Generative Recommendation Models: Progress and Directions

Yupeng Hou[1], An Zhang[2], Leheng Sheng[2], Zhengyi Yang[3], Xiang Wang[3], Tat-Seng Chua[2], Julian McAuley[1]

Proxy Speaker: Jiancan Wu

[1]UC San Diego [2]NUS [3]USTC





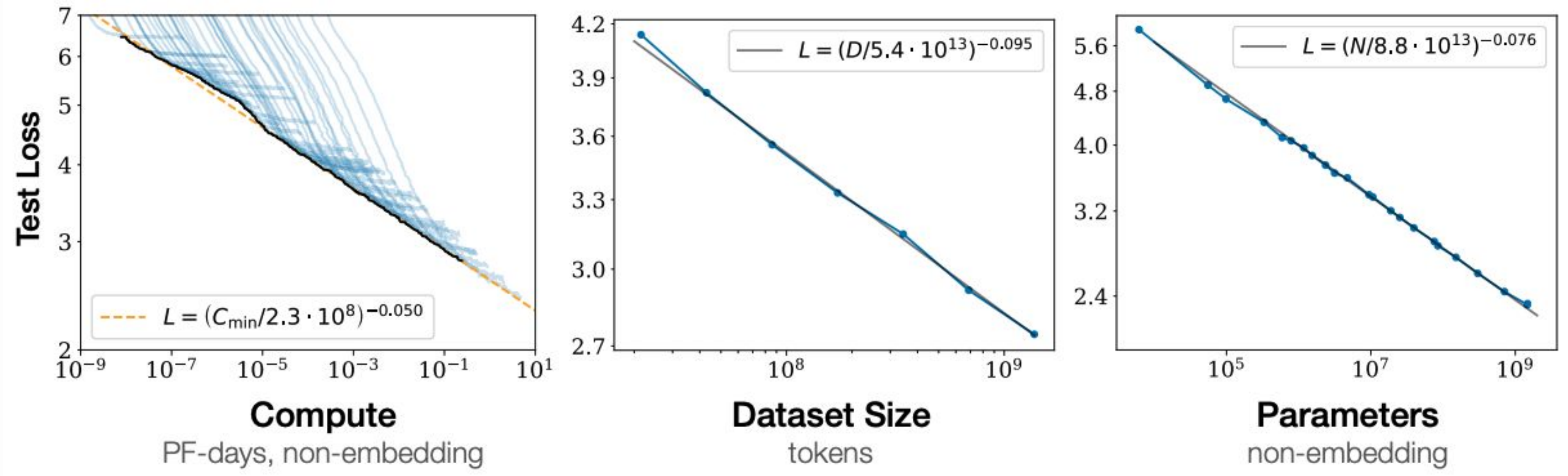


01

Introduction

of Generative Recommendation

Scaling Law as a Pathway towards AGI



Scaling laws provide a framework for understanding how **model size**, **data volume**, and **test-time computing** might lead to advanced AI capabilities.

Understanding Scaling Laws for Recommendation Models. Arxiv 2022.

However …

Language Modeling

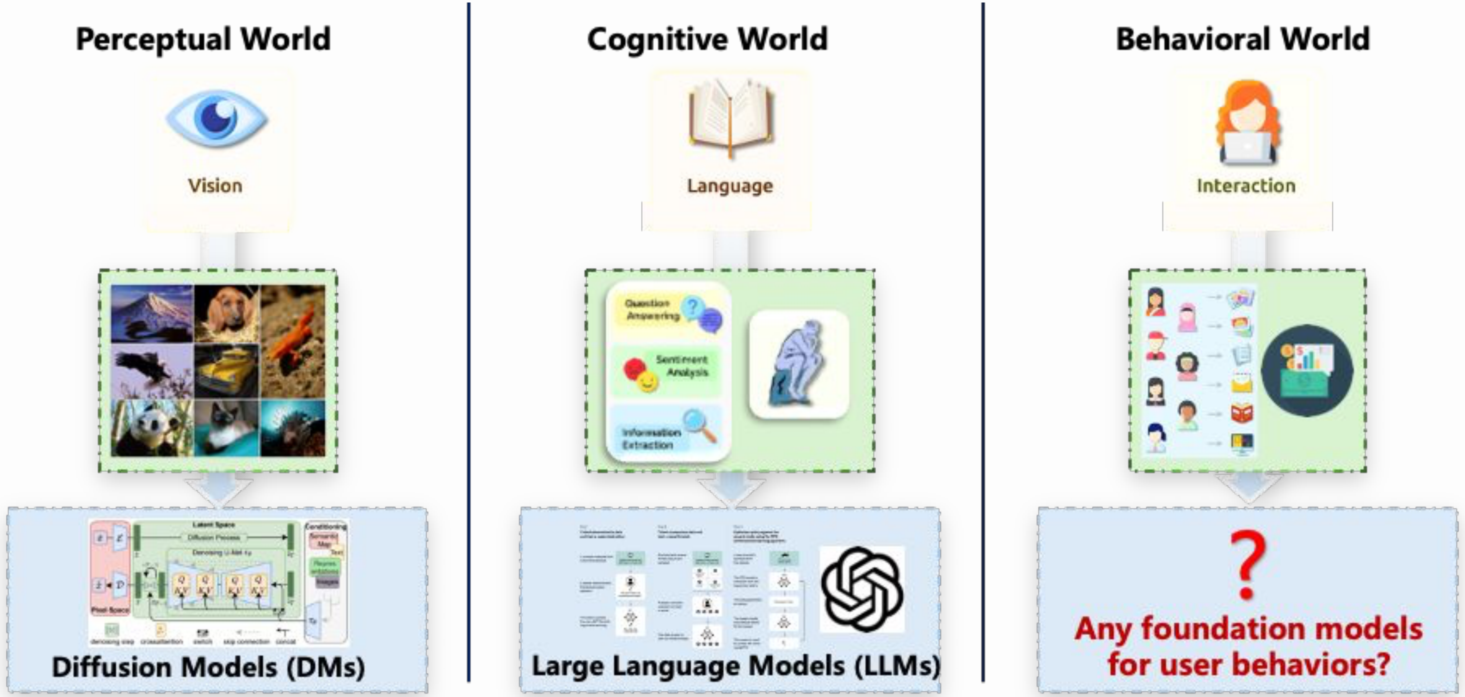
- Dense world knowledge
- Text tokens (Ten thousands level)

User Behavior Modeling

- Sparse user-item interactions
- Items (Billion to trillion level)

Scaling laws rarely apply to traditional recommendation models.

As the Reflection of Real World,



What are Generative Models & Why？

A generative model learns the underlying distribution of data and can generate new samples from it.

Learning →

$p_\theta(x)$

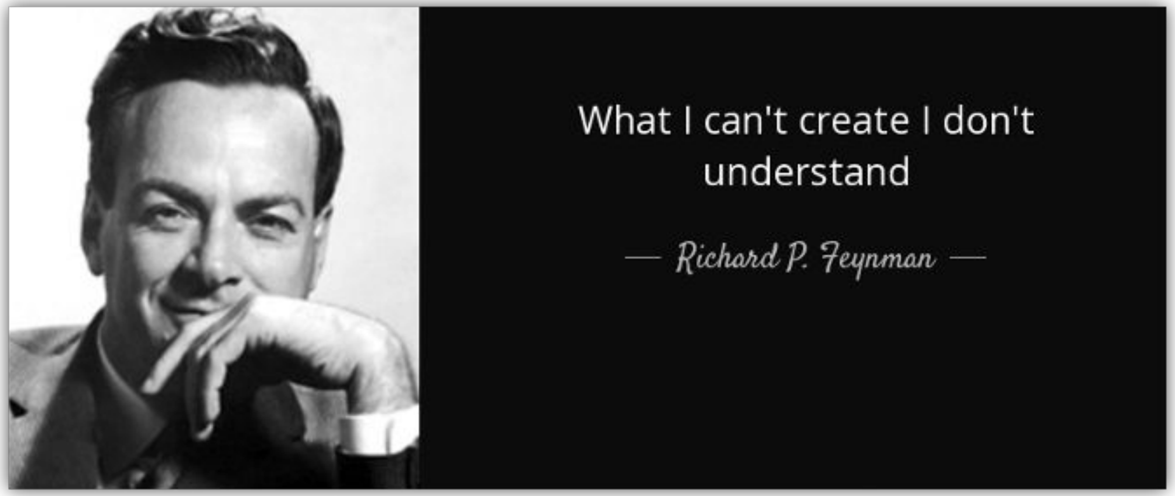
Generating →

Training data~$p_{data}(x)$

Data generation distribution

New samples

A Potential Solution: “Generative” Recommendation



“What User Behaviors LLMs can not Generate, LLMs do not Understand.”

Where are We Now?

In language and vision:

- Large language/diffusion models have been established.
- Scaling law has been witnessed.

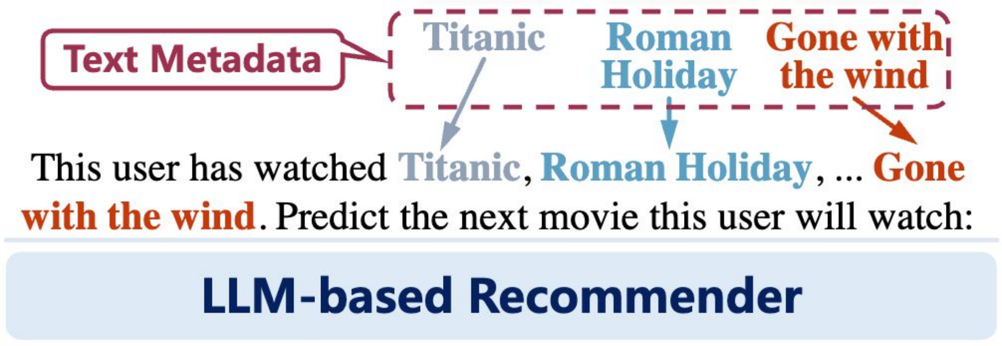
In recommendation:

- Incorporat generative components in traditional recommender.
- Initial attempts on generative recommendation.

Pathways towards Scalable Generative Recommendation

Adapt Pre-trained Models

- Large Language Models



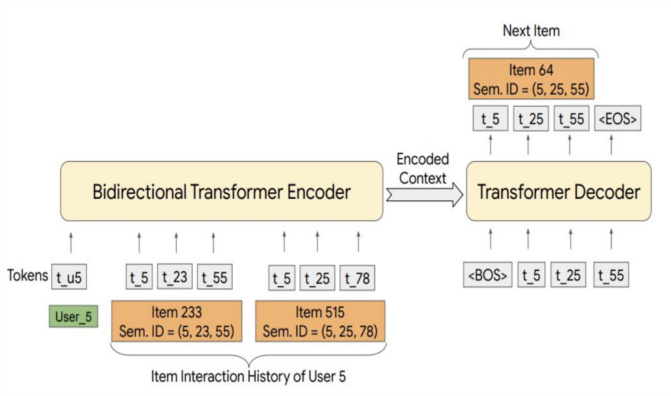
Adapting LLMs for recommendation task

Liao et al. LLaRA: Large Language-Recommendation Assistant. SIGIR 2024.

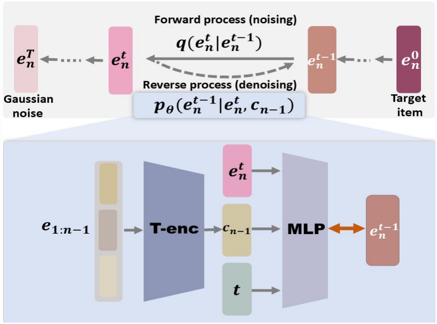
Pathways towards Scalable Generative Recommendation

Train from Scratch

- Autoregressive Models
  - Semantic ID



- Diffusion Models



Schedule Overview

Time (AEST)	Session	Presenter
9:00 - 9:10	Part 1: Background and Introduction	Tat-Seng Chua
9:10 - 10:10	Part 2: LLM-based Generative Recommendation	Leheng Sheng
10:10 - 10:30	Part 3.1: Introduction of Semantic IDs	Yupeng Hou
10:30 - 11:00	Coffee Break & QA Session	
11:00 - 11:40	Part 3.2: SemID-based Generative Recommendation	Yupeng Hou
11:40 - 12:10	Part 4: Diffusion-based Generative Recommendation	Jiancan Wu (proxy speaker of Zhengyi)
12:10 - 12:30	Part 5: Open Challenges and Beyond	Yupeng Hou

02

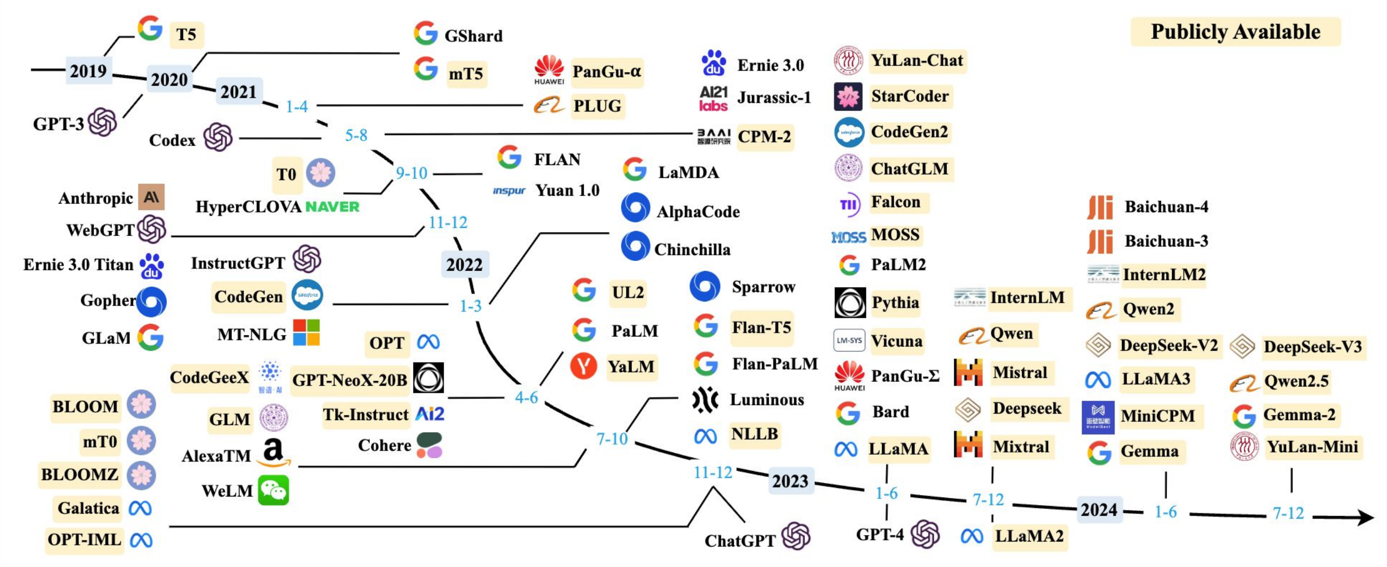
LLM

-based Generative Recommendation

The Rise of Large Language Models

Transformer

2017



O3, R1...

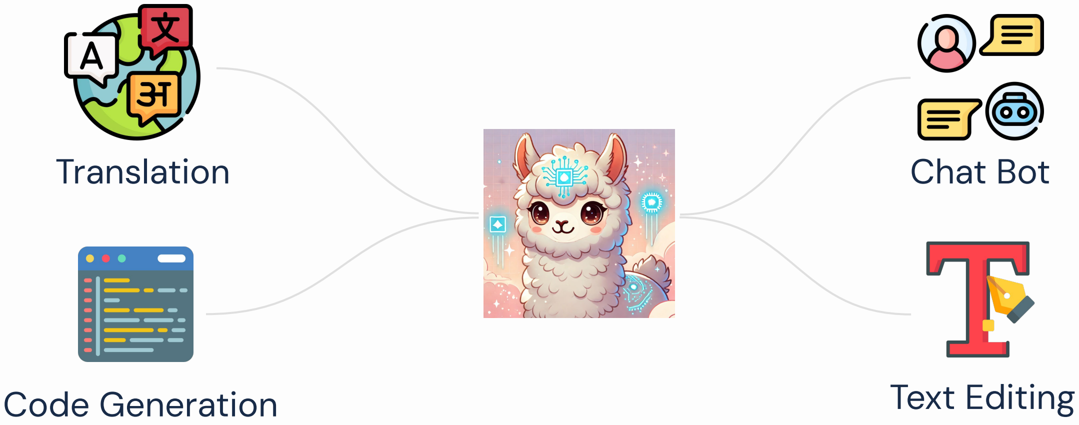
2025

LLMs are developing so fast recently...

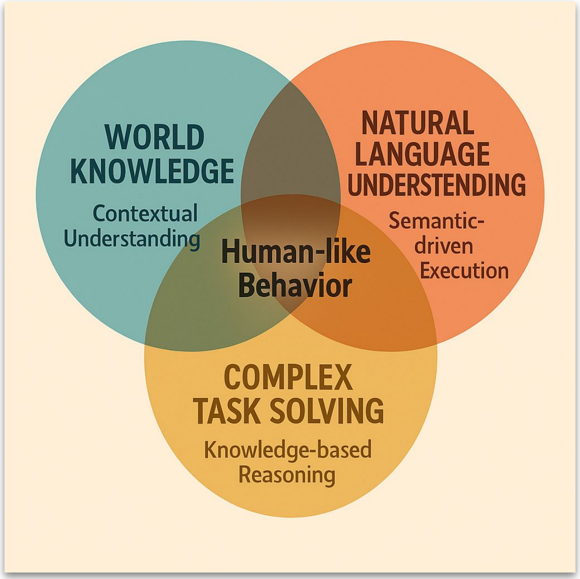
arXiv.2303.18223

Large Language Models

LLMs are machine learning models that can perform a variety of natural language processing (NLP) tasks



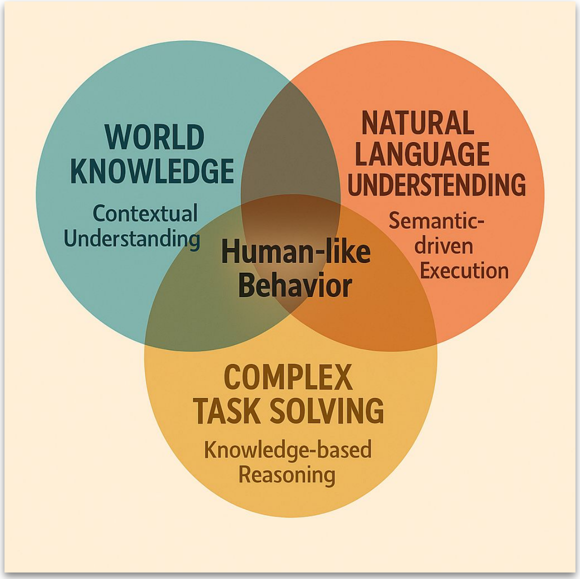
Large Language Models



Key features of LLMs:

- World knowledge.
- Natural language understanding.
- Human-like behavior.

Large Language Models



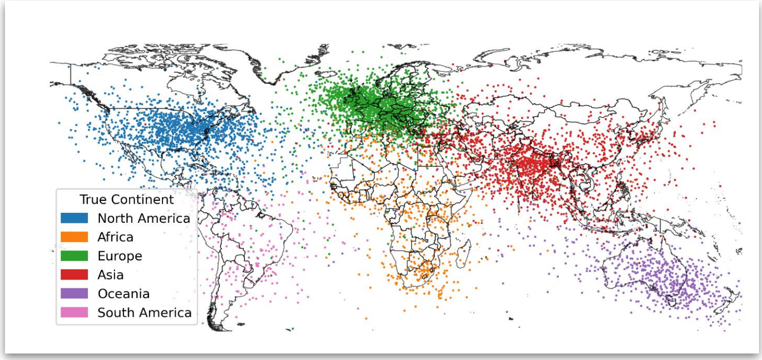
Key features of LLMs:

- World knowledge.
- Natural language understanding.
- Human-like behavior.

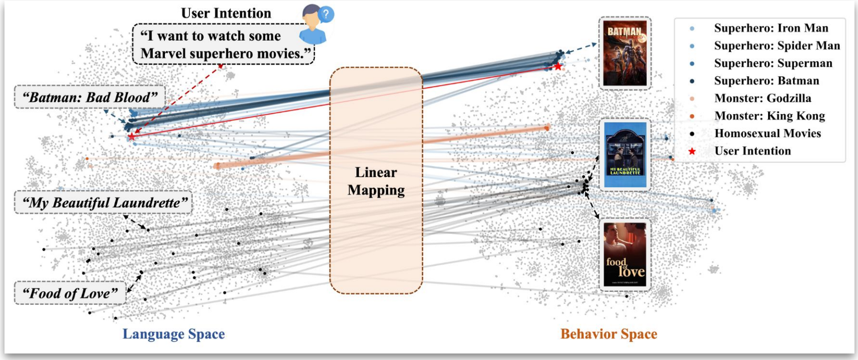
How can these features benefit recommender systems?

Benefits of LLMs for Recommendation

(1) **World knowledge** – from pretraining



In space



In recommendation

Gurnee et al. Language Models Represent Space and Time. ICLR 2024.
Sheng et al. Language Representations Can be What Recommenders Need: Findings and Potentials. ICLR 2025.

Benefits of LLMs for Recommendation

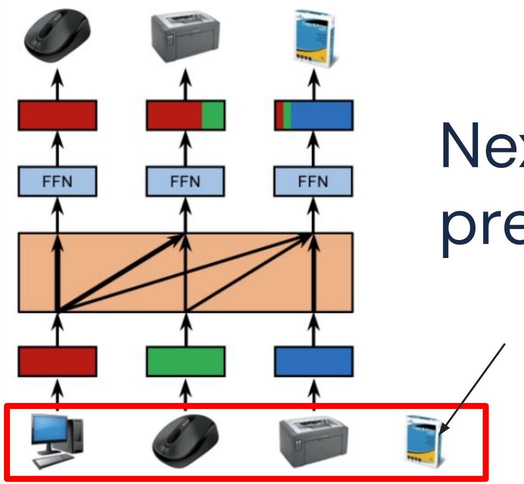
(1) World knowledge

LLM as sequential recommender

-> Alleviating the data sparsity of ID-based interactions in recommendation

Benefits of LLMs for Recommendation

(1) World knowledge



Next ID prediction

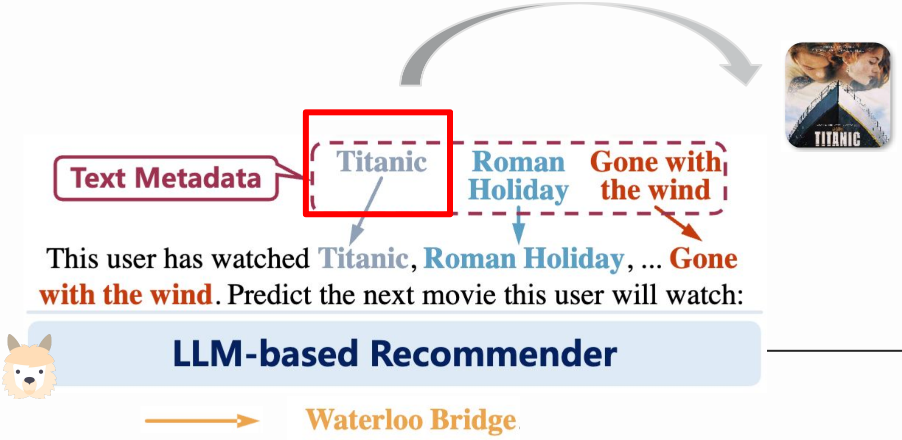
Item IDs

ID-based item modeling lack semantic meanings

Example: SASRec [*ICDM'18*]

Benefits of LLMs for Recommendation

(1) World knowledge

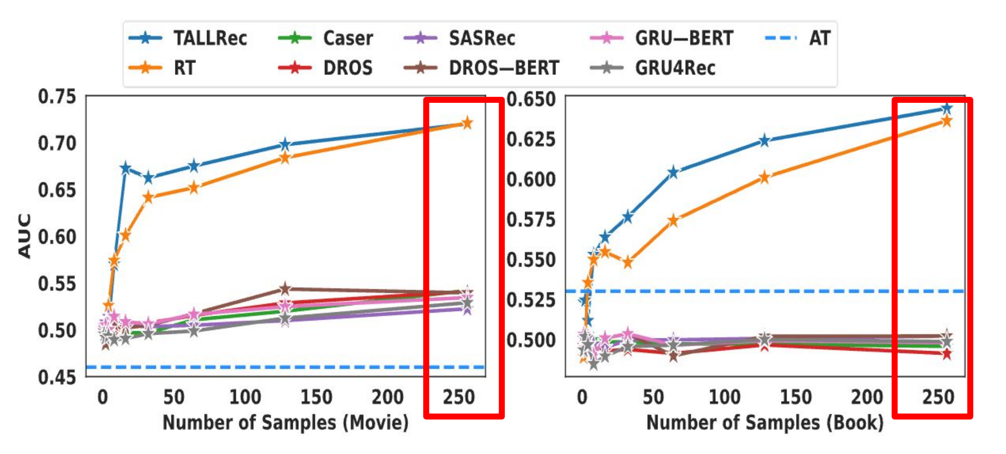


Titanic is a 1997 epic romance and disaster film directed by James Cameron, telling the tragic love story between Jack and Rose aboard the ill-fated RMS Titanic. It blends historical events with fictional drama, becoming one of the most iconic and emotionally powerful films of all time.

Abundant prior knowledge about items

Benefits of LLMs for Recommendation

(1) World knowledge



Few data -> a good recommender

Benefits of LLMs for Recommendation

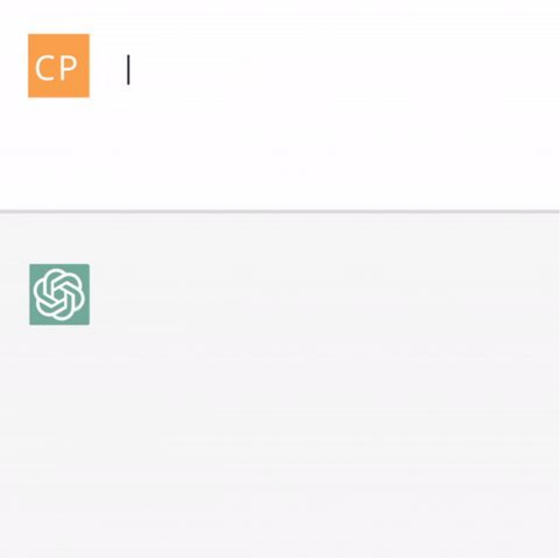
(1) World knowledge

LLM as sequential recommender

Lower data requirement
Cross-domain ability
Cold-start ability
...

Benefits of LLMs for Recommendation

(2) Natural language understanding & generation



LLMs can interact with users fluently

Benefits of LLMs for Recommendation

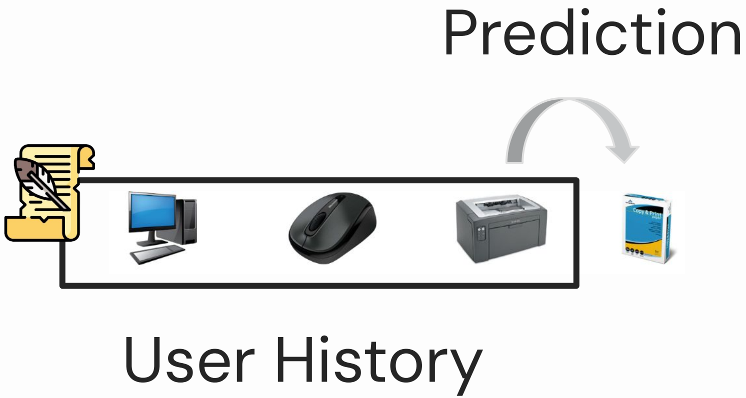
(2) Natural language understanding & generation

LLM as conversational recommender

-> Towards more interactive recommender systems

Benefits of LLMs for Recommendation

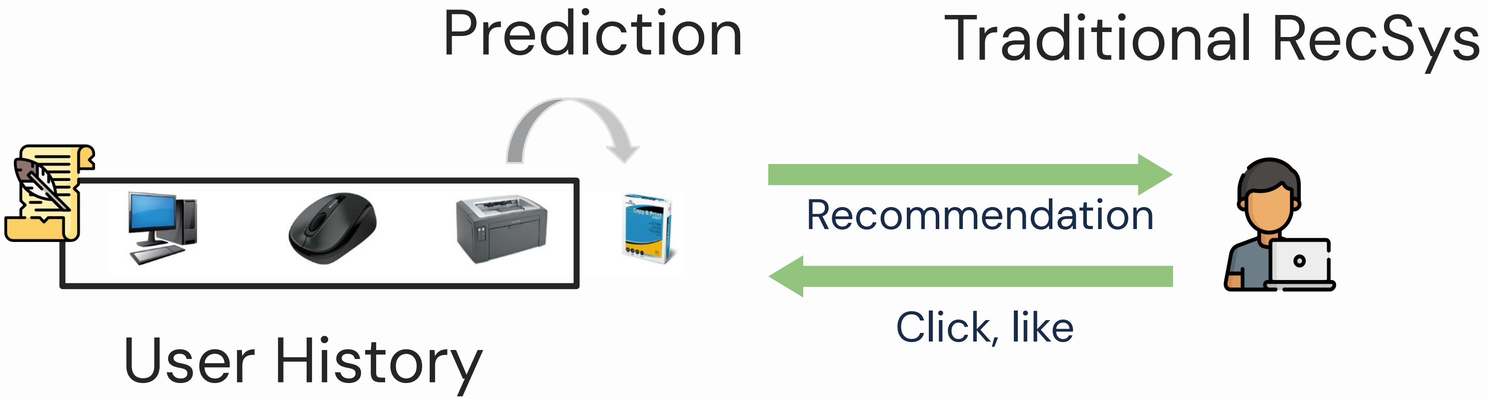
(2) Natural language understanding & generation



Traditional RecSys

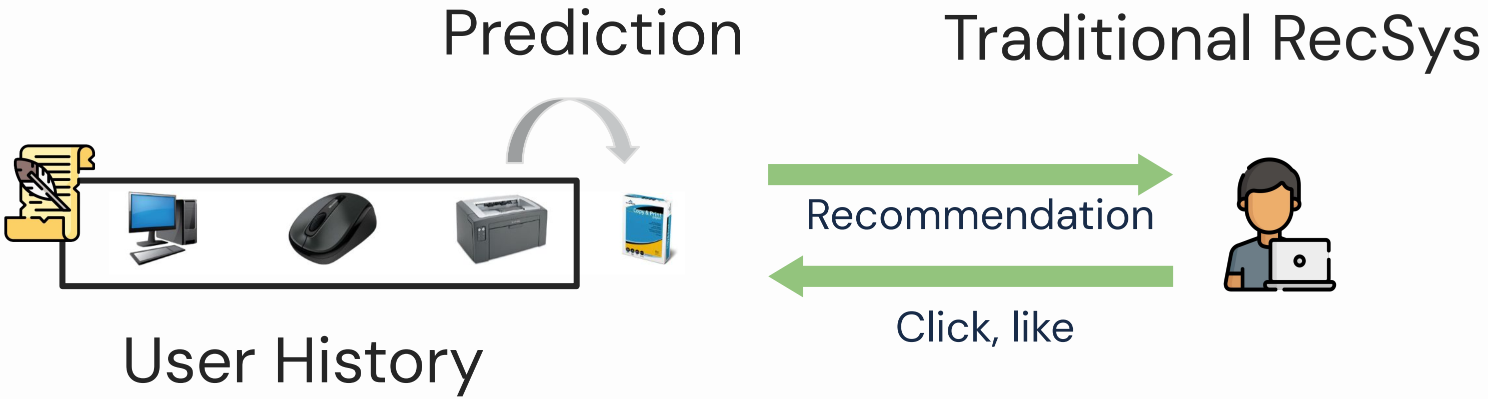
Benefits of LLMs for Recommendation

(2) Natural language understanding & generation



Benefits of LLMs for Recommendation

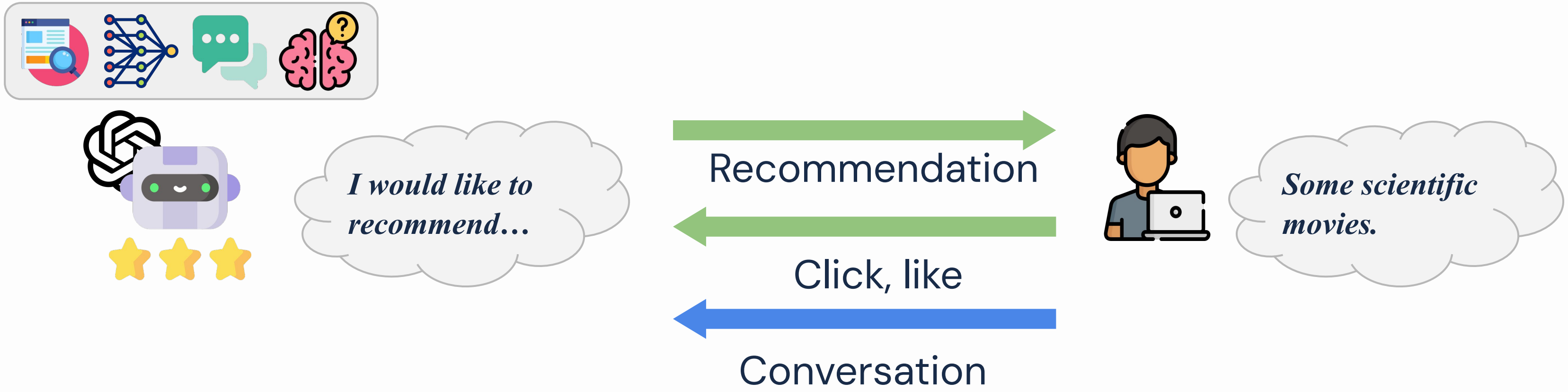
(2) Natural language understanding & generation



Passive recommendation!

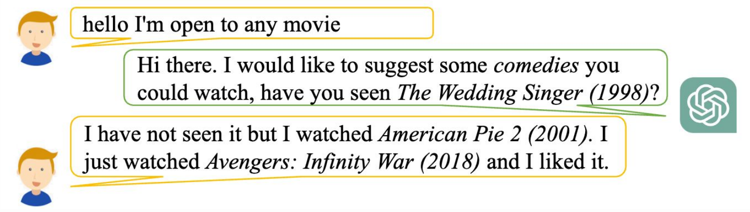
Benefits of LLMs for Recommendation

(2) Natural language understanding & generation



Benefits of LLMs for Recommendation

(2) Natural language understanding & generation



LLM as conversational recommender

Interactive
User-friendly
More accurate
...

Benefits of LLMs for Recommendation

(3) Human-like behavior

Benefits of LLMs for Recommendation

(3) Human-like behavior

Generative Agents can (mostly) simulate human behaviors

- Cooperation
- Organization

Park et al. Generative Agents: Interactive Simulacra of Human Behavior. UIST 2023

Benefits of LLMs for Recommendation

(3) Human-like behavior

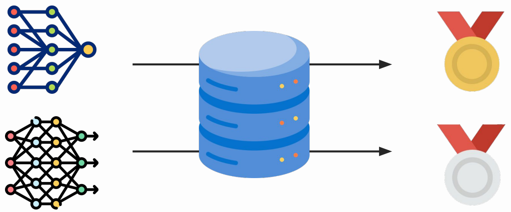
LLM as user simulator

-> Simulating user behaviors for evaluating recommenders.

Benefits of LLMs for Recommendation

(3) Human-like behavior

Offline recommender evaluation

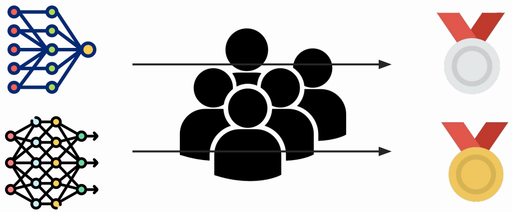


Inaccurate, but affordable

Benefits of LLMs for Recommendation

(3) Human-like behavior

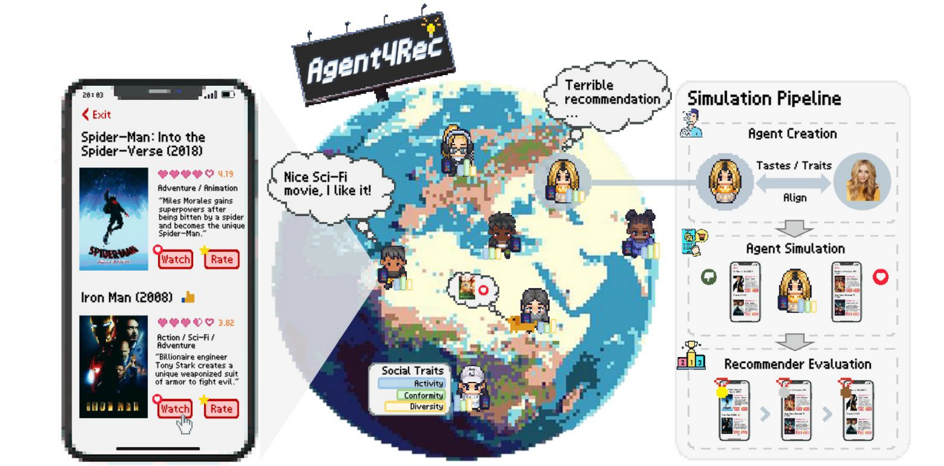
Online recommender evaluation



Accurate, but costly

Benefits of LLMs for Recommendation

(3) Human-like behavior



[LLM as user simulator]

Faithful
Affordable
Controllable
...

Zhang et al. On Generative Agents in Recommendation. SIGIR 2024

Part 1: LLM as Sequential Recommender

(i) **Early efforts**: Pretrained LLMs for recommendation;

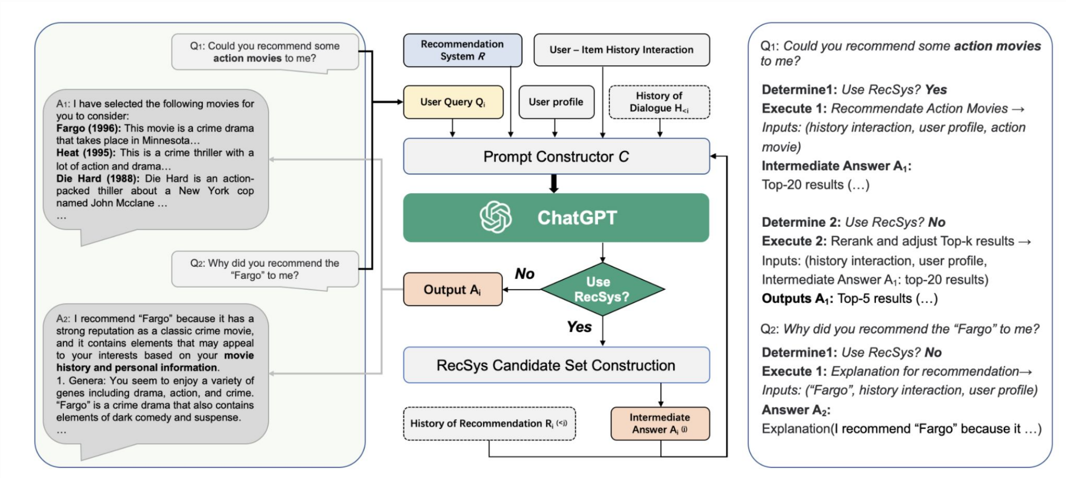
Early efforts

- Directly use freezed LLMs (e.g., GPT 4) for recommendation.

Early efforts

Prompt Engineering + In-Context Learning (ChatRec)

Key idea: LLMs as the recsys controller

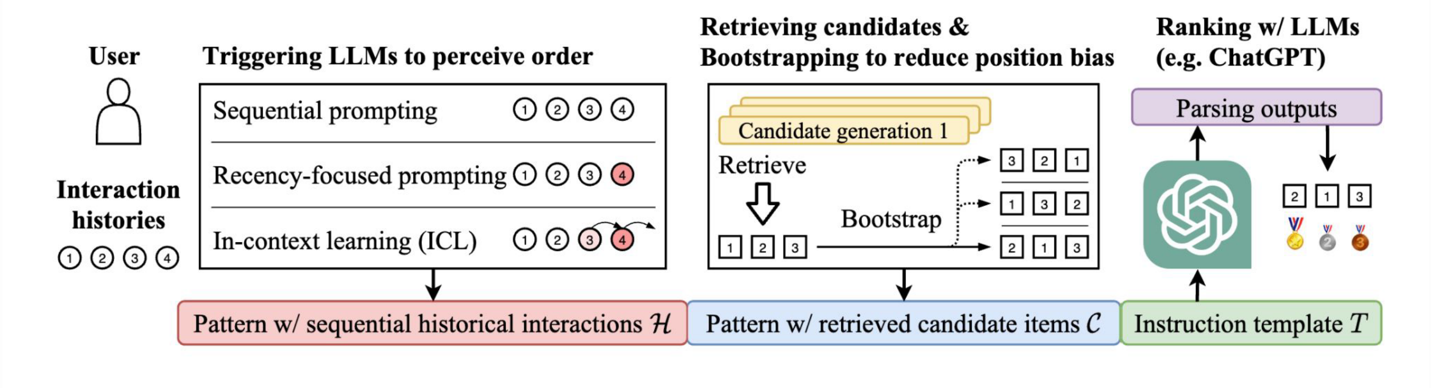


Gao et al. Chat-REC: Towards Interactive and Explainable LLMs-Augmented Recommender System. arXiv 2303.14524.

Early efforts

Prompt Engineering + In-Context Learning (LLMRank)

Key idea: LLMs as the reranker



Early efforts

- Directly use freezed LLMs (e.g., GPT 4) for recommendation.
- A performance gap compared to traditional recommenders exists.

Early efforts

Sub-optimal performance comparing to SASRec!

Performance of LLMRank

	Method	ML-1M N@1	ML-1M N@5	ML-1M N@10	ML-1M N@20	Games N@1	Games N@5	Games N@10	Games N@20
full	Pop	0.08	1.20	4.13	5.79	0.13	1.00	2.27	2.62
full	BPRMF [49]	0.26	1.69	4.41	6.04	0.55	1.98	**2.96**	**3.19**
full	SASRec [33]	**3.76**	**9.79**	**10.45**	**10.56**	**1.33**	**3.55**	**4.02**	**4.11**
zero-shot	BM25 [50]	0.26	0.87	2.32	5.28	0.18	1.07	1.80	2.55
zero-shot	UniSRec [30]	0.88	3.46	5.30	6.92	0.00	1.86	2.03	2.31
zero-shot	VQ-Rec [29]	0.20	1.60	3.29	5.73	0.20	1.21	1.91	2.64
zero-shot	Ours	**1.74**	**5.22**	**6.91**	**7.90**	**0.90**	**2.26**	2.80	3.08

Hou et al. Large Language Models are Zero-Shot Rankers for Recommender Systems. ECIR 2024.

Early efforts

Sub-optimal performance comparing to SASRec!

Aligning LLMs for recommendation tasks is necessary!

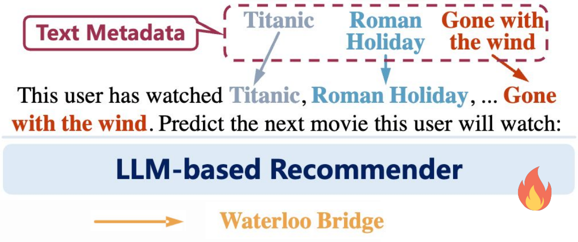
	SASRec [illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
zero-shot	BM25 [50]	0.26	0.87	2.32	5.28	0.18	1.07	1.80	2.55
	UniSRec [30]	0.88	3.46	5.30	6.92	0.00	1.86	2.03	2.31
	VQ-Rec [29]	0.20	1.60	3.29	5.73	0.20	1.21	1.91	2.64
	Ours	**1.74**	**5.22**	**6.91**	**7.90**	**0.90**	**2.26**	2.80	3.08

Hou et al. Large Language Models are Zero-Shot Rankers for Recommender Systems. ECIR 2024.

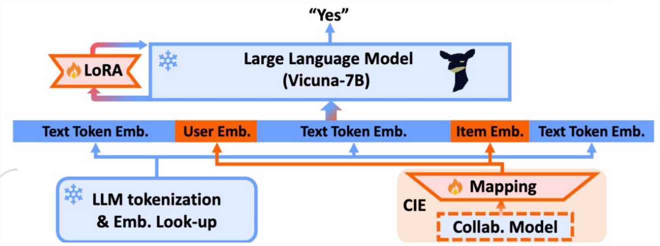
Part 1: LLM as Sequential Recommender

(i) Early efforts: Pretrained LLMs for recommendation;
(ii) **Aligning** LLMs for recommendation;

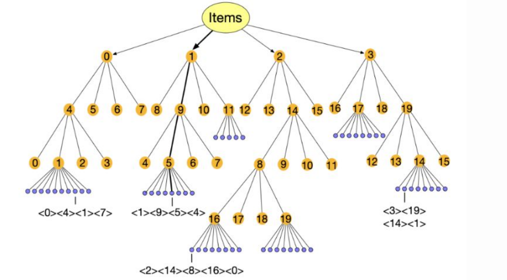
Aligning LLMs for recommendation



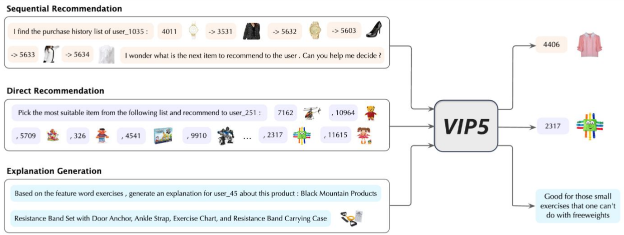
Pure text-based



+ Collaborative embeddings

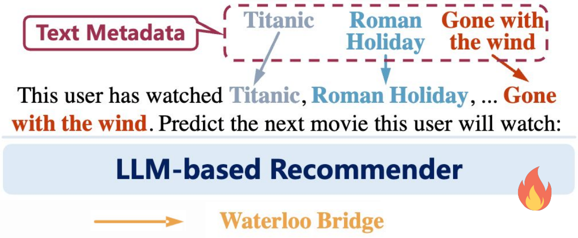


+ External item tokens

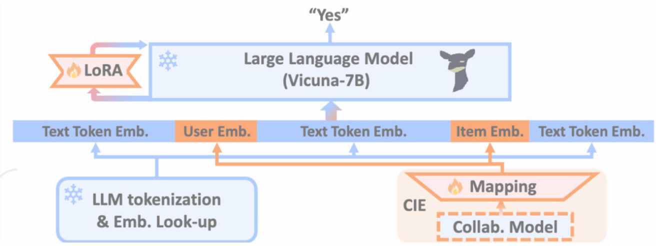


+ Multimodal information

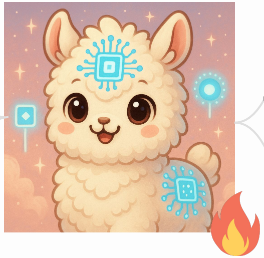
Aligning LLMs for recommendation

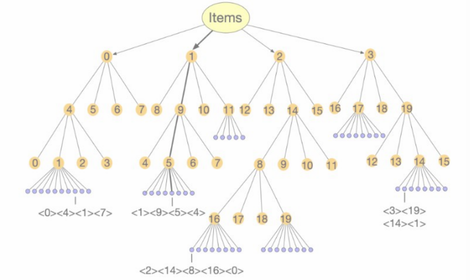


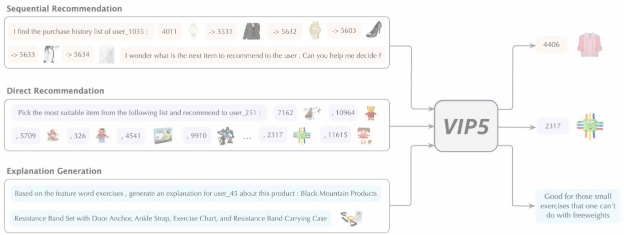
Pure text-based



+ Collaborative embeddings





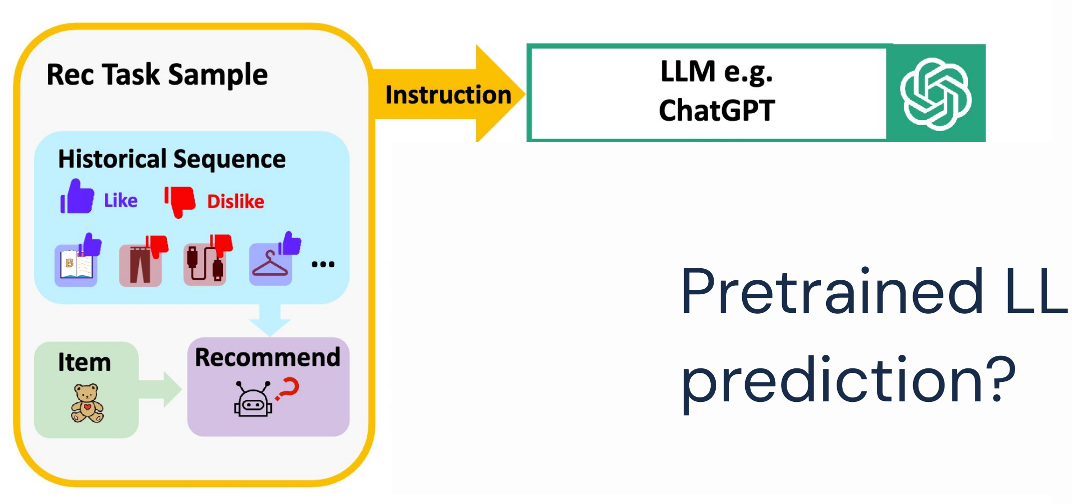


+ External item tokens

+ Multimodal information

Aligning LLMs for recommendation

(1) Pure text-based (TALLRec)

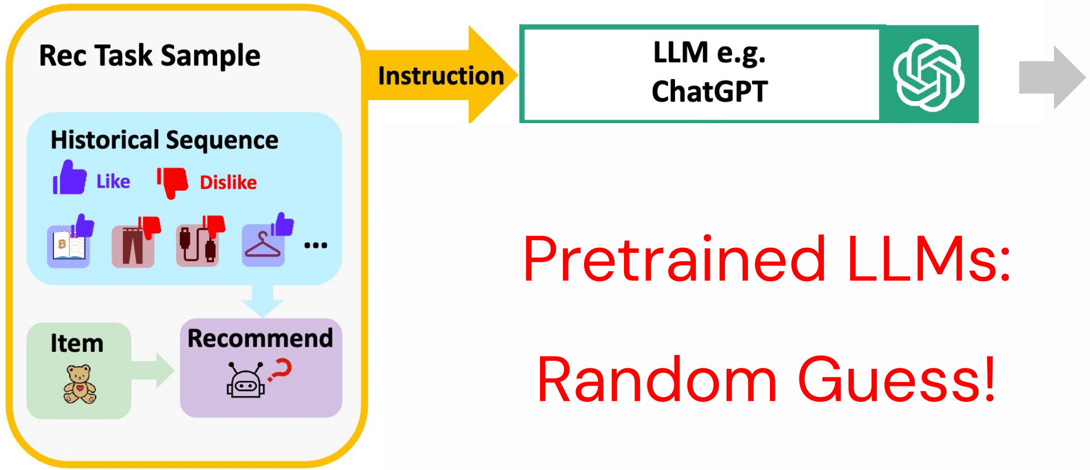


Pretrained LLMs for CTR prediction?

Bao et al. TALLRec: An Effective and Efficient Tuning Framework to Align Large Language Model with Recommendation. RecSys 2023.

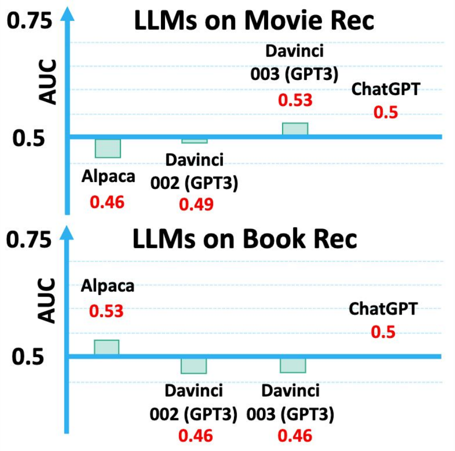
Aligning LLMs for recommendation

(1) Pure text-based (TALLRec)



Pretrained LLMs:

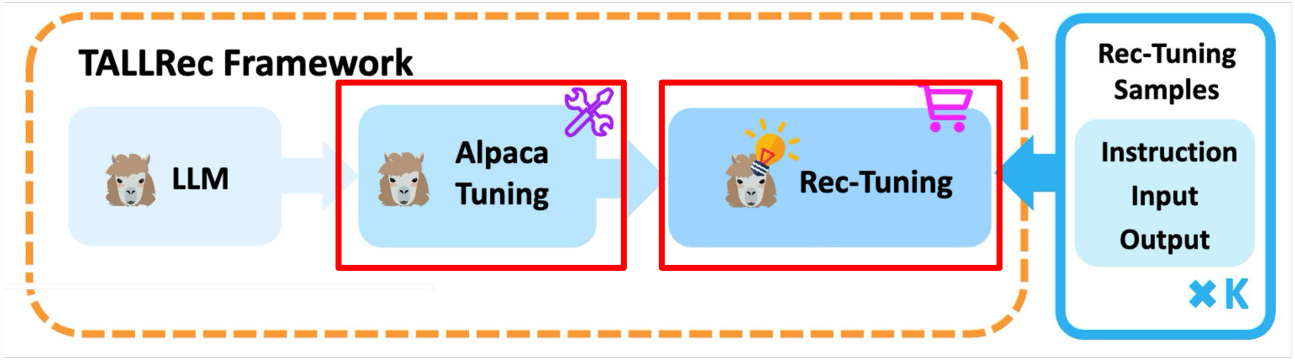
Random Guess!



Bao et al. TALLRec: An Effective and Efficient Tuning Framework to Align Large Language Model with Recommendation. RecSys 2023.

Aligning LLMs for recommendation

(1) Pure text-based (TALLRec)

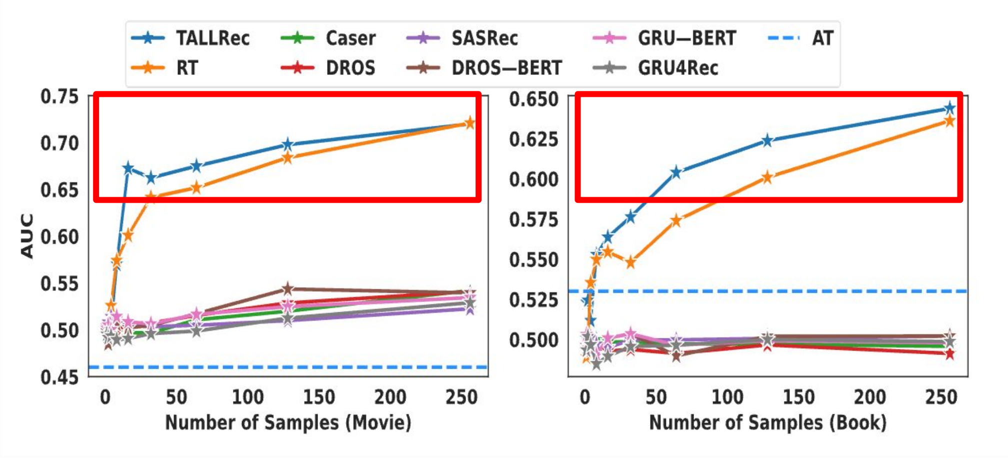


General task alignment -> Recommendation alignment

Bao et al. TALLRec: An Effective and Efficient Tuning Framework to Align Large Language Model with Recommendation. RecSys 2023.

Aligning LLMs for recommendation

(1) Pure text-based (TALLRec)

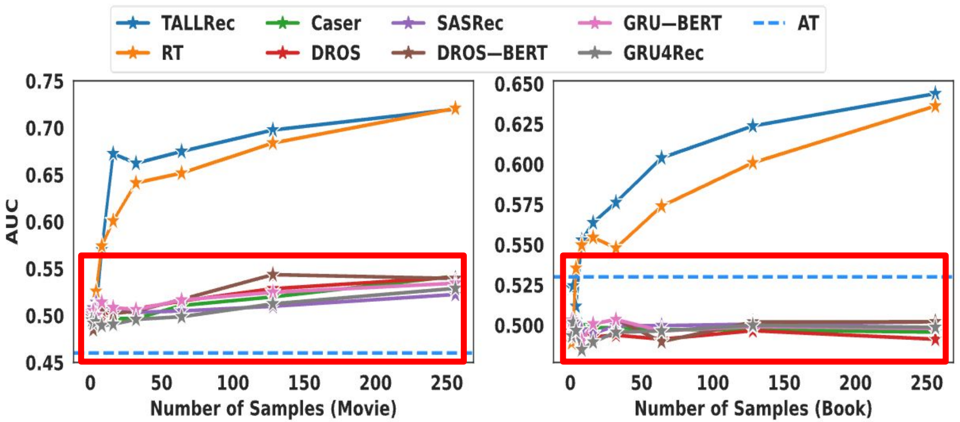


Few training data -> Huge improvements

Bao et al. TALLRec: An Effective and Efficient Tuning Framework to Align Large Language Model with Recommendation. RecSys 2023.

Aligning LLMs for recommendation

(1) Pure text-based (TALLRec)



Traditional recommenders: suffer from too-sparse supervision signals

Aligning LLMs for recommendation

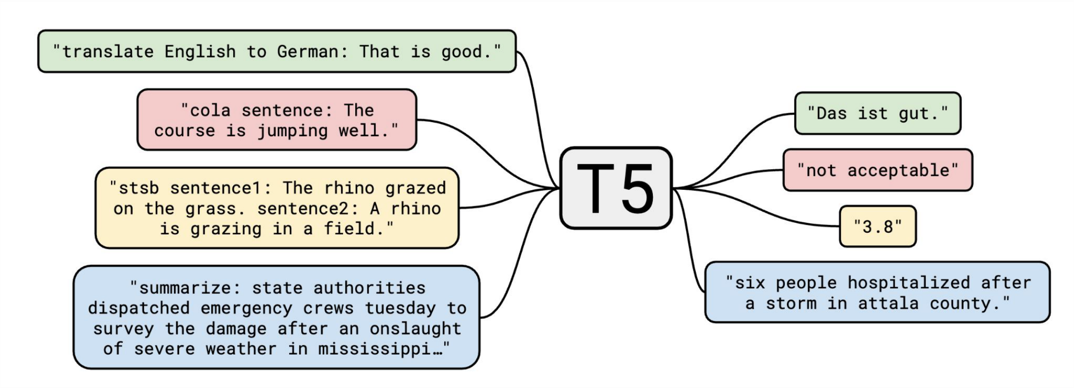
(1) Pure text-based (TALLRec)

Cross-domain generalization

Bao et al. TALLRec: An Effective and Efficient Tuning Framework to Align Large Language Model with Recommendation. RecSys 2023.

Aligning LLMs for recommendation

(1) Pure text-based – Multiple rec taks

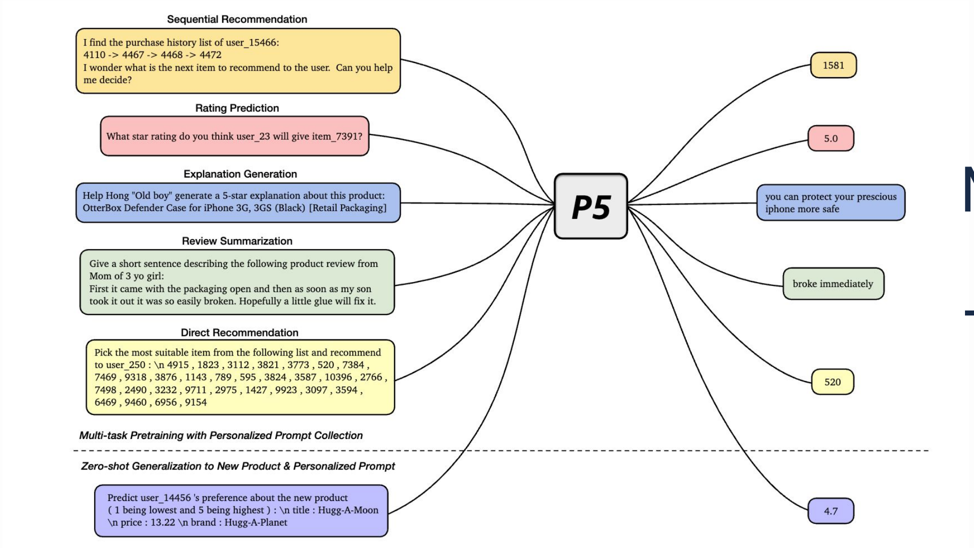


Unified language modeling in NLP

Raffel et al. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. JMLR 2020.

Aligning LLMs for recommendation

(1) Pure text-based – Multiple rec taks



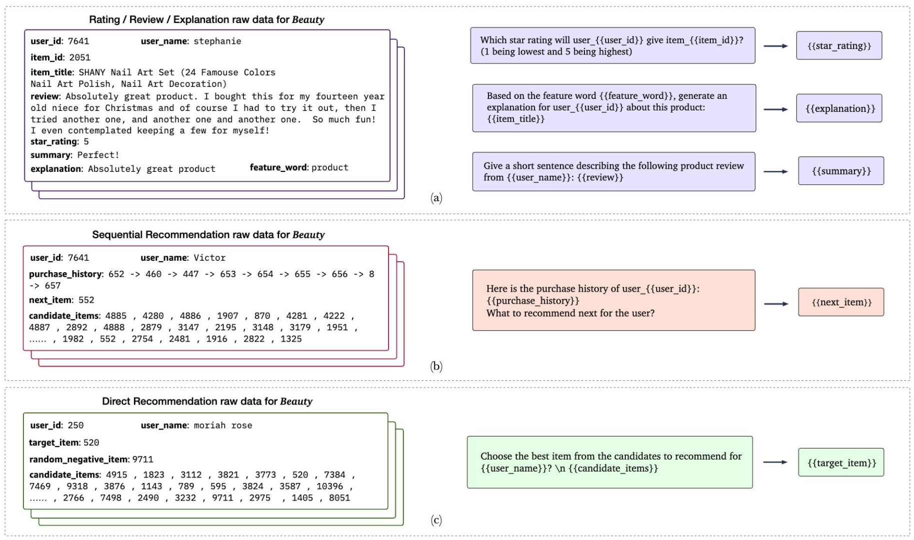
Multi-task alignment (P5)

-> general recommender

Geng et al. Recommendation as Language Processing (RLP): A Unified Pretrain, Personalized Prompt & Predict Paradigm (P5). RecSys 2022.

Aligning LLMs for recommendation

(1) Pure text-based – Multiple rec taks



Training on different task prompts -> multiple recommendation abilities.

Geng et al. Recommendation as Language Processing (RLP): A Unified Pretrain, Personalized Prompt & Predict Paradigm (P5). RecSys 2022.

Aligning LLMs for recommendation

(1) Pure text-based – Multiple rec taks

Table 6: Performance comparison on review summarization (%).

Methods	Sports				Beauty				Toys			
	BLUE2	ROUGE1	ROUGE2	ROUGEL	BLUE2	ROUGE1	ROUGE2	ROUGEL	BLUE2	ROUGE1	ROUGE2	ROUGEL
T0 (4-1)	2.1581	2.2695	0.5694	1.6221	1.2871	1.2750	0.3904	0.9592	2.2296	2.4671	0.6482	1.8424
GPT-2 (4-1)	0.7779	4.4534	1.0033	1.9236	0.5879	3.3844	0.6756	1.3956	0.6221	3.7149	0.6629	1.4813
P5-S (4-1)	2.4962	11.6701	2.7187	10.4819	**2.1225**	**8.4205**	**1.6676**	**7.5476**	**2.4752**	**9.4200**	**1.5975**	**8.2618**
P5-B (4-1)	**2.6910**	**12.0314**	**3.2921**	**10.7274**	1.9325	8.2909	1.4321	7.4000	1.7833	8.7222	1.3210	7.6134

Table 7: Performance comparison on direct recommendation.

Methods	Sports					Beauty					Toys				
	HR@1	HR@5	NDCG@5	HR@10	NDCG@10	HR@1	HR@5	NDCG@5	HR@10	NDCG@10	HR@1	HR@5	NDCG@5	HR@10	NDCG@10
BPR-MF	0.0314	0.1404	0.0848	0.2563	0.1220	0.0311	0.1426	0.0857	0.2573	0.1224	0.0233	0.1066	0.0641	0.2003	0.0940
BPR-MLP	0.0351	0.1520	0.0927	0.2671	0.1296	0.0317	0.1392	0.0848	0.2542	0.1215	0.0252	0.1142	0.0688	0.2077	0.0988
SimpleX	0.0331	**0.2362**	**0.1505**	0.3290	0.1800	0.0325	0.2247	0.1441	0.3090	0.1711	0.0268	**0.1958**	**0.1244**	**0.2662**	**0.1469**
P5-S (5-1)	0.0638	0.2096	0.1375	0.3143	0.1711	0.0600	0.2021	0.1316	0.3121	0.1670	0.0405	0.1538	0.0969	0.2405	0.1248
P5-B (5-1)	0.0245	0.0816	0.0529	0.1384	0.0711	0.0224	0.0904	0.0559	0.1593	0.0780	0.0187	0.0827	0.0500	0.1543	0.0729
P5-S (5-4)	0.0701	0.2241	0.1483	**0.3313**	**0.1827**	**0.0862**	**0.2448**	**0.1673**	**0.3441**	**0.1993**	0.0413	0.1411	0.0916	0.2227	0.1178
P5-B (5-4)	0.0299	0.1026	0.0665	0.1708	0.0883	0.0506	0.1557	0.1033	0.2350	0.1287	0.0435	0.1316	0.0882	0.2000	0.1102
P5-S (5-5)	0.0574	0.1503	0.1050	0.2207	0.1276	0.0601	0.1611	0.1117	0.2370	0.1360	0.0440	0.1282	0.0865	0.2011	0.1098
P5-B (5-5)	0.0641	0.1794	0.1229	0.2598	0.1488	0.0588	0.1573	0.1089	0.2325	0.1330	0.0386	0.1122	0.0756	0.1807	0.0975
P5-S (5-8)	0.0567	0.1514	0.1049	0.2196	0.1269	0.0571	0.1566	0.1078	0.2317	0.1318	**0.0451**	0.1322	0.0889	0.2023	0.1114
P5-B (5-8)	**0.0726**	0.1955	0.1355	0.2802	0.1627	0.0608	0.1564	0.1096	0.2300	0.1332	0.0389	0.1147	0.0767	0.1863	0.0997

Table 3: Performance comparison on sequential recommendation.

Methods	Sports				Beauty				Toys			
	HR@5	NDCG@5	HR@10	NDCG@10	HR@5	NDCG@5	HR@10	NDCG@10	HR@5	NDCG@5	HR@10	NDCG@10
Caser	0.0116	0.0072	0.0194	0.0097	0.0205	0.0131	0.0347	0.0176	0.0166	0.0107	0.0270	0.0141
HGN	0.0189	0.0120	0.0313	0.0159	0.0325	0.0206	0.0512	0.0266	0.0321	0.0221	0.0497	0.0277
GRU4Rec	0.0129	0.0086	0.0204	0.0110	0.0164	0.0099	0.0283	0.0137	0.0097	0.0059	0.0176	0.0084
BERT4Rec	0.0115	0.0075	0.0191	0.0099	0.0203	0.0124	0.0347	0.0170	0.0116	0.0071	0.0203	0.0099
FDSA	0.0182	0.0122	0.0288	0.0156	0.0267	0.0163	0.0407	0.0208	0.0228	0.0140	0.0381	0.0189
SASRec	0.0233	0.0154	0.0350	0.0192	0.0387	0.0249	0.0605	0.0318	0.0463	0.0306	0.0675	0.0374
S^3-Rec	0.0251	0.0161	0.0385	0.0204	0.0387	0.0244	0.0647	0.0327	0.0443	0.0294	0.0700	0.0376
P5-S (2-3)	0.0272	0.0169	0.0361	0.0198	0.0503	0.0370	0.0659	0.0421	**0.0648**	**0.0567**	**0.0709**	**0.0587**
P5-B (2-3)	0.0364	0.0296	0.0431	0.0318	**0.0508**	**0.0379**	**0.0664**	**0.0429**	0.0608	0.0507	0.0688	0.0534
P5-S (2-13)	0.0258	0.0159	0.0346	0.0188	0.0490	0.0358	0.0646	0.0409	0.0647	0.0566	0.0705	0.0585
P5-B (2-13)	**0.0387**	**0.0312**	**0.0460**	**0.0336**	0.0493	0.0367	0.0645	0.0416	0.0587	0.0486	0.0675	0.0536

Table 4: Performance comparison on explanation generation (%).

Methods	Sports				Beauty				Toys			
	BLUE4	ROUGE1	ROUGE2	ROUGEL	BLUE4	ROUGE1	ROUGE2	ROUGEL	BLUE4	ROUGE1	ROUGE2	ROUGEL
Attn2Seq	0.5305	12.2800	1.2107	9.1312	0.7889	12.6590	1.6820	9.7481	1.6238	13.2245	2.9942	10.7398
NRT	0.4793	11.0723	1.1304	7.6674	0.8295	12.7815	1.8543	9.9477	1.9084	13.5231	3.6708	11.1867
PETER	0.7112	12.8944	1.3283	9.8635	1.1541	14.8497	2.1413	11.4143	1.9861	14.2716	3.6718	11.7010
P5-S (3-3)	**1.0447**	**14.9048**	**2.1297**	**11.1778**	**1.2237**	**17.6938**	**2.2489**	**12.8606**	2.2892	**15.4505**	3.6974	**12.1718**
P5-B (3-3)	1.0407	14.1589	2.1220	10.6096	0.9742	16.4530	1.8858	11.8765	**2.3185**	15.3474	**3.7209**	12.1312
PETER+	**2.4627**	**24.1181**	5.1937	**18.4105**	**3.2606**	25.5541	5.9668	19.7168	**4.7919**	28.3083	9.4520	**22.7017**
P5-S (3-9)	1.4101	23.5619	**5.4196**	17.6245	1.9788	**25.6253**	**6.3678**	**19.9497**	4.1222	**28.4088**	9.5432	22.6064
P5-B (3-9)	1.4689	23.5476	5.3926	17.5852	1.8765	25.1183	6.0764	19.4488	3.8933	27.9916	9.5896	22.2178
P5-S (3-12)	1.3212	23.2474	5.3461	17.3780	1.9425	25.1474	6.0551	19.5601	4.2764	28.1897	9.1327	22.2514
P5-B (3-12)	1.4303	23.3810	5.3239	17.4913	1.9031	25.1763	6.1980	19.5188	3.5861	28.1369	**9.7562**	22.3056

Single LLM -> Effective on various recommendation tasks

Geng et al. Recommendation as Language Processing (RLP): A Unified Pretrain, Personalized Prompt & Predict Paradigm (P5). RecSys 2022.

Aligning LLMs for recommendation

(1) Pure text-based (P5)

Multi-scenario Recommendation: The items the user has recently clicked on are as follows: {USER BEHAVIOR SEQUENCE}. In scenario {SCENE}, please recommend items.

Multi-objective Recommendation: The items the user has recently clicked on are as follows: {USER BEHAVIOR SEQUENCE}. Please find items that the user will {ACTION}.

Long-tail Item Recommendation: The items the user has recently clicked on are as follows: {USER BEHAVIOR SEQUENCE}. Please recommend long-tail items.

Serendipity Recommendation: The items the user has recently clicked on are as follows: {USER BEHAVIOR SEQUENCE}. Please recommend some new item categories.

Long-term Recommendation: The items the user has recently clicked on are as follows: {USER BEHAVIOR SEQUENCE}. Please find items that match the user's long-term interests.

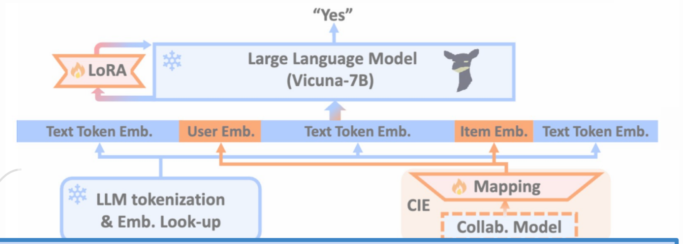
Search Problem: The items the user has recently clicked on are as follows: {USER BEHAVIOR SEQUENCE}. Please recommend items that match {QUERY}.

URM:

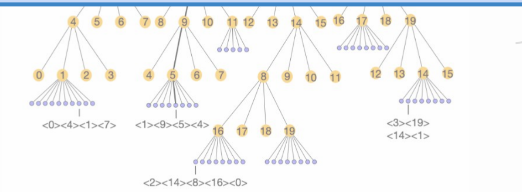
Unify recommendation & search

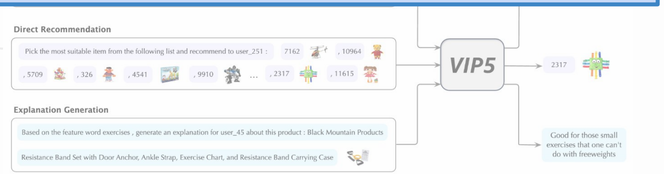
Jiang et al. Large Language Models Are Universal Recommendation Learners. arXiv: 2502.03041.

Aligning LLMs for recommendation



Is textual information enough for alignment?

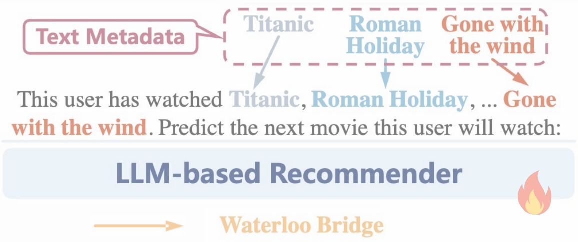




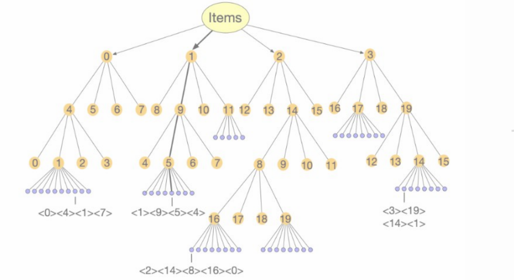
+ External item tokens

+ Multimodal information

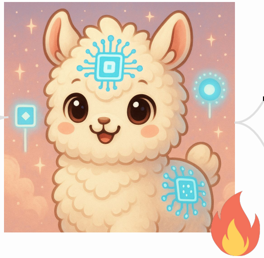
Aligning LLMs for recommendation

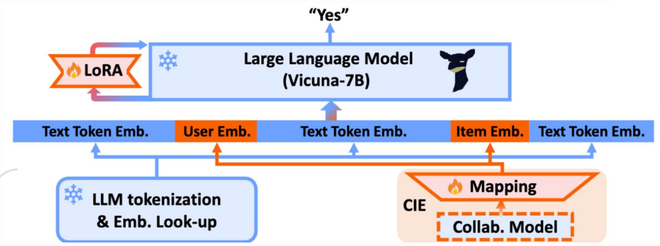


Pure text-based

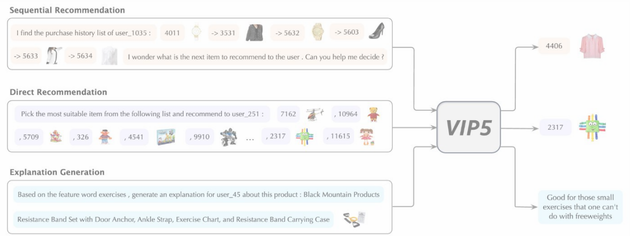


+ External item tokens





+ Collaborative embeddings



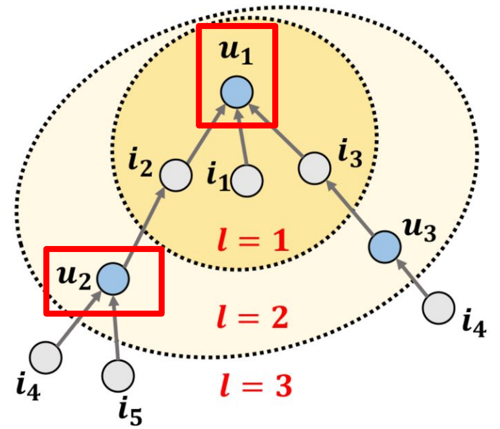
+ Multimodal information

Aligning LLMs for recommendation

(2) + Collaborative embeddings

Motivation:

Language modeling may not capture collaborative information



High-order Connectivity for u_1

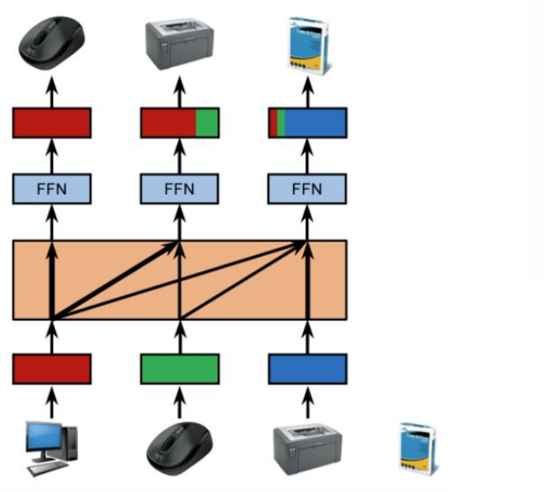
Wang et al. Neural Graph Collaborative Filtering. SIGIR 2019.

Aligning LLMs for recommendation

(2) + Collaborative embeddings

Solution:

Aligning LLMs with embeddings from traditional recommenders

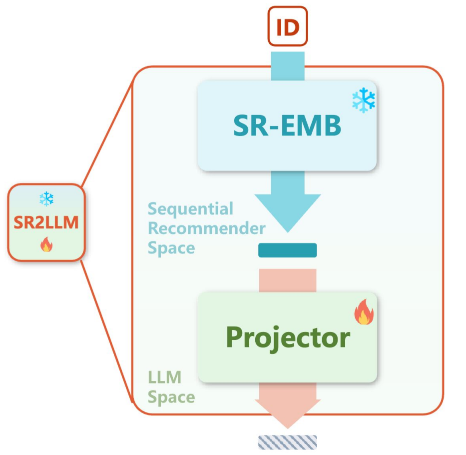


Aligning LLMs for recommendation

(2) + Collaborative embeddings (LLaRA)

+ Pretrained item embeddings

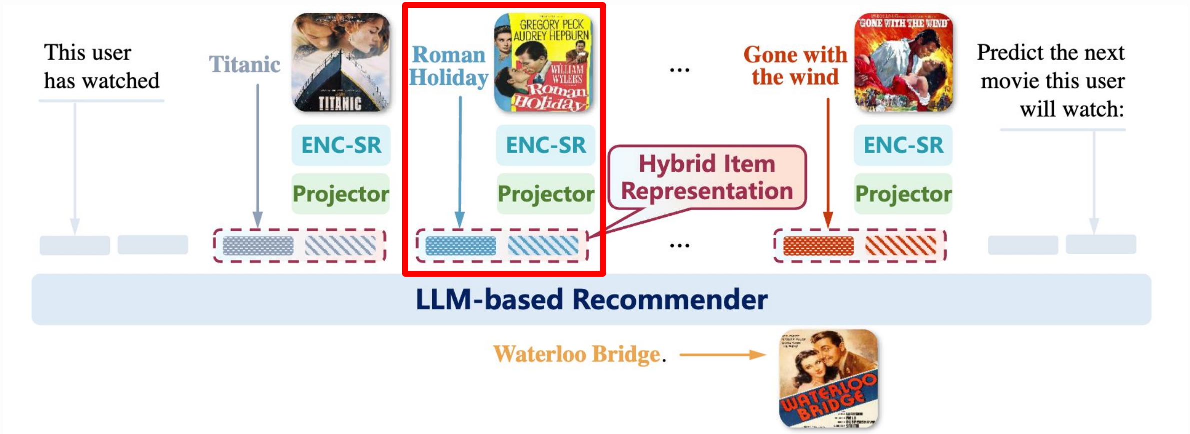
(a) Text-only prompting method.	(b) Hybrid prompting method.
Input: This user has watched Titanic [PH], Roman Holiday [PH], ... Gone with the wind [PH] in the previous. Please predict the next movie this user will watch. The movie title candidates are The Wizard of Oz [PH], Braveheart [PH],…, Waterloo Bridge [PH],… Batman & Robin [PH]. Choose only one movie from the candidates. The answer is	**Input:** This user has watched Titanic [emb_s^{14}], Roman Holiday [emb_s^{20}], ... Gone with the wind [emb_s^{37}] in the previous. Please predict the next movie this user will watch. The movie title candidates are The Wizard of Oz [emb_s^{5}], Braveheart [emb_s^{42}],…, Waterloo Bridge [emb_s^{20}],… Batman & Robin [emb_s^{19}]. Choose only one movie from the candidates. The answer is
Output: Waterloo Bridge.	**Output:** Waterloo Bridge.



Aligning LLMs for recommendation

(2) + Collaborative embeddings (LLaRA)

- Pretrained item embeddings

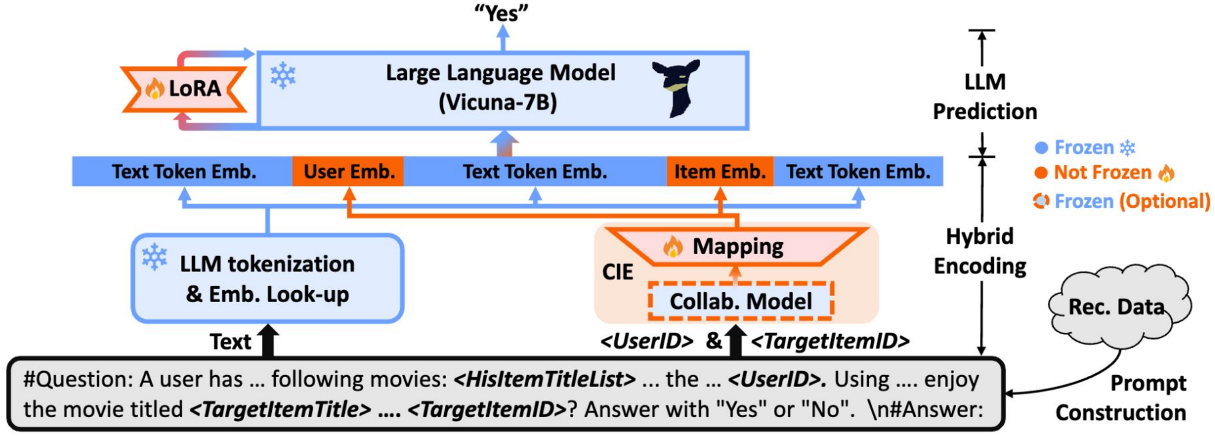


Liao et al. LLaRA: Large Language-Recommendation Assistant. SIGIR 2024.

Aligning LLMs for recommendation

(2) + Collaborative embeddings (CoLLM)

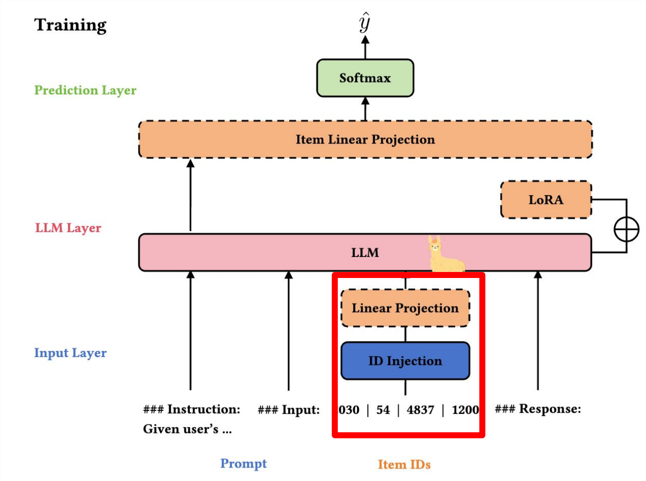
+ Pretrained item embeddings + user embeddings



Zhang et al. CoLLM: Integrating Collaborative Embeddings Into Large Language Models for Recommendation. TKDE 2025.

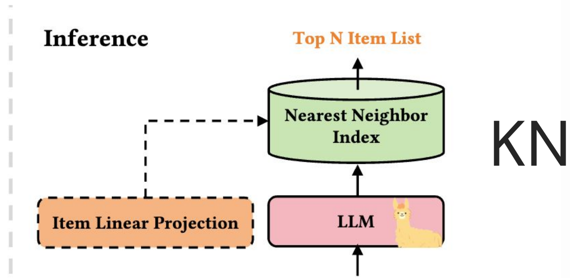
Aligning LLMs for recommendation

(2) + Collaborative embeddings (E4SRec)



Discard text;

Collaborative embeddings only

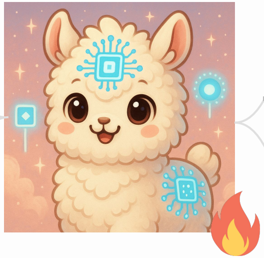


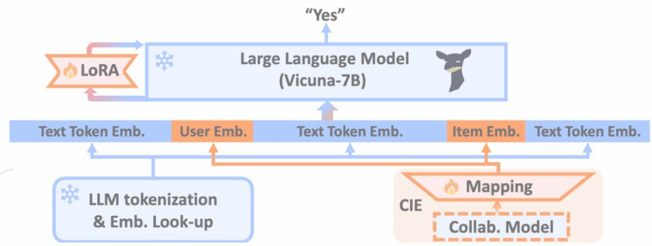
KNN for inference

Li et al. E4SRec: An Elegant Effective Efficient Extensible Solution of Large Language Models for Sequential Recommendation. WWW 2024.

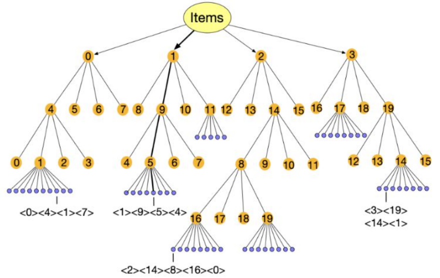
Aligning LLMs for recommendation

Pure text-based

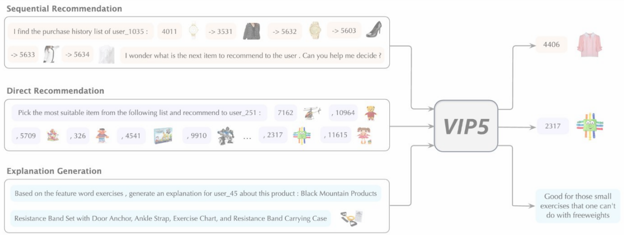




+ Collaborative embeddings



+ External item tokens



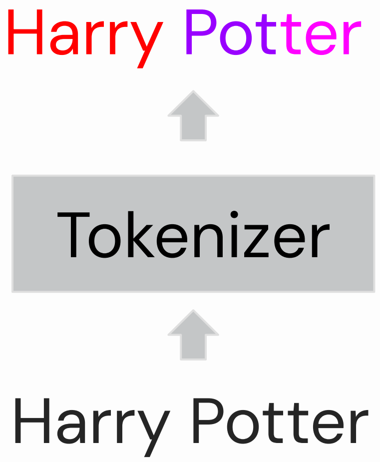
+ Multimodal information

Aligning LLMs for recommendation

(3) + External item tokens

Motivation:

Tokens for language modeling are not optimal for recommendation.



Aligning LLMs for recommendation

(3) + External item tokens

Motivation:

Tokens for language modeling are not optimal for recommendation.

Maybe better?

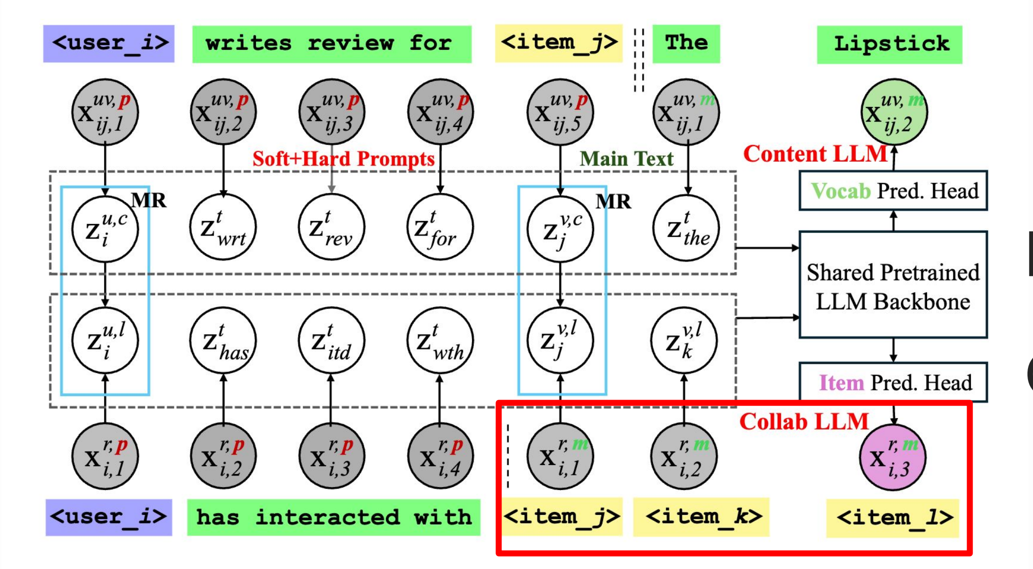
Harry Potter

Tokenizer

Harry Potter

Aligning LLMs for recommendation

(3) + External item tokens (CLLM4Rec)



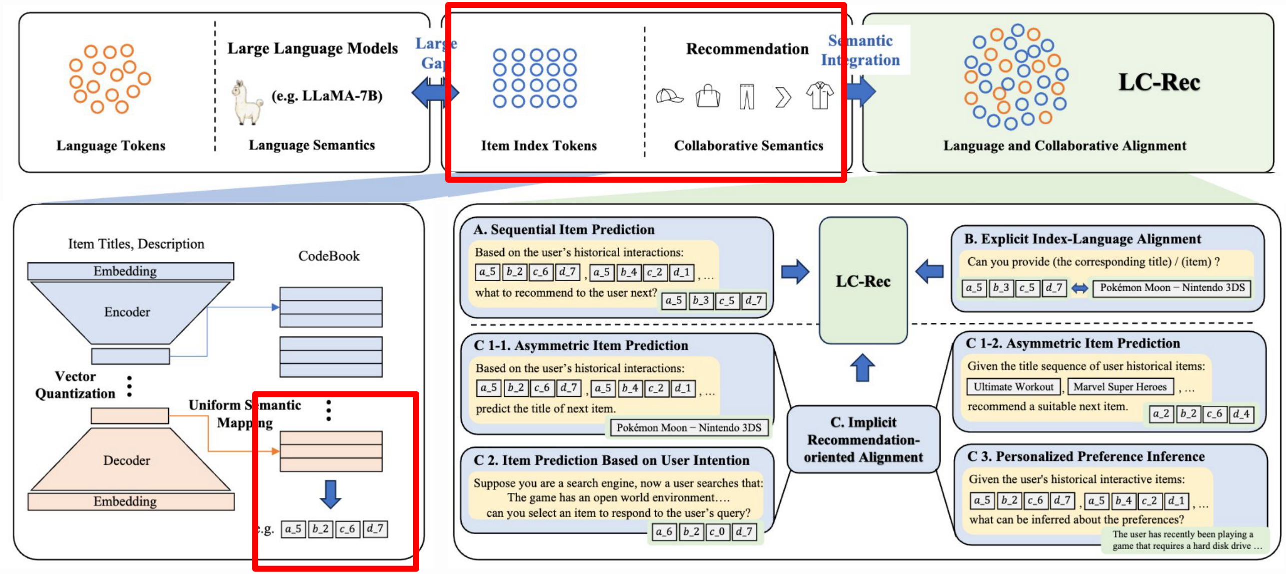
Naive approach:

One ID for each item

Zhu et al. Collaborative Large Language Model for Recommender Systems. WWW 2024.

Aligning LLMs for recommendation

(3) + External item tokens (LC-Rec)



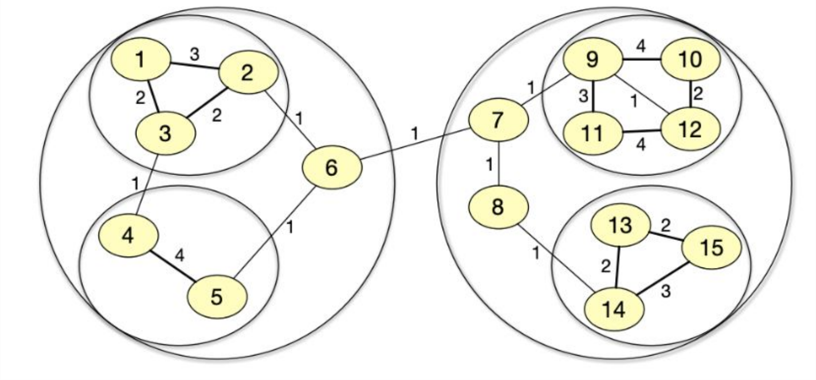
+ Semantic IDs

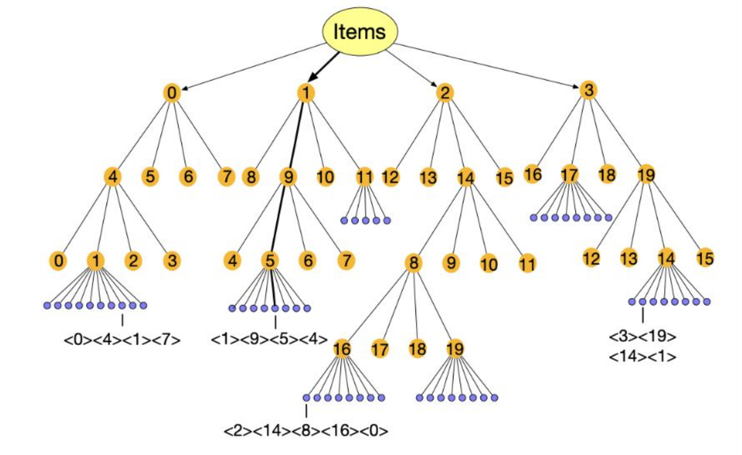
(Similar items have similar IDs)

Zheng et al. Adapting Large Language Models by Integrating Collaborative Semantics for Recommendation. ICDE 2024.

Aligning LLMs for recommendation

(3) + External item tokens

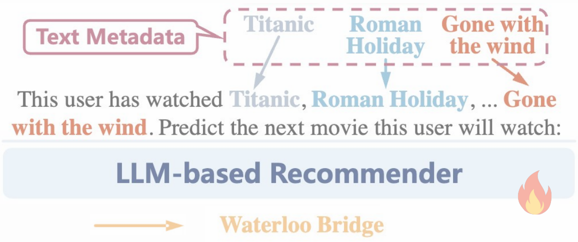




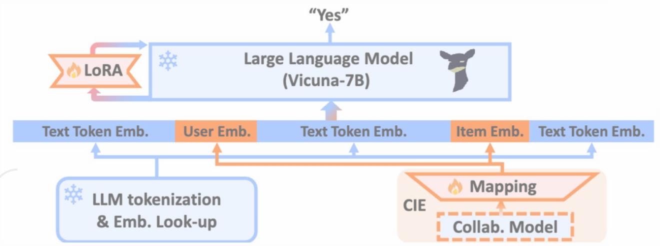
More complicated item tokens design

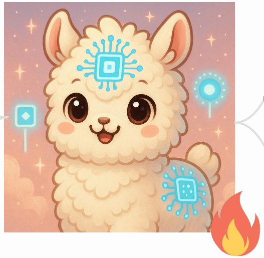
Hua et al. How to Index Item IDs for Recommendation Foundation Models. SIGIR-AP 2023

Aligning LLMs for recommendation

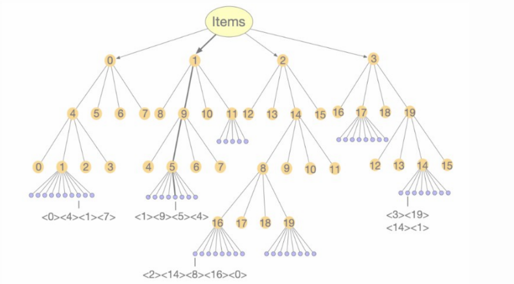


Pure text-based

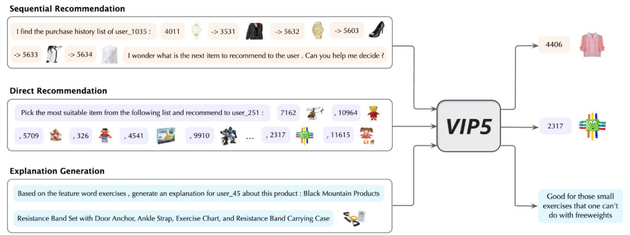




+ Collaborative embeddings



+ External item tokens



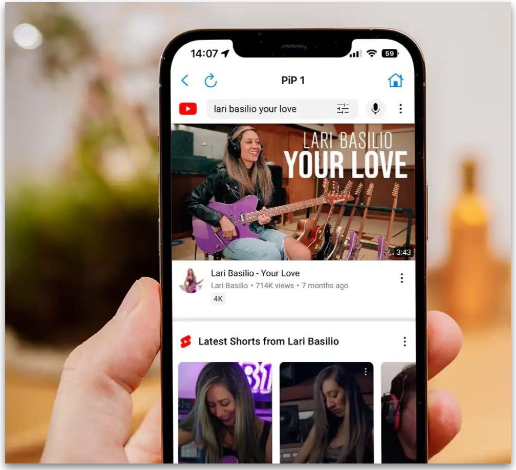
+ Multimodal information

Aligning LLMs for recommendation

(4) + Multimodal information

Motivation:

Human make decisions with multimodal information.

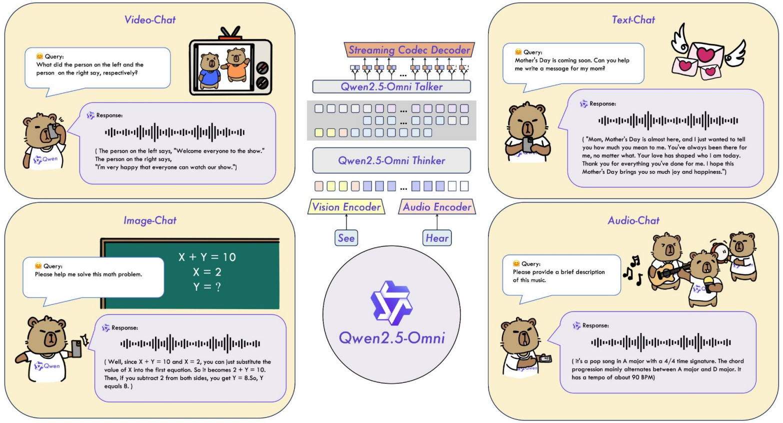


Aligning LLMs for recommendation

(4) + Multimodal information

Motivation:

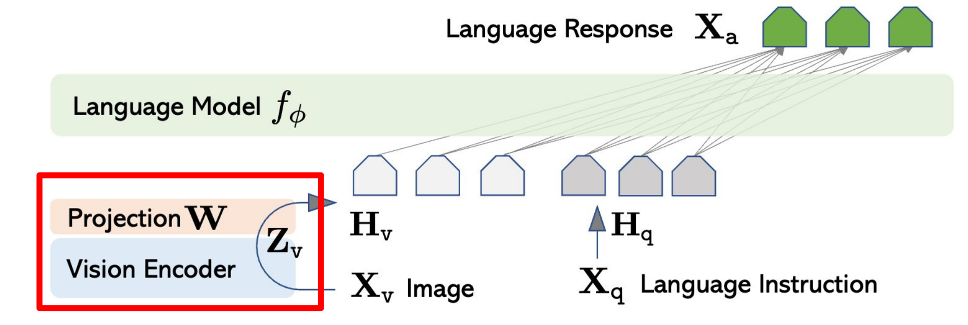
Post-trained LLM can understand multimodal information



Qwen2.5-Omni Technical Report.

Aligning LLMs for recommendation

(4) + Multimodal information

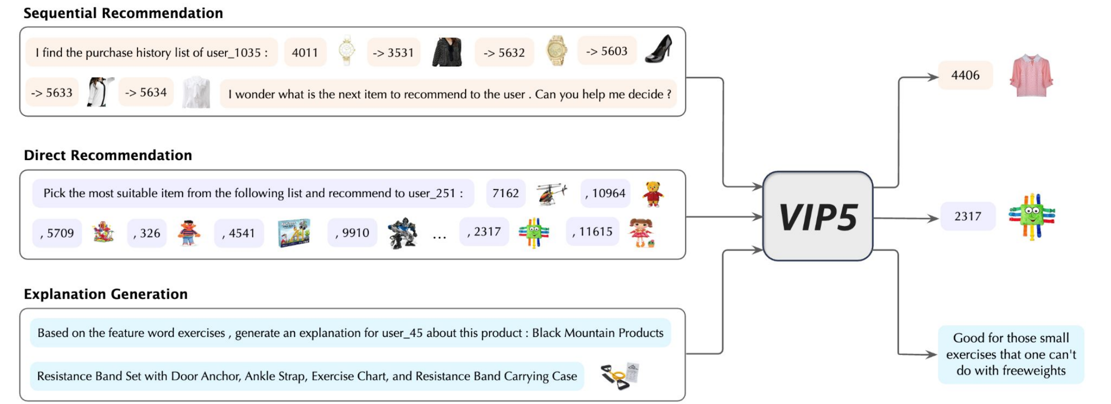


Aligning vision and language with a projector

Liu et al. Visual Instruction Tuning. NeurIPS 2023.

Aligning LLMs for recommendation

(4) + Multimodal information (VIP5)



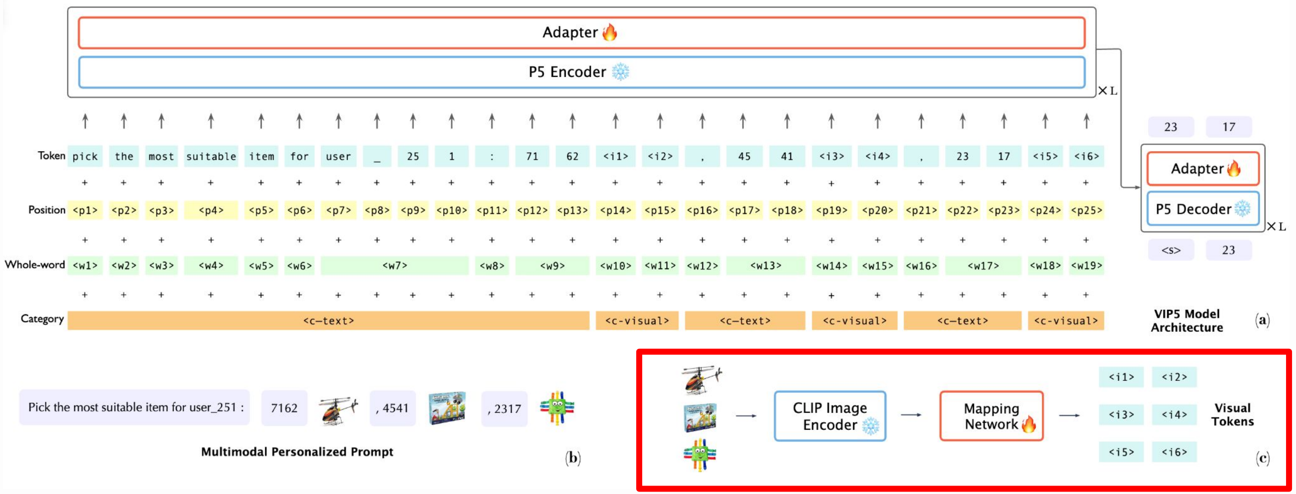
Diff between P5:

Pair text with its image

Gen et al. VIP5: Towards Multimodal Foundation Models for Recommendation. EMNLP 2023 (Findings).

Aligning LLMs for recommendation

(4) + Multimodal information (VIP5)



Alignment with projector

Aligning LLMs for recommendation

(4) + Multimodal information (VIP5)

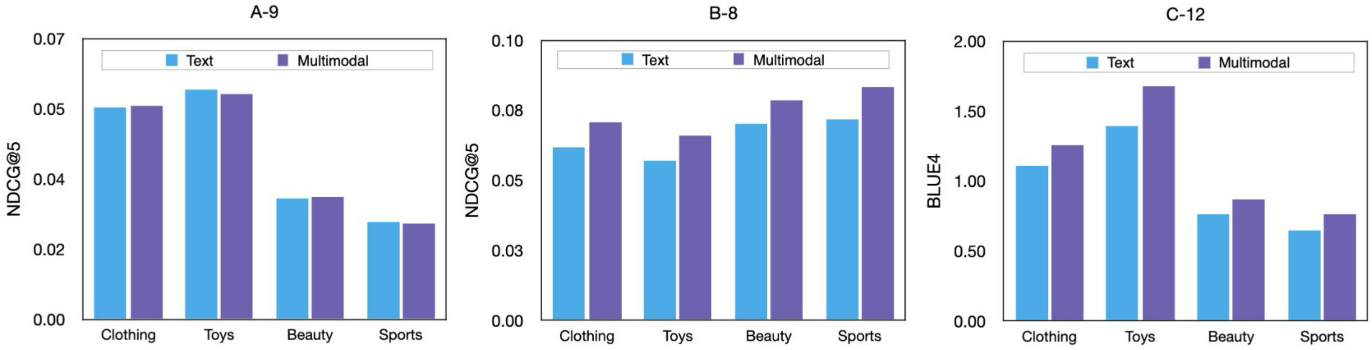
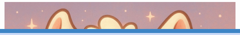


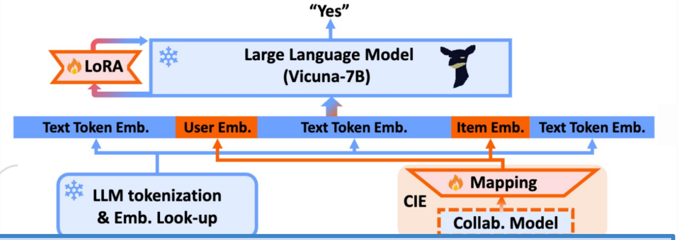
Figure 3: Performance comparison between text-based prompt and multimodal prompt.

Multimodal information is important

Gen et al. VIP5: Towards Multimodal Foundation Models for Recommendation. EMNLP 2023 (Findings).

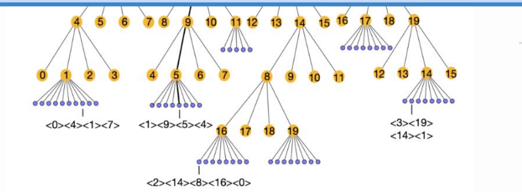
Aligning LLMs for recommendation

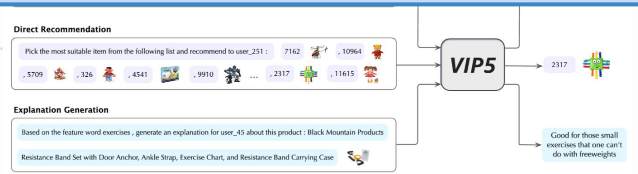




ngs

Information tailored for recommendation matters!





+ External item tokens

+ Multimodal information

Part 1: LLM as Sequential Recommender

(i) Early efforts: Pretrained LLMs for recommendation;
(ii) Aligning LLMs for recommendation;
(iii) **Training** objective & **inference**

Training objective

(1) Supervised finetuning (SFT)

I have watched Titanic, Roman Holiday, … Gone with the wind. Predict the next movie I will watch:

Training objective

(1) Supervised finetuning (SFT)

I have watched Titanic, Roman Holiday, … Gone with the wind. Predict the next movie I will watch:

Waterloo Bridge.

Prediction

Training objective

(1) Supervised finetuning (SFT)

I have watched Titanic, Roman Holiday, … Gone with the wind. Predict the next movie I will watch:

Waterloo Bridge.

Prediction

Training objective

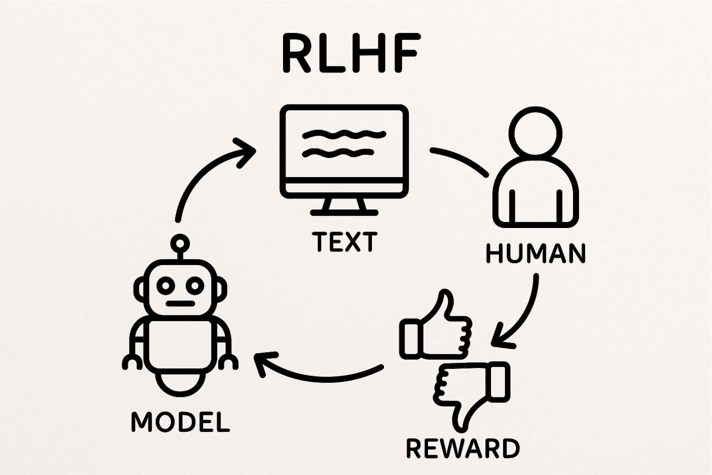
(1) Supervised finetuning (SFT)

$$\mathcal{L}_{\mathrm{SFT}}(\theta) = -\mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\sum_{t=1}^{T} \log \boxed{P_\theta(y_t \mid y_{<t})}\right]$$

Always predict the next token

Training objective

(2) Preference learning



LLMs are trained to align human preferences

Recommendation is about user preferences

Training objective

(2) Preference learning

I have watched Titanic, Roman Holiday, … Gone with the wind. Predict the next movie I will watch:

Waterloo Bridge > Harry Potter

Rafailov et al. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. NeurIPS 2023.

Training objective

(2) Preference learning

$$\mathcal{L}_{\mathrm{DPO}} = -\mathbb{E}_{(x_u, e_p, e_d)}\left[\log \sigma\left(\beta \log \frac{\pi_\theta(e_p|x_u)}{\pi_{\mathrm{ref}}(e_p|x_u)} - \beta \log \frac{\pi_\theta(e_d|x_u)}{\pi_{\mathrm{ref}}(e_d|x_u)}\right)\right],$$

Direct Preference Optimization!

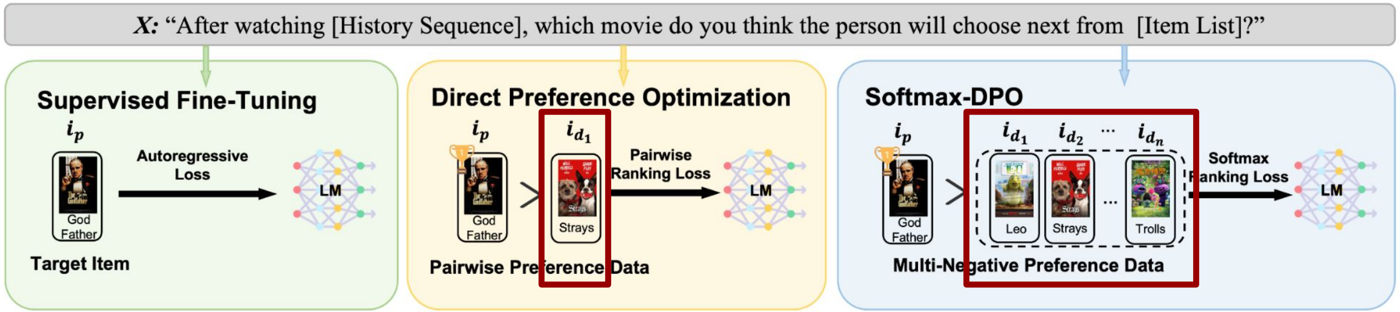
I have watched Titanic, Roman Holiday, … Gone with the wind. Predict the next movie I will watch:

Waterloo Bridge > Harry Potter

Rafailov et al. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. NeurIPS 2023.

Training objective

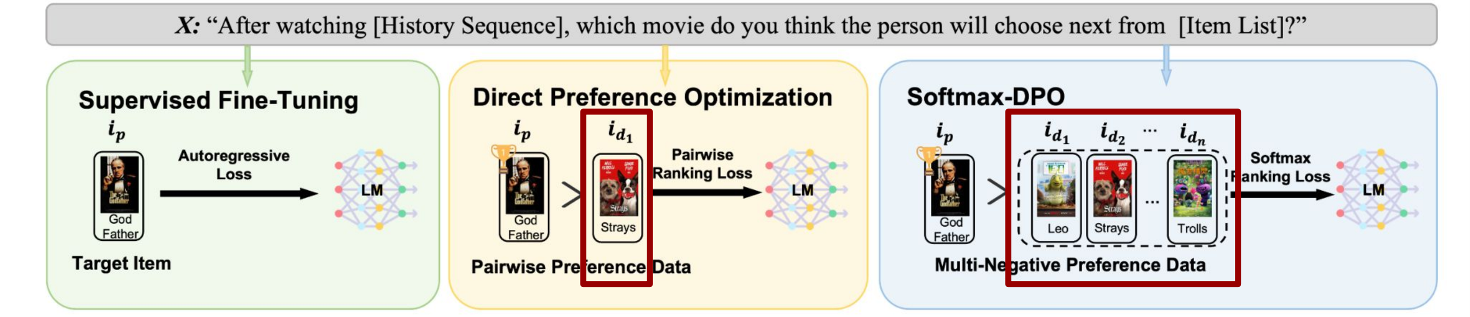
(2) Preference learning



Single negative → Multiple negatives

Training objective

(2) Preference learning



$$\mathcal{L}_{\mathrm{S-DPO}}(\pi_\theta;\pi_{\mathrm{ref}}) = -\mathbb{E}_{(x_u,e_p,\mathcal{E}_d)\sim\mathcal{D}}\left[\log\sigma\left(-\log\sum_{e_d\in\mathcal{E}_d}\exp\left(\beta\log\frac{\pi_\theta(e_d|x_u)}{\pi_{\mathrm{ref}}(e_d|x_u)}-\beta\log\frac{\pi_\theta(e_p|x_u)}{\pi_{\mathrm{ref}}(e_p|x_u)}\right)\right)\right].$$

Chen et al. On Softmax Direct Preference Optimization for Recommendation. NeurIPS 2024.

Training objective

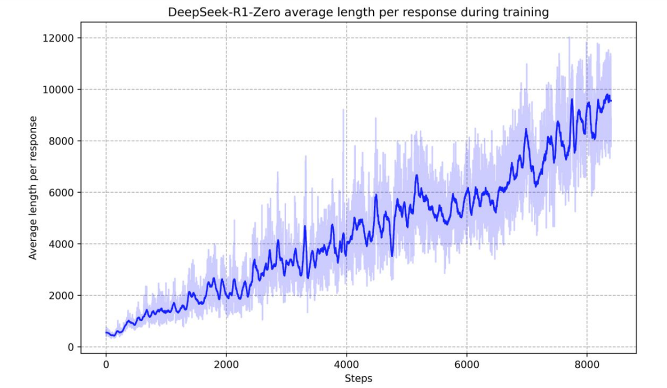
(3) Reinforce learning

Emergent reasoning capabilities through RL

$$\mathcal{J}_{GRPO}(\theta) = \mathbb{E}[q \sim P(Q), \{o_i\}_{i=1}^{G} \sim \pi_{\theta_{old}}(O|q)]$$

$$\frac{1}{G}\sum_{i=1}^{G}\left(\min\left(\frac{\pi_\theta(o_i|q)}{\pi_{\theta_{old}}(o_i|q)}A_i, \text{clip}\left(\frac{\pi_\theta(o_i|q)}{\pi_{\theta_{old}}(o_i|q)}, 1-\varepsilon, 1+\varepsilon\right)A_i\right) - \beta\mathbb{D}_{KL}\left(\pi_\theta||\pi_{ref}\right)\right), \tag{1}$$

$$\mathbb{D}_{KL}\left(\pi_\theta||\pi_{ref}\right) = \frac{\pi_{ref}(o_i|q)}{\pi_\theta(o_i|q)} - \log\frac{\pi_{ref}(o_i|q)}{\pi_\theta(o_i|q)} - 1, \tag{2}$$



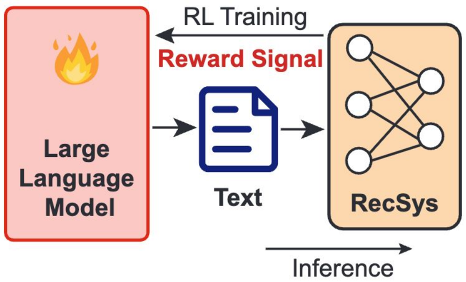
Question: If $a > 1$, then the sum of the real solutions of $\sqrt{a - \sqrt{a + x}} = x$ is equal to

Response: <think>
To solve the equation $\sqrt{a - \sqrt{a + x}} = x$, let's start by squaring both $\cdots$
$\left(\sqrt{a - \sqrt{a + x}}\right)^2 = x^2 \implies a - \sqrt{a + x} = x^2.$
Rearrange to isolate the inner square root term:
$(a - x^2)^2 = a + x \implies a^2 - 2ax^2 + (x^2)^2 = a + x \implies x^4 - 2ax^2 - x + (a^2 - a) = 0$
$\ldots$
Wait, wait. Wait. That's an aha moment I can flag here.
Let's reevaluate this step-by-step to identify if the correct sum can be $\cdots$
We started with the equation:
$\sqrt{a - \sqrt{a + x}} = x$
First, let's square both sides:
$a - \sqrt{a + x} = x^2 \implies \sqrt{a + x} = a - x^2$
Next, I could square both sides again, treating the equation: $\cdots$
$\ldots$

DeepSeek-AI et al. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. arXiv: 2501.12948.

Training objective

(3) Reinforce learning



$$\max_{\theta} \mathbb{E}_{s\sim p(s), a\sim \pi_\theta(a|s)} \boxed{[f(a|s)]}.$$

Maximize the reward from recommender system

Prompt Template for Rec-R1 + Dense Retriever (Product Search)

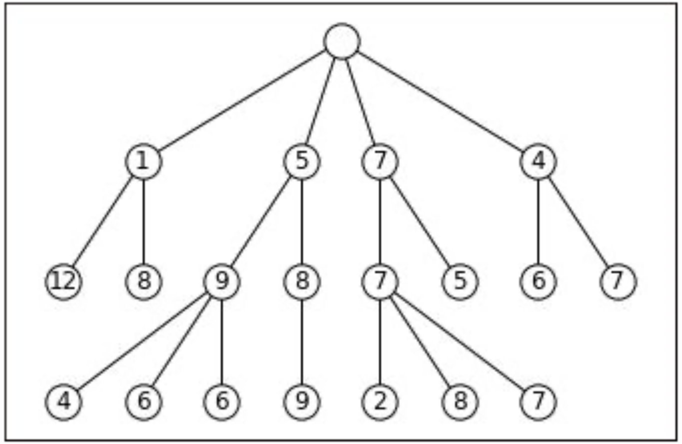
```
You are an expert in generating queries for dense retrieval. Given a customer
query, your task is to retain the original query while expanding it with
additional semantically relevant information, retrieve the most relevant
products, ensuring they best meet customer needs. If no useful expansion is
needed, return the original query as is.

Below is the query:
``` {user_query} ```

<|im_start|>system
You are a helpful AI assistant. You first think about the reasoning process
in the mind and then provide the user with the answer.
<|im_end|>
<|im_start|>user
[PROMPT as above]
Show your work in <think>\think> tags. Your final response must be in JSON
format within <answer>\answer> tags. For example,
<answer>
{ "query": xxx }
</answer>.
<|im_end|>
<|im_start|>assistant
Let me solve this step by step.
<think>
```
```

Inference

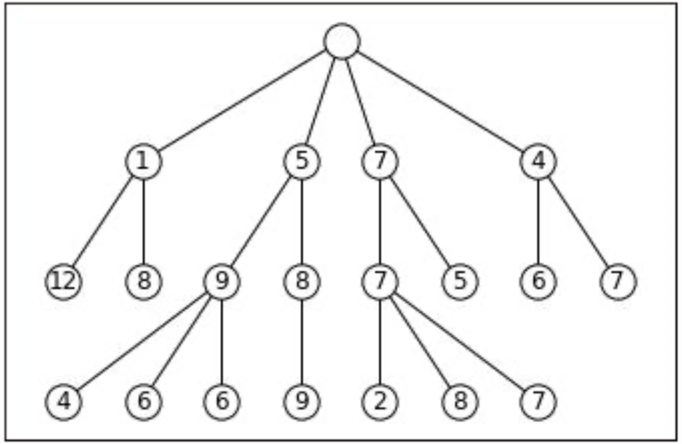
(1) Beam Search



Generating answers with the top-k highest scored beams

Inference

(1) Beam Search



It may generate invalid items

In RecSys :
No Hallucination permitted!

Inference

(2) Constrained Beam Search

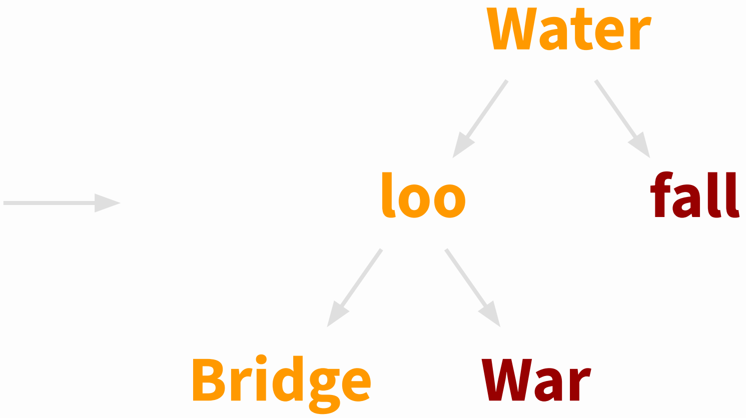
Valid items:
Waterloo Bridge, Waterfall Story, and Waterloo War

How to make the generated items always valid?

Inference

(2) Constrained Beam Search

Valid items:
Waterloo Bridge, Waterfall Story, and Waterloo War



Constrained search tree

Inference

(2) Constrained Beam Search

I have watched Titanic, Roman Holiday, …
Gone with the wind. Predict the next movie
I will watch:

$\xrightarrow{P = 1}$ **Water**

Inference

(2) Constrained Beam Search

I have watched Titanic, Roman Holiday, … Gone with the wind. Predict the next movie I will watch:

P = 1

Water

P = 0.8

P = 0.2

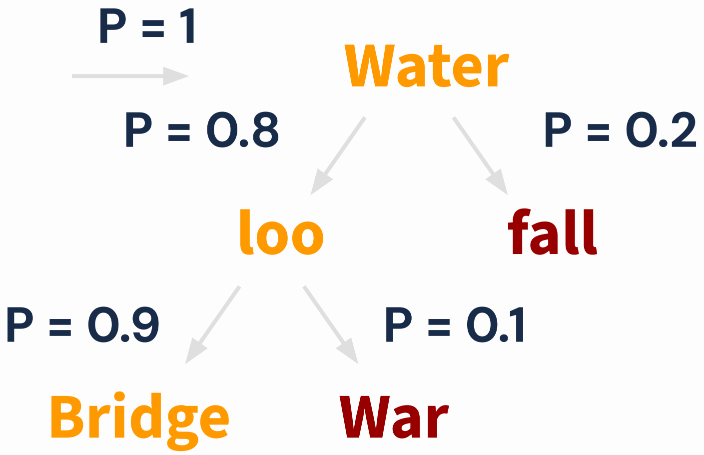
loo

fall

Inference

(2) Constrained Beam Search

**I have watched Titanic, Roman Holiday, …
Gone with the wind. Predict the next movie
I will watch:**

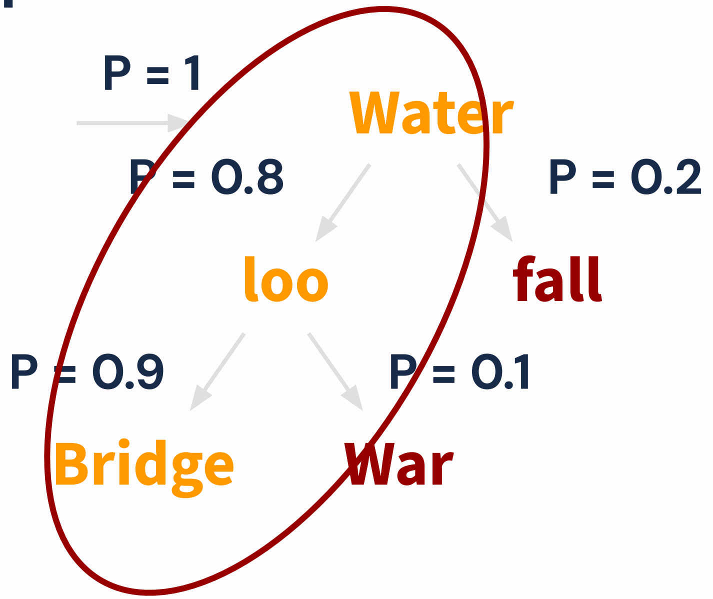


Inference

(2) Constrained Beam Search

I have watched Titanic, Roman Holiday, … Gone with the wind. Predict the next movie I will watch:

Valid Item!



Inference

(3) Special design

$$\mathcal{S}(h_{\leq t}) = \mathcal{S}(h_{\leq t-1}) + \log(p(h_t|x, h_{\leq t-1})),$$

$$\mathcal{S}(h) = \mathcal{S}(h)\boxed{/h_L^{\alpha},}$$

Length penalty in beam search;
Human does not like over long sentences.

Redundant for recommendation

Bao et al. Decoding Matters: Addressing Amplification Bias and Homogeneity Issue for LLM-based Recommendation. EMNLP 2024.

Inference

(3) Special design

$$\mathcal{S}(h_{\leq t}) = \mathcal{S}(h_{\leq t-1}) + \log(p(h_t|x, h_{\leq t-1})),$$

$$\mathcal{S}(h) = \mathcal{S}(h)/\cancel{L},$$

Remove length penalty

	Instruments	Books	CDs	Sports	Toys	Games
Baseline	0.1062	0.0308	0.0956	0.1171	0.0965	0.0610
D^3	**0.1111**	**0.0354**	**0.1190**	**0.1215**	**0.1025**	**0.0767**
- RLN	0.1093	0.0353	0.1000	0.1200	0.0975	0.0659
- TFA	0.1086	0.0309	0.1115	0.1192	0.1006	0.0732

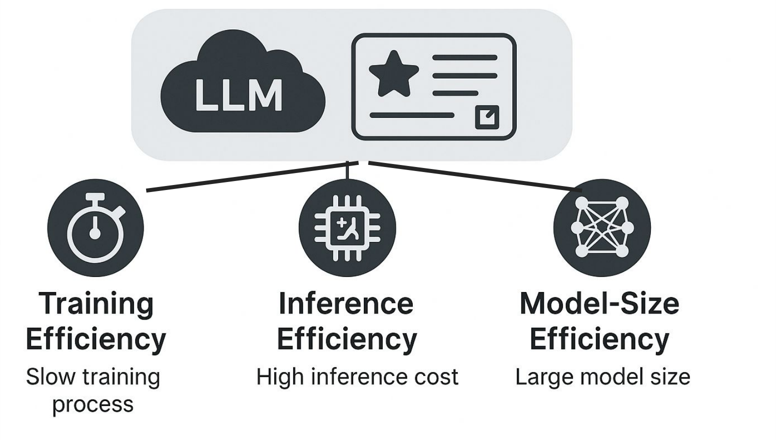
Imp when removing

Part 1: LLM as Sequential Recommender

(i) Early efforts: Pretrained LLMs for recommendation;
(ii) Aligning LLMs for recommendation;
(iii) Training objective & inference
(iiii) **Efficiency**

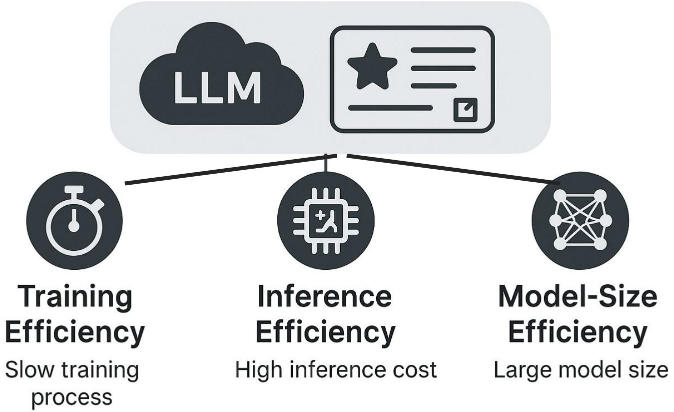
Efficiency

A crucial question in real-world deployment



Efficiency

A crucial question in real-world deployment



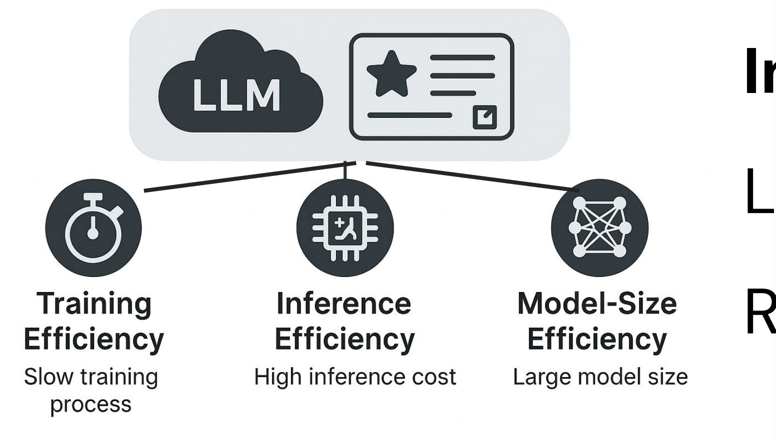
Training efficiency:

LLM: update by months

Recommender: update by hours

Efficiency

A crucial question in real-world deployment



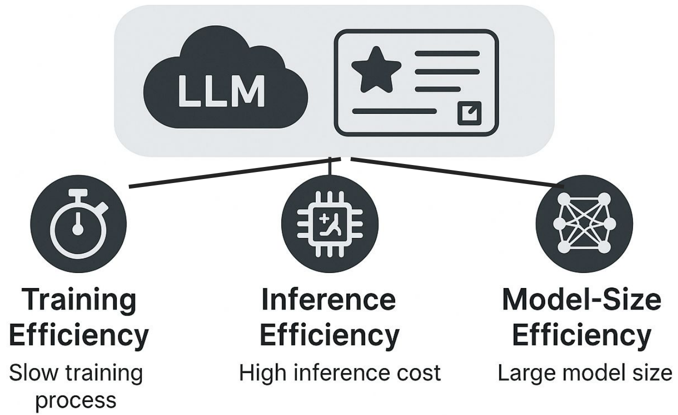
Inference efficiency:

LLM: wait for seconds

Recommender: wait for milliseconds

Efficiency

A crucial question in real-world deployment



Model-size efficiency:

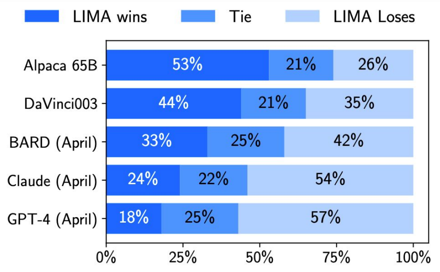
LLM: serve for millions

Recommender: serve for billions

Efficiency

(1) Training efficiency

Source	#Examples
Training	
Stack Exchange (STEM)	200
Stack Exchange (Other)	200
wikiHow	200
Pushshift r/WritingPrompts	150
Natural Instructions	50
Paper Authors (Group A)	200
Dev	
Paper Authors (Group A)	50
Test	
Pushshift r/AskReddit	70
Paper Authors (Group B)	230



Less is more for alignment

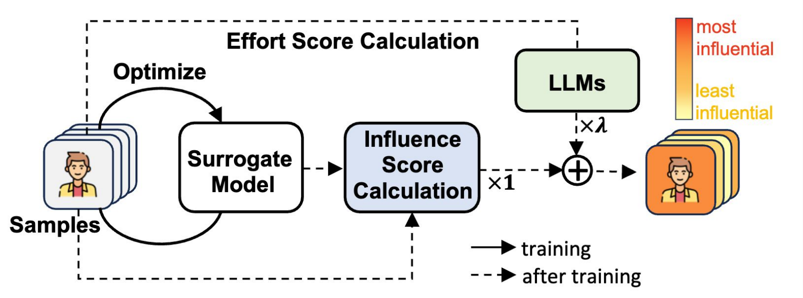
1k high quality examples ->

Surpass large scale training

Zhou et al. LIMA: Less Is More for Alignment. NeurIPS 2023.

Efficiency

(1) Training efficiency



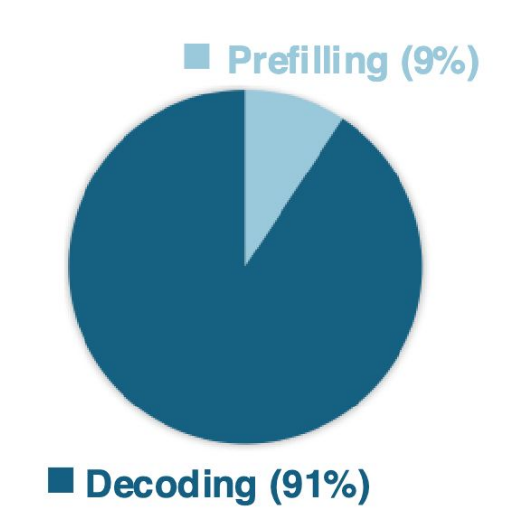
	Games				
	R@10↑	R@20↑	N@10↑	N@20↑	Time↓
Full	0.0169	0.0233	0.0102	0.0120	36.87h
DEALRec	0.0181	0.0276	0.0115	0.0142	1.67h
% Improve.	**7.10%**	**18.45%**	**12.75%**	**18.33%**	**-95.47%**

Select the most informative examples ->

Reducing 95% training time

Efficiency

(2) Inference efficiency



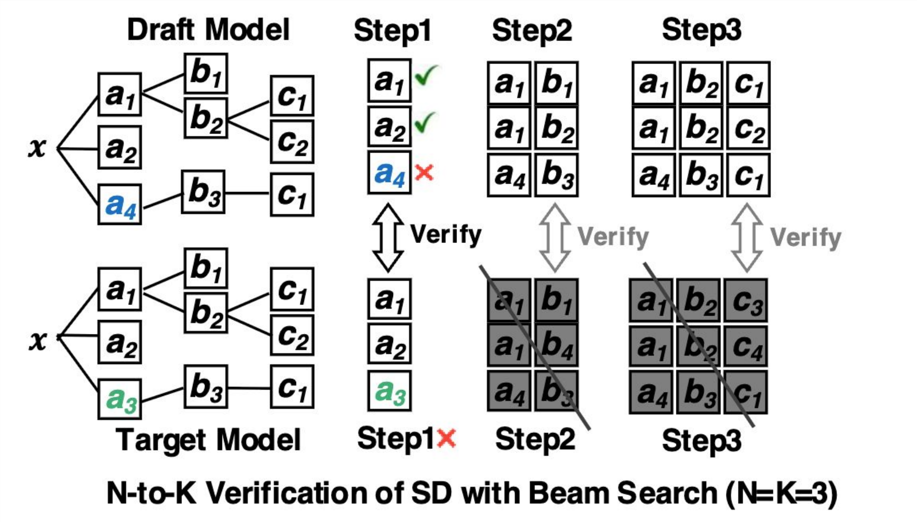
Autoregressive paradigm in LLM

-> huge time on the decoding stage

Lin et al. Efficient Inference for Large Language Model-based Generative Recommendation. ICLR 2025.

Efficiency

(2) Inference efficiency



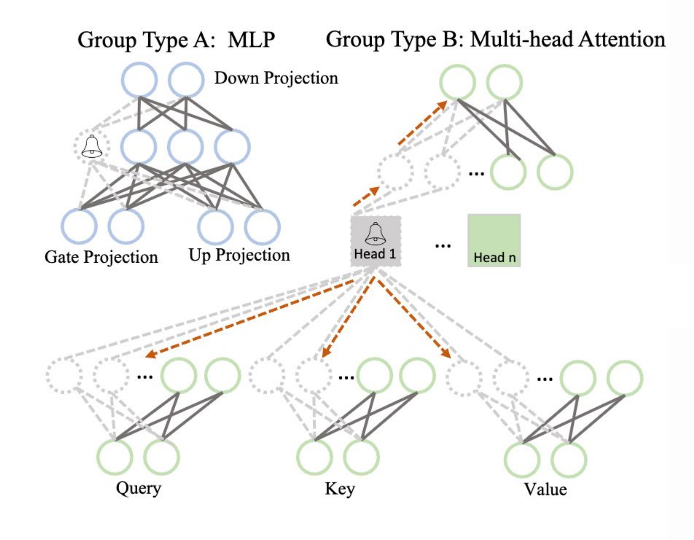
Speculative decoding:

Decoder acceleration with a small-size draft model

Lin et al. Efficient Inference for Large Language Model-based Generative Recommendation. ICLR 2025.

Efficiency

(3) Model-size efficiency – Pruning



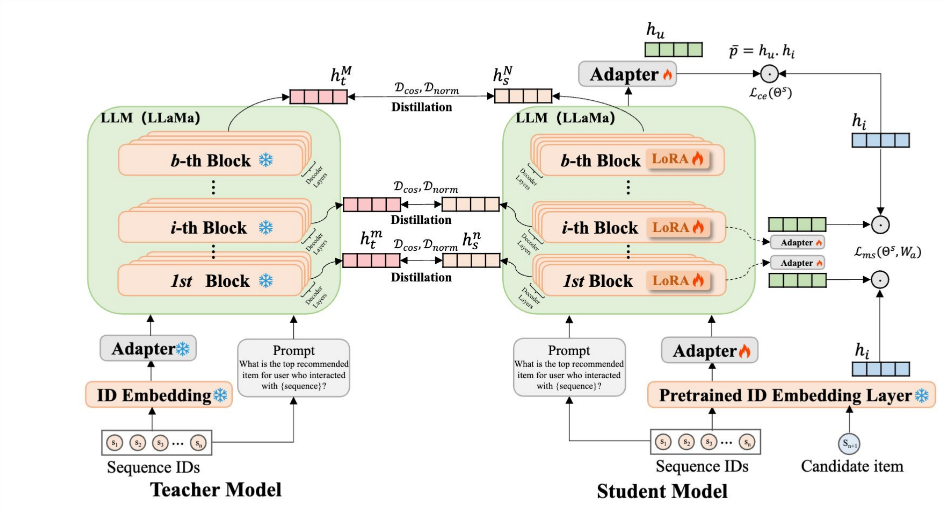
Similar performance with 0.6% parameters

		Tasks		
Method	**#Params**	**TNEWS↑**	**IFLYTEK↑**	**CSL↑**
M6-base	327M	0.598	0.631	0.852
ALBERT-zh-base	12M	0.550	0.564	0.785
M6-Edge	10M	**0.552**	**0.586**	**0.831**
ALBERT-zh-tiny	4M	0.534	0.488	0.750
M6-Edge, Pruned	2M	**0.537**	**0.559**	**0.798**

Ma et al. LLM-Pruner: On the Structural Pruning of Large Language Models. NeurIPS 2023
Cui et al. M6-Rec: Generative Pretrained Language Models are Open-Ended Recommender Systems. arXiv: 2205.08084.

Efficiency

(3) Model-size efficiency – Distillation



SLM learns from LLM

With Hard label + soft label

Efficiency

(3) Model-size efficiency – Distillation

Table 3: Experimental results (%) on the Music and Sport dataset.

Model	Music HR@1	Music HR@5	Music NDCG@5	Music MRR	Sport HR@1	Sport HR@5	Sport NDCG@5	Sport MRR	Rank
Caser	0.71	3.28	1.96	2.29	1.05	3.75	2.39	2.84	13.50
GRU4Rec	1.89	3.22	2.57	3.08	5.26	7.75	6.52	7.08	10.13
BERT4Rec	2.10	3.16	2.64	3.11	4.81	6.70	5.79	6.26	10.63
SASRec	1.82	5.72	3.79	4.51	4.70	8.43	6.59	7.24	8.75
HGN	2.01	5.49	3.82	4.17	3.42	6.24	4.83	5.30	10.50
LightSANs	1.05	4.06	2.54	3.00	5.18	8.94	7.07	7.72	8.25
S^3-Rec	2.48	7.37	4.94	4.68	4.14	8.49	6.89	7.35	6.88
DuoRec	1.84	4.50	3.19	3.04	4.13	8.81	7.03	6.64	9.13
MAERec	2.19	6.35	4.67	3.96	4.01	8.35	6.65	6.98	8.63
Open-P5	4.35	8.12	6.74	-	5.49	8.50	6.92	-	5.33
E4SRec	5.62	9.29	7.50	7.98	6.40	9.67	8.05	8.70	1.75
E4SRec_8	5.46	8.86	7.21	7.74	5.48	8.63	7.06	7.76	3.63
E4SRec_4	5.33	8.75	7.08	7.59	5.41	8.65	7.04	7.72	4.50
$\text{SLMRec}_{4\leftarrow8}$	**5.72**	**9.15**	**7.48**	**8.03**	**6.62**	**9.83**	**8.25**	**8.89**	**1.25**

Method	Tr time(h)	Inf time(h)	Tr params (B)	Inf params (B)
$\text{Open-P5}_{\text{LLaMa}}$	0.92	4942	0.023	7.237
E4SRec	3.95	0.415	0.023	6.631
$\textbf{SLMRec}_{4\leftarrow8}$	0.60	0.052	0.003	0.944

Reduced model-size;

Reduced inference time

Xu et al..SLMRec: Distilling Large Language Models into Small for Sequential Recommendation. ICLR 2025.

Part 1: LLM as Sequential Recommender

(1) Early efforts: pretrained LLMs for rec

(2) Aligning LLMs for recommendation

- Pure text-based
- External item tokens
- Collaborative embeddings
- Multimodal information

(3) Training objective & inference

Training: SFT, DPO, RL; **Inference**: (constrained) beam search

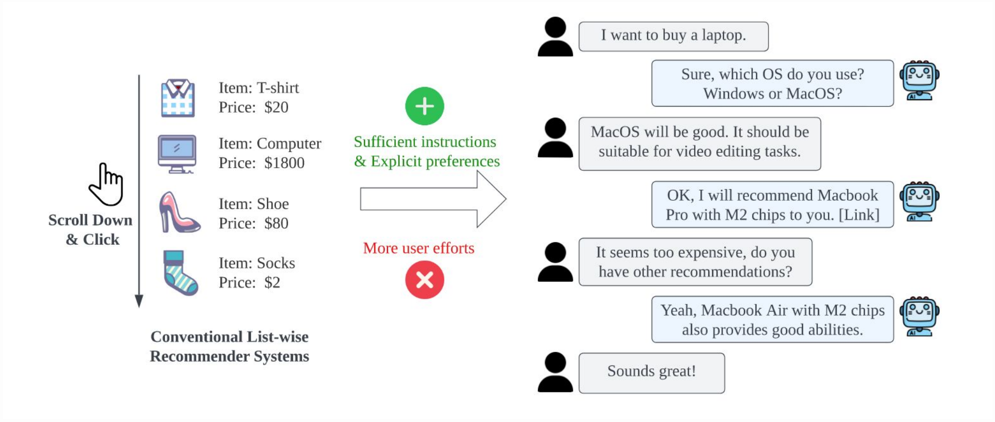
(4) Efficiency

Data efficiency; Inference efficiency; Model-size efficiency

Part 2: LLM as Conversational Recommender

Conversational Recommender System (CRS)

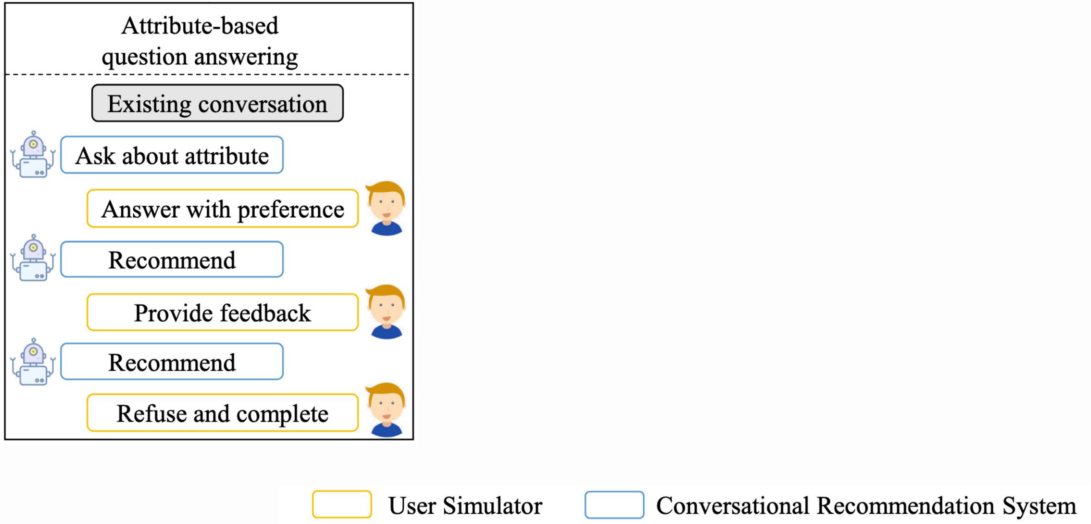
- Recommendations with multiple turns conversation
- Interactive; engaging users in the loop



Chen et al. All Roads Lead to Rome: Unveiling the Trajectory of Recommender Systems Across the LLM Era. arXiv.2407.10081

Paradigms of CRS before the era of LLM

Attribute-based

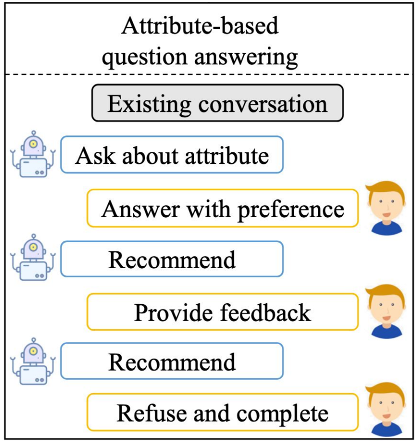


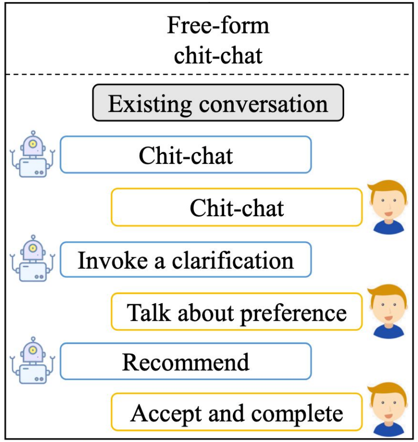
Wang et al. Rethinking the Evaluation for Conversational Recommendation in the Era of Large Language Models. EMNLP 2023.

Paradigms of CRS before the era of LLM

Attribute-based

Free-form





User Simulator | Conversational Recommendation System

Wang et al. Rethinking the Evaluation for Conversational Recommendation in the Era of Large Language Models. EMNLP 2023.

Paradigms of CRS before the era of LLM

Features: Task-specific conversational recommenders, trained on limited conversation data.

Paradigms of CRS before the era of LLM

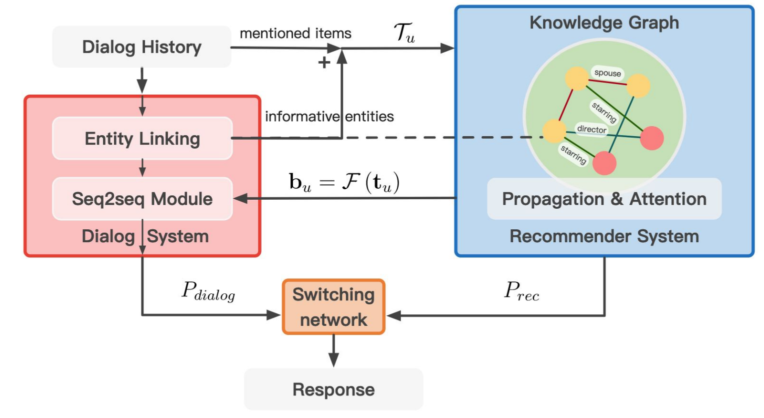
Features: Task-specific conversational recommenders, trained on limited conversation data.

- Lack of world knowledge.
- Requirement of complicated strategies.
- Incompatible natural language generation abilities.
- Lack of generalization capabilities.

Paradigms of CRS before the era of LLM

Traditional CRS: KBRD

- End-to-end conversational recommender system
- Switching between conversation and recommendation
- External knowledge from knowledge graph



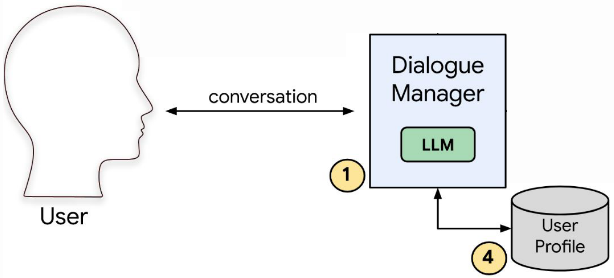
Chen et al. Towards Knowledge-Based Recommender Dialog System. EMNLP 2019.

Example

LLM as conversational recommender

LLM as Conversational Recommender

Framework (RecLLM)

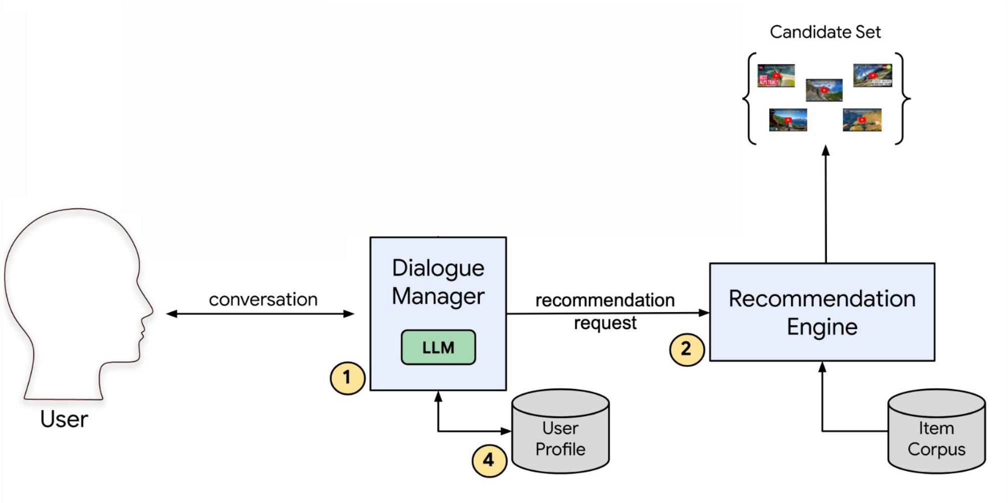


Conversation with users via LLMs

Friedman et al. Leveraging Large Language Models in Conversational Recommender Systems. arXiv:2305.07961.

LLM as Conversational Recommender

Framework (RecLLM)

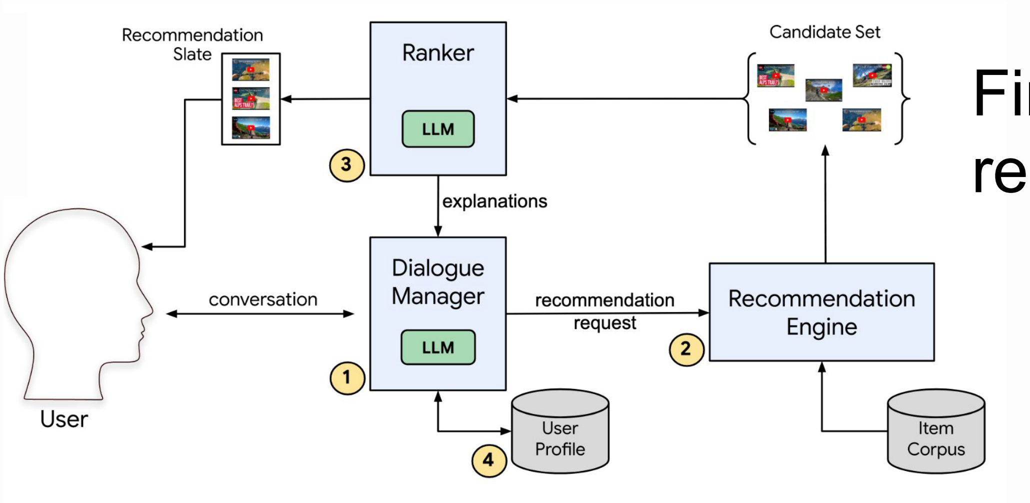


Recommendation via tools

Friedman et al. Leveraging Large Language Models in Conversational Recommender Systems. arXiv:2305.07961.

LLM as Conversational Recommender

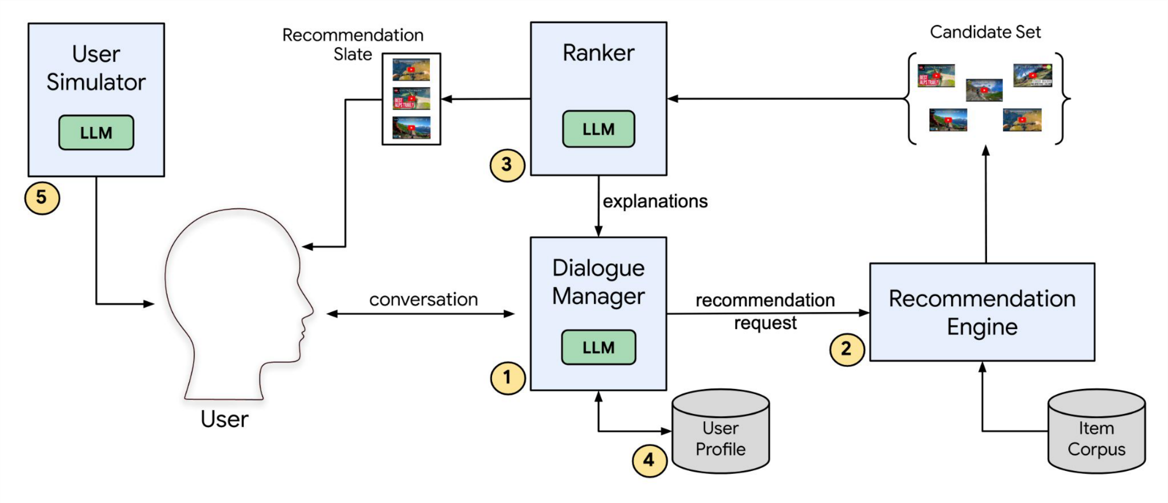
Framework (RecLLM)



Fine-grained reranking via LLMs

LLM as Conversational Recommender

Framework (RecLLM)

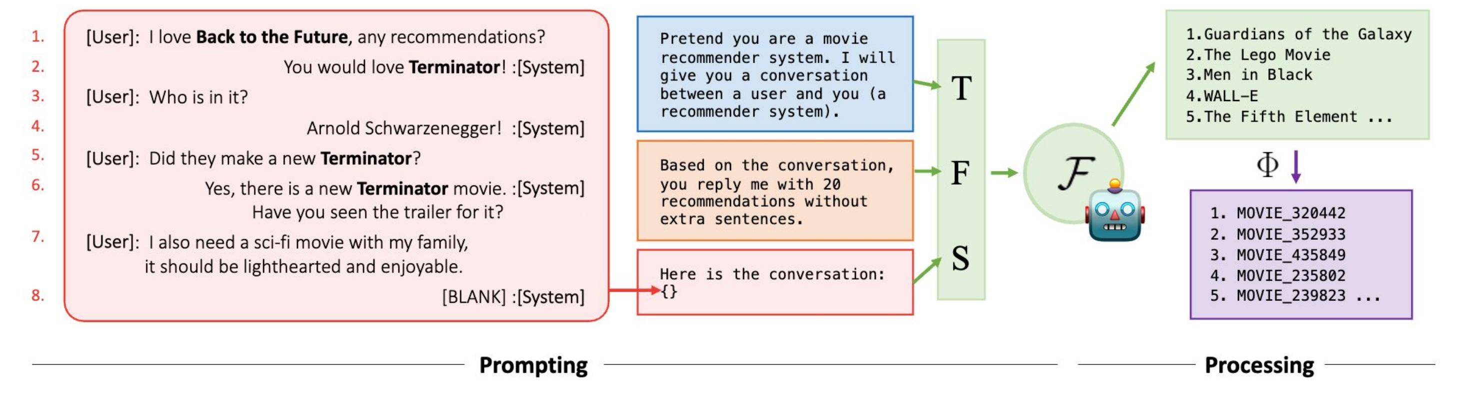


Evaluation via LLMs

Friedman et al. Leveraging Large Language Models in Conversational Recommender Systems. arXiv:2305.07961.

LLM as Conversational Recommender

LLMs as zero-shot CRS

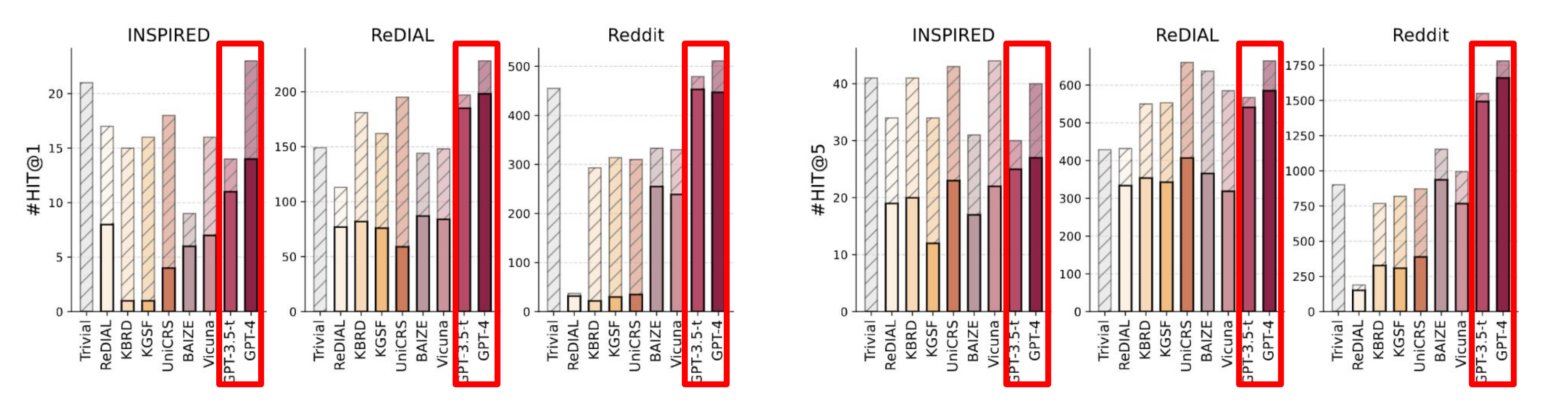


How powerful are LLMs for zero-shot CRS?

He et al. Large Language Models as Zero-Shot Conversational Recommenders. CIKM 2023.

LLM as Conversational Recommender

LLMs as zero-shot CRS

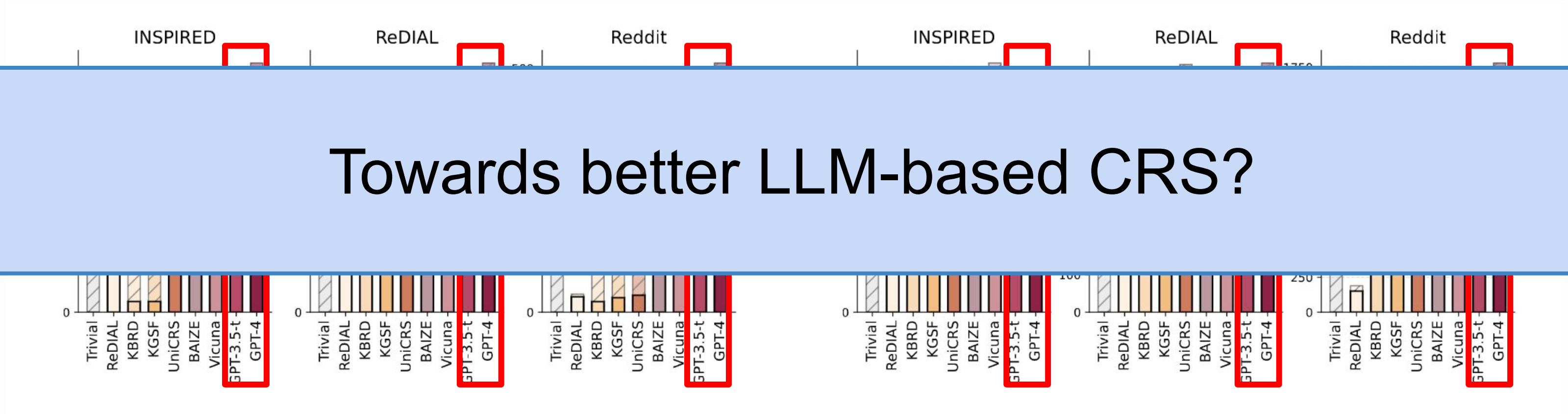


Can surpass traditional CRSs!

He et al. Large Language Models as Zero-Shot Conversational Recommenders. CIKM 2023.

LLM as Conversational Recommender

LLMs as zero-shot CRS



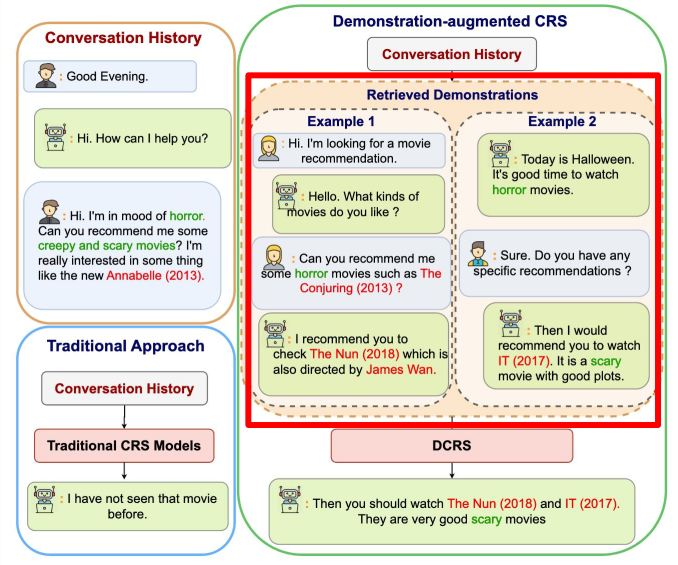
Towards better LLM-based CRS?

Can surpass traditional CRSs!

He et al. Large Language Models as Zero-Shot Conversational Recommenders. CIKM 2023.

LLM as Conversational Recommender

+ Demonstration



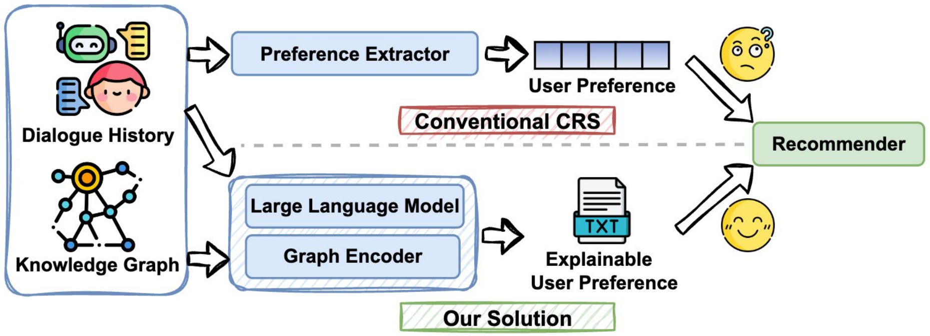
Prompting with previously successful conversation

Relevant conversation history helps!

Dao et al. Broadening the View: Demonstration-augmented Prompt Learning for Conversational Recommendation. SIGIR 2024.

LLM as Conversational Recommender

+ **Knowledge graph**

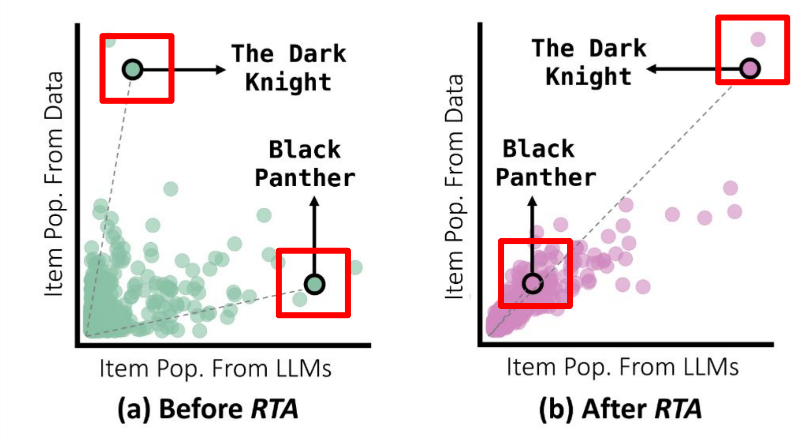


Recommendation-specific knowledge graph helps

Qiu et al. Unveiling User Preferences: A Knowledge Graph and LLM-Driven Approach for Conversational Recommendation. arXiv:2411.14459

LLM as Conversational Recommender

+ Collaborative information



Collaborative information (e.g., popularity) helps LLMs fit the real distribution in CRS

He et al. Reindex-Then-Adapt: Improving Large Language Models for Conversational Recommendation. WSDM 2025.

LLM as Conversational Recommender

Challenges – Datasets

Public datasets for CRS are limited, due to the scarcity of conversational products and real-world CRS datasets

LLM as Conversational Recommender

Challenges – Evaluation

Traditional metrics like NDCG and BLEU are often insufficient to assess user experience

LLM as Conversational Recommender

Challenges – Product

What is the form of LLM-based CRS products?

ChatBot? Search bar? Independent App?

Part 2: LLM as Conversational Recommender

(1) LLMs show potential in CRS

(2) LLM-based CRS can be improved with:

demonstration, collaborative information …

(3) Challenges in LLM-based CRSs:

dataset, evaluation, and product

Part 3: LLM as User Simulator

User simulators before the era of LLM

RL-based user simulator

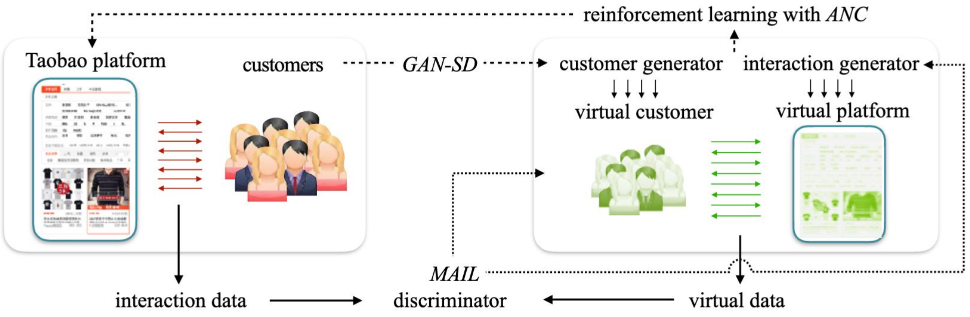


Figure 1: Virtual-Taobao architecture for reinforcement learning.

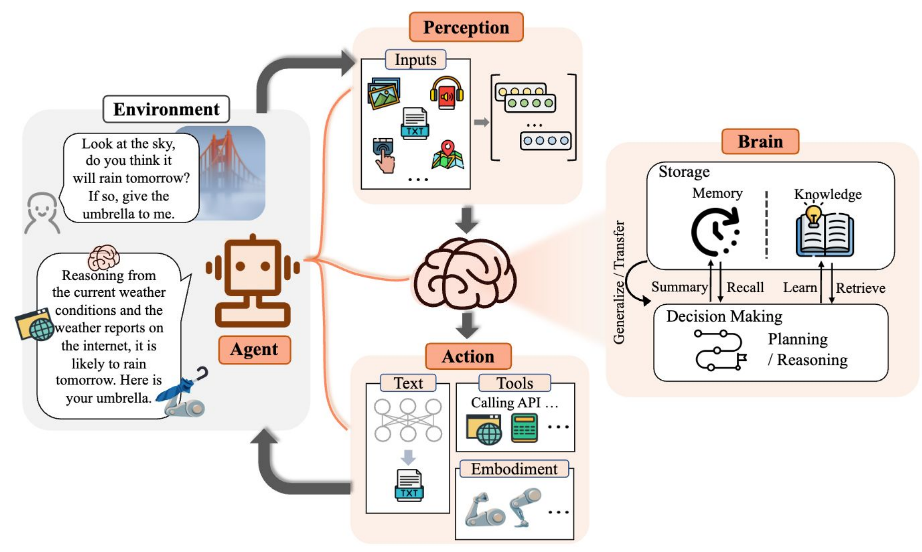
High sampling cost
Overfitting risks
Training instability
Limited action space

...

Shi et al. Virtual-Taobao: Virtualizing Real-world Online Retail Environment for Reinforcement Learning. AAAI 2019.

LLM as User Simulator

Generative agents



Perception

Planning

Memory

Action

...

Xi et al.The Rise and Potential of Large Language Model Based Agents: A Survey. arXiv: 2309.07864.

LLM as User Simulator

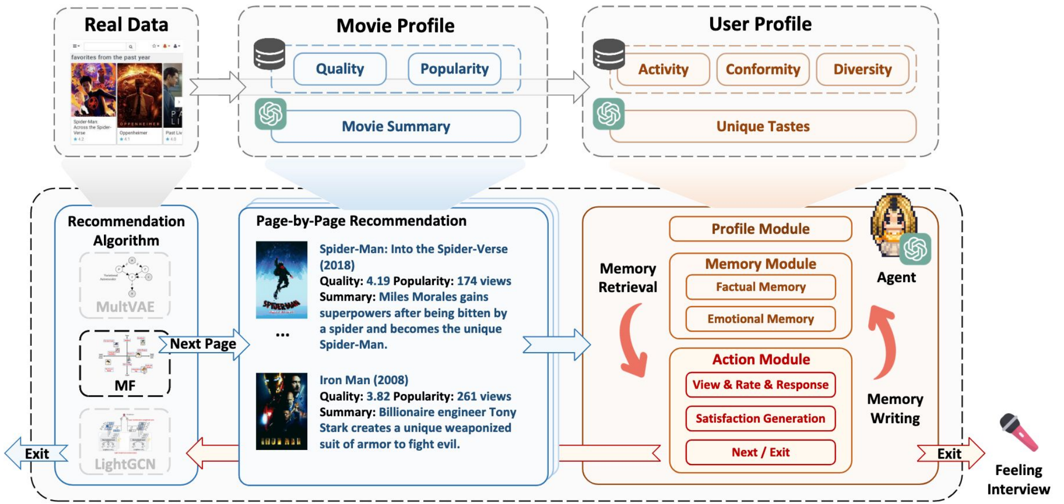
Generative agents for recommendation

Human-like behavior
Abundant action space
Reduced training cost
…

Zhang et al. On generative agents in recommendation. SIGIR 2024.

LLM as User Simulator

Generative agents for recommendation



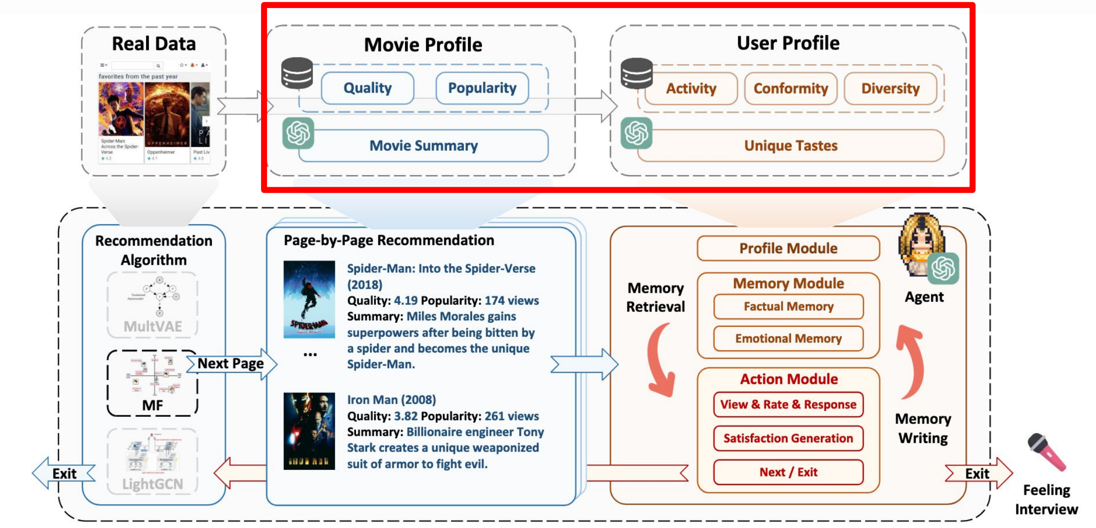
Realworld-like simulation paradigm

- 1000 users
- Page-by-page simulation

Zhang et al. On generative agents in recommendation. SIGIR 2024.

LLM as User Simulator

Generative agents for recommendation



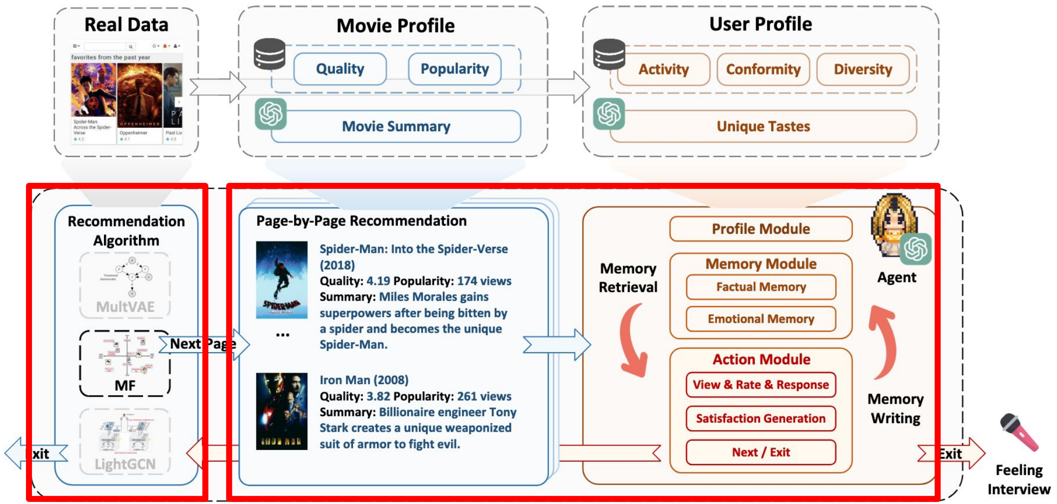
Realworld-like simulation paradigm

- 1000 users
- Page-by-page simulation

Zhang et al. On generative agents in recommendation. SIGIR 2024.

LLM as User Simulator

Generative agents for recommendation



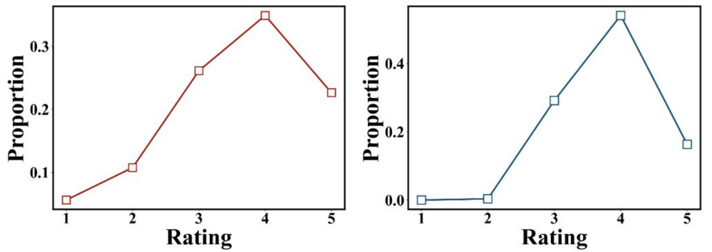
Realworld-like simulation paradigm

- 1000 users
- Page-by-page simulation

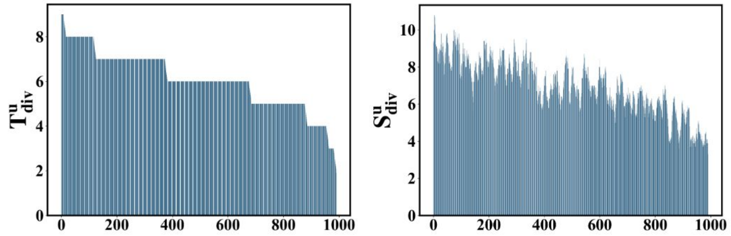
Zhang et al. On generative agents in recommendation. SIGIR 2024.

LLM as User Simulator

Generative agents for recommendation



(a) Distribution on MovieLens (b) Agent-simulated distribution



(a) Ground-truth diversity (b) Simulated diversity

Aligned user preferences
& Recommender evaluation

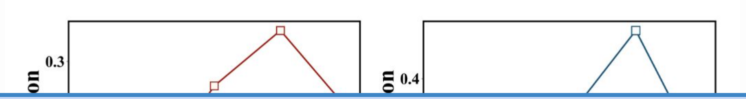
Table 2: Recommendation strategies evaluation.

	$\overline{P}_{view}$	$\overline{N}_{like}$	$\overline{P}_{like}$	$\overline{N}_{exit}$	$\overline{S}_{sat}$
Random	0.312	3.3	0.269	2.99	2.93
Pop	0.398	4.45	0.360	3.01	3.42
MF	0.488	**6.07***	0.462	**3.17***	3.80
MultVAE	0.495	5.69	0.452	3.10	3.75
LightGCN	**0.502***	5.73	**0.465***	3.02	**3.85***

Zhang et al. On generative agents in recommendation. SIGIR 2024.

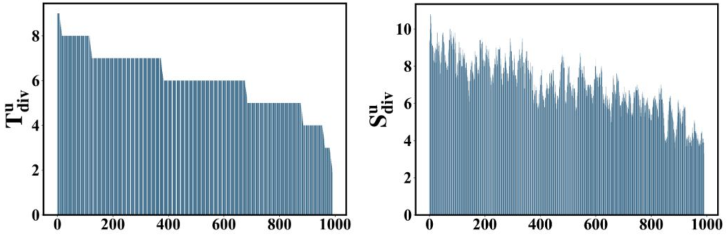
LLM as User Simulator

Generative agents for recommendation



Aligned user preferences

How to make the simulation more faithful?



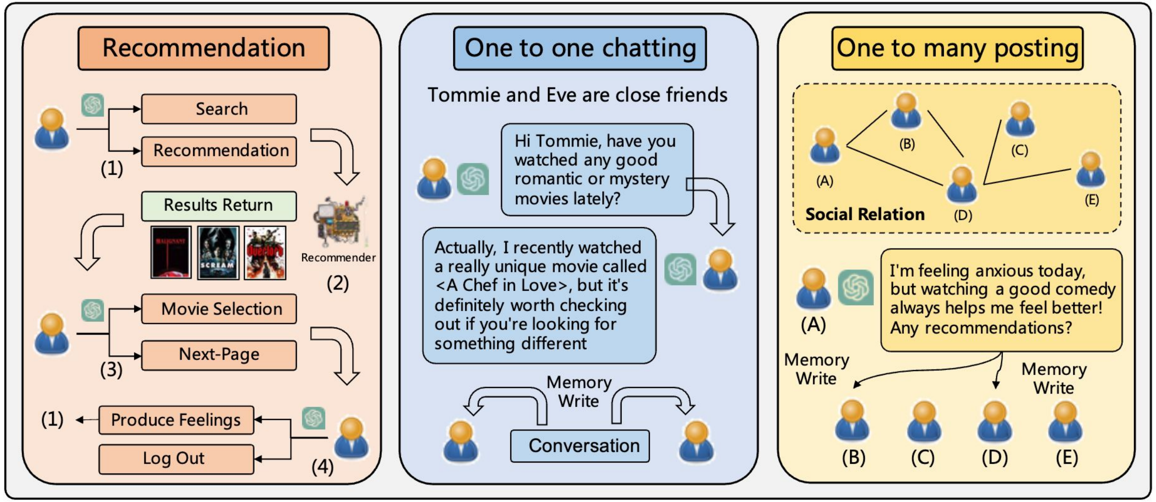
(a) Ground-truth diversity (b) Simulated diversity

	$\overline{P}_{view}$	$\overline{N}_{like}$	$\overline{P}_{like}$	$\overline{N}_{exit}$	$\overline{S}_{sat}$
Random	0.312	3.3	0.269	2.99	2.93
Pop	0.398	4.45	0.360	3.01	3.42
MF	0.488	**6.07***	0.462	**3.17***	3.80
MultVAE	0.495	5.69	0.452	3.10	3.75
LightGCN	**0.502***	5.73	**0.465***	3.02	**3.85***

Zhang et al. On generative agents in recommendation. SIGIR 2024.

LLM as User Simulator

+ **Social behaviors**



Recommendation
Chat
Networking

LLM as User Simulator

+ Multiturn conversation

User ID: A1EXXXXXDUE6B0

Historical Interactions: ["Robin Williams: Live On Broadway", "**Mission Impossible on VHS**", "**Solaris**", "**Elysium**", "Wall Street", "Mystic River", ...]

Conversation 1:

[User] Hi, I recently watched **Mission Impossible on VHS** and it was a fantastic high-tech spy movie! Tom Cruise gave …

[Agent] That's great to hear! I have a movie recommendation for you based on your likes. How about checking out World War Z?

[User] Oh, I'm definitely interested. Can you tell me more about the movie?

[Agent] Certainly! World War Z is a good zombie war flick…

Conversation 2:

[User] Hi, I recently watched the movie "**Solaris**" and I have to say, I didn't enjoy it at all. It felt overly melodramatic and lacked substance…

[Agent] I understand why you didn't like "Solaris". I can recommend a movie you might enjoy instead. How about "**Elysium**"? It's a bilingual film that…

[User] Oh, I've actually already watched "Elysium" and it was better than I expected, but still not great.…

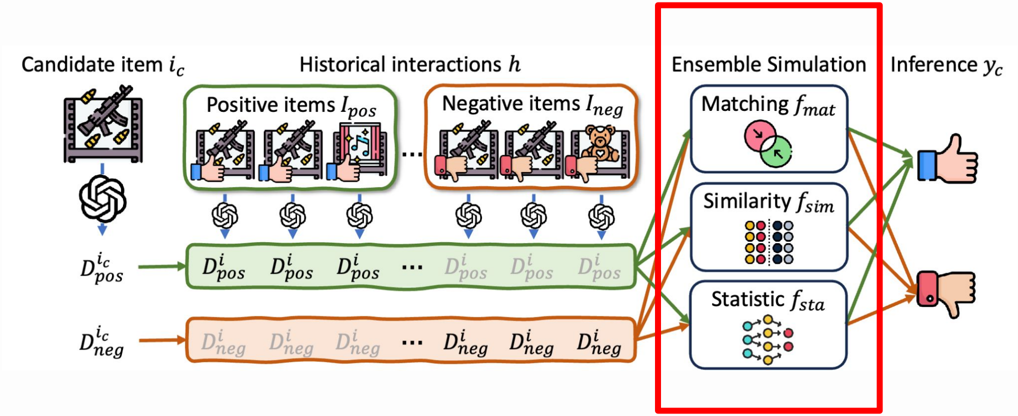
[Agent] How about giving "Wrecking Crew" a try? It's another movie you might like based on your previous preferences…

[User] Sure, that sounds interesting…

Simulating users in the conversational scenarios

LLM as User Simulator

+ **Multi-facet simulation objective**



Category matching
Fine-grained similarity
Statistic information

LLM as User Simulator

+ Multi-facet simulation objective

Dataset	Metric	PPO	TRPO	A2C	DQN
Yelp	A. Rwd↑	9.97	13.45	24.15	**27.56**
	T. Rwd↑	141.57	157.42	267.60	**330.98**
	Liking%↑	34.59	40.07	48.35	**49.43**
Amazon Music	A. Rwd↑	10.49	11.31	13.45	**16.70**
	T. Rwd↑	129.03	140.15	141.03	**181.42**
	Liking%↑	29.30	32.46	29.54	**33.18**
Amazon Games	A. Rwd↑	18.72	21.35	**27.56**	26.43
	T. Rwd↑	208.43	242.26	**317.56**	269.02
	Liking%↑	33.15	37.64	**43.52**	40.73
Amazon Movie	A. Rwd↑	29.42	27.47	31.72	**38.60**
	T. Rwd↑	310.69	301.40	354.34	**416.18**
	Liking%↑	38.59	36.70	42.37	**44.50**
Anime	A. Rwd↑	14.12	14.58	**21.50**	18.03
	T. Rwd↑	155.74	163.44	**242.95**	201.94
	Liking%↑	25.46	24.27	**31.52**	30.67

Reliable environment for RL-based recommenders

Zhang et al. LLM-Powered User Simulator for Recommender System. AAAI 2025.

Part 3: LLM as User Simulator

(1) RL-based simulators are limited in action space, action space, and training instability

(2) LLMs open up a new paradigm for simulating users

(3) They can give feedback for RL-based recommenders

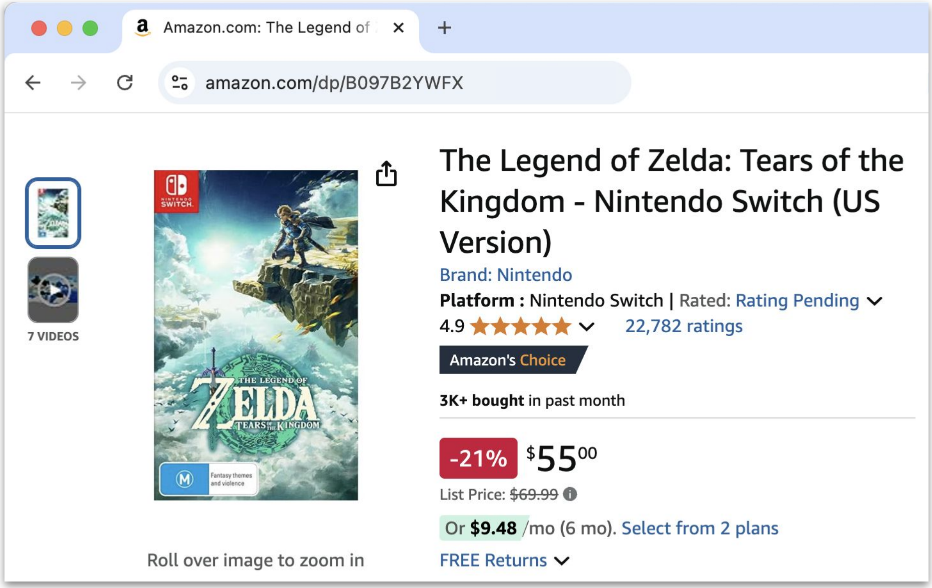
(4) Challenges:
scaling, training, industry deployment

03

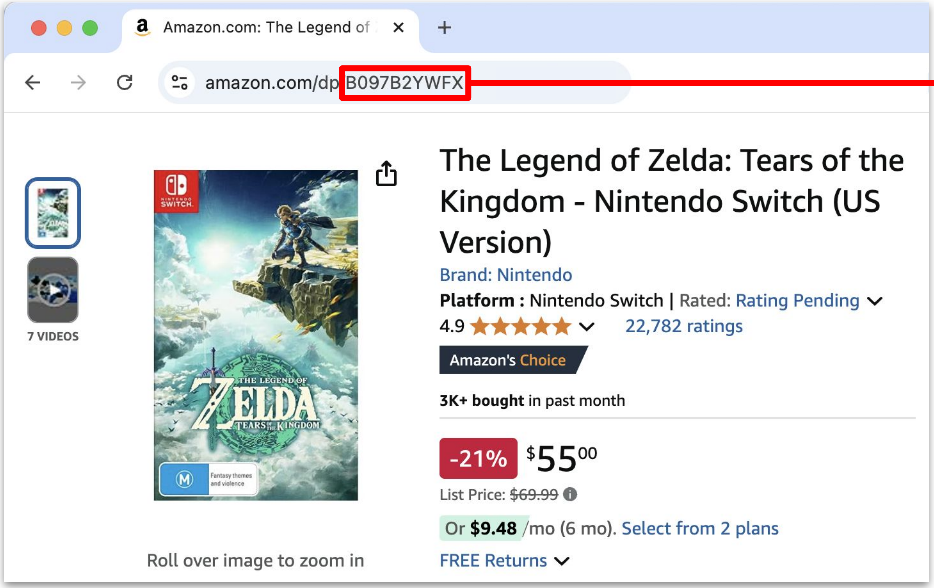
Semantic ID

–based Generative Recommendation

How to Index an Item in RecSys?



How to Index an Item in RecSys?

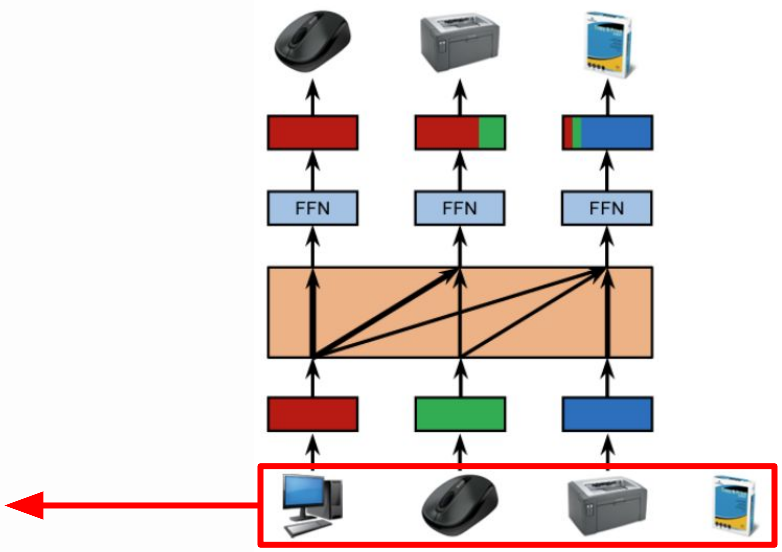


Item ID:

B097B2YWFX

How to Index an Item in RecSys?

Example: SASRec [*ICDM'18*]



Each item is indexed by a unique **item ID**

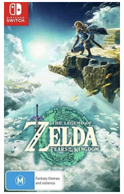
How to Index an Item in LLMs

I wanna some popular Nintendo games

How to Index an Item in LLMs

I wanna some popular Nintendo games

LLMs

How about 

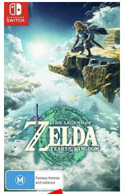
or 

?

How to Index an Item in LLMs

I wanna some popular Nintendo games

LLMs

How about 

or 

?

How to index items in LLMs? **Item ID**?

How to Index an Item in LLMs

How many tokens in LLMs?

Meta	Llama 3	~128,000
OpenAI	GPT-4o	~200,000
Google DeepMind	Gemma 2	~256,000

How to Index an Item in LLMs

How many tokens in LLMs?

Meta	Llama 3	~128,000
OpenAI	GPT-4o	~200,000
Google DeepMind	Gemma 2	~256,000

How many item IDs?

Amazon-Reviews-2023	~48,200,000

How to Index an Item in LLMs

How many tokens/item IDs in LLMs/RecSys?

Difficult to align these vocabularies given so many tokens

~128,000

~200,000

~256,000

~48,200,000

How to Index an Item in LLMs

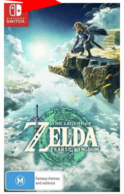
Is there a way to
index a large volume of items
using a compact vocabulary?

Semantic IDs

(also called: SemID or SID)

A few tokens that jointly index one item.

t3, t321, t643, t1011



Semantic IDs

(also called: SemID or SID)

A few tokens that jointly index one item.

t3, t321, t643, t1011

{t257, t258, …, t320, t321, t322, …, t511, t512}

Each token from a vocabulary shared by all items

Semantic IDs
(also called: SemID or SID)

A few tokens that jointly index one item.

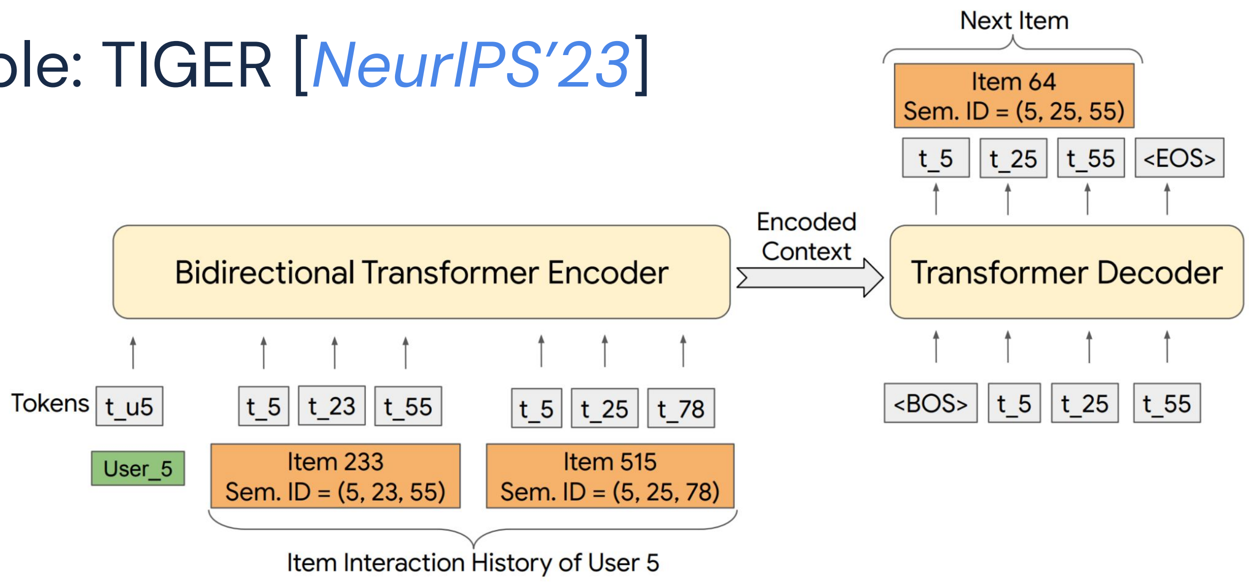
`t3, t321, t643, t1011`

Can index maximally $256^4 \approx 4.3 \times 10^9$ items with 1024 tokens

(4 tokens per item, each from a vocabulary of 256)

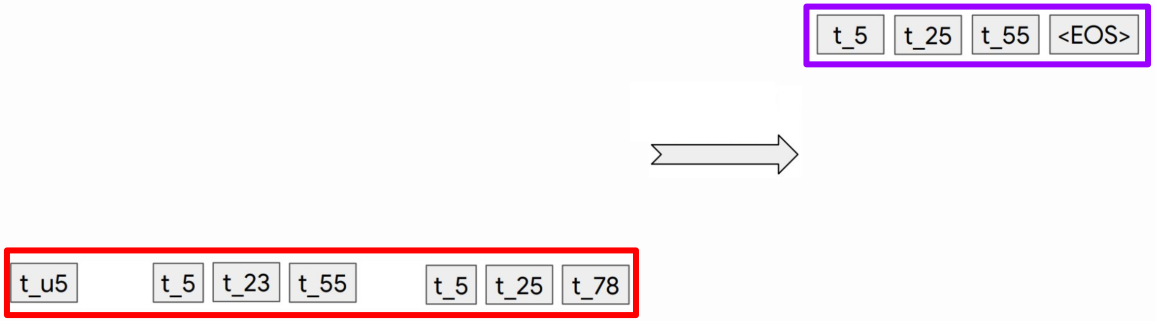
Generative Models based on Semantic IDs

Example: TIGER [*NeurIPS'23*]



Generative Models based on Semantic IDs

Example: TIGER [*NeurIPS'23*]



Recommendation as a **seq**-to-**seq** generation problem

Rajput et al. Recommender Systems with Generative Retrieval. NeurIPS 2023.

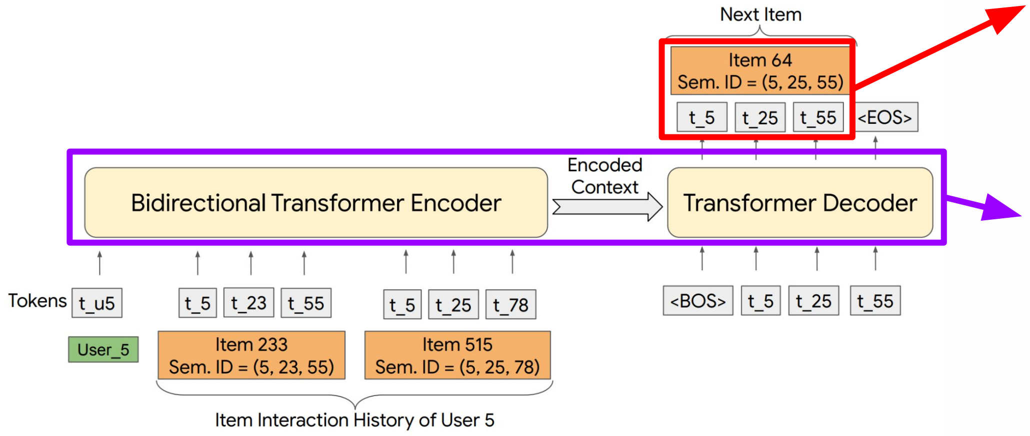
Generative Models based on Semantic IDs

Recommendation as a **seq**-to-**seq** generation problem

Input: user interacted items $\{c_{11}, c_{12}, c_{13}, c_{14}, c_{21}, c_{22}, \ldots\}$

Output: next item $\{c_{t1}, c_{t2}, c_{t3}, c_{t4}\}$

SemID-based Generative Recommendation



Part 1:

How to construct SIDs

Part 2:

How to build SID-based Generative Rec Models

Rajput et al. Recommender Systems with Generative Retrieval. NeurIPS 2023.

Part 1: Semantic ID Construction

Semantic ID Construction

Input: all data associated with the item (description, title, interactions, features, …)

Output: mapping between items ⇔ Semantic IDs

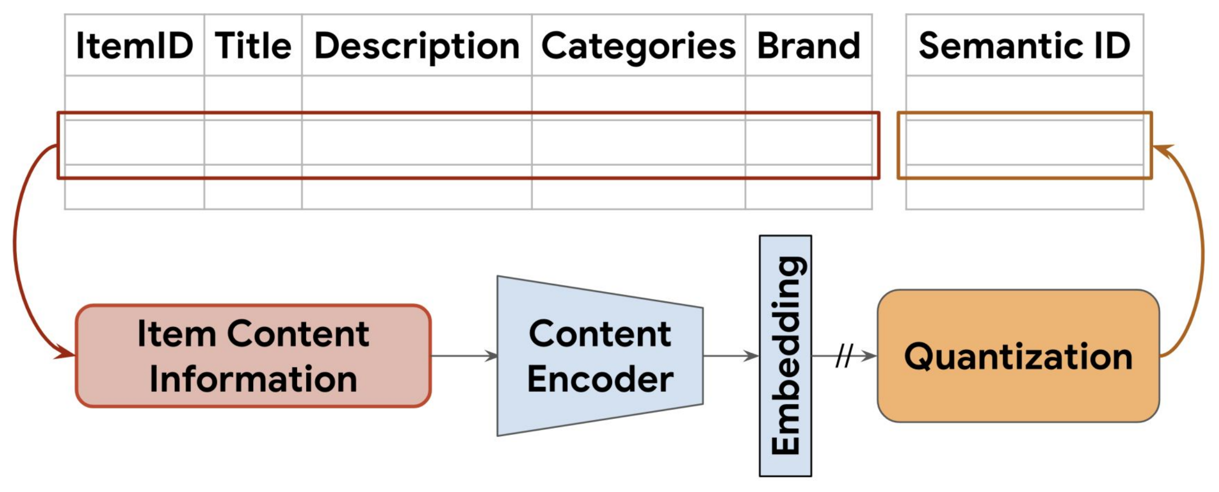
B097B2YWFX ⇔ {t3, t321, t643, t1011}

Part 1: Semantic ID Construction

(i) **First example**: TIGER and RQ-VAE-based SemIDs;

SemID Construction – First Example: TIGER

Input: concatenated text features

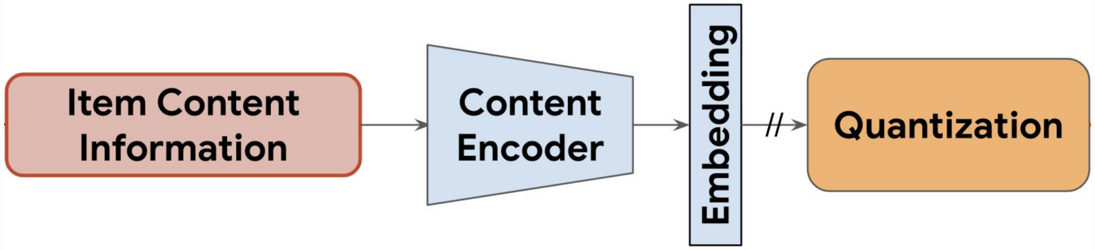


Output: mapping between items ⇔ Semantic IDs

Rajput et al. Recommender Systems with Generative Retrieval. NeurIPS 2023.

SemID Construction – First Example: TIGER

Text ➤ Vector ➤ IDs



SemID Construction – First Example: TIGER

1. Item Content Information (Text)

ItemID Title Description

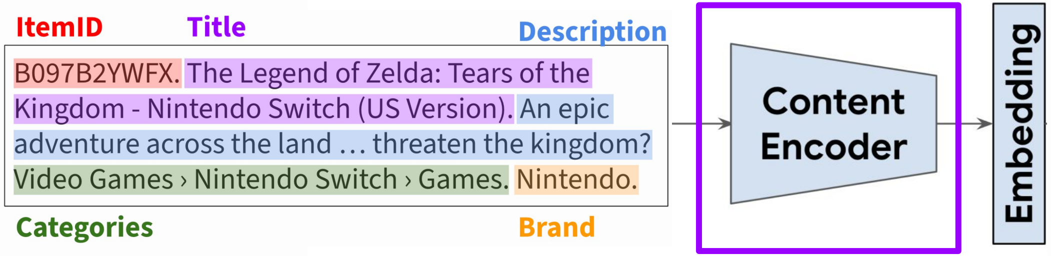
B097B2YWFX. The Legend of Zelda: Tears of the Kingdom - Nintendo Switch (US Version). An epic adventure across the land … threaten the kingdom? Video Games › Nintendo Switch › Games. Nintendo.

Categories Brand

SemID Construction – First Example: TIGER

2. Content Encoder + Embedding (Text ➤ Vector)

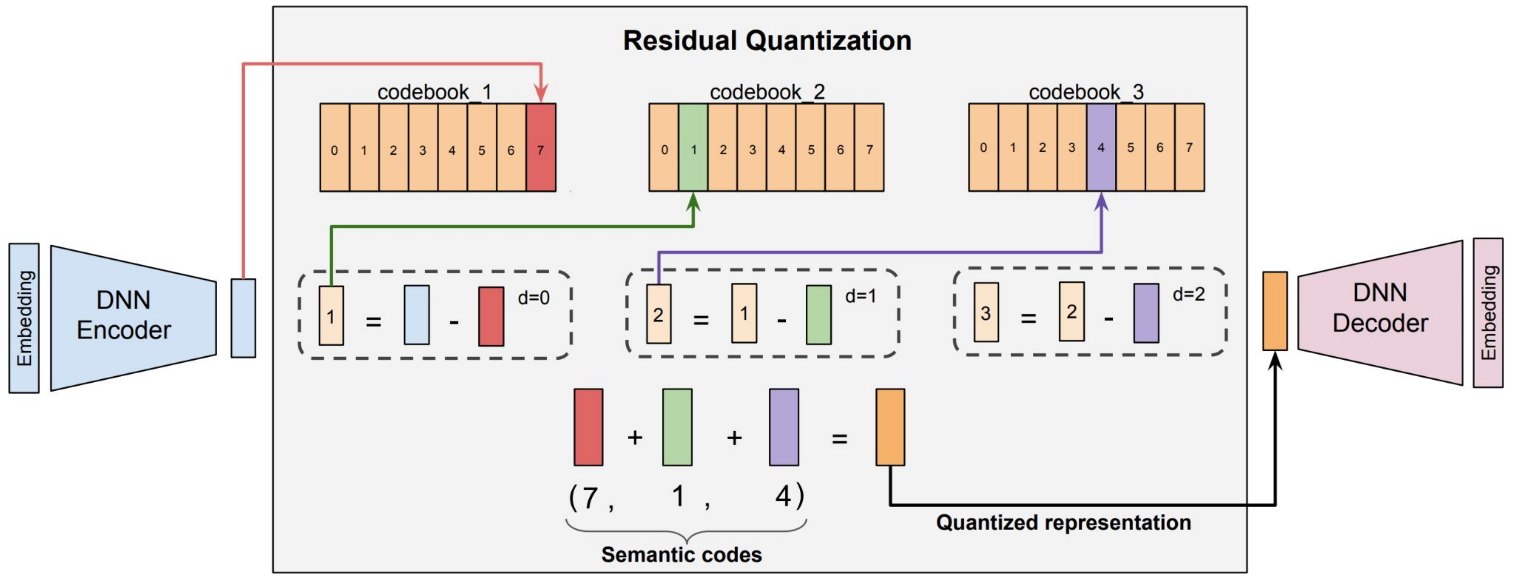
Pre-trained (fixed) sentence embedding model
(**SentenceT5**)



Ni et al. Sentence-T5: Scalable Sentence Encoders from Pre-trained Text-to-Text Models. Findings of ACL 2022.
Rajput et al. Recommender Systems with Generative Retrieval. NeurIPS 2023.

SemID Construction – First Example: TIGER

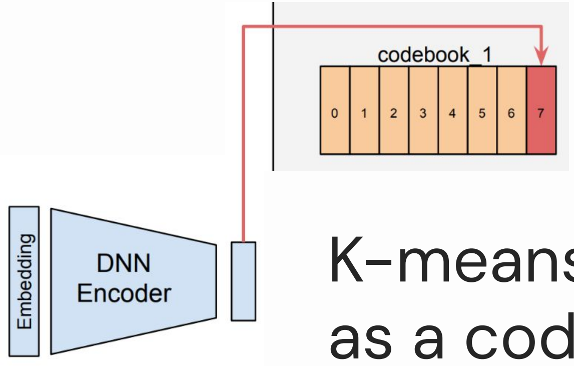
3. RQ-VAE Quantization (Vector ➤ IDs)



Zeghidour et al. SoundStream: An End-to-End Neural Audio Codec. TASLP 2022.
Rajput et al. Recommender Systems with Generative Retrieval. NeurIPS 2023.

SemID Construction – First Example: TIGER

3. RQ-VAE Quantization (Vector ➤ IDs)

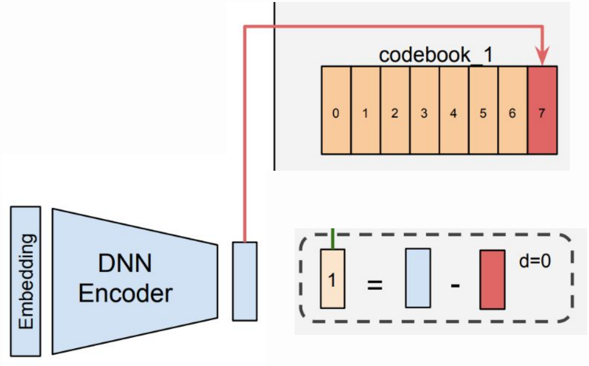


K-means: cluster center ID as a code in the codebook

Zeghidour et al. SoundStream: An End-to-End Neural Audio Codec. TASLP 2022.
Rajput et al. Recommender Systems with Generative Retrieval. NeurIPS 2023.

SemID Construction – First Example: TIGER

3. RQ-VAE Quantization (Vector ➤ IDs)

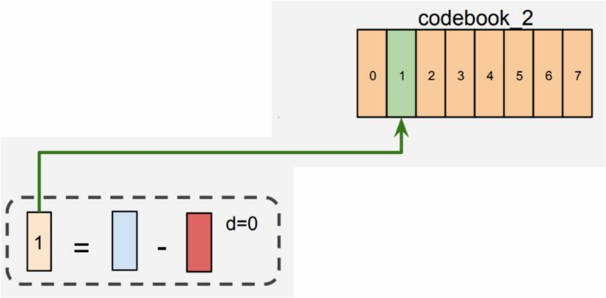


Residual of “input vector” and “clustering center vector”

Zeghidour et al. SoundStream: An End-to-End Neural Audio Codec. TASLP 2022.
Rajput et al. Recommender Systems with Generative Retrieval. NeurIPS 2023.

SemID Construction – First Example: TIGER

3. RQ-VAE Quantization (Vector ➤ IDs)

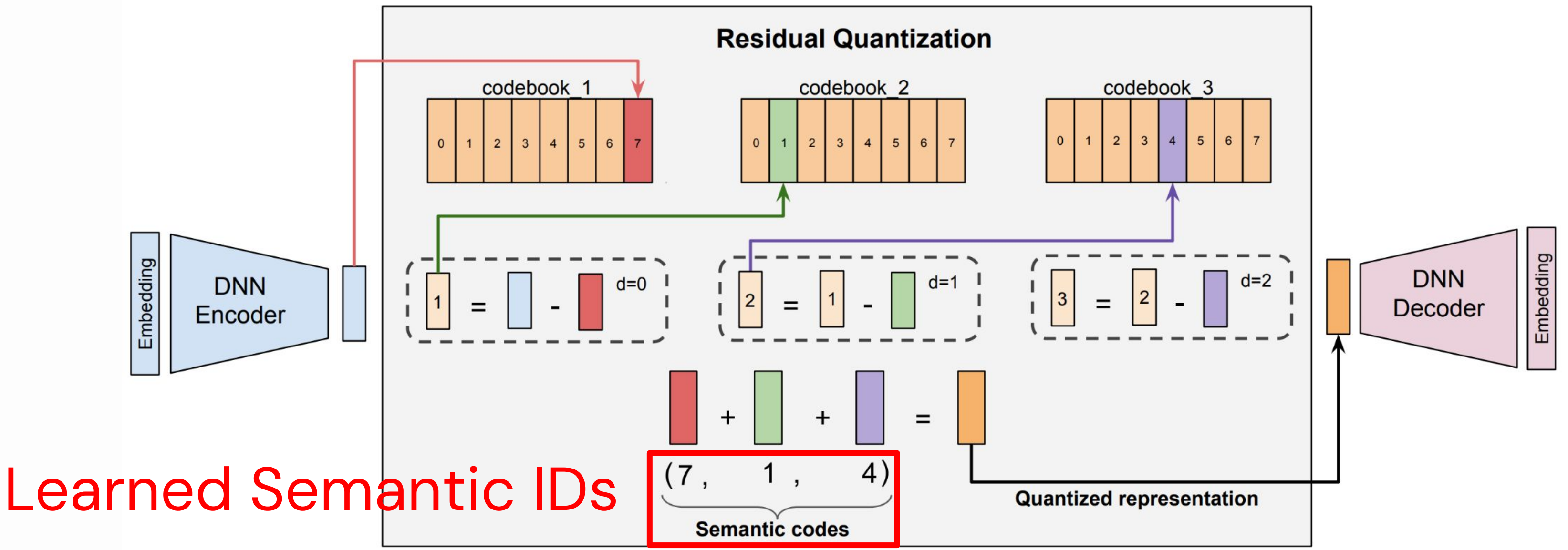


Residual as next level's input

Zeghidour et al. SoundStream: An End-to-End Neural Audio Codec. TASLP 2022.
Rajput et al. Recommender Systems with Generative Retrieval. NeurIPS 2023.

SemID Construction – First Example: TIGER

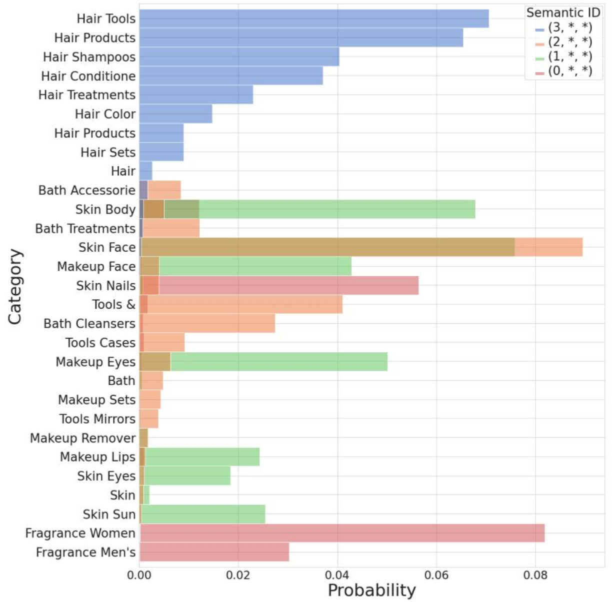
3. RQ-VAE Quantization (Vector ➤ IDs)



SemID Construction – First Example: TIGER

Properties of **RQ-VAE**-based SemIDs

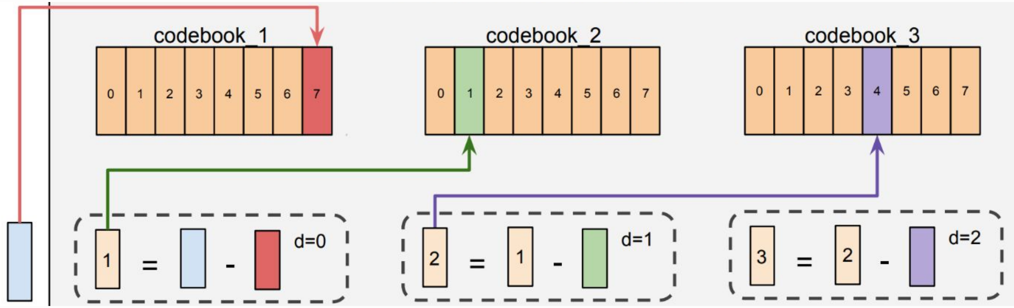
1. Semantic;



SemID Construction – First Example: TIGER

Properties of **RQ-VAE**-based SemIDs

1. Semantic;
2. Ordered / sequential dependent;



SemID Construction – First Example: TIGER

Collisions

(12, 24, 52)

(12, 24, 52)

SemID Construction – First Example: TIGER

Collisions

(12, 24, 52, 0)

(12, 24, 52, 1)

One extra token
to avoid conflicts

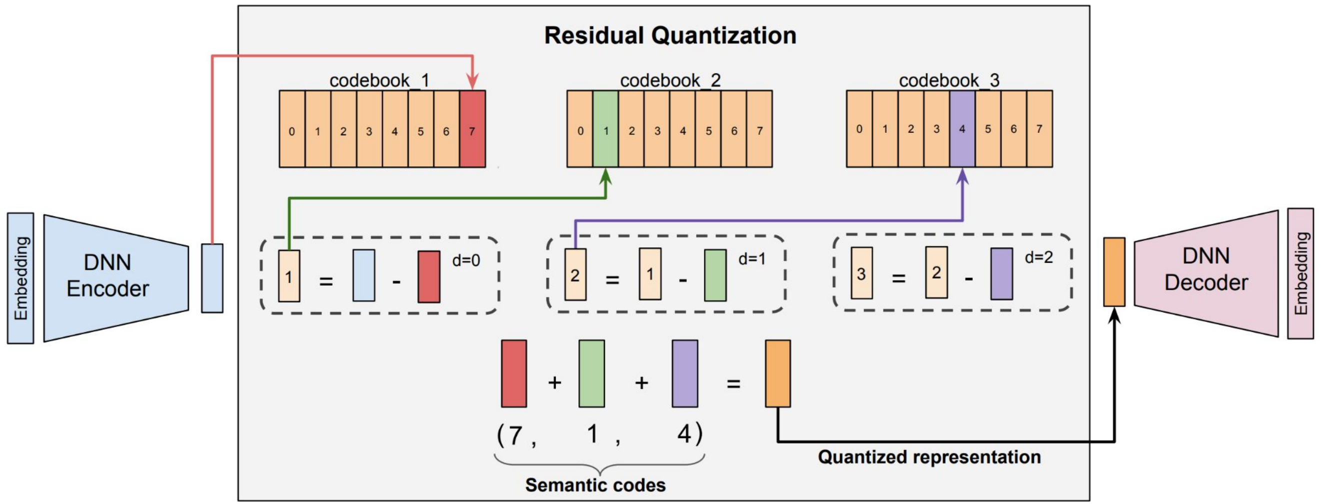
Rajput et al. Recommender Systems with Generative Retrieval. NeurIPS 2023.

Part 1: Semantic ID Construction

(i) First example: TIGER and RQ-VAE-based SemIDs;
(ii) **Techniques** to construct SemIDs;

Techniques to Construct SemIDs

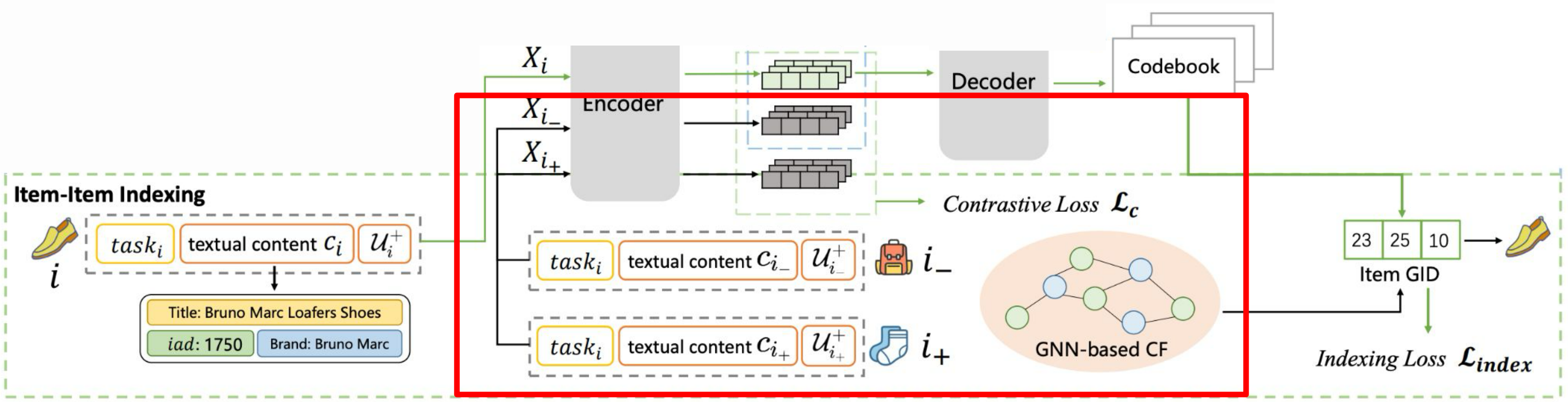
Residual Quantization



Rajput et al. Recommender Systems with Generative Retrieval. NeurIPS 2023.

Techniques to Construct SemIDs

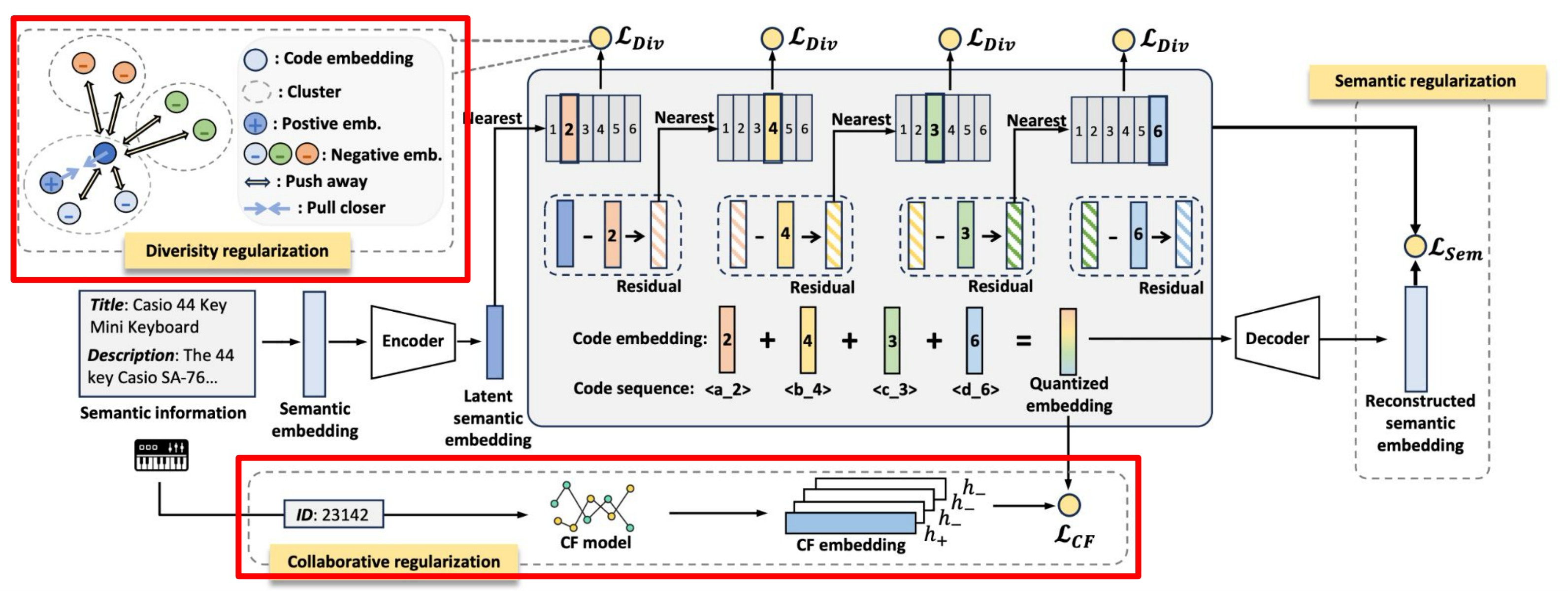
Residual Quantization + Item-level Regularization



Wang et al. Content-Based Collaborative Generation for Recommender Systems. CIKM 2024.

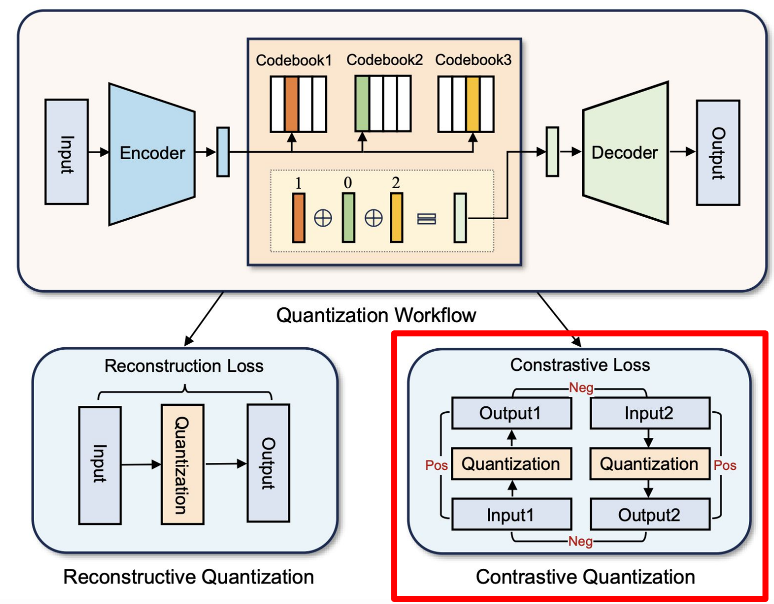
Techniques to Construct SemIDs

Residual Quantization + Item-level Regularization



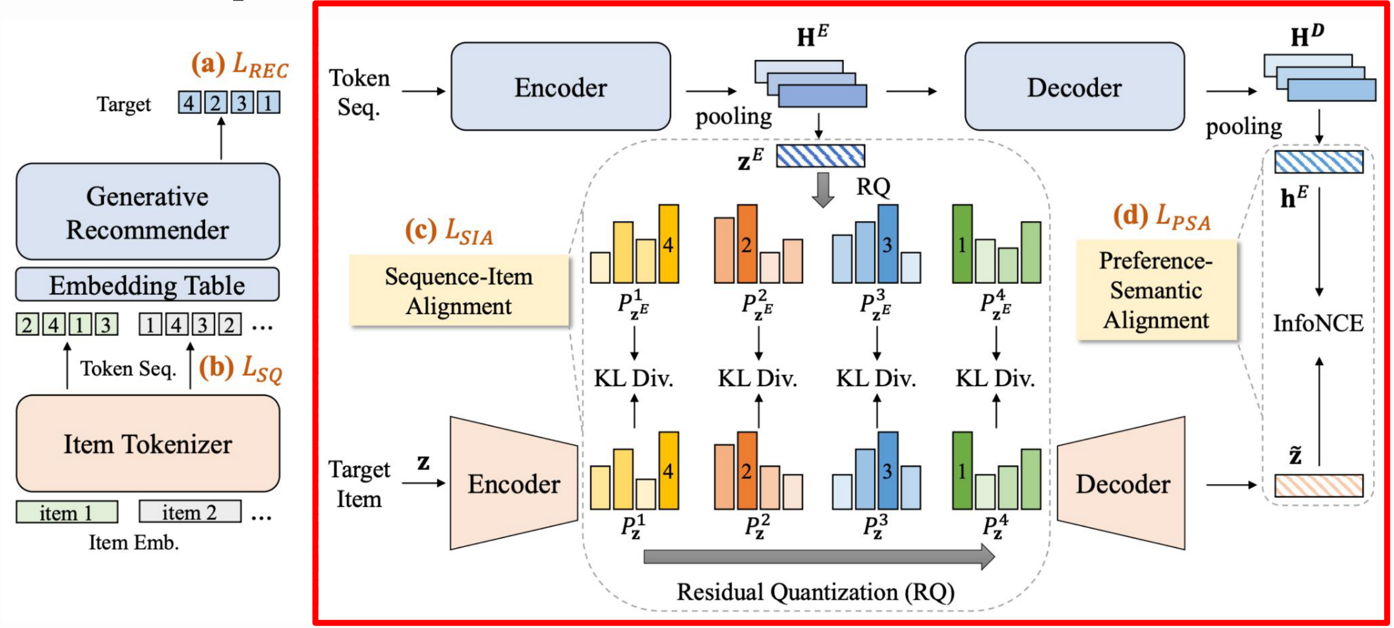
Techniques to Construct SemIDs

Residual Quantization + Item-level Regularization



Techniques to Construct SemIDs

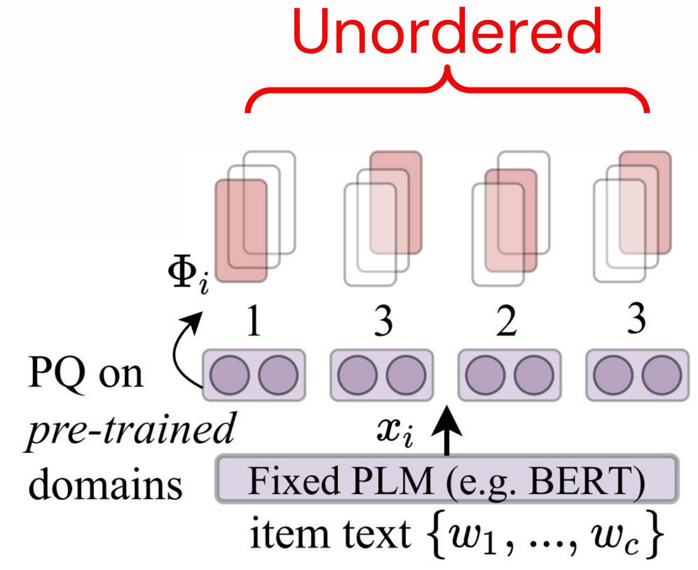
Residual Quantization + Recommendation Loss



Liu et al. End-to-End Learnable Item Tokenization for Generative Recommendation. arXiv:2409.05546.

Techniques to Construct SemIDs

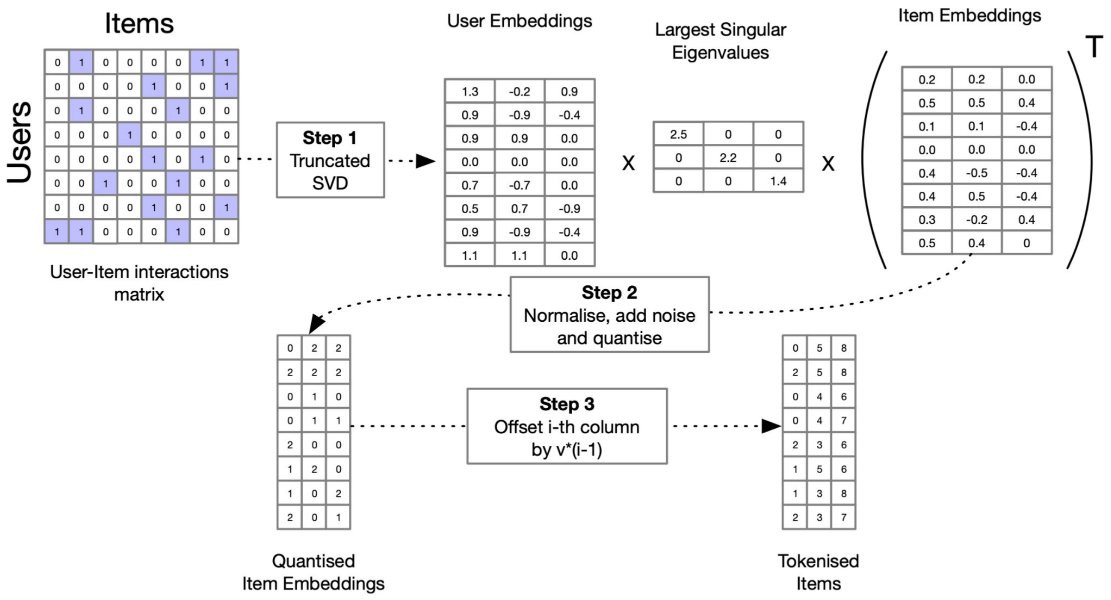
Product Quantization



Hou et al. Learning Vector-Quantized Item Representation for Transferable Sequential Recommenders. WWW 2023.

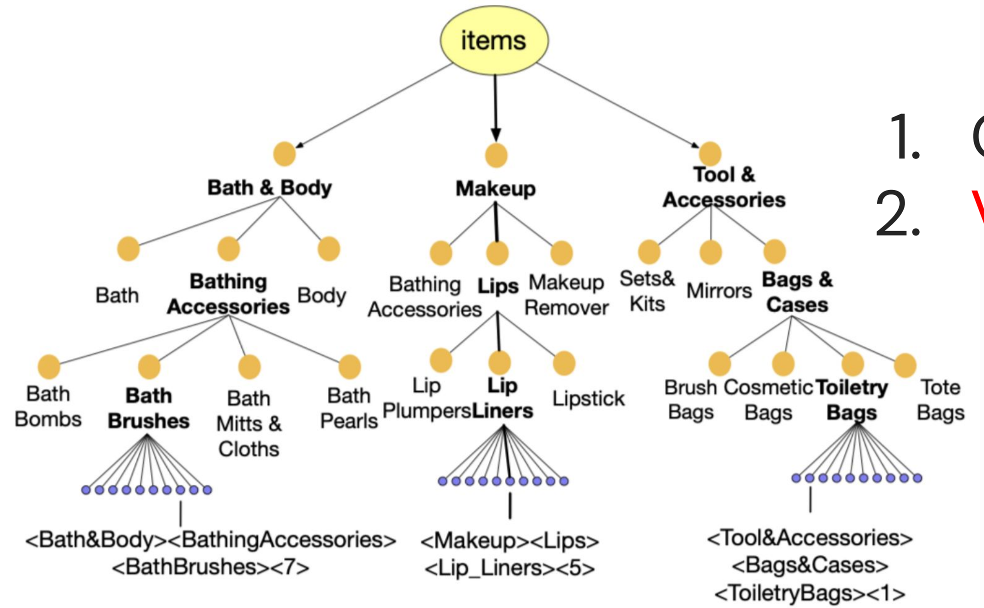
Techniques to Construct SemIDs

Product Quantization



Techniques to Construct SemIDs

Hierarchical Clustering (Heuristics-based)

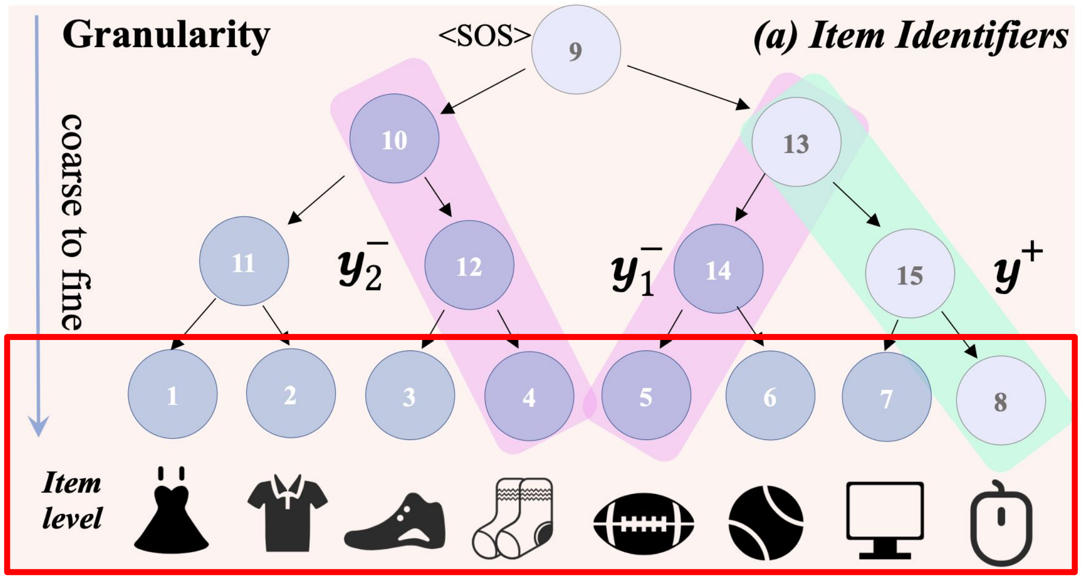


1. Ordered;
2. Variable-length SemIDs;

Hua et al. How to Index Item IDs for Recommendation Foundation Models. SIGIR-AP 2023.

Techniques to Construct SemIDs

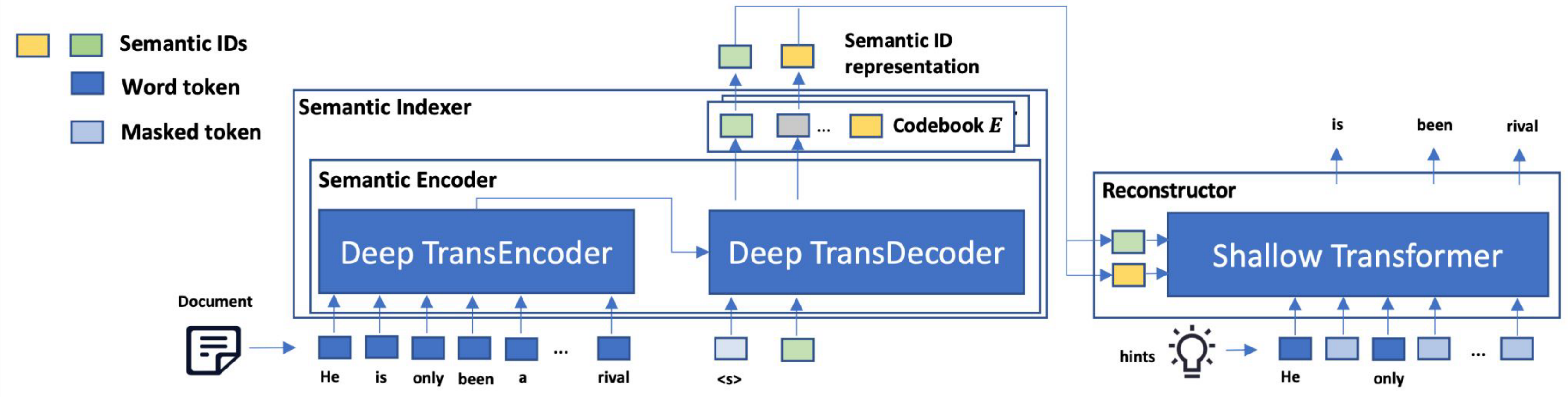
Hierarchical Clustering (Latent-based)



Si et al. Generative Retrieval with Semantic Tree-Structured Item Identifiers via Contrastive Learning. SIGIR-AP 2024.

Techniques to Construct SemIDs

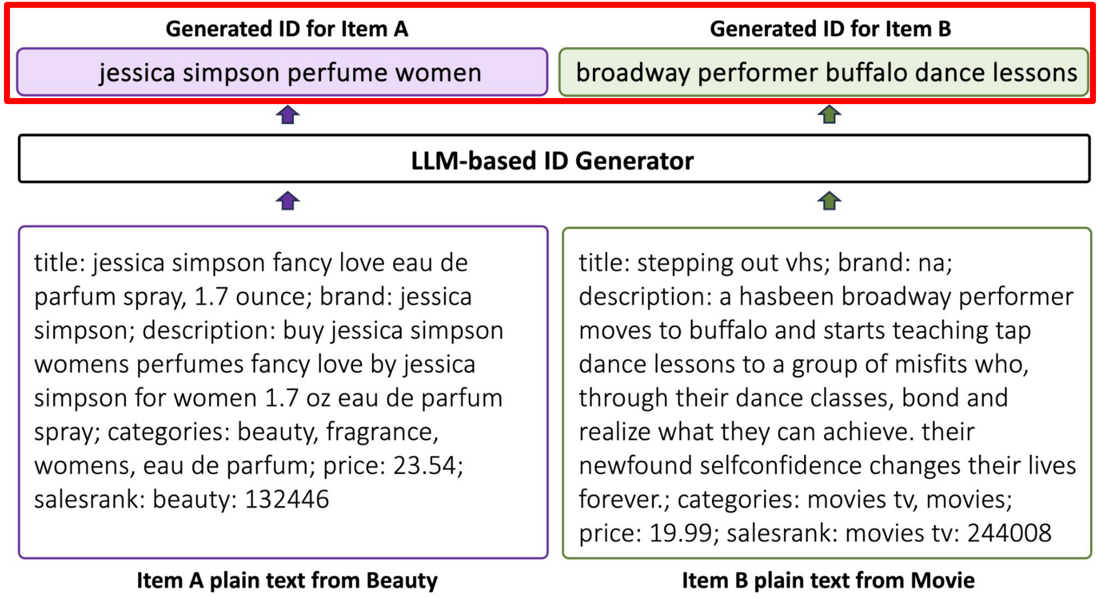
Language Model-based ID Generator



Natural language as inputs;
SemIDs as outputs

Techniques to Construct SemIDs

Language Model-based ID Generator



Words as SemIDs (like tagging)

Tan et al. IDGenRec: LLM-RecSys Alignment with Textual ID Learning. SIGIR 2024.

Techniques to Construct SemIDs

Context-independent

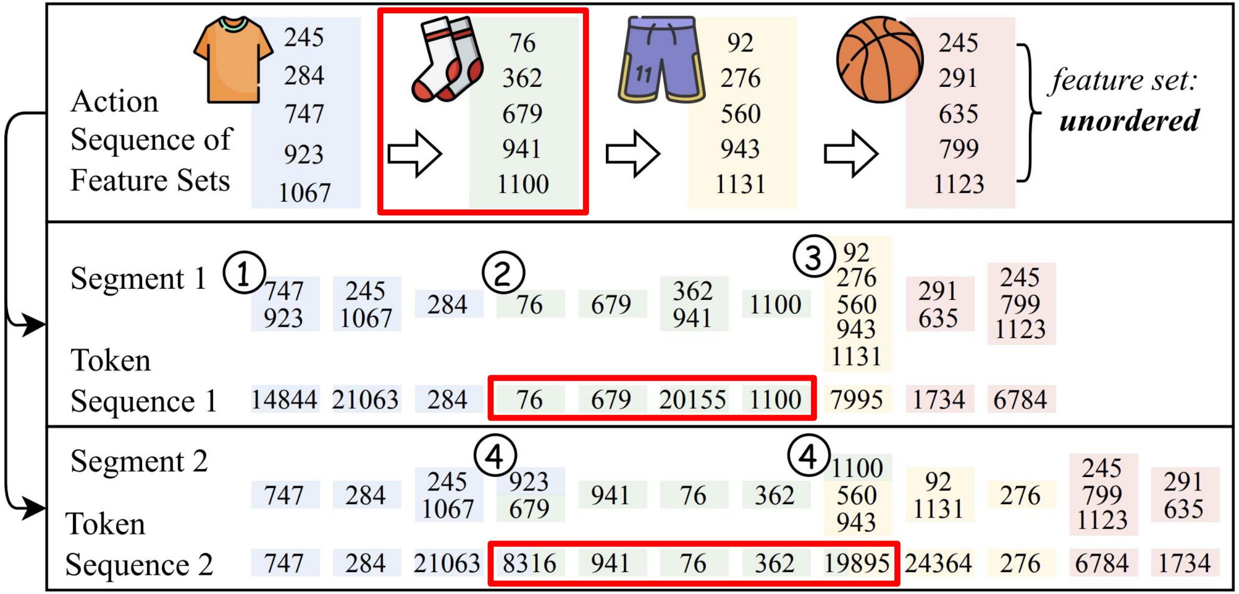
Action Tokenization	Example	Contextual	Unordered
Product Quantization	VQ-Rec (Hou et al., 2023)	✗	✓
Hierarchical Clustering	P5-CID (Hua et al., 2023)	✗	✗
Residual Quantization	TIGER (Rajput et al., 2023)	✗	✗
Text Tokenization	LMIndexer (Jin et al., 2024)	✗	✗
Raw Features	HSTU (Zhai et al., 2024)	✗	✗
SentencePiece	SPM-SID (Singh et al., 2024)	✗	✗

Same item ⇒ fixed semIDs in all sequences

Hou et al. ActionPiece: Contextually Tokenizing Action Sequences for Generative Recommendation. arXiv:2502.13581.

Techniques to Construct SemIDs

Context-independent ⇒ Context-aware

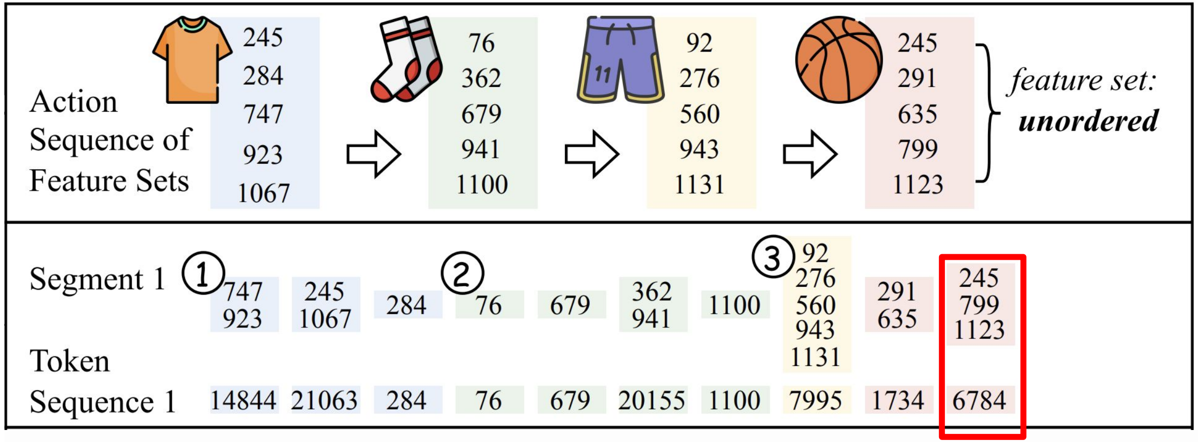


Same item ⇒ different semIDs based on context

Hou et al. ActionPiece: Contextually Tokenizing Action Sequences for Generative Recommendation. arXiv:2502.13581.

Techniques to Construct SemIDs

Context-independent ⇒ Context-aware



Core Idea:

Merge frequently co-occurring features as new tokens

(**ActionPiece**: "WordPiece" tokenization for generative rec)

Hou et al. ActionPiece: Contextually Tokenizing Action Sequences for Generative Recommendation. arXiv:2502.13581.

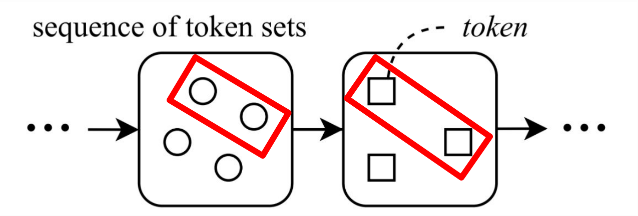
Techniques to Construct SemIDs

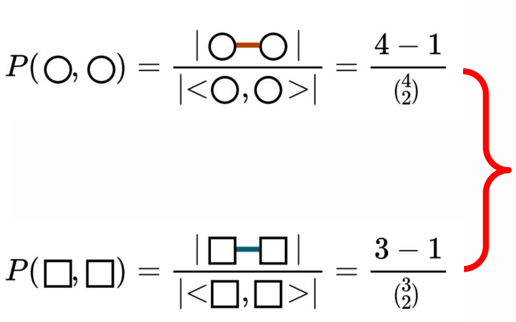
Context-independent ⇒ Context-aware

Algorithm 1 ActionPiece Vocabulary Construction

input Sequence corpus $\mathcal{S}'$, initial tokens $\mathcal{V}_0$, target size Q
output Merge rules $\mathcal{R}$, constructed vocabulary $\mathcal{V}$

Initialize vocabulary $\mathcal{V} \leftarrow \mathcal{V}_0$ # each initial token corresponds to one unique item feature
$\mathcal{R} \leftarrow \emptyset$
while $|\mathcal{V}| < Q$ **do**
  # ***Count:*** accumulate weighted token co-occurrences
  $\mathrm{count}(\cdot,\cdot) \leftarrow \mathrm{Count}(\mathcal{S}', \mathcal{V})$ # Algorithm 2
  # ***Update:*** Merge a frequent token pair into a new token
  Select $(c_u, c_v) \leftarrow \arg\max_{(c_i, c_j)} \mathrm{count}(c_i, c_j)$
  $\mathcal{S}' \leftarrow \mathrm{Update}(\mathcal{S}', \{(c_u, c_v) \rightarrow c_{\mathrm{new}}\})$ # Algorithm 3
  $\mathcal{R} \leftarrow \mathcal{R} \cup \{(c_u, c_v) \rightarrow c_{\mathrm{new}}\}$ # new merge rule
  $\mathcal{V} \leftarrow \mathcal{V} \cup \{c_{\mathrm{new}}\}$ # add new token to the vocabulary
end while
return $\mathcal{R}, \mathcal{V}$





Features co-occurring within items

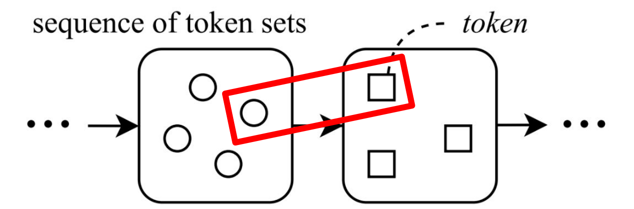
Techniques to Construct SemIDs

Context-independent ⇒ Context-aware

Algorithm 1 ActionPiece Vocabulary Construction

input Sequence corpus $\mathcal{S}'$, initial tokens $\mathcal{V}_0$, target size Q
output Merge rules $\mathcal{R}$, constructed vocabulary $\mathcal{V}$

Initialize vocabulary $\mathcal{V} \leftarrow \mathcal{V}_0$ # each initial token corresponds to one unique item feature
$\mathcal{R} \leftarrow \emptyset$
while $|\mathcal{V}| < Q$ **do**
 # ***Count:*** accumulate weighted token co-occurrences
 $\text{count}(\cdot,\cdot) \leftarrow \text{Count}(\mathcal{S}', \mathcal{V})$ # Algorithm 2
 # ***Update:*** Merge a frequent token pair into a new token
 Select $(c_u, c_v) \leftarrow \arg\max_{(c_i, c_j)} \text{count}(c_i, c_j)$
 $\mathcal{S}' \leftarrow \text{Update}(\mathcal{S}', \{(c_u, c_v) \rightarrow c_{\text{new}}\})$ # Algorithm 3
 $\mathcal{R} \leftarrow \mathcal{R} \cup \{(c_u, c_v) \rightarrow c_{\text{new}}\}$ # new merge rule
 $\mathcal{V} \leftarrow \mathcal{V} \cup \{c_{\text{new}}\}$ # add new token to the vocabulary
end while
return $\mathcal{R}, \mathcal{V}$



$$P(\bigcirc,\bigcirc) = \frac{|\bigcirc\!-\!\bigcirc|}{|<\bigcirc,\bigcirc>|} = \frac{4-1}{\binom{4}{2}}$$

$$P(\bigcirc,\square) = \frac{|\bigcirc\!-\!\square|}{|\bigcirc| \times |\square|} = \frac{1}{4 \times 3}$$

$$P(\square,\square) = \frac{|\square\!-\!\square|}{|<\square,\square>|} = \frac{3-1}{\binom{3}{2}}$$

Features co-occurring within or across items

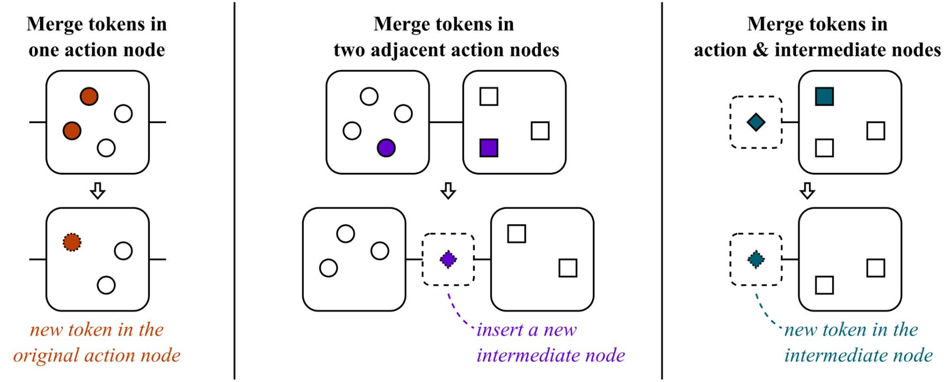
Techniques to Construct SemIDs

Context-independent ⇒ Context-aware

Algorithm 1 ActionPiece Vocabulary Construction

input Sequence corpus $\mathcal{S}'$, initial tokens $\mathcal{V}_0$, target size Q
output Merge rules $\mathcal{R}$, constructed vocabulary $\mathcal{V}$

Initialize vocabulary $\mathcal{V} \leftarrow \mathcal{V}_0$ # each initial token corresponds to one unique item feature
$\mathcal{R} \leftarrow \emptyset$
while $|\mathcal{V}| < Q$ **do**
  # ***Count:*** accumulate weighted token co-occurrences
  $\text{count}(\cdot,\cdot) \leftarrow \text{Count}(\mathcal{S}', \mathcal{V})$ # Algorithm 2
  # ***Update:*** Merge a frequent token pair into a new token
  Select $(c_u, c_v) \leftarrow \arg\max_{(c_i, c_j)} \text{count}(c_i, c_j)$
  $\mathcal{S}' \leftarrow \text{Update}(\mathcal{S}', \{(c_u, c_v) \rightarrow c_{\text{new}}\})$ # Algorithm 3
  $\mathcal{R} \leftarrow \mathcal{R} \cup \{(c_u, c_v) \rightarrow c_{\text{new}}\}$ # new merge rule
  $\mathcal{V} \leftarrow \mathcal{V} \cup \{c_{\text{new}}\}$ # add new token to the vocabulary
end while
return $\mathcal{R}, \mathcal{V}$



Summary of
Techniques to Construct SemIDs

Context-independent

- Residual Quantization (+ regularization)
- Product Quantization
- Hierarchical Clustering
- LM-based ID Generator

Summary of Techniques to Construct SemIDs

Context-independent

- Residual Quantization (+ regularization)
- Product Quantization
- Hierarchical Clustering
- LM-based ID Generator

Context-aware

Part 1: Semantic ID Construction

(i) First example: TIGER and RQ-VAE-based SemIDs;
(ii) Techniques to construct SemIDs;
(iii) **Inputs** for SemID construction;

Inputs for SemID Construction

Input: all data associated with the item

Inputs for SemID Construction

Input: all data associated with the item

What exactly does “all data” mean? 🤷

Inputs for SemID Construction

Text or Multimodal Features

Text/Visual/Acoustic ➤ Vector ➤ IDs

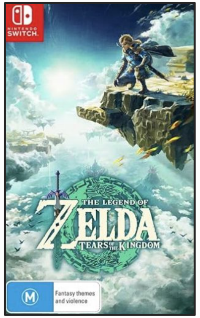
Pretrained Encoder Quantization

ItemID **Title** **Description**

B097B2YWFX. The Legend of Zelda: Tears of the Kingdom - Nintendo Switch (US Version). An epic adventure across the land … threaten the kingdom? Video Games › Nintendo Switch › Games. Nintendo.

Categories **Brand**

Text



Multimodal

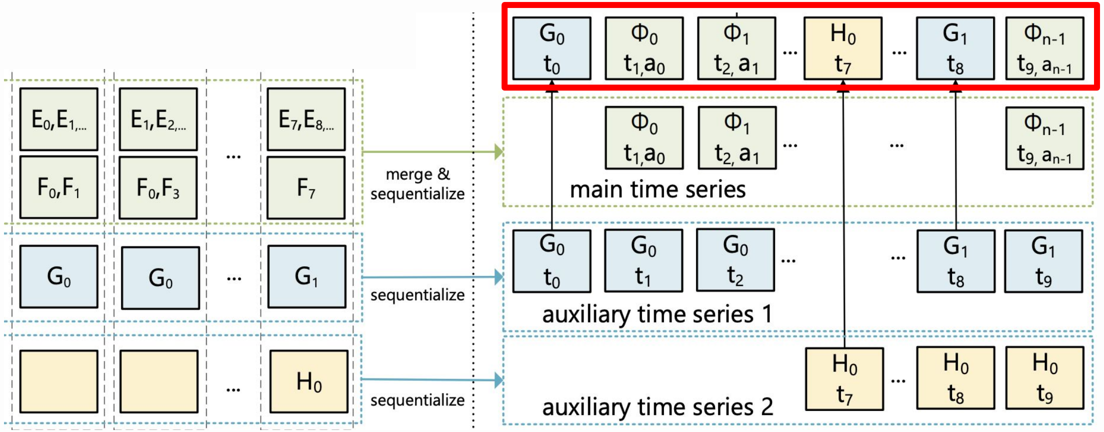
Deng et al. OneRec: Unifying Retrieve and Rank with Generative Recommender and Preference Alignment. arXiv:2502.18965.
Liu et al. MMGRec: Multimodal Generative Recommendation with Transformer Model. arXiv:2404.16555.

Inputs for SemID Construction

Categorical Features

Categorical Features ➤ IDs

Merge & Sequentialize



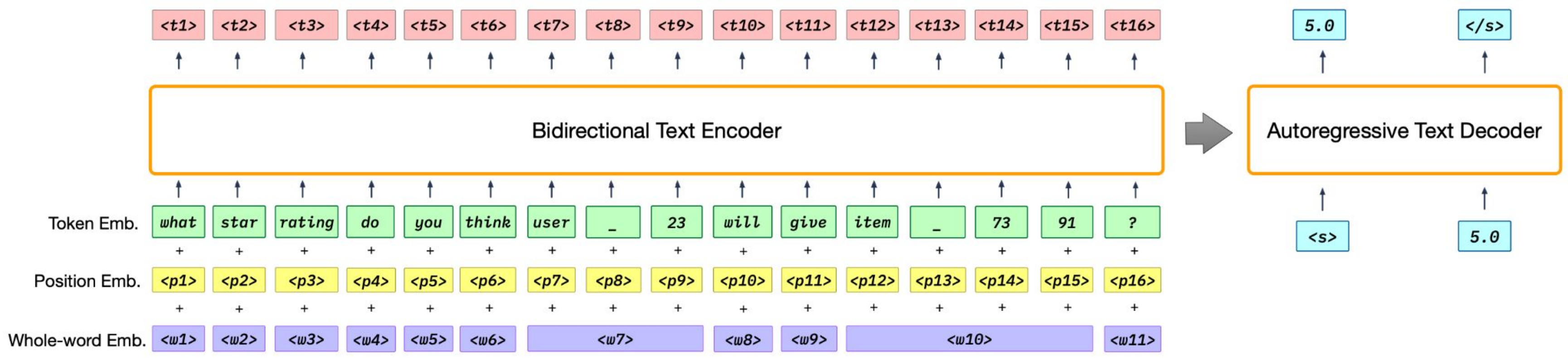
Zhai et al. Actions Speak Louder than Words: Trillion-Parameter Sequential Transducers for Generative Recommendations. ICML 2024.

Inputs for SemID Construction

No Features

Item ID ➤ IDs

Text Tokenizer



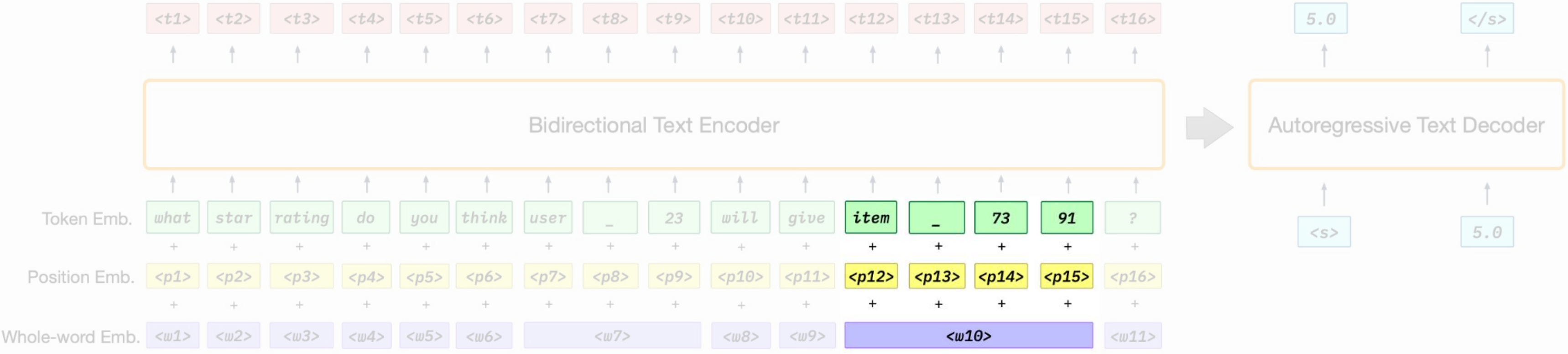
Geng et al. Recommendation as Language Processing (RLP): A Unified Pretrain, Personalized Prompt & Predict Paradigm (P5). RecSys 2022.

Inputs for SemID Construction

No Features

Item ID ➤ **IDs**

Text Tokenizer

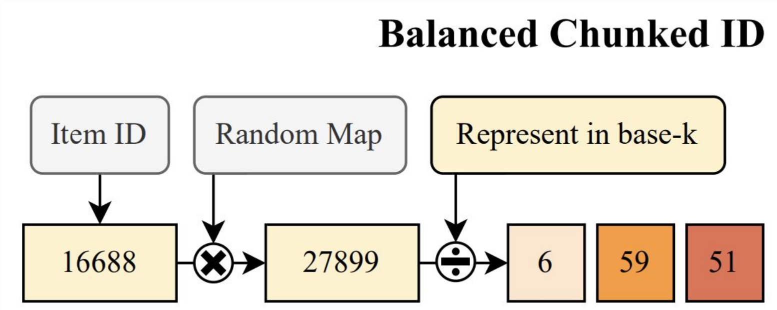


Geng et al. Recommendation as Language Processing (RLP): A Unified Pretrain, Personalized Prompt & Predict Paradigm (P5). RecSys 2022.

Inputs for SemID Construction

No Features

Random IDs



Inputs for SemID Construction

Input: all data associated with the item

(1) Item Metadata

Text / Multimodal / Categorical / No Features

Inputs for SemID Construction

Input: all data associated with the item

(1) Item Metadata

Text / Multimodal / Categorical / No Features

(2) Item Metadata + Behaviors

Inputs for SemID Construction

Input: all data associated with the item

(1) Item Metadata

Text / Multimodal / Categorical / No Features

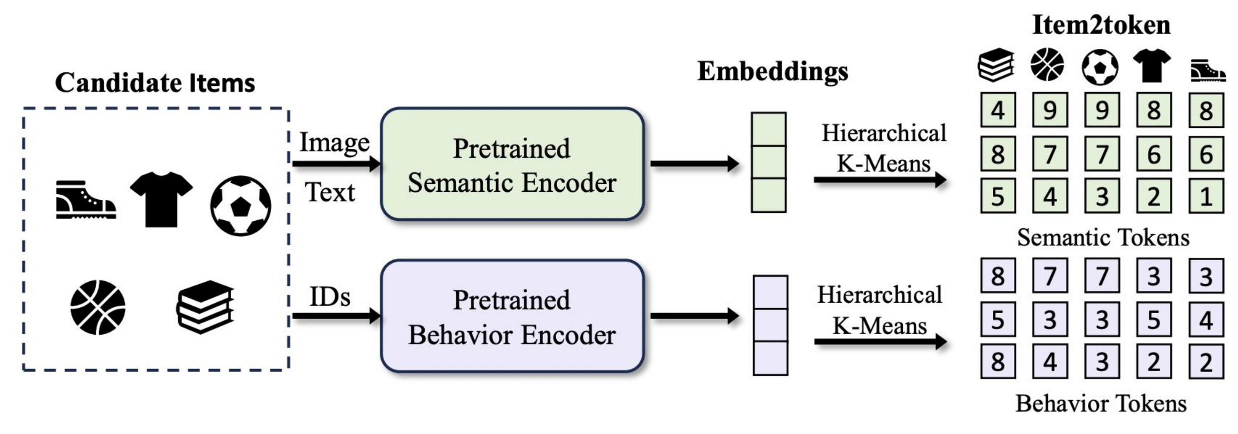
(2) Item Metadata + Behaviors

But how?

Inputs for SemID Construction

Item Metadata + Behaviors

Fused Semantic IDs

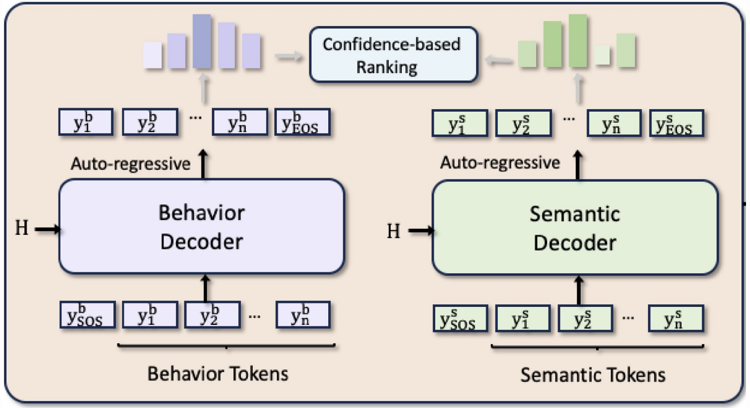


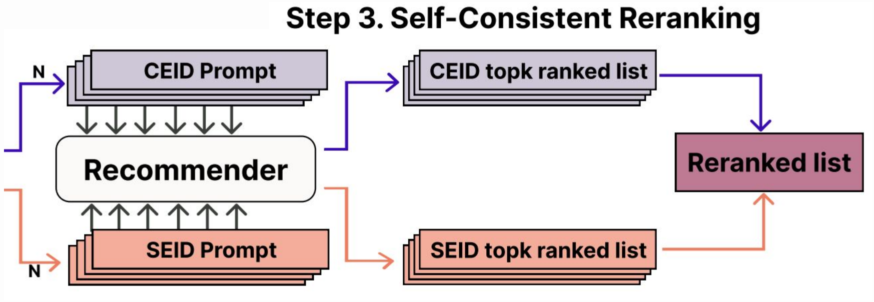
Wang et al. EAGER: Two-Stream Generative Recommender with Behavior-Semantic Collaboration. KDD 2024.

Inputs for SemID Construction

Item Metadata + Behaviors

Fused Semantic IDs + Two-stream Generation





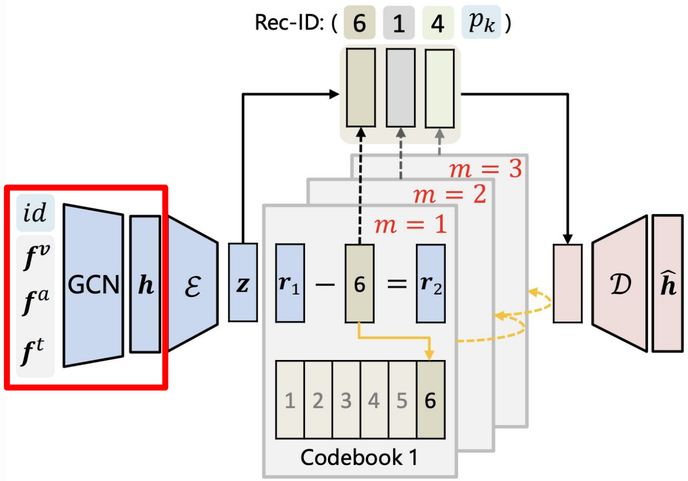
Wang et al. EAGER: Two-Stream Generative Recommender with Behavior-Semantic Collaboration. KDD 2024.
Kim et al. SC-Rec: Enhancing Generative Retrieval with Self-Consistent Reranking for Sequential Recommendation. arXiv:2408:08686.

Inputs for SemID Construction

Item Metadata + Behaviors

Fused Representations

User–Item Graph +
Semantic Features



Liu et al. MMGRec: Multimodal Generative Recommendation with Transformer Model. arXiv:2404.16555.

Inputs for SemID Construction

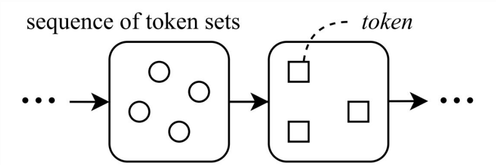
Item Metadata + Behaviors

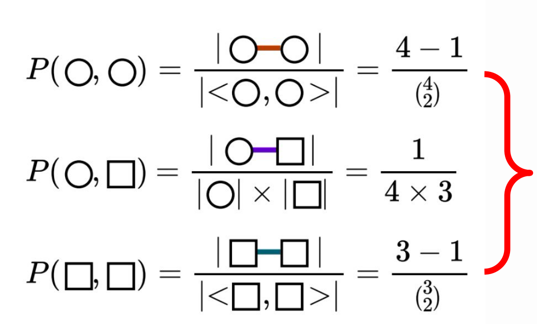
Train Tokenizer on Behavior Sequence Corpus

Algorithm 1 ActionPiece Vocabulary Construction

input Sequence corpus $\mathcal{S}'$, initial tokens $\mathcal{V}_0$, target size Q
output Merge rules $\mathcal{R}$, constructed vocabulary $\mathcal{V}$

Initialize vocabulary $\mathcal{V} \leftarrow \mathcal{V}_0$ # each initial token corresponds to one unique item feature
$\mathcal{R} \leftarrow \emptyset$
while $|\mathcal{V}| < Q$ **do**
  $\boxed{\text{\# \textit{Count:} accumulate weighted token co-occurrences}}$
  $\text{count}(\cdot,\cdot) \leftarrow \text{Count}(\mathcal{S}', \mathcal{V})$ # Algorithm 2
  # ***Update:*** Merge a frequent token pair into a new token
  Select $(c_u, c_v) \leftarrow \arg\max_{(c_i, c_j)} \text{count}(c_i, c_j)$
  $\mathcal{S}' \leftarrow \text{Update}(\mathcal{S}', \{(c_u, c_v) \rightarrow c_{\text{new}}\})$ # Algorithm 3
  $\mathcal{R} \leftarrow \mathcal{R} \cup \{(c_u, c_v) \rightarrow c_{\text{new}}\}$ # new merge rule
  $\mathcal{V} \leftarrow \mathcal{V} \cup \{c_{\text{new}}\}$ # add new token to the vocabulary
end while
return $\mathcal{R}, \mathcal{V}$





Features co-occurring within or across items

Inputs for SemID Construction

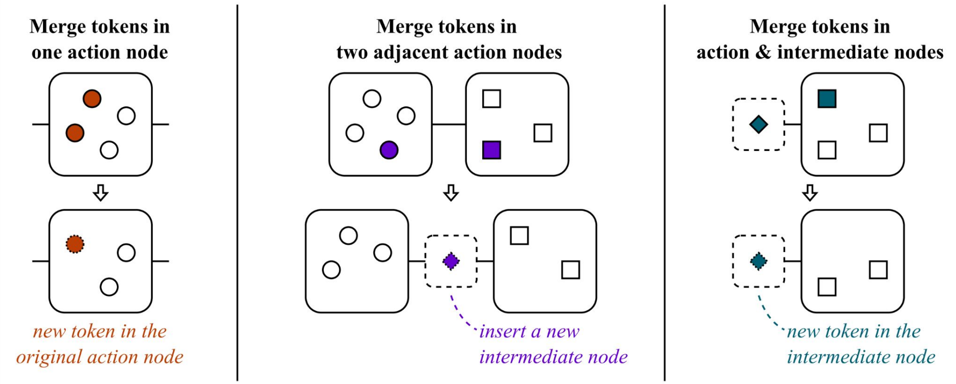
Item Metadata + Behaviors

Train Tokenizer on Behavior Sequence Corpus

Algorithm 1 ActionPiece Vocabulary Construction

input Sequence corpus $\mathcal{S}'$, initial tokens $\mathcal{V}_0$, target size Q
output Merge rules $\mathcal{R}$, constructed vocabulary $\mathcal{V}$

```
Initialize vocabulary V ← V_0 # each initial token corresponds to one unique item feature
R ← ∅
while |V| < Q do
    # Count: accumulate weighted token co-occurrences
    count(·,·) ← Count(S', V) # Algorithm 2
    # Update: Merge a frequent token pair into a new token
    Select (c_u, c_v) ← arg max_(c_i,c_j) count(c_i, c_j)
    S' ← Update(S', {(c_u, c_v) → c_new}) # Algorithm 3
    R ← R ∪ {(c_u, c_v) → c_new} # new merge rule
    V ← V ∪ {c_new} # add new token to the vocabulary
end while
return R, V
```



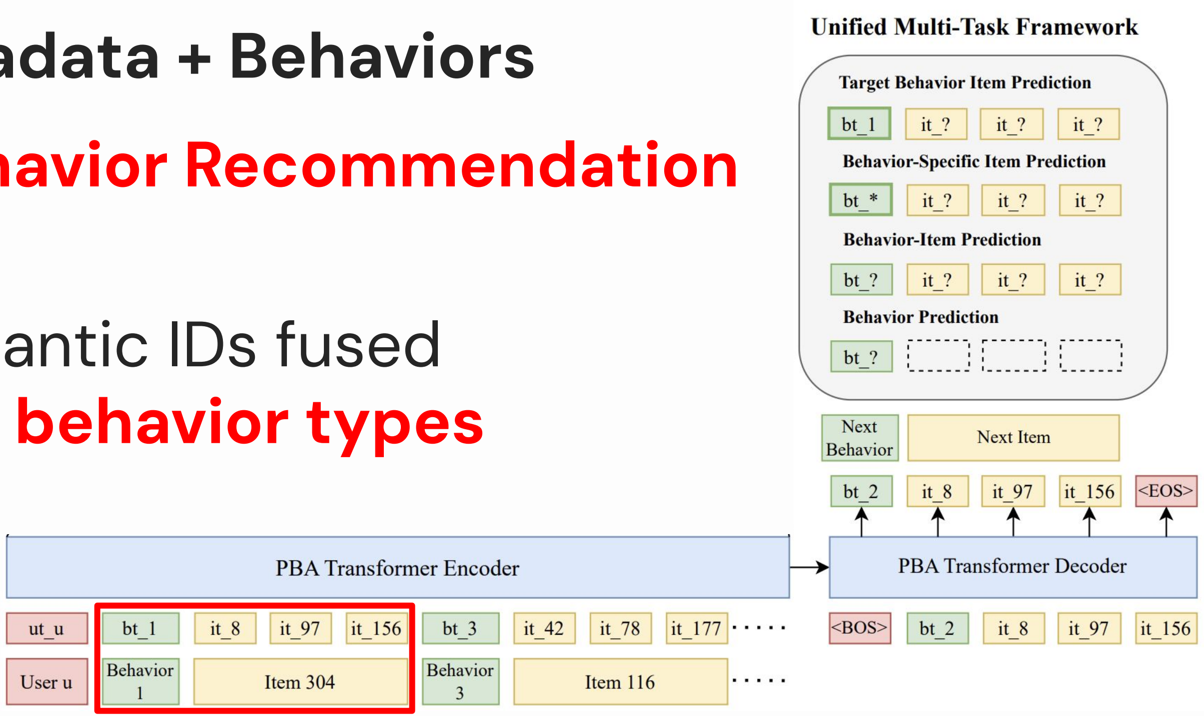
Hou et al. ActionPiece: Contextually Tokenizing Action Sequences for Generative Recommendation. arXiv:2502.13581.

Inputs for SemID Construction

Item Metadata + Behaviors

Multi-Behavior Recommendation

Semantic IDs fused
with **behavior types**



Liu et al. Multi-Behavior Generative Recommendation. CIKM 2024.

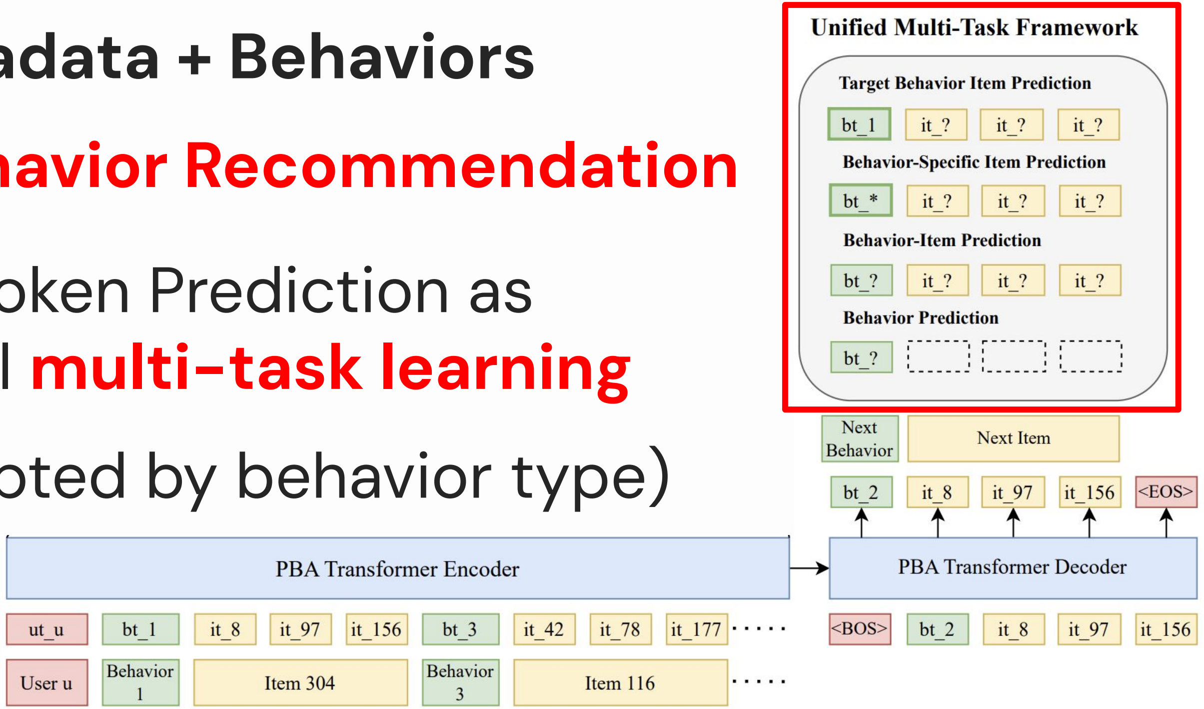
Inputs for SemID Construction

Item Metadata + Behaviors

Multi-Behavior Recommendation

Next Token Prediction as natural **multi-task learning**

(prompted by behavior type)



Liu et al. Multi-Behavior Generative Recommendation. CIKM 2024.

Inputs for SemID Construction

Input: all data associated with the item

(1) Item Metadata

Text / Multimodal / Categorical / No Features

(2) Item Metadata + Behaviors

Fused semantic IDs & Representations

Tokenizer trained on behavior sequences

Part 1 Summary – SemID Construction

(1) First Example: TIGER

(2) Construction Techniques

(3) Inputs

Part 1 Summary – SemID Construction

(1) First Example: TIGER

(2) Construction Techniques

Context-independent (PQ, RQ, Clustering, LM-based generator) -> Context-aware

(3) Inputs

Part 1 Summary – SemID Construction

(1) First Example: TIGER

(2) Construction Techniques

Context-independent (PQ, RQ, Clustering, LM-based generator) -> Context-aware

(3) Inputs

Item Metadata (Text, Multimodal, Features)

+ Behaviors (Fused SemIDs / Representations)

Part 2: SemID-based Generative Recommendation Model Architecture

SemID-based Recommender Architecture

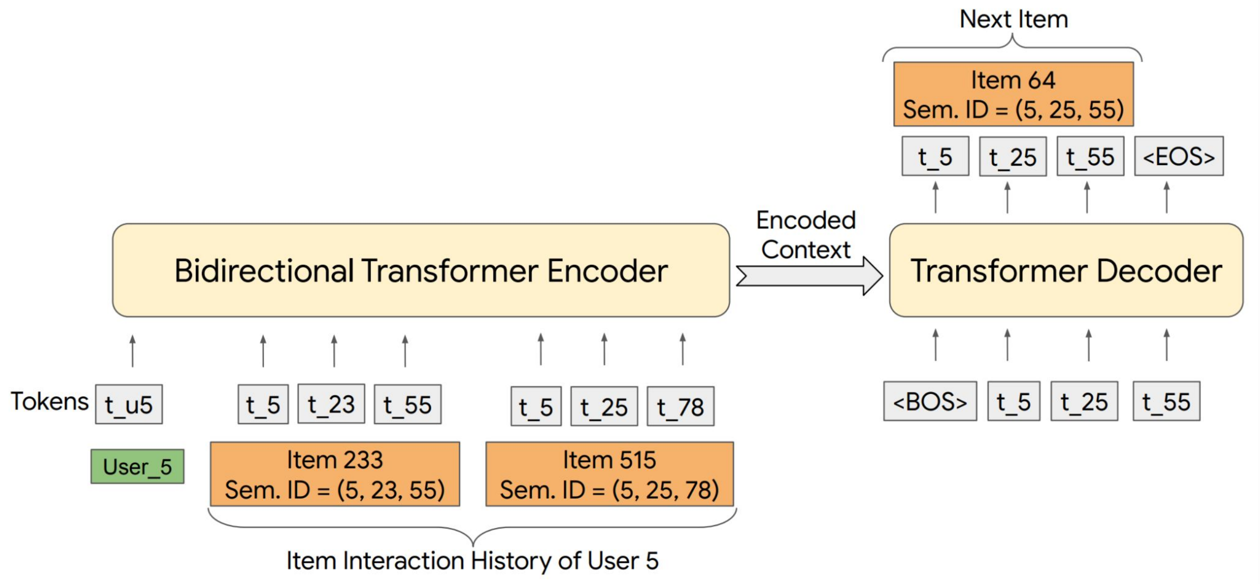
Recommendation as a **seq**-to-**seq** generation problem

Input: user interacted items $\{c_{11}, c_{12}, c_{13}, c_{14}, c_{21}, c_{22}, \ldots\}$

Output: next item $\{c_{t1}, c_{t2}, c_{t3}, c_{t4}\}$

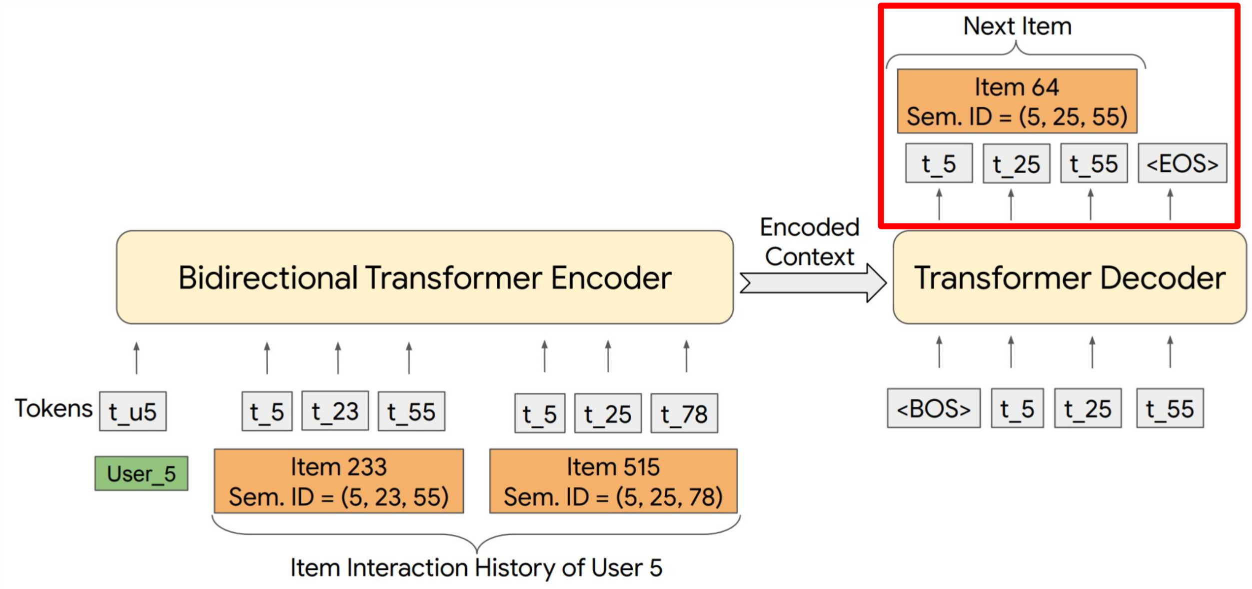
SemID-based Recommender Architecture

Architecture: Decoder-Only / Encoder-Decoder



SemID-based Recommender Architecture

Objective: Next-Token Prediction



SemID-based Recommender Architecture

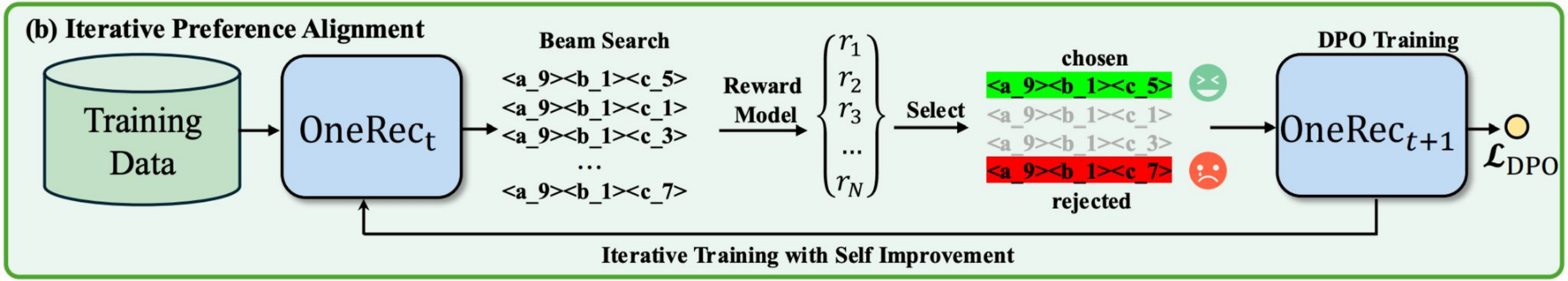
Objective: Next-Token Prediction

Could we add negative samples like BPR loss?

SemID-based Recommender Architecture

Objective: Preference Alignment Objective

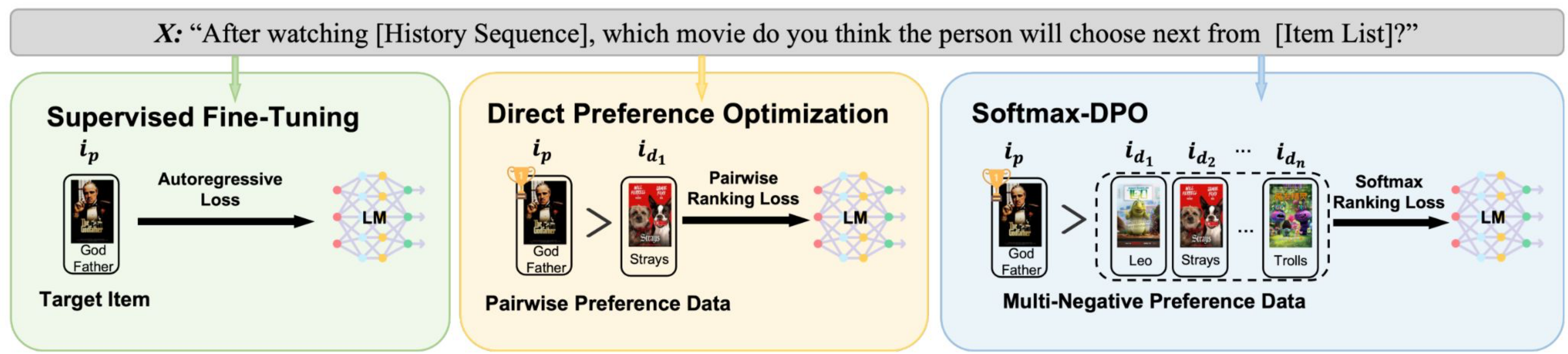
One negative sample per instance



SemID-based Recommender Architecture

Objective: Preference Alignment Objective

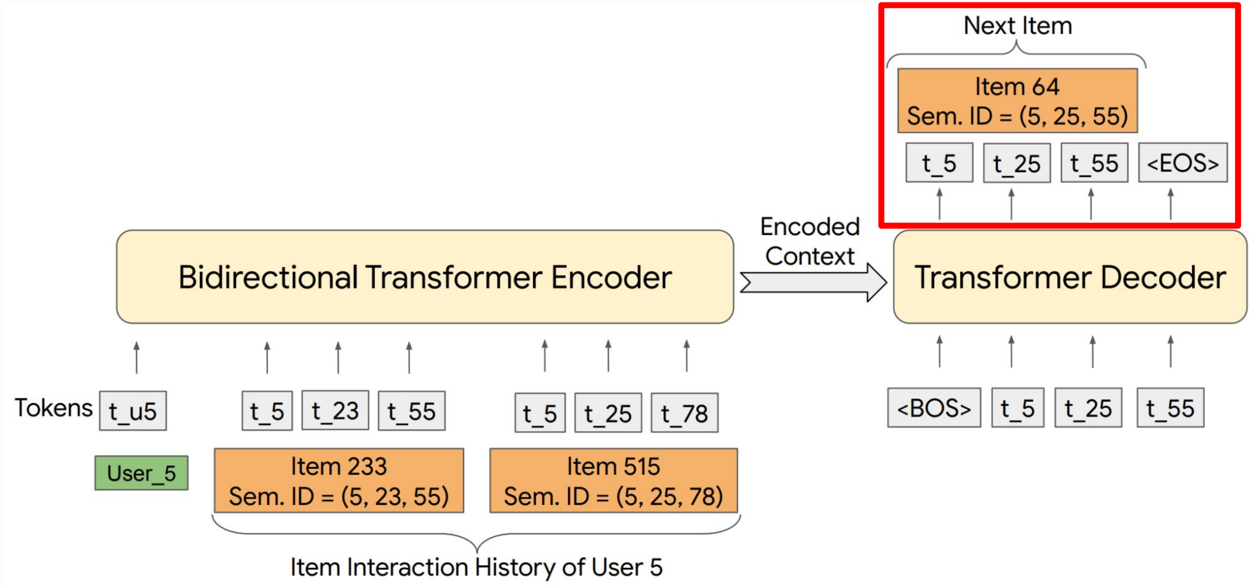
Multiple negative samples per instance



Chen et al. On Softmax Direct Preference Optimization for Recommendation. NeurIPS 2024.

SemID-based Recommender Architecture

Inference: How to get a ranking list?

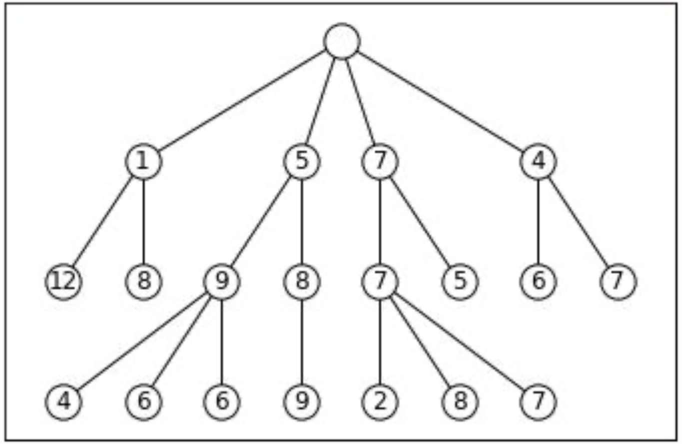


Rajput et al. Recommender Systems with Generative Retrieval. NeurIPS 2023.

SemID-based Recommender Architecture

Inference: How to get a ranking list?

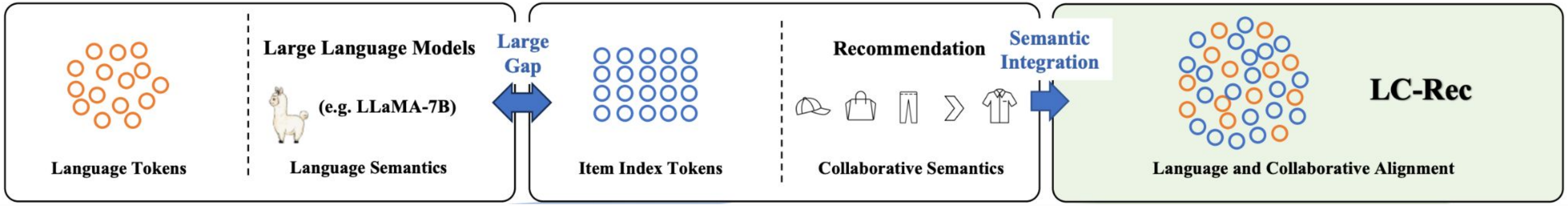
(Constrained) Beam Search



https://en.wikipedia.org/wiki/Beam_search

SemID-based Recommender Architecture

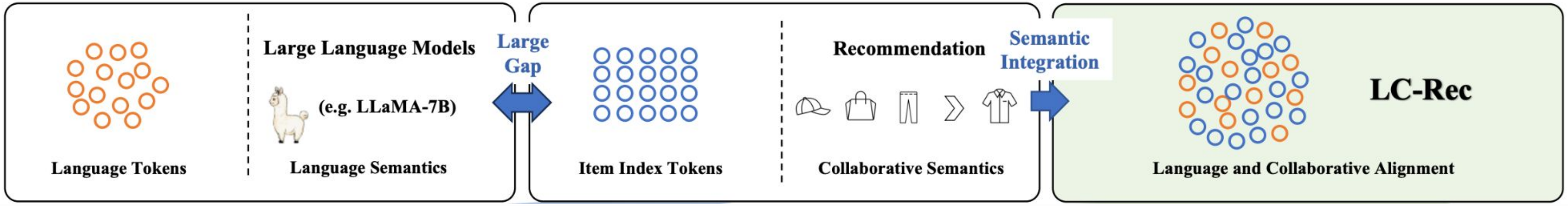
Align with LLMS - LC-Rec



Zheng et al. Adapting Large Language Models by Integrating Collaborative Semantics for Recommendation. ICDE 2024.

SemID-based Recommender Architecture

Align with LLMS - LC-Rec



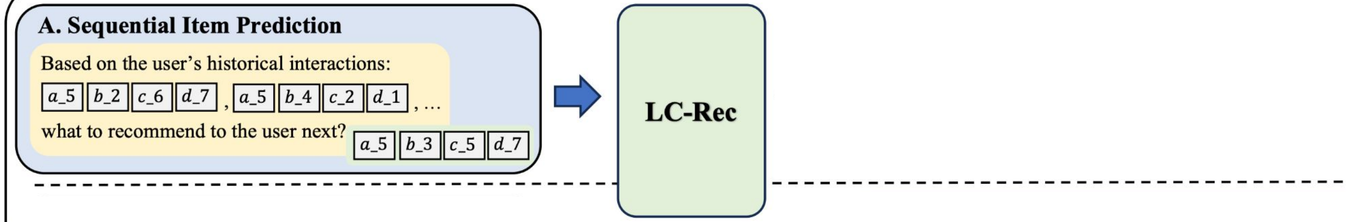
Core Idea:

Construct instructions containing both Semantic IDs and language tokens

Zheng et al. Adapting Large Language Models by Integrating Collaborative Semantics for Recommendation. ICDE 2024.

SemID-based Recommender Architecture

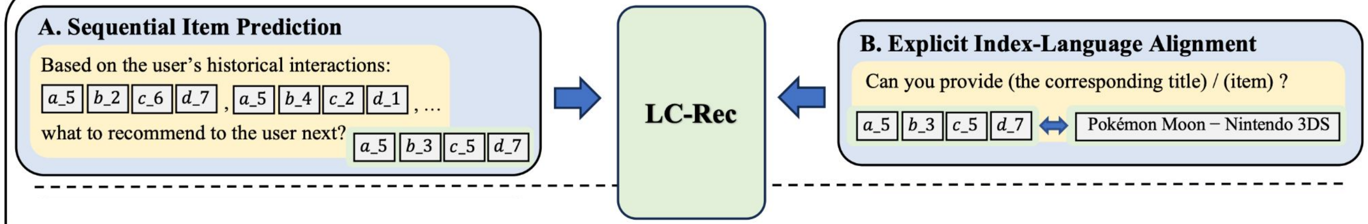
Align with LLMS - LC-Rec



SemID-based seq2seq task

SemID-based Recommender Architecture

Align with LLMS - LC-Rec

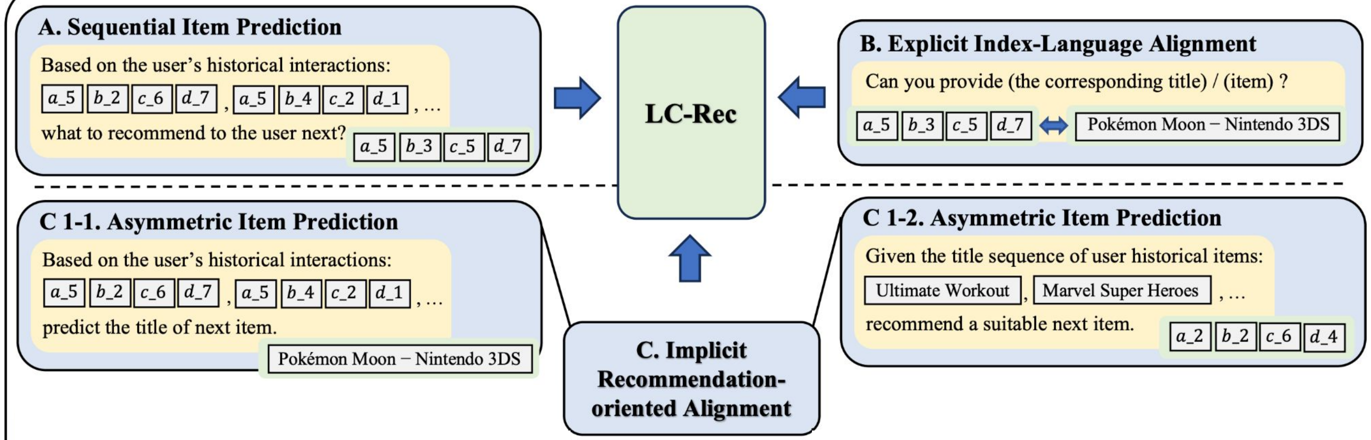


Translation between SemIDs and titles

Zheng et al. Adapting Large Language Models by Integrating Collaborative Semantics for Recommendation. ICDE 2024.

SemID-based Recommender Architecture

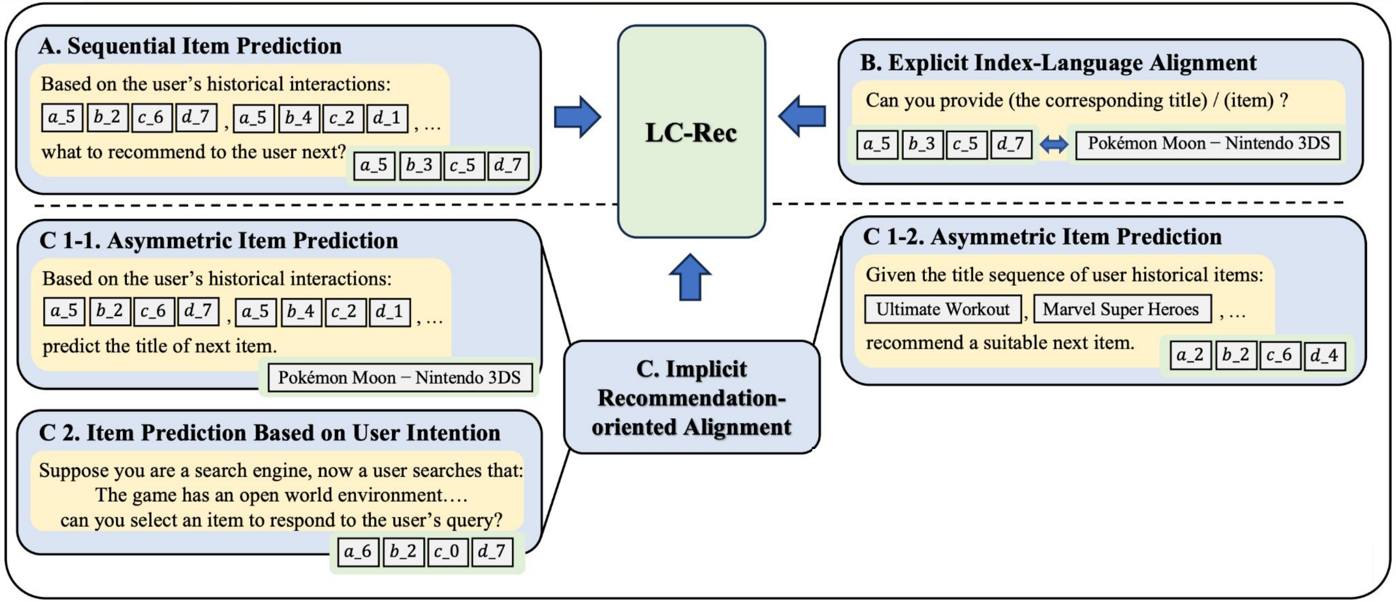
Align with LLMS - LC-Rec



Implicit Translation between SemIDs and titles

SemID-based Recommender Architecture

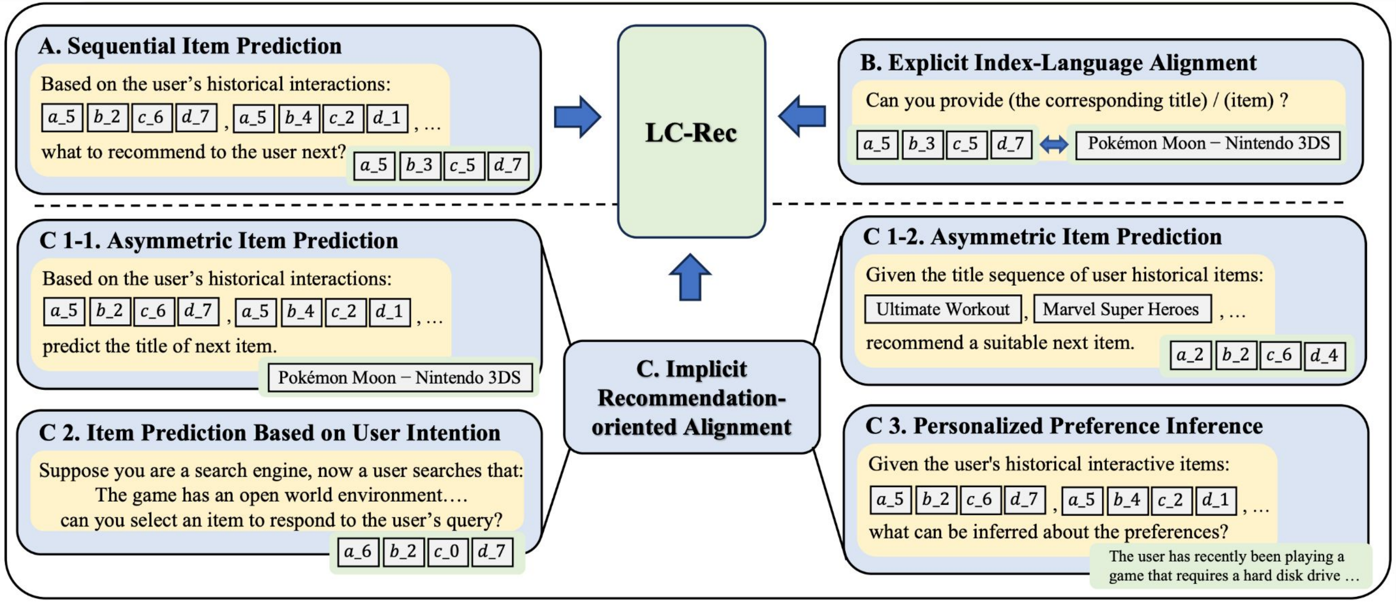
Align with LLMS - LC-Rec



Zheng et al. Adapting Large Language Models by Integrating Collaborative Semantics for Recommendation. ICDE 2024.

SemID-based Recommender Architecture

Align with LLMS - LC-Rec



Zheng et al. Adapting Large Language Models by Integrating Collaborative Semantics for Recommendation. ICDE 2024.

Part 2 Summary – Architecture

(1) Train from Scratch

(2) Align with LLMs

Part 2 Summary - Architecture

(1) Train from Scratch

Objective (NTP, DPO, S-DPO)

Inference (Beam Search)

(2) Align with LLMs

Part 2 Summary – Architecture

(1) Train from Scratch

Objective (NTP, DPO, S-DPO)

Inference (Beam Search)

(2) Align with LLMs

LC-Rec: Instructions containing both semIDs and language tokens

04

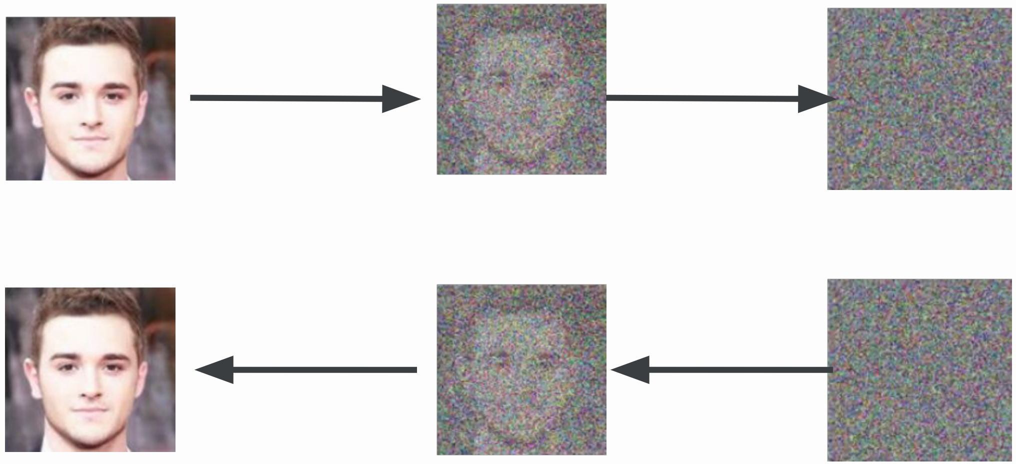
Diffusion Model

-based Generative Recommendation

What is Diffusion

Forward process

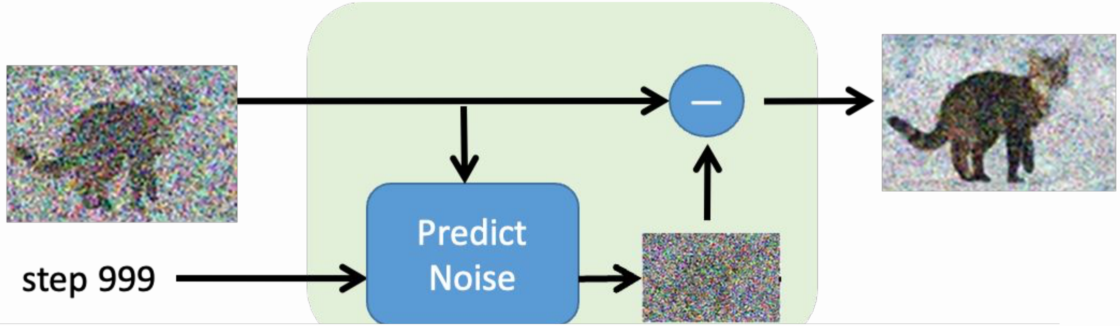
Reverse process



Build the mapping between data sample and Gaussian sample

Denoising Diffusion Probabilistic Models. NeurIPS 2020

What is Diffusion



Algorithm 1 Training

1: **repeat**
2: $\quad \mathbf{x}_0 \sim q(\mathbf{x}_0)$
3: $\quad t \sim \mathrm{Uniform}(\{1, \ldots, T\})$
4: $\quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
5: $\quad$ Take gradient descent step on
$$\nabla_\theta \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t) \right\|^2$$
6: **until** converged

Algorithm 2 Sampling

1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: **for** $t = T, \ldots, 1$ **do**
3: $\quad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
4: $\quad \mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) \right) + \sigma_t \mathbf{z}$
5: **end for**
6: **return** $\mathbf{x}_0$

Remove the noise step by step from a Gaussian sample.

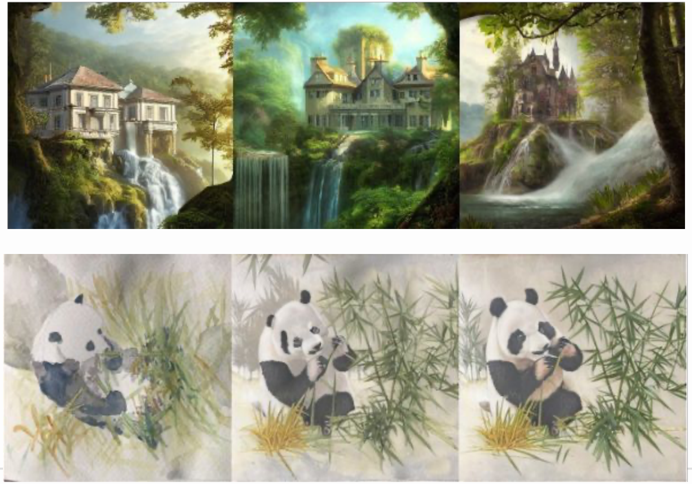
Denoising Diffusion Probabilistic Models. NeurIPS 2020

Diffusion in CV

Diffusion is at the core of visual content generation.

Image generation

Stable Diffusion, DALL-E...



Video generation

Sora, Hunyuan-Video, Keling...

Diffusion for recommendation

Use diffusion to enhance traditional recommender

- More robust representation
- Data augmentation

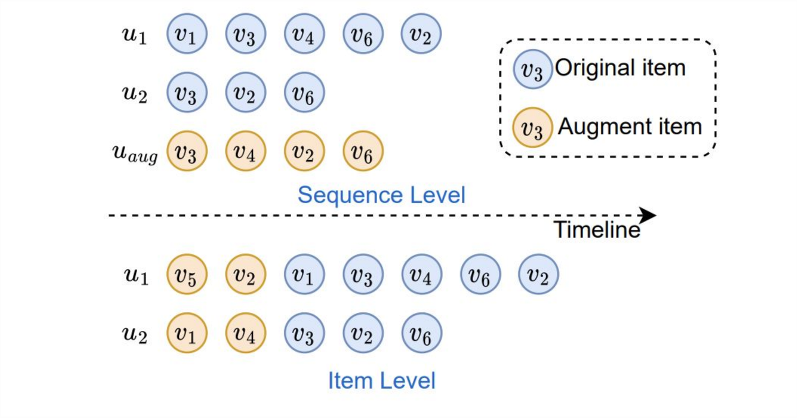
Diffusion as recommender

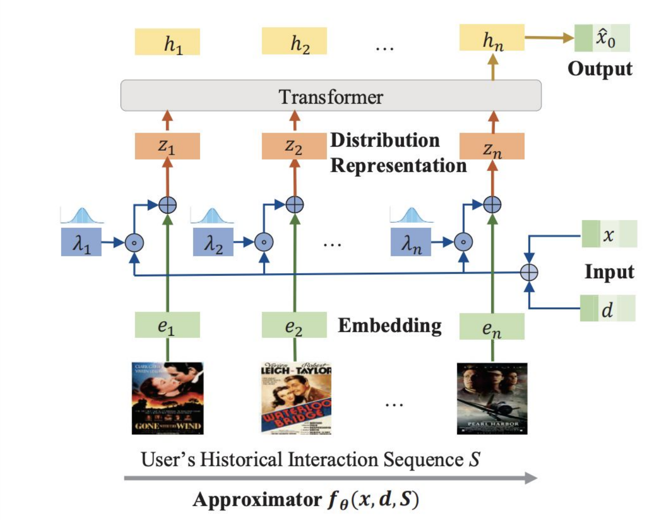
- Diffuse on the user interaction vector
- Diffuse on item representation
- Discrete diffusion

Diffusion for personalized content generation

- Personalized try-on, image,....

Diffusion as enhancer





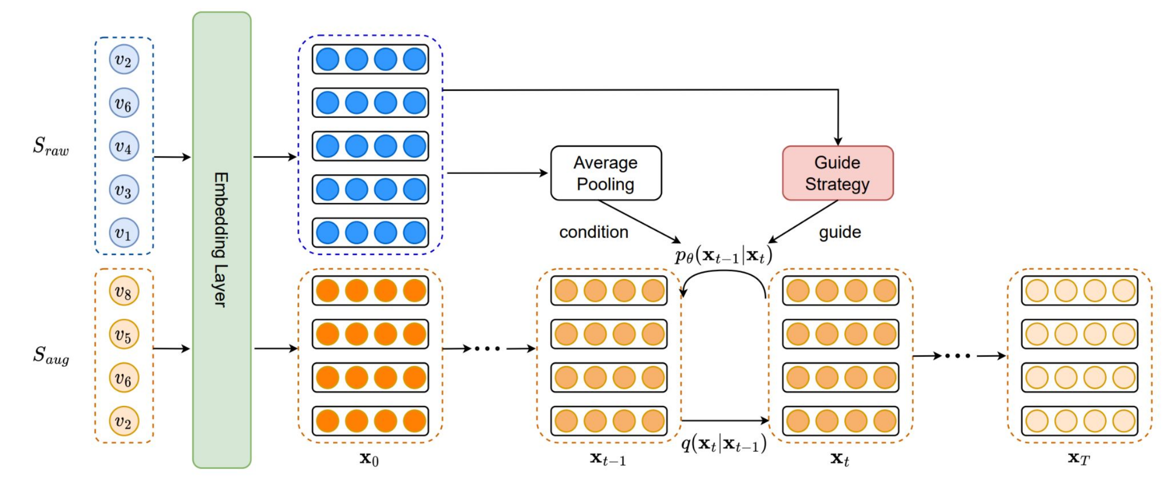
Generate more interaction or sequences

Enhance the robustness of embeddings

Diffusion Augmentation for Sequential Recommendation, in CIKM 2023.
DiffuRec: A Diffusion Model for Sequential Recommendation, in TOIS 2024

Pseudo sequence generation (I)

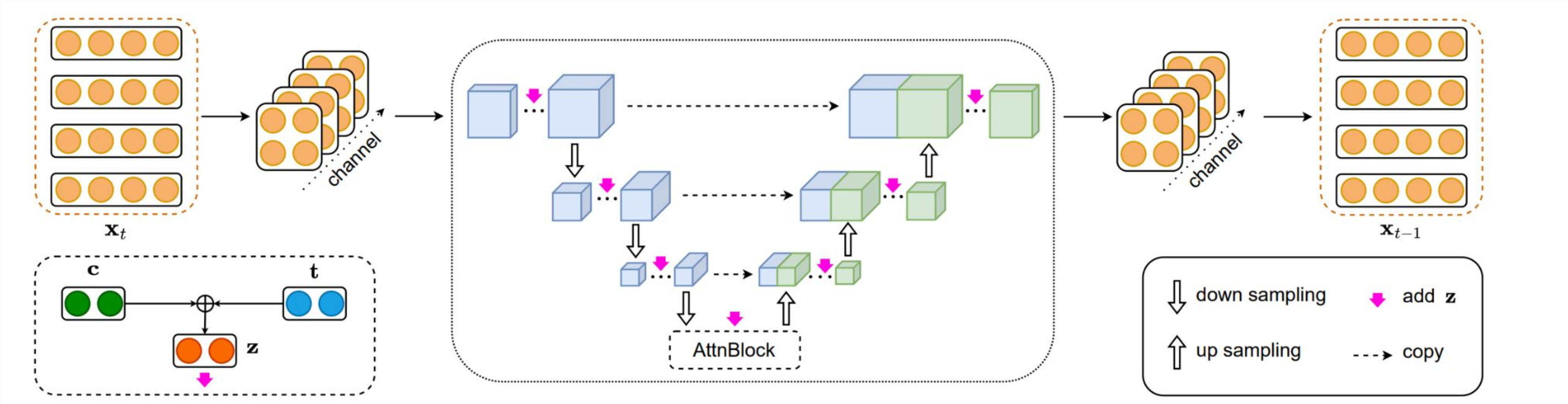
Generate pseudo sequence embeddings conditioned on historical interaction sequence



Diffusion Augmentation for Sequential Recommendation, in CIKM 2023.

Pseudo sequence generation (II)

The model architecture is adopted from U-Net



Diffusion Augmentation for Sequential Recommendation, in CIKM 2023.

Diffusion for recommendation

Use diffusion to enhance traditional recommender

- More robust representation
- Data augmentation

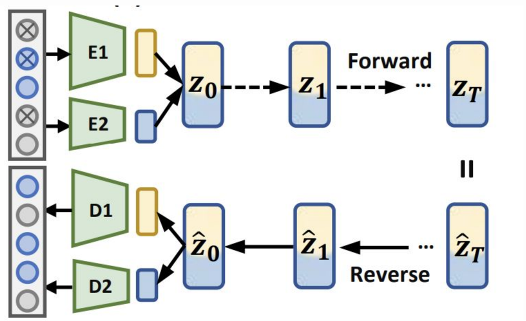
Diffusion as recommender

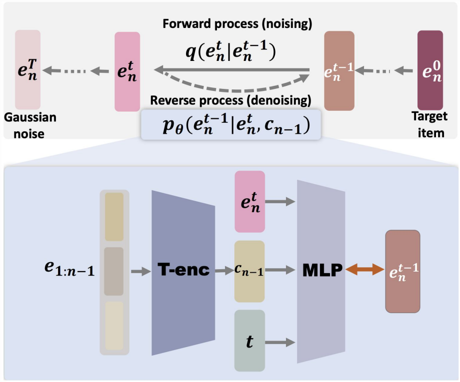
- Diffuse on the user interaction vector
- Diffuse on item representation
- Discrete diffusion

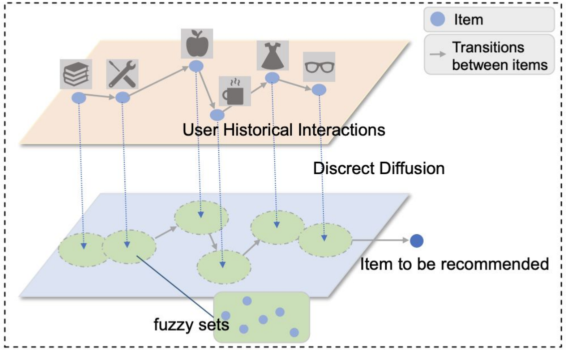
Diffusion for personalized content generation

- Personalized try-on, image,....

Diffusion as recommender







Diffuse on the user interaction vector

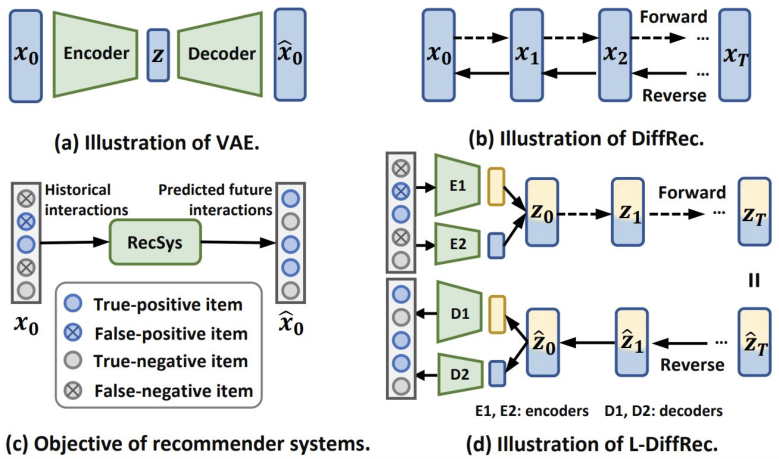
Diffuse on item representation

Discrete diffusion

Interaction vector completion (I)

Motivation – limitation of GANs and VAEs:

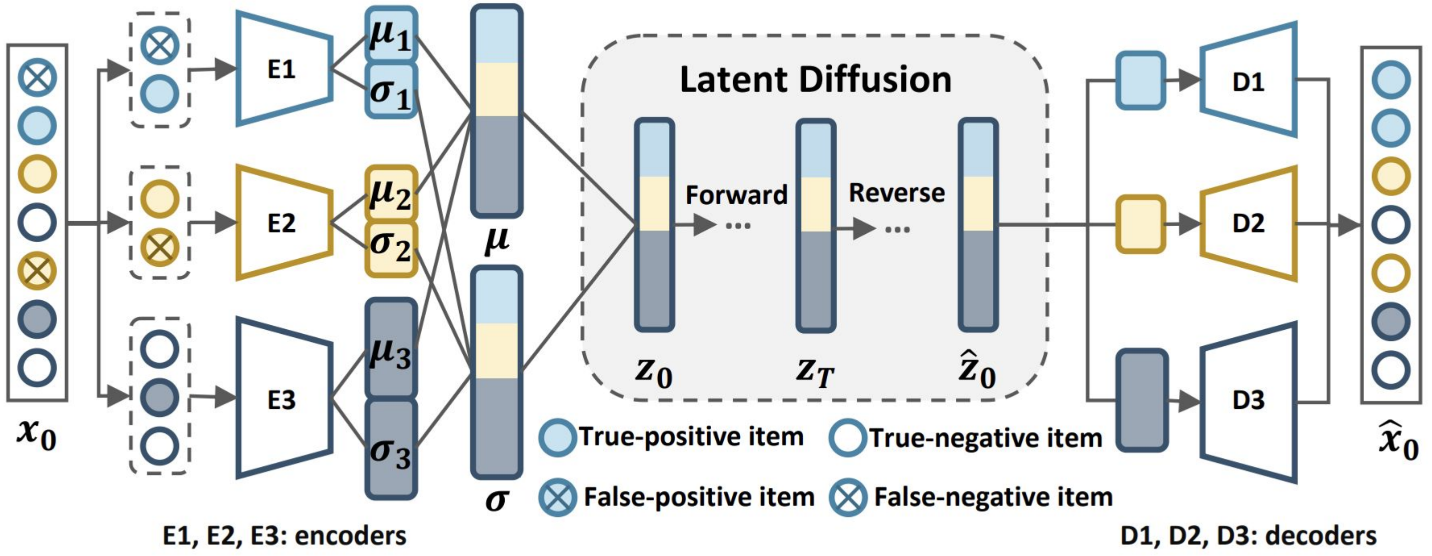
GAN- and VAE-based recommenders suffers from issues like **instability and representation collapse**.



Diffusion Recommender Model, in SIGIR 2023.

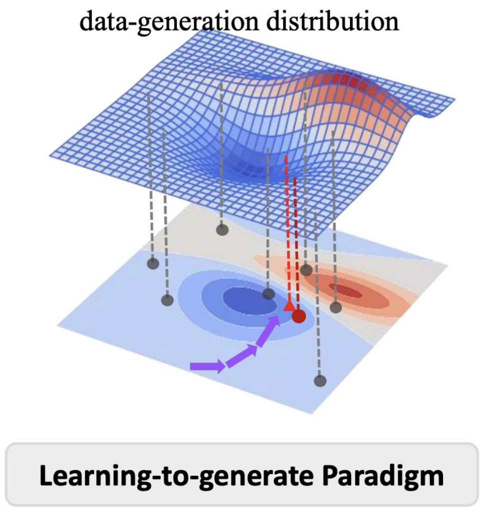
Interaction vector completion (II)

Forward: **corrupt the interaction vector** into gaussian noise
Reverse: **recover the interaction vector** from the gaussian



Generate item embedding

There exists an implicit distribution, from which target item embedding can be generated.



Challenge:

- The data-generation distribution is complicated and unknown.

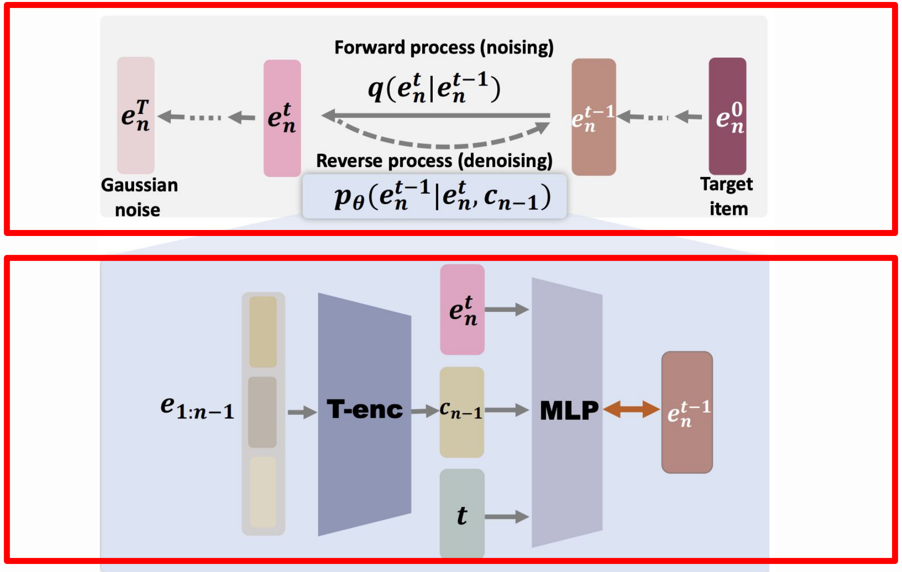
Solution:

- Capture the data-generation distribution by connecting it with Gaussian distribution.
- This can be achieved by diffusion.

Generate What You Prefer: Reshaping Sequential Recommendation via Guided Diffusion, in NeurIPS 2023

Generate item embedding

- Diffusion on target item embeddings.
- Guided by user interaction sequence for personalization.



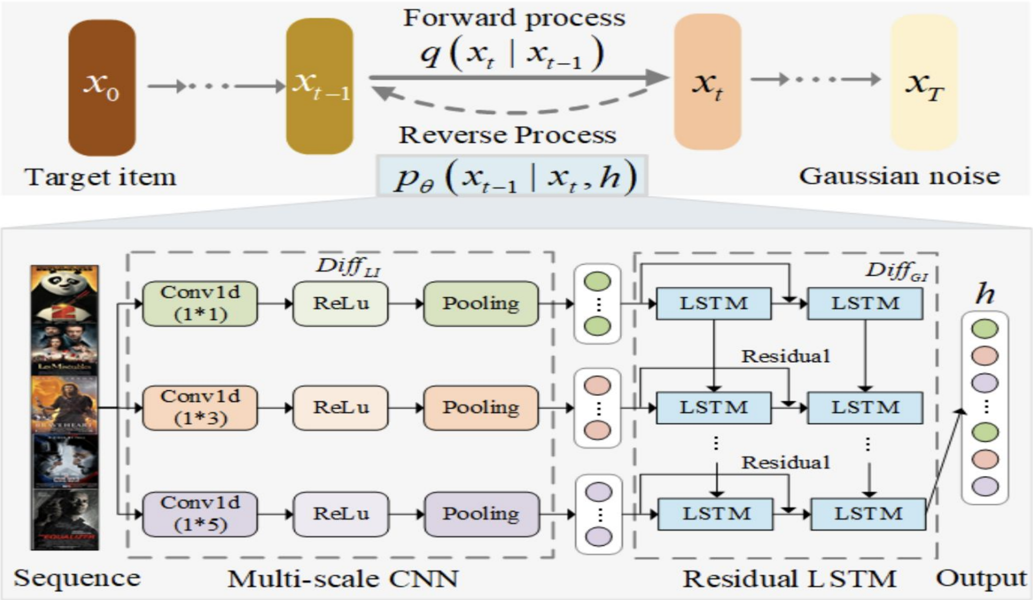
Diffusion process

Guidance

Generate What You Prefer: Reshaping Sequential Recommendation via Guided Diffusion, in NeurIPS 2023

Generate item embedding

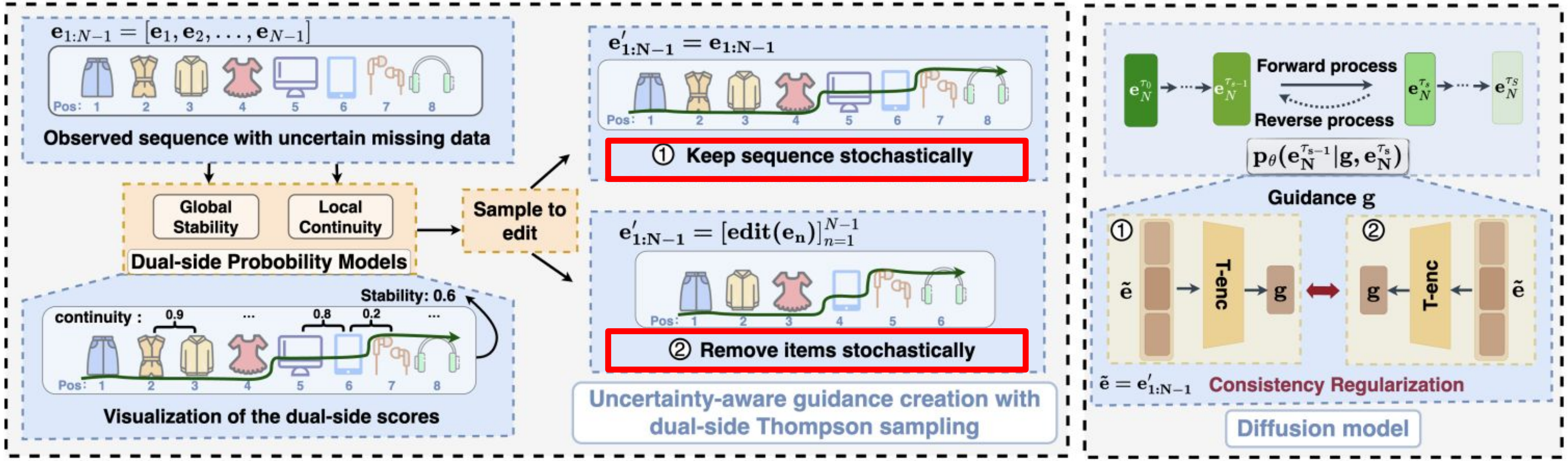
- Different sequence encoder



Diffusion Recommendation with Implicit Sequence Influence, in WWW 2024

Generate item embedding

- Uncertainty-aware guidance



Uncertainty-aware Guided Diffusion for Missing Data in Sequential Recommendation, in SIGIR 2025

Generate item embedding

- Incorporate preference optimization

$$\mathcal{L}_{\text{Simple}} = \mathbb{E}_{(\mathbf{e}_0^+, \mathbf{c}, t)} \left[\left\| \mathcal{F}_\theta(\mathbf{e}_t^+, t, \mathcal{M}(\mathbf{c})) - \mathbf{e}_0^+ \right\|_2^2 \right] ,$$

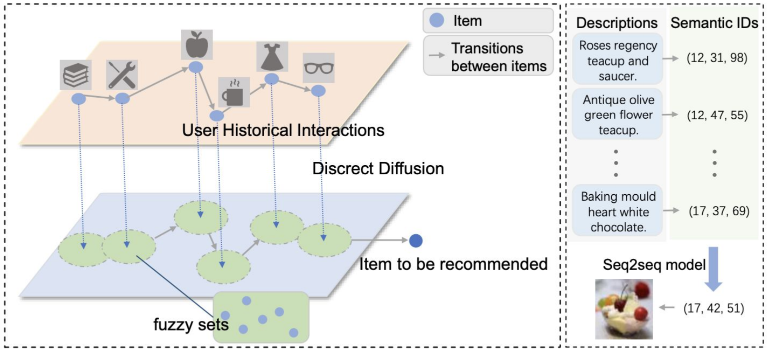
$$\mathcal{L}_{\text{BPR-Diff-C}} = -\log \sigma(-|\mathcal{H}| \cdot [S(\hat{\mathbf{e}}_0^+, \mathbf{e}_0^+) - S(\mathcal{F}_\theta(\bar{\mathbf{e}}_t^-, t, \mathcal{M}(\mathbf{c})), \bar{\mathbf{e}}_0^-)]) .$$

$$\mathcal{L}_{\text{PerferDiff}} = \underbrace{\lambda \mathcal{L}_{\text{Simple}}}_{\text{Learning Generation}} + \underbrace{(1-\lambda)\mathcal{L}_{\text{BPR-Diff-C}}}_{\text{Learning Preference}} .$$

Preference Diffusion for Recommendation, in ICLR 2025

Discrete diffusion

State transitions occur under discrete conditions for the entire interaction sequence.



- Represent interaction sequence as one-hot vector through semantic ID.
- Conduct discrete diffusion on interaction sequence.

Breaking Determinism: Fuzzy Modeling of Sequential Recommendation Using Discrete State Space Diffusion Model, in NeurIPS 2024.

Discrete diffusion

Semantic IDs

Algorithm 1 Training of DDSR.

Input: historical interaction sequence $v_{1:n-1} = c_{1:n-1;1:m}$; target item $v_n = c_{n;1:m}$; transition matrix $\boldsymbol{Q}_t$; Approximator $f_\theta(\cdot)$.

Output: well-trained Approximator $f_\theta(\cdot)$.

While not converged do:

Sample Diffusion Time: $t \sim [0, 1, \ldots, T]$;

Calculate t-step transition probability: $\overline{\boldsymbol{Q}_t} = \boldsymbol{Q}_1 \boldsymbol{Q}_2 \cdots \boldsymbol{Q}_t$;

Convert $c_{n;1:m}$ to one-hot encoding $\boldsymbol{x}^0_{n;1:m}$;

Obtain the discrete state $x^t_{n;1:m}$ after t steps by Equation 2, thereby obtaining the 'fuzzy set' $c^t_{1:n-1;1:m}$;

Modeling $c_{2;n;1:m}$ based on 'fuzzy sets' through Equation 5;

Take gradient descent step on $\nabla L_{CE}\left(\hat{c}_{2:n;1:m}, c_{2:n;1:m}\right)$.

Forward process

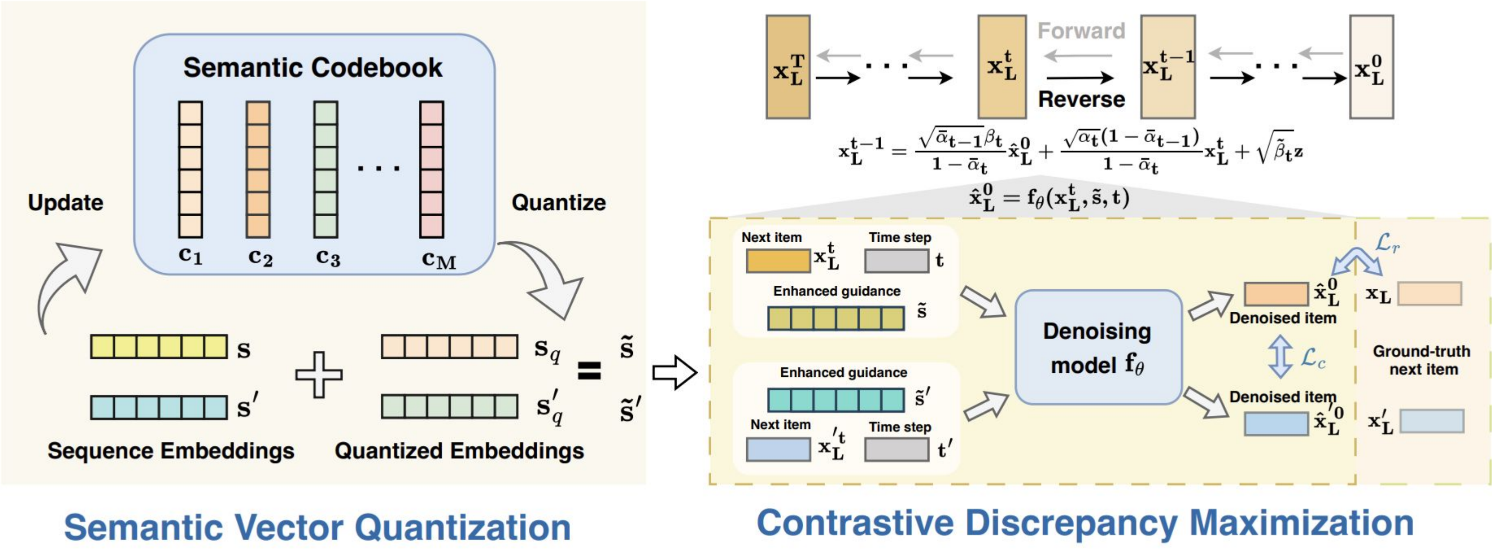
$$[\boldsymbol{Q}_t]_{ij} = \begin{cases} (1-\beta_t)/(|\mathcal{V}|-1) & \text{if } i \neq j \\ \beta_t & \text{if } i = j \end{cases}.$$

$$\hat{c}_{2:n;1:m} = f_\theta(c^t_{1:n-1;1:m}, t).$$

Breaking Determinism: Fuzzy Modeling of Sequential Recommendation Using Discrete State Space Diffusion Model, in NeurIPS 2024.

Discrete diffusion

- Quantization embedding with continuous diffusion.



Distinguished Quantized Guidance for Diffusion-based Sequence Recommendation, in WWW 2025..

Diffusion for recommendation

Use diffusion to enhance traditional recommender

- More robust representation
- Data augmentation

Diffusion as recommender

- Diffuse on the user interaction vector
- Diffuse on item representation
- Discrete diffusion

Diffusion for personalized content generation

- Personalized try-on, image,....

Personalized content generation

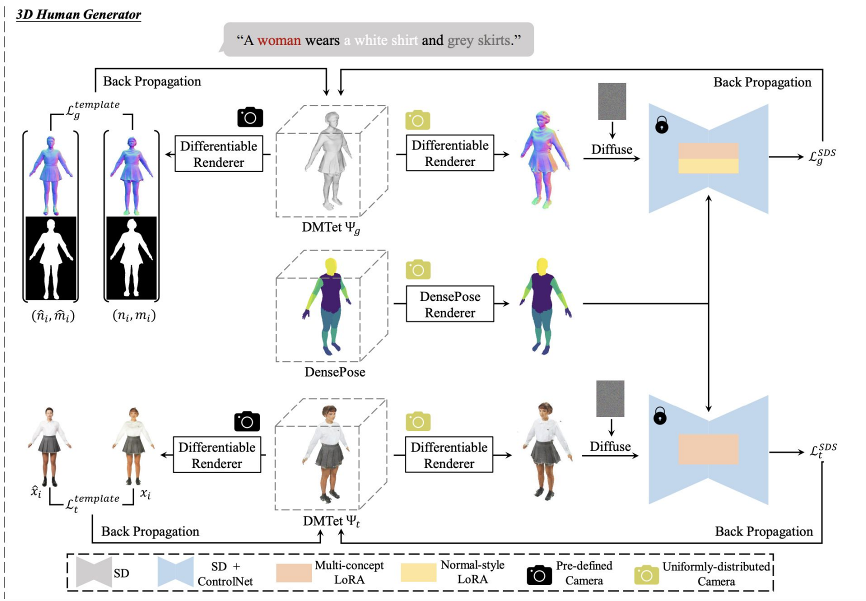
Personalized try-on

Personalized image

OOTDiffusion: Outfitting Fusion based Latent Diffusion for Controllable Virtual Try-on, in arXiv
InstantBooth: Personalized Text-to-Image Generation without Test-Time Finetuning. In CVPR 2024.

Personalized Try-on

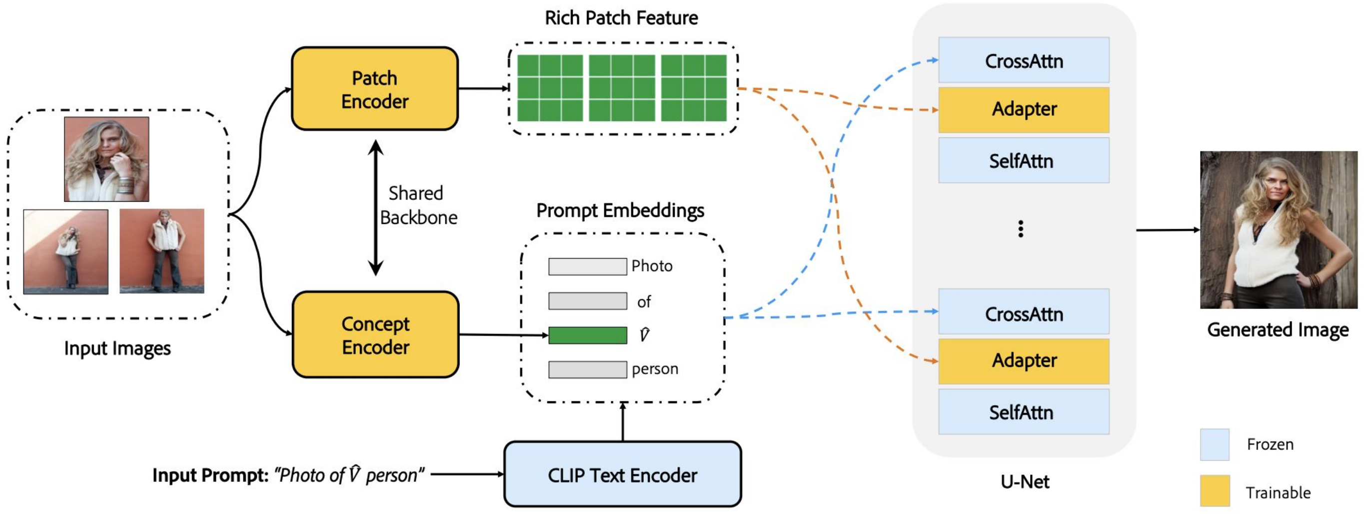
Generate realistic 3D try-on given person images, clothes images, and a text prompt.



DreamVTON: Customizing 3D Virtual Try-on with Personalized Diffusion Models. MM 2024

Personalized Image

Generate personalized image given person images and the desired concept.



InstantBooth: Personalized Text-to-Image Generation without Test-Time Finetuning. CVPR 2024

05

Summary

Open Challenges and Beyond

Summary

Scaling Law:

- **larger** model + **larger** dataset -> **better** performance

Most large models are generative

- (LLMs, Text2Video Models)

Summary

Scaling Law:

- **larger** model + **larger** dataset -> **better** performance

Most large models are generative

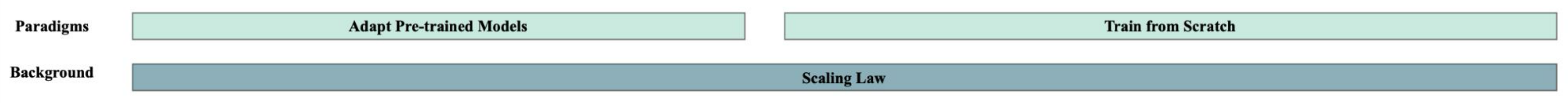
- (LLMs, Text2Video Models)

💡 **Large generative rec models?**

Summary

🤔 How to get a large generative rec model?

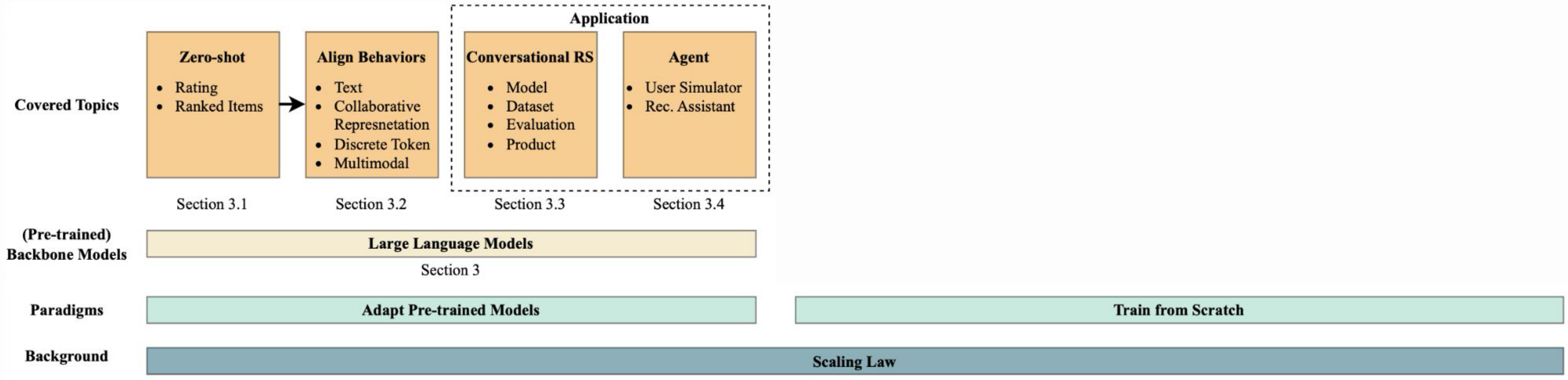
- Pre-trained model (e.g., LLMs) -> Adaptation;
- From scratch;



Summary

Adaptation

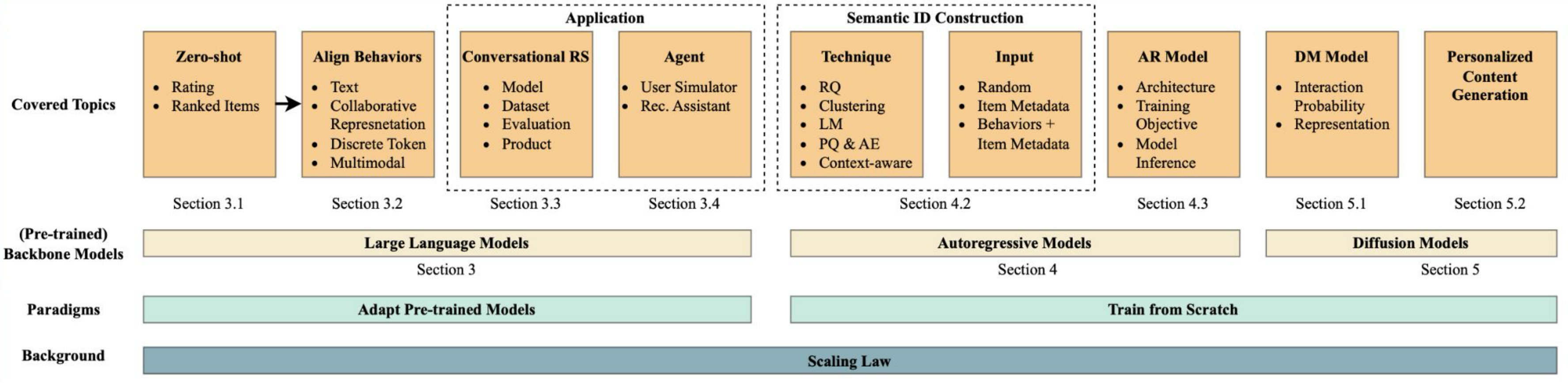
Mainly LLM-based recommendations



Summary

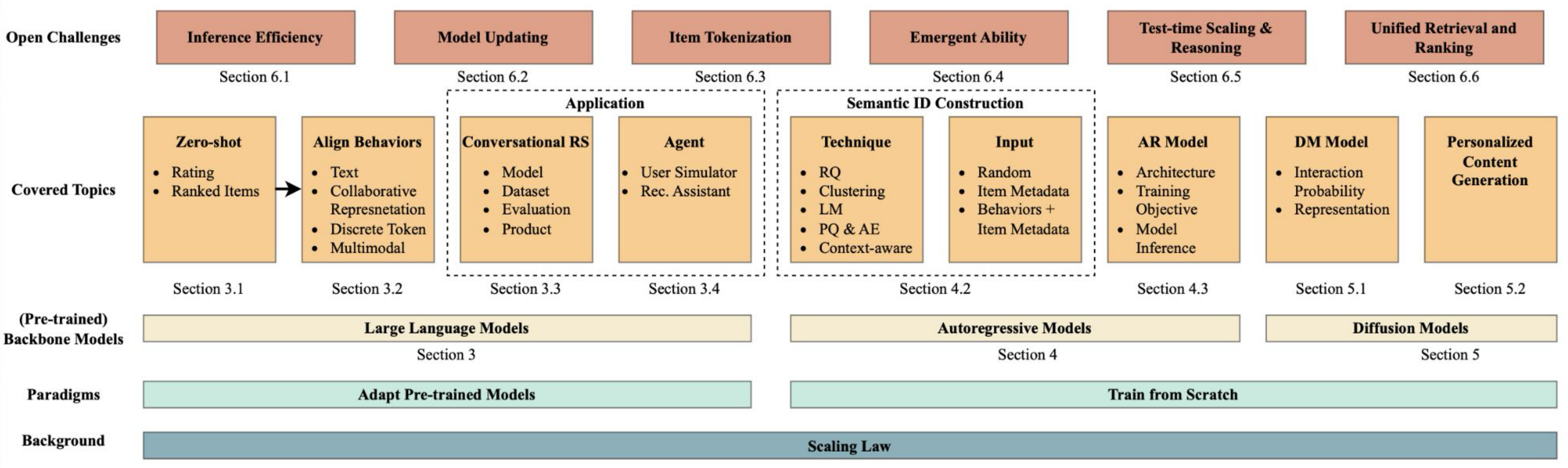
From Scratch

- Autoregressive models (e.g., semantic ID-based);
- Diffusion models;



Summary

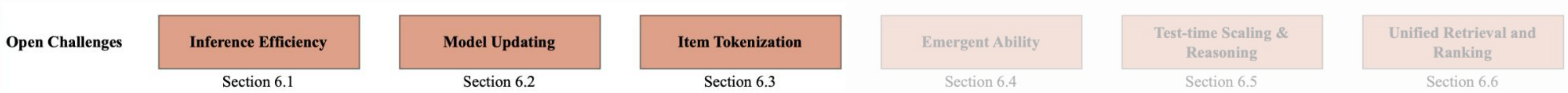
Open Challenges



Open Challenges

Part 1: What becomes harder?

Comparing to traditional RecSys, what challenges may large generative models face?



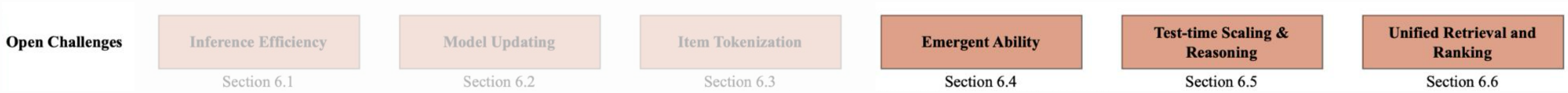
Open Challenges

Part 1: What becomes harder?

Comparing to traditional RecSys, what challenges may large generative models face?

Part 2: What becomes possible?

What new opportunities may large generative models unlock for recommender systems?



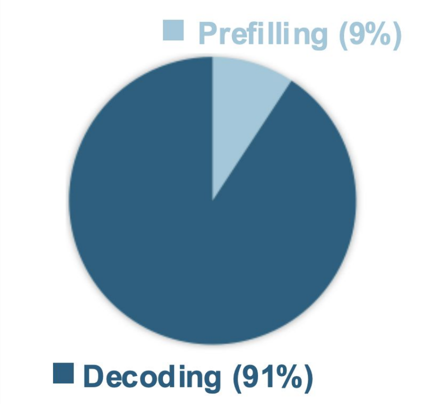
Part 1: What Becomes Harder?

Comparing to traditional RecSys, what challenges may large generative models face?

Inference Efficiency

Retrieval Models: **K Nearest Neighbor Search**

Generative Models (e.g., AR models): **Beam Search**

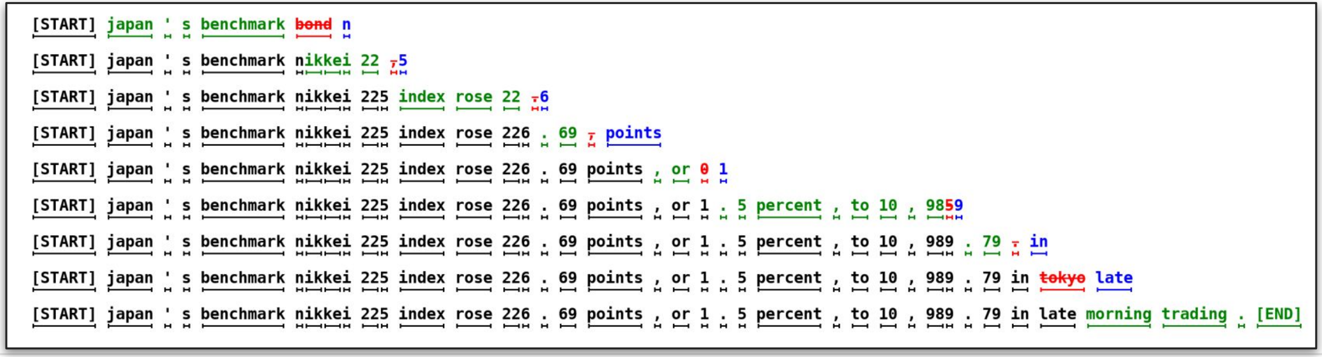


Lin et al. Efficient Inference for Large Language Model-based Generative Recommendation. ICLR 2025.

Inference Efficiency

How to accelerate LLMs? **Speculative Decoding**

- Use a "cheap" model to generate candidates
- "Expensive" model can accept or reject (and perform inference if necessary)

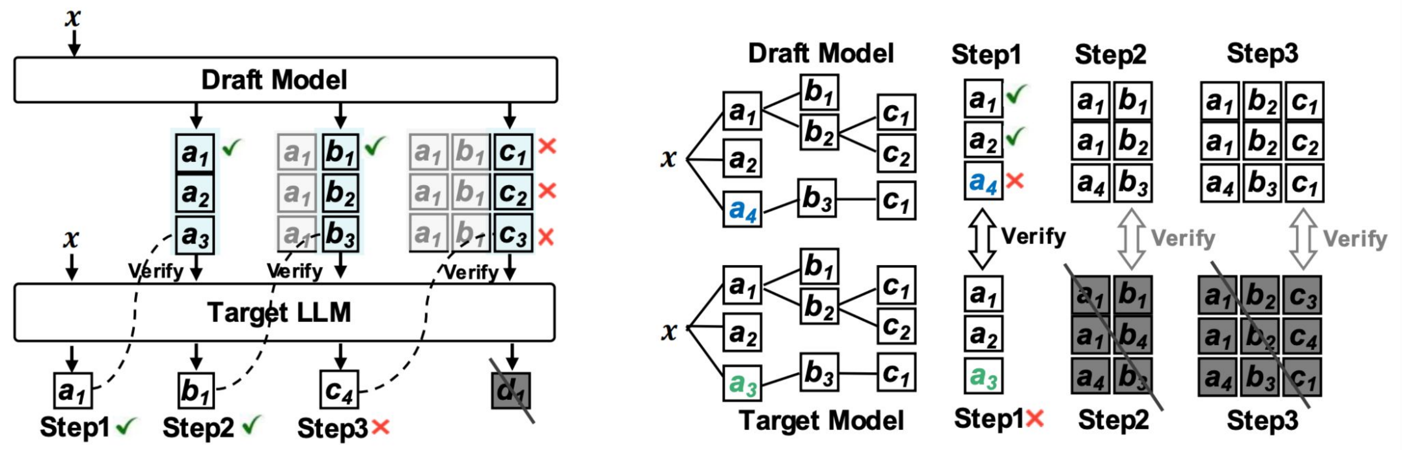


Leviathan et al. Fast Inference from Transformers via Speculative Decoding. ICML 2023.

Inference Efficiency

Speculative decoding for generative rec? ❌

N-to-K verification



(b) N-to-1 Verification of Traditional SD (N=3)

(c) N-to-K Verification of SD with Beam Search (N=K=3)

Lin et al. Efficient Inference for Large Language Model-based Generative Recommendation. ICLR 2025.

Inference Efficiency

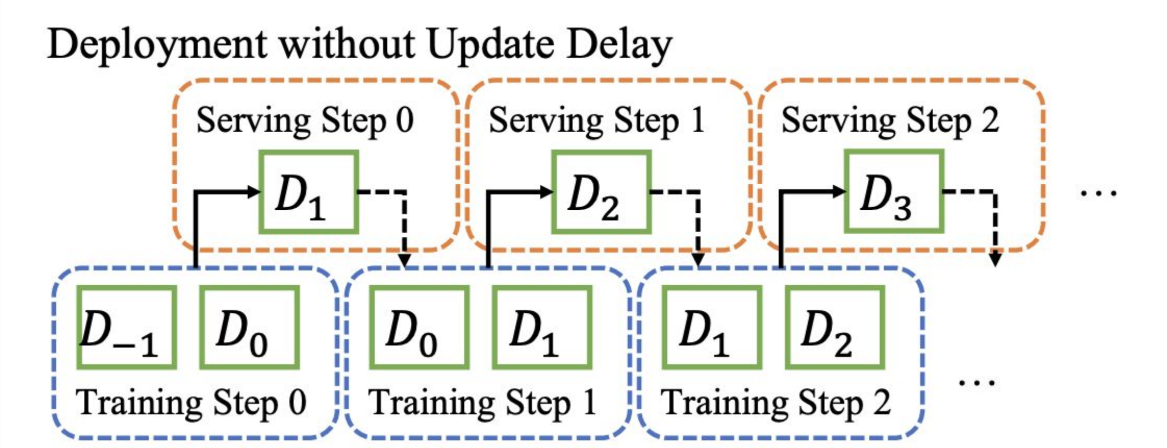
In addition to single-model acceleration methods, what about "serving throughout"?

Example: vLLM offers solutions for high-throughput and memory-efficient inference and serving

What's unique for generative rec?

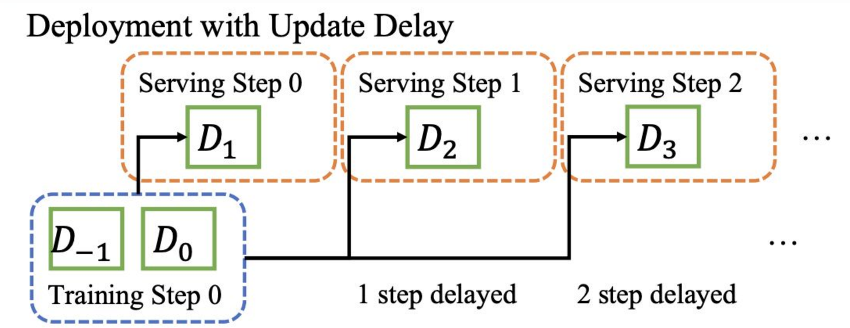
Timely Model Update

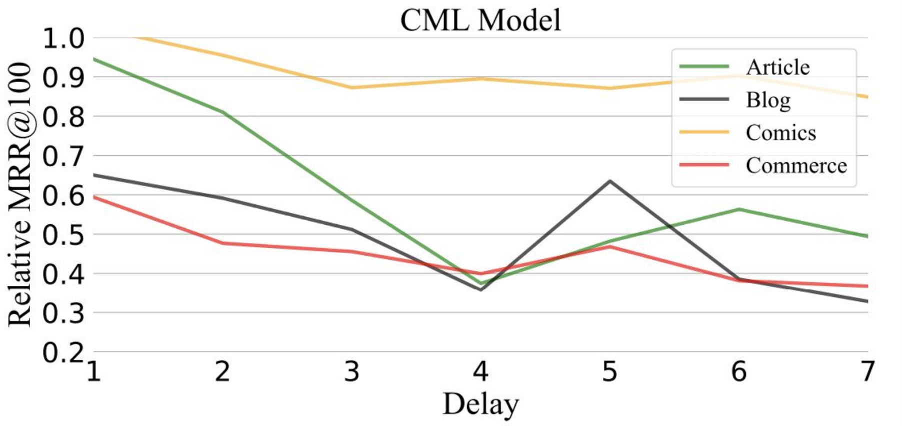
Recommendation models favor timely updates



Timely Model Update

Delayed updates lead to performance degradation





Lee et al. How Important is Periodic Model Update in Recommender Systems? SIGIR 2023.

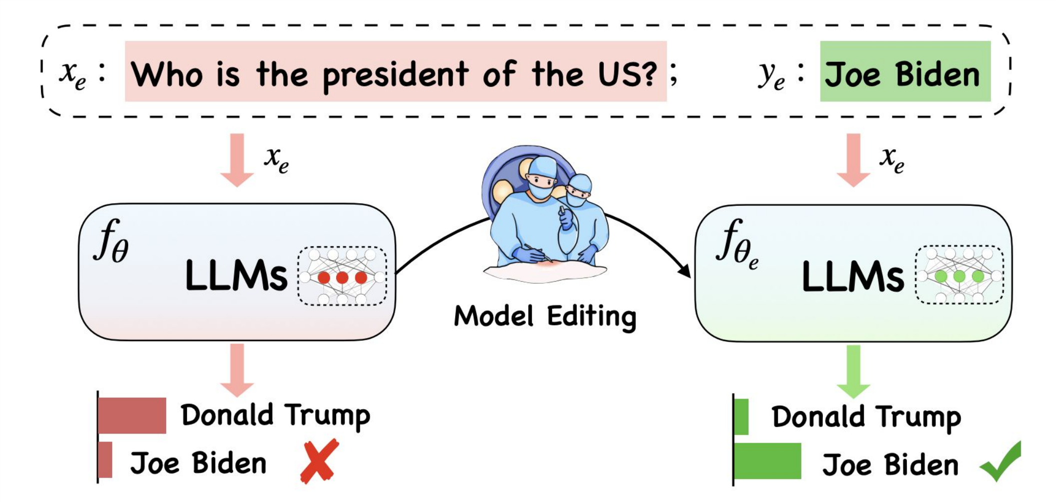
Timely Model Update

How to update large generative rec models timely?

(Frequently retraining large generative models may be resource consuming)

Timely Model Update

How to update large generative rec models timely?

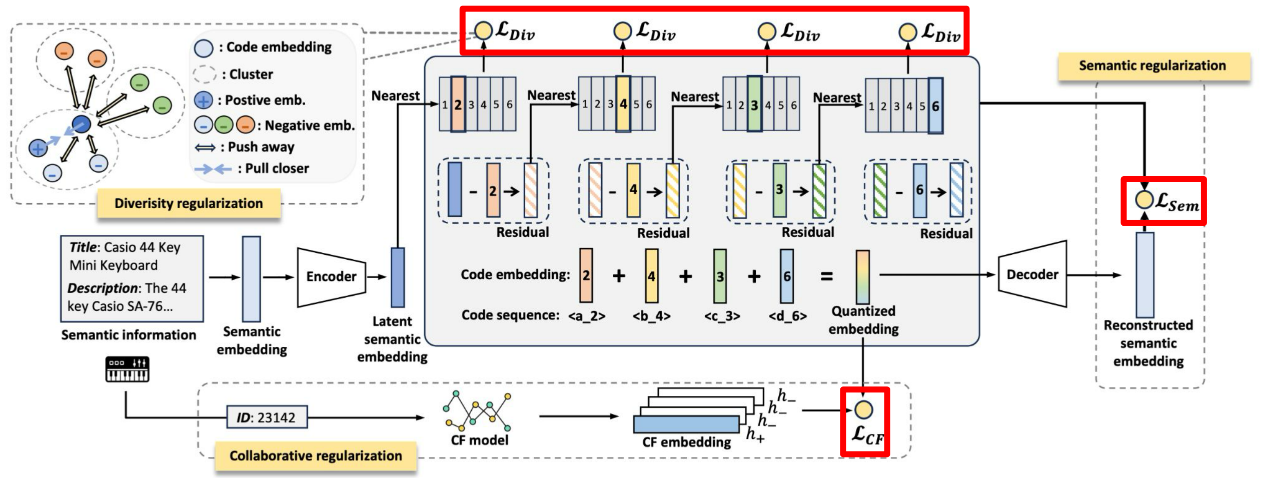


Knowledge editing?

Yao et al. Editing Large Language Models: Problems, Methods, and Opportunities. EMNLP 2023.

Item Tokenization

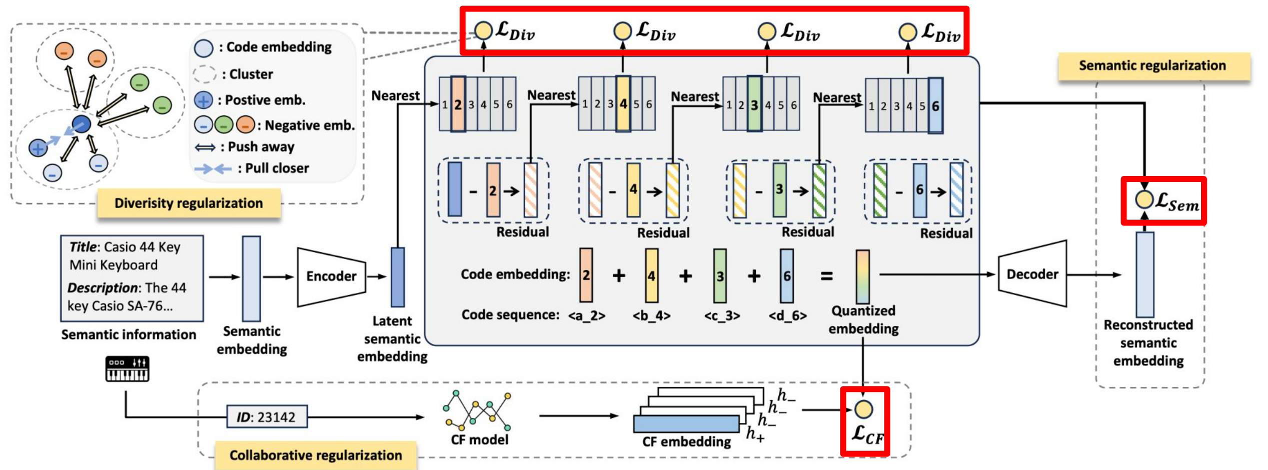
Multiple objectives for optimizing item tokenization …



Item Tokenization

Multiple objectives for optimizing item tokenization …

But **none** of them is **directly related to rec performance**



Item Tokenization

reconstruction loss ≠ downstream performance

How to connect tokenization objective with recommendation performance?

Zipf's distribution? Entropy? Linguistic metrics?

Item Tokenization

Language Tokenization

2014~2015:

Word / Char

Context-independent ⇒ Context-aware

Item Tokenization

Language Tokenization

2014~2015:
Word / Char

2016~present:
BPE / WordPiece

Context-independent ⇒ Context-aware

Item Tokenization

Language Tokenization

2014~2015:
Word / Char

2016~present:
BPE / WordPiece

Context-independent ⇒ Context-aware

SemID Construction

2023~2024:
RQ / PQ / Clustering /
LM-based Generator

Item Tokenization

Language Tokenization

2014~2015:
Word / Char

2016~present:
BPE / WordPiece

Context-independent ⇒ Context-aware

SemID Construction

2023~2024:
RQ / PQ / Clustering / LM-based Generator

2025:
ActionPiece / **?**

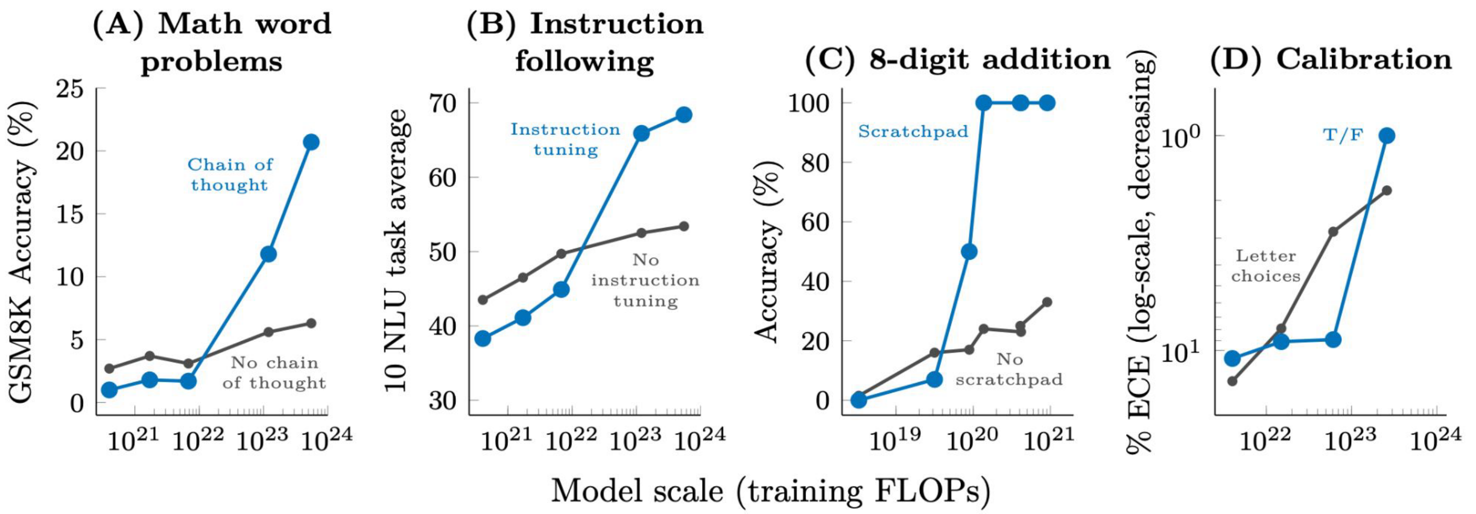
Hou et al. ActionPiece: Contextually Tokenizing Action Sequences for Generative Recommendation. arXiv:2502.13581.

Part 2: What Becomes Possible?

What new opportunities may large generative models unlock for recommender systems?

Emergent Ability

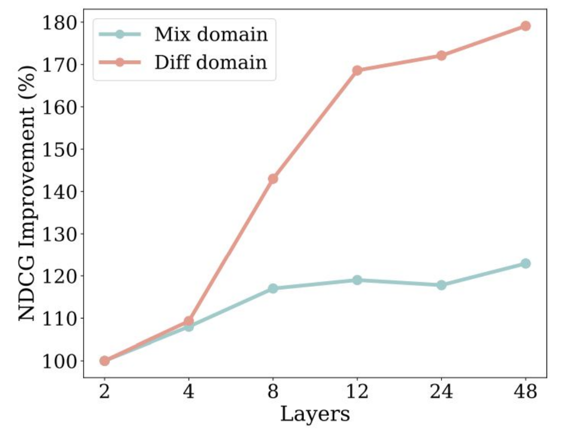
Abilities not present in smaller models but is present in larger models



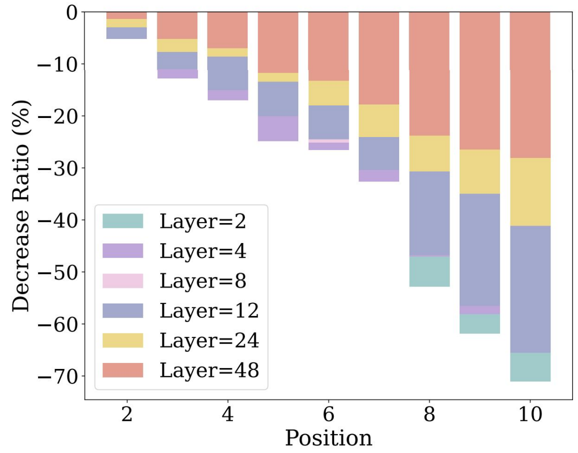
Wei et al. Emergent Abilities of Large Language Models. TMLR.

Emergent Ability

Do we have emergent abilities in large generative recommendation models?



Cross-domain

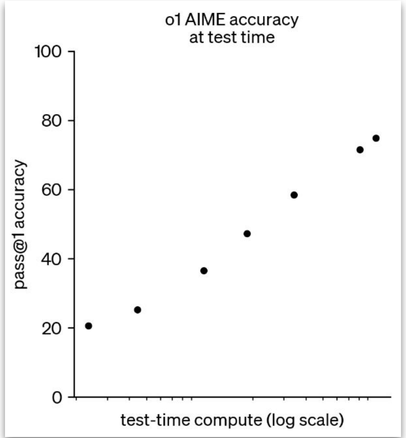


Trajectory Prediction

Zhang et al. Scaling Law of Large Sequential Recommendation Models. RecSys 2024.

Test-time Scaling

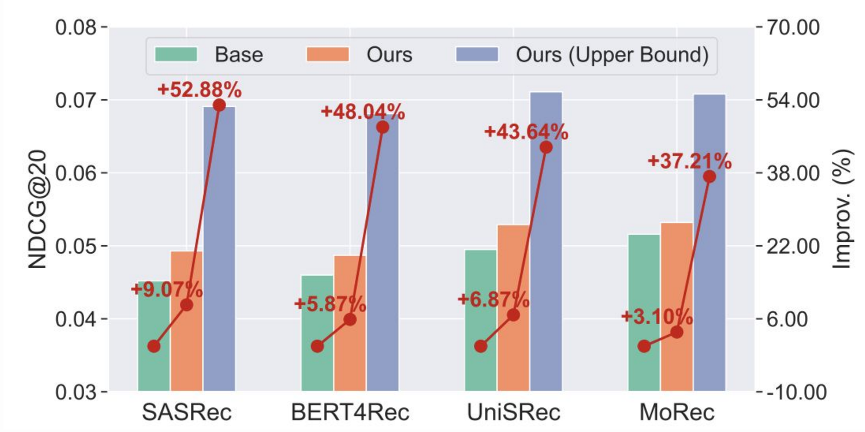
There have been explorations on model / data scaling of recommendation models

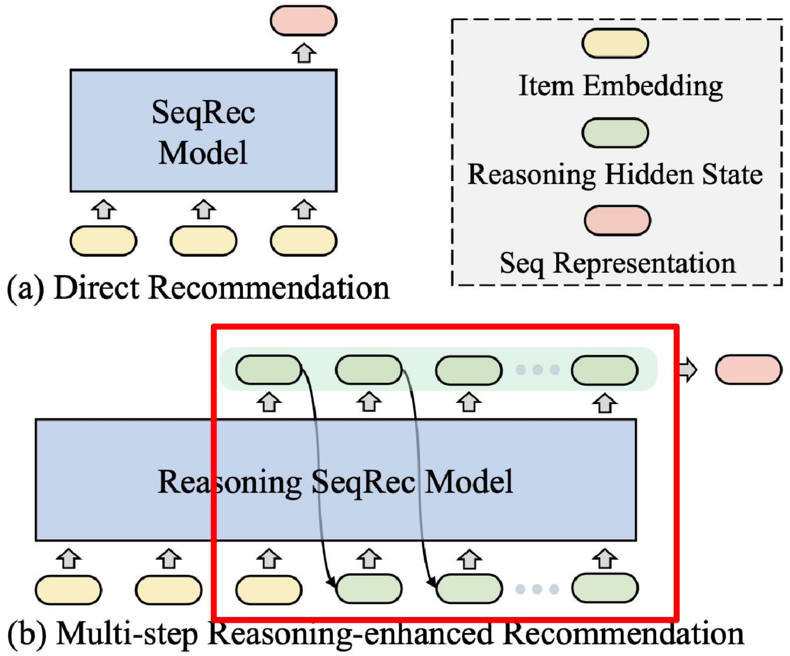


Test-time scaling is still under exploration

Test-time Scaling

Reasoning over latent hidden states to scale up test-time computation.





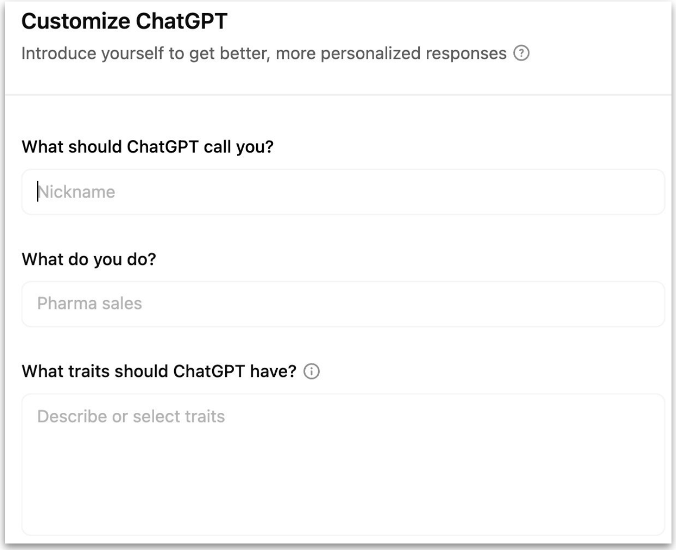
Large Diffusion Model

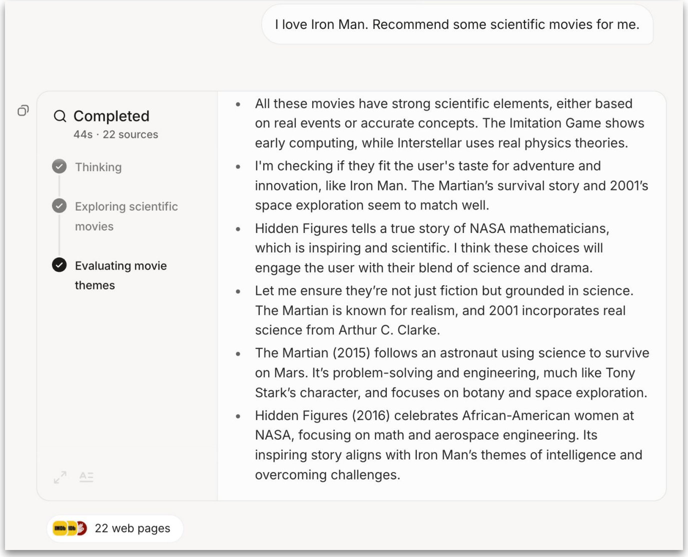
End-to-end personalized content generation with diffusion model.

- Existing work largely uses **pretrained diffusion model** to generate personalized content.
- An **end-to-end training** could bring mere personalization.

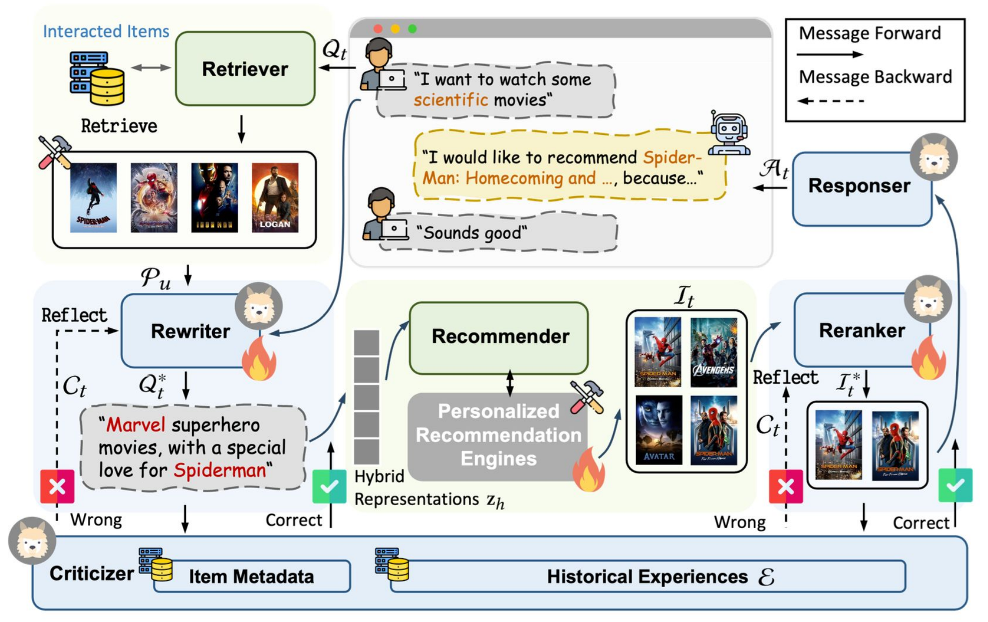
Agentic

Next: personalization & agentic system



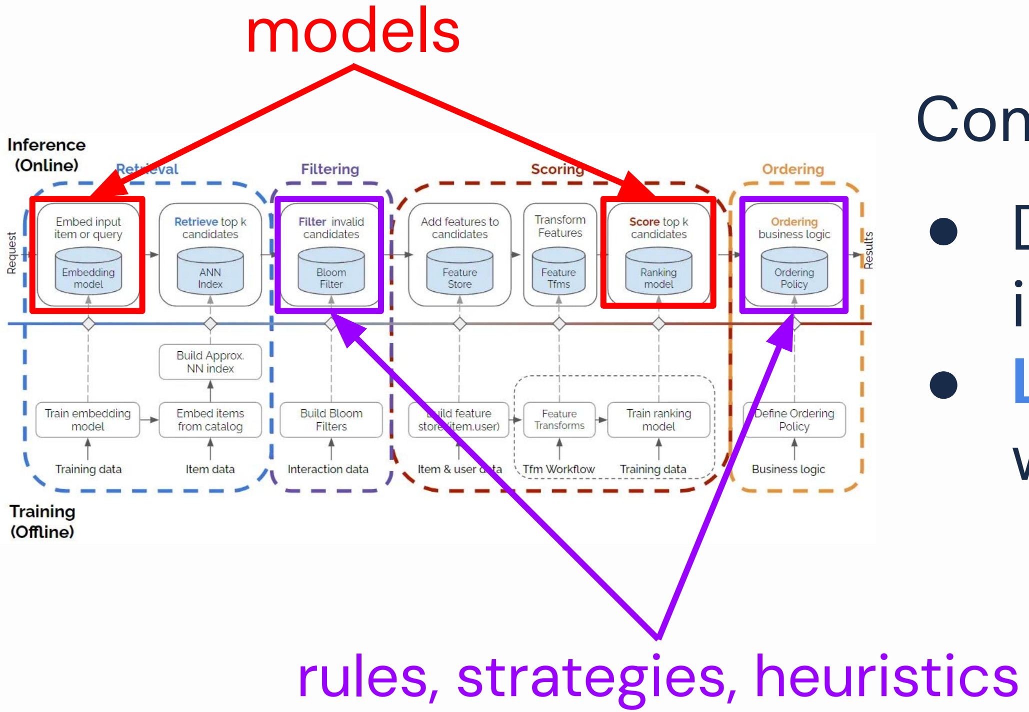


Agentic



Large agentic system for precise user modeling and better recommendation

Unified Retrieval & Ranking

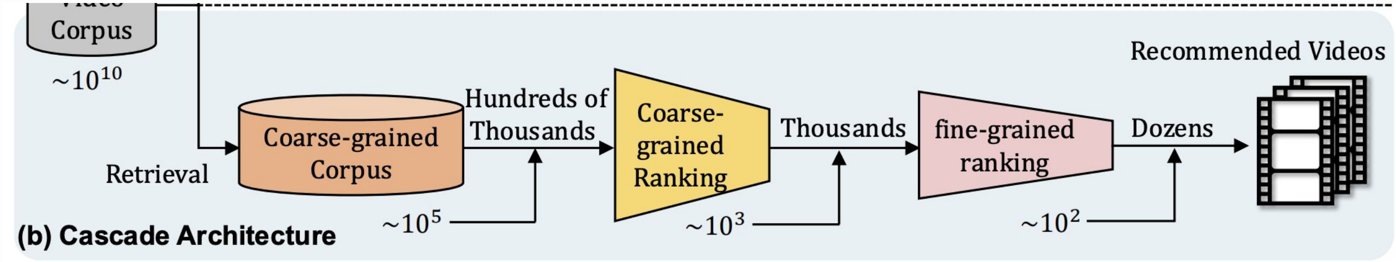


Complicated Architecture

- Difficult to be optimized in an **end-to-end** way
- **Latency** between / within different modules

Unified Retrieval & Ranking

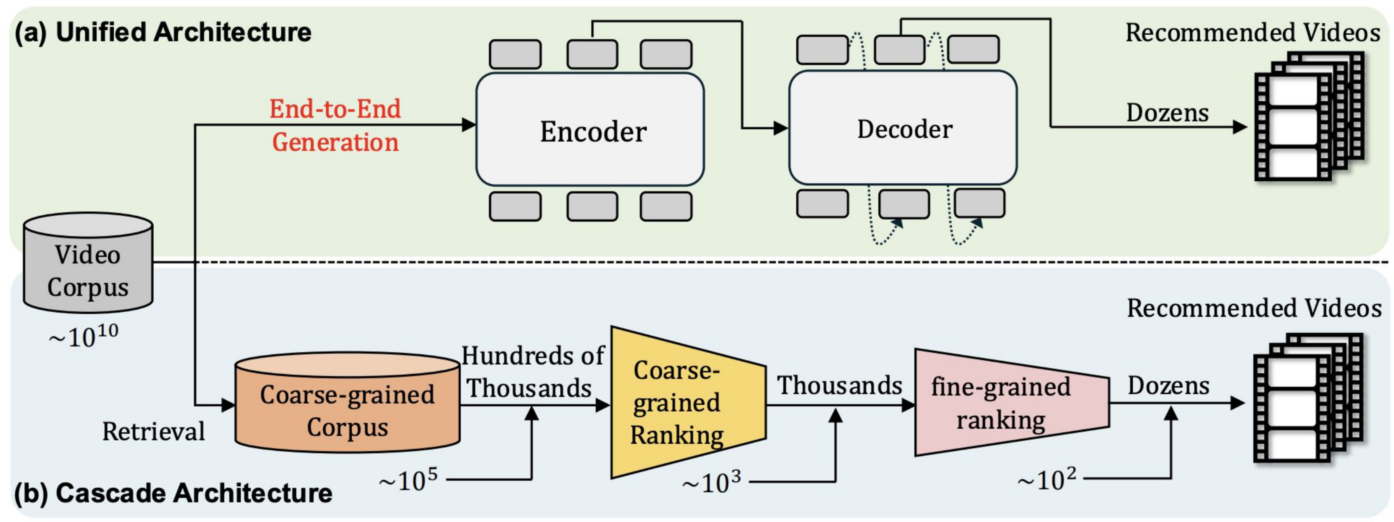
Is it possible to replace traditional cascade architecture



(b) Cascade Architecture

Unified Retrieval & Ranking

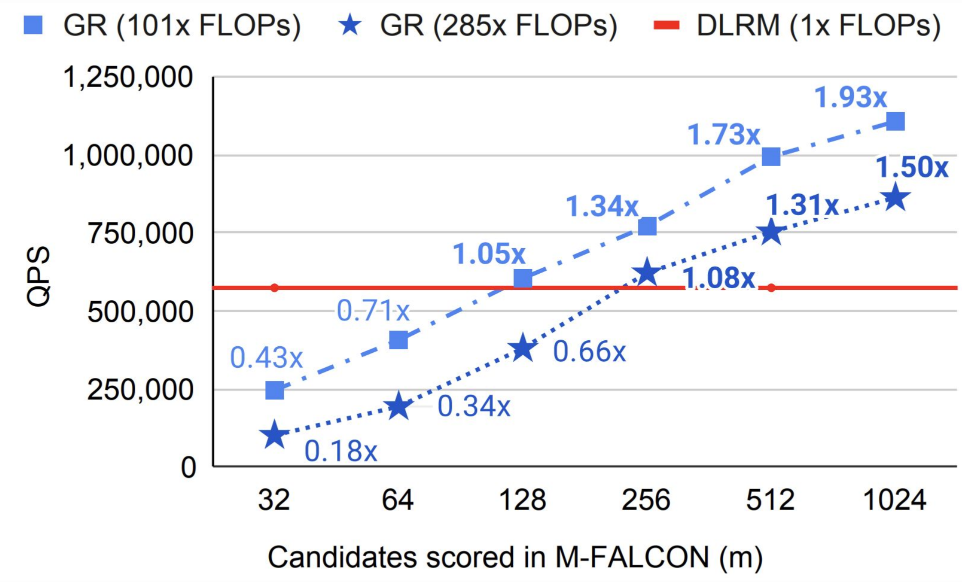
Is it possible to replace traditional cascade architecture with a **unified generative model**?

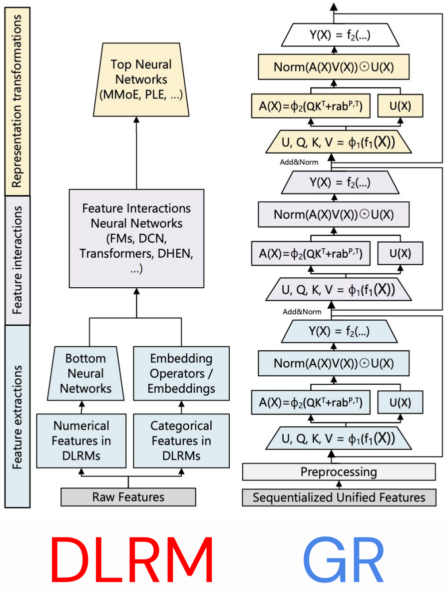


Deng et al. OneRec: Unifying Retrieve and Rank with Generative Recommender and Iterative Preference Alignment. arXiv:2502.18965.

Unified Retrieval & Ranking

Better throughout when ranking more candidates





Q & A

Thank you for coming!

Please refer to

large-genrec.github.io

for *slides*, *paper list*, ……

- Papers
  - Surveys
  - LLM-based Generative Recommendation
    - LLM as Sequential Recommender
      - Early Efforts: Zero-shot Recommendation with LLMs
      - Aligning LLMs for Recommendation
      - Training Objectives & Inference
    - LLM as Conversational Recommender & Recommendation Assistant
    - LLM as User Simulator
  - Semantic ID-based Generative Recommendation
    - Semantic ID Construction
      - Quantization
      - Hierarchical Clustering
      - Contextual Action Tokenization
      - Behavior-aware Tokenization
      - Language Model-based Generator
    - Architecture
      - Dense & Generative Retrieval
      - Unified Retrieval and Ranking
    - Aligning with LLMs
  - Diffusion Model-based Generative Recommendation
    - Diffusion-enhanced Recommendation
    - Diffusion as Recommender
    - Personalized Content Generation with Diffusion
- Resources