***	**0.8265***	**0.6250***	**0.7074***
AITM	0.7236	0.6154	0.7398	0.6329	0.6039	0.6126	0.6046
AITM_DN	0.7243	0.6172	0.7398	0.6318	0.6057	0.6151	0.6166
AITM_EP	0.7237	0.6177	0.7389	0.6337	0.6064	0.6163	0.6238
AITM_ours	**0.7273***	**0.6178***	**0.7407***	**0.6349***	**0.8196***	**0.6178***	**0.7089***
Shared Bottom	0.7228	0.6169	0.7389	0.6321	0.6073	0.6077	0.6268
Shared Bottom_DN	0.7250	0.6176	0.7396	0.6323	0.6081	0.6253	0.6294
Shared Bottom_EP	0.7243	**0.6182**	0.7392	0.6331	0.6268	0.6174	0.6233
Shared Bottom_ours	**0.7269***	**0.6182***	**0.7400***	**0.6341***	**0.8323***	**0.6262***	**0.7182***

RLMRec

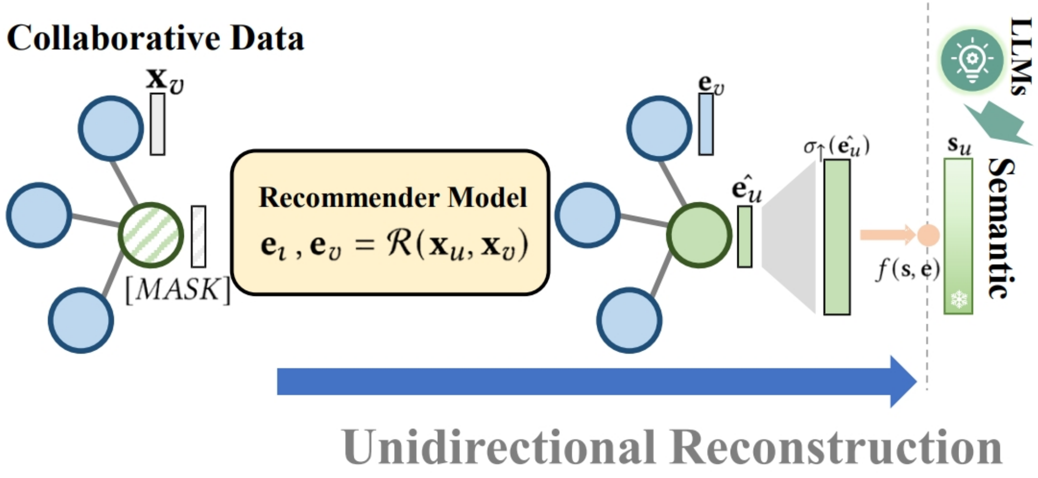
- **Generating profile**
- **Extra alignment loss**



(a) Profile Generation via Reasoning



(b) Contrastive Alignment



(c) Generative Alignment

Ren, Xubin, et al. "Representation Learning with Large Language Models for Recommendation." WWW 2024.

Summary

- Model §4
  - Model Initialization §4.1
    - Whole
      - CTRL [37]; FLIP [79];
    - Embedding
      - LLM2X [22]; SAID [24]; LEARN [30]; TSLRec [42]; LLMEmb [48]; LLMInit [109]
  - Model Distillation §4.2
    - Feature
      - LEADER [51]; SLMRec [97]
    - Response
      - DLLM2Rec [9]
  - Embedding Utilization §4.3
    - User Only
      - LLM-CF [73]; LLMGPR [55]
    - Item Only
      - TedRec [96]; LSVCR [117]; LRD [100]; NoteLLM [105]; LLM-ESR [49]; CELA [82]; NoteLLM-2 [106]; AlphaRec [69]; Flare [23]; HLLM [5]; QARM [60]; Molar [61]; PRE-CISE [70]; PAD [84]; RecG [39]
    - User & Item
      - SINGLE [54]; LoID [98]; BAHE [19]; DynLLM [116]; EmbSum [104]; Laser [40]; LARR [77]
  - Embedding Guidance §4.4
    - User Only
      - LLM4SBR [65]; LLM-ESR [49]; LLM4MSR [87]; LLM4CDSR [53]
    - User & Item
      - RLMRec [66]; DaRec [102]; HARec [63]; IRLLRec [85]

Agenda

1 Introduction

Zijian Zhang

2 Knowledge Enhancement

Pengyue Jia

3 Interaction Enhancement

Ziwei Liu

4.1 Model Enhancement 1

Maolin Wang

4.2 Model Enhancement 2

Yuhao Wang

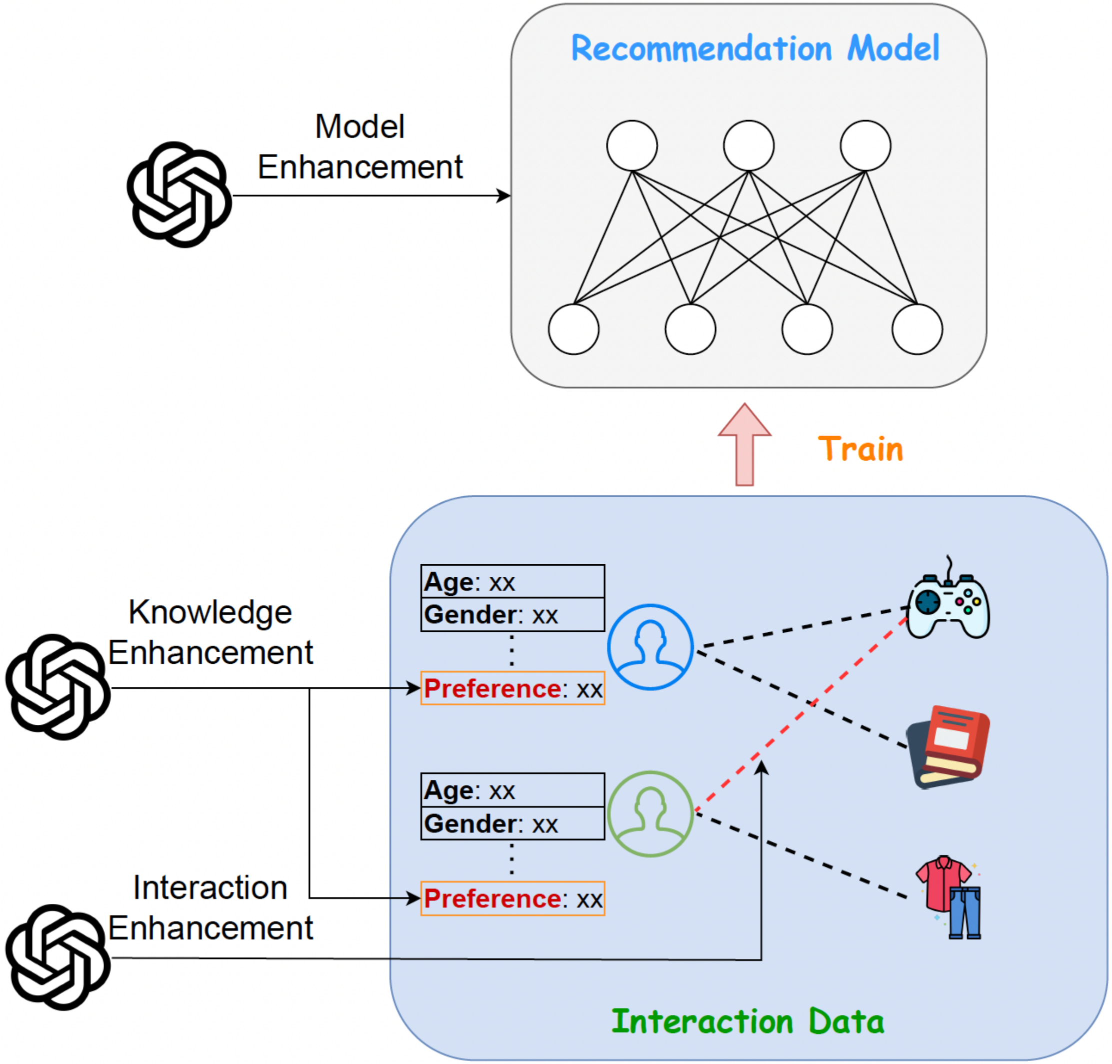
5 Conclusion

Yejing Wang

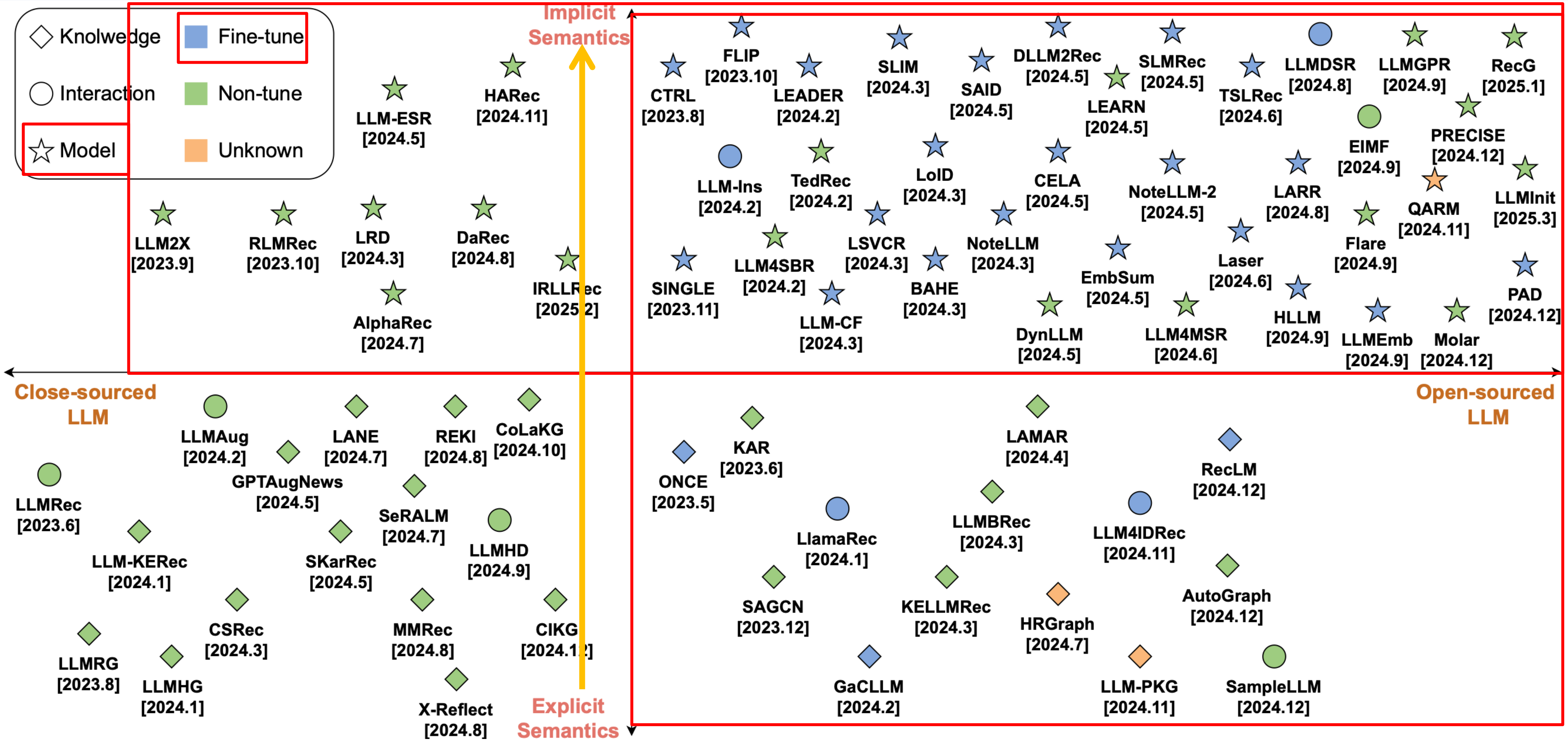
Conclusion

LLM-enhanced RS (LLMERS)

- Knowledge enhancement
  - World Knowledge, Reasoning
  - Explicit semantic -> requiring further encoding
- Interaction enhancement
  - Generation
  - Highly scalable
- Model enhancement
  - Representation, implicit semantic
  - (Small) pre-encoding & storage issues



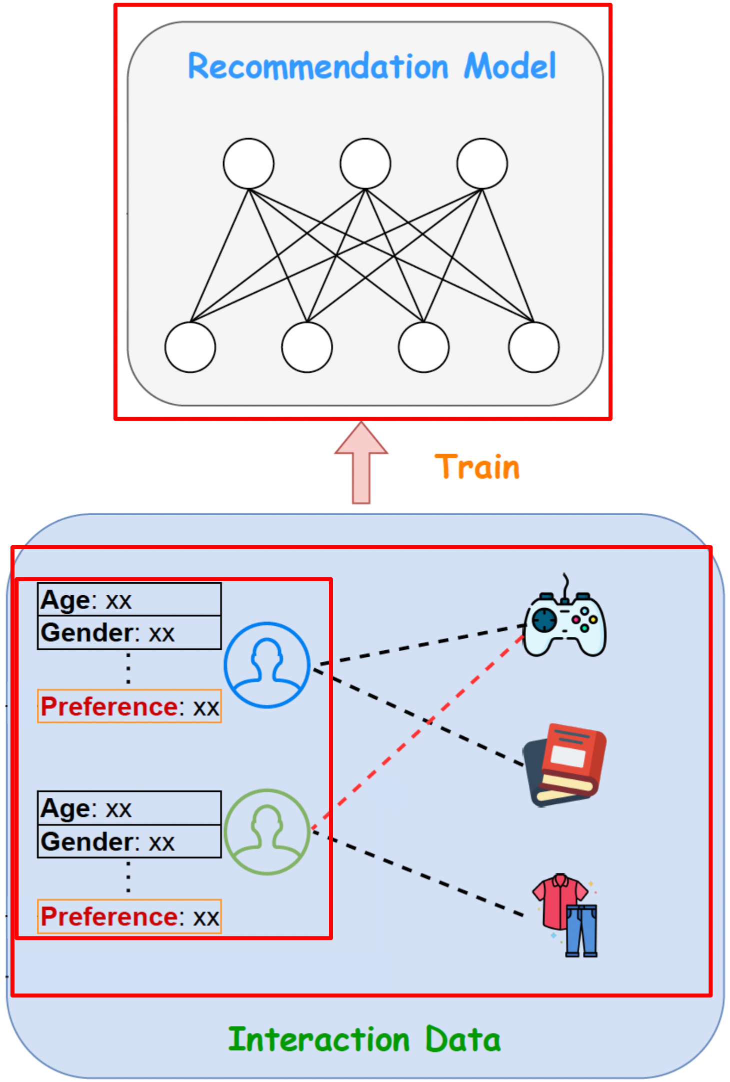
Trend



- Semantics: explicit -> implicit, model enhancement solutions are dominant
- Fine-tuned open-source LLMs are more popular for better adaptability

Future Directions

- Extended to more recommendation tasks
  - Focused on collaborative filtering and sequential recommendation
  - Potential in other tasks like multi-task, cross-domain…
- Support for multimodal/text-free RS
  - Multimodal input: MLLM -> adaptive representation
    - E.g., phone case vs. dresses on E-commerce platforms
  - Text-free RS: tabular data comprehension
- User-centric enhancement
  - Relying on item description/representations -> lengthy prompt/behavior-unaware integration
  - Item-free user enhancement, behavior-aware integration



Future Directions

- Scalability
  - More efficient than “LLM as RS” solutions
  - Huge computational & storage burden for extremely numerous items, especially knowledge & model enhancement -> Limited ROI
- Explainability
  - Not addressed by LLM
  - Generating explanations while enhancing RS
- Evaluations
  - Only evaluated combined with traditional RS
  - Lack of benchmarks and metrics for LLM enhancement