THE HONG KONG POLYTECHNIC UNIVERSITY
香港理工大學

Learning to Tokenize ID for LLM-based Recommendations

Wenqi Fan (范文琦)

Department of Computing (COMP) & Department of Management and Marketing (MM)

The Hong Kong Polytechnic University (PolyU)

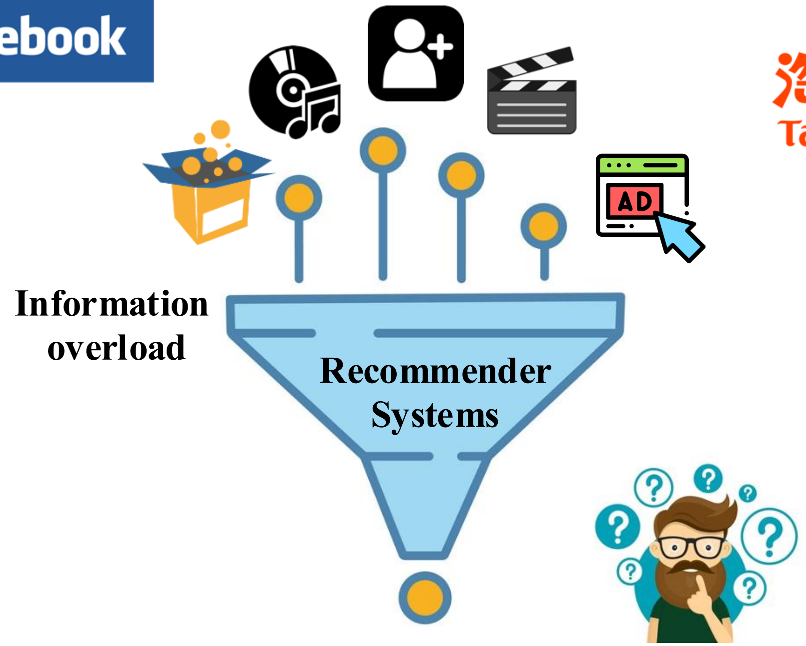
Email: wenqi.fan@polyu.edu.hk, Homepage: https://wenqifan03.github.io

Haohao Qu, Wenqi Fan, Zihuai Zhao, and Qing Li.
TokenRec: Learning to Tokenize ID for LLM-based Generative Recommendation. arXiv:2406.10450, 2024.

Recommender Systems (RecSys)

Age of Information Explosion

amazon JD.COM LinkedIn facebook 淘宝网 Taobao.com



Recommend item X to user

Items can be: Products, Friends, News, Movies, Videos, etc.

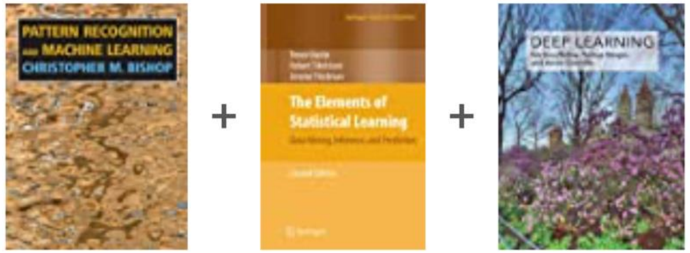
Recommender Systems (RecSys)

- Recommendation has been widely applied in online services:
  - E-commerce, Content Sharing, Social Networking, …

amazon ebay 淘宝网 Taobao.com JD.COM

Product Recommendation

Frequently bought together



A + B + C

Total price: $208.9

Add all three to Cart

Add all three to List

Amazon's recommendation algorithm drives **35%** of its sales [from McKinsey, 2012]

Recommender Systems (RecSys)

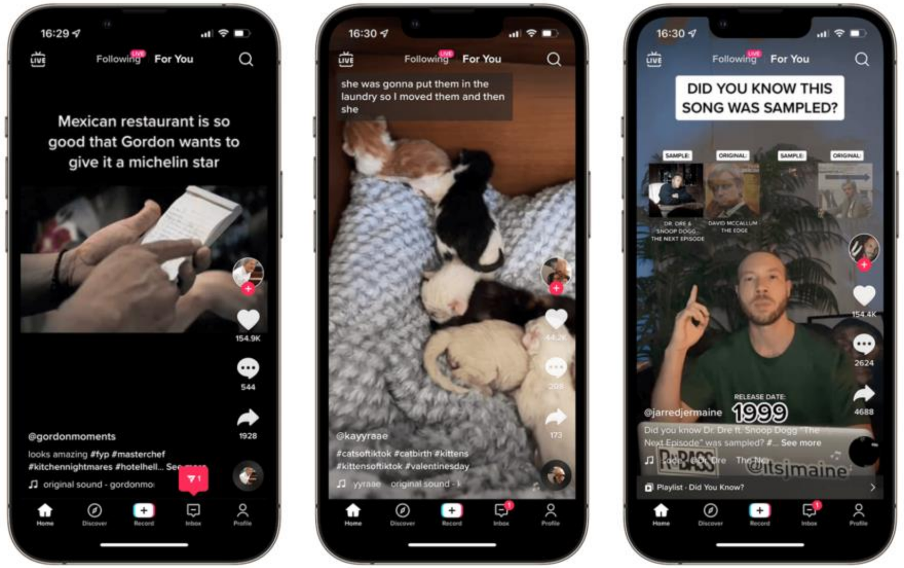
- ❑ Recommendation has been widely applied in online services:
  - ❖ E-commerce, Content Sharing, Social Networking ...

News/Video/Image Recommendation

TikTok's recommendation algorithm

Top 10 Global Breakthrough Technologies in 2021

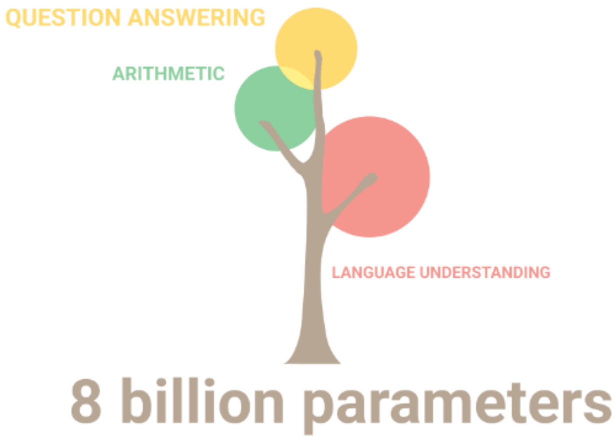
MIT Technology Review



Large Language Models (LLMs)

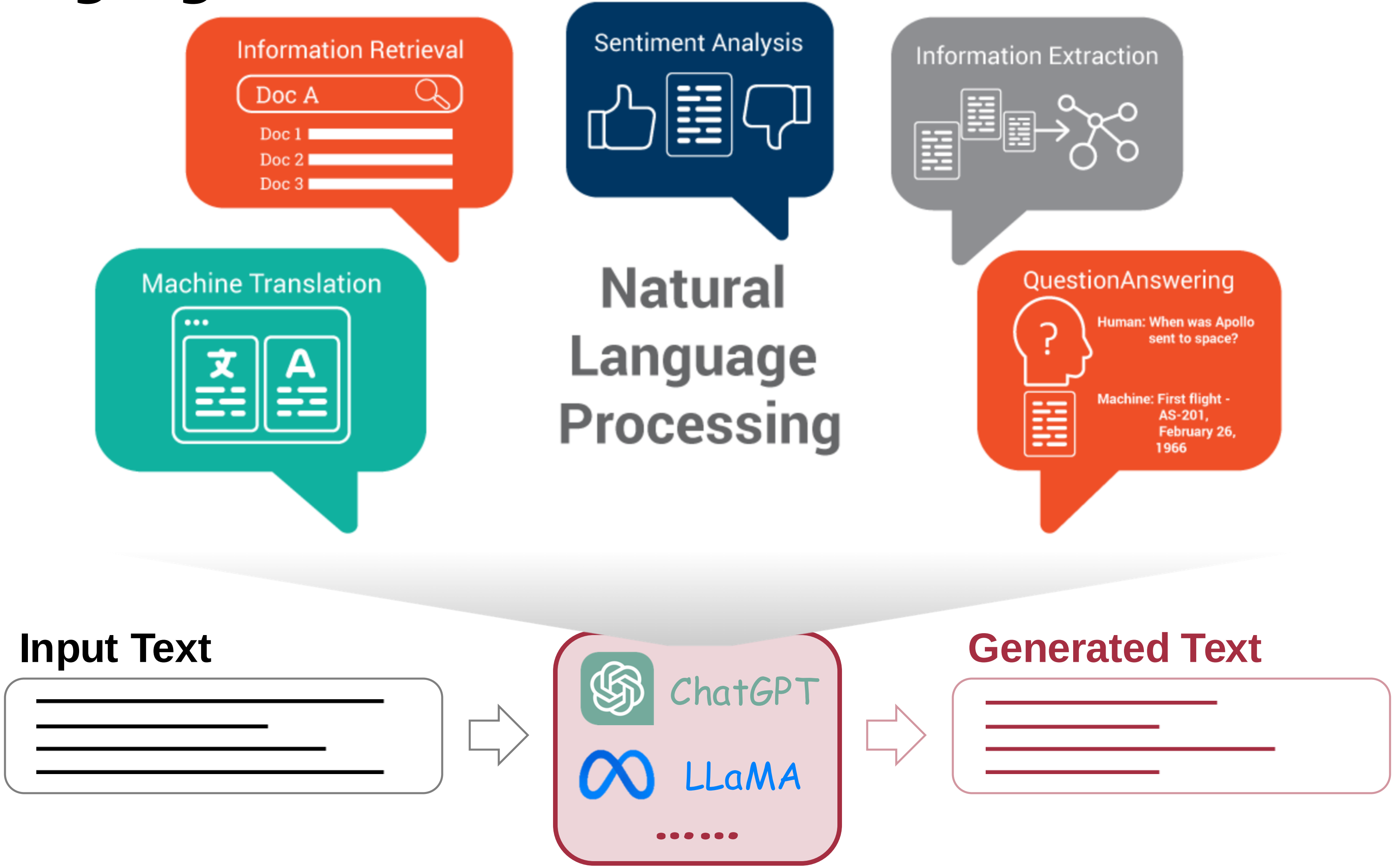
They Are Changing Our Lives !

OpenAI | Meta | Google AI | T5 | ……



https://github.com/Hannibal046/Awesome-LLM/tree/main

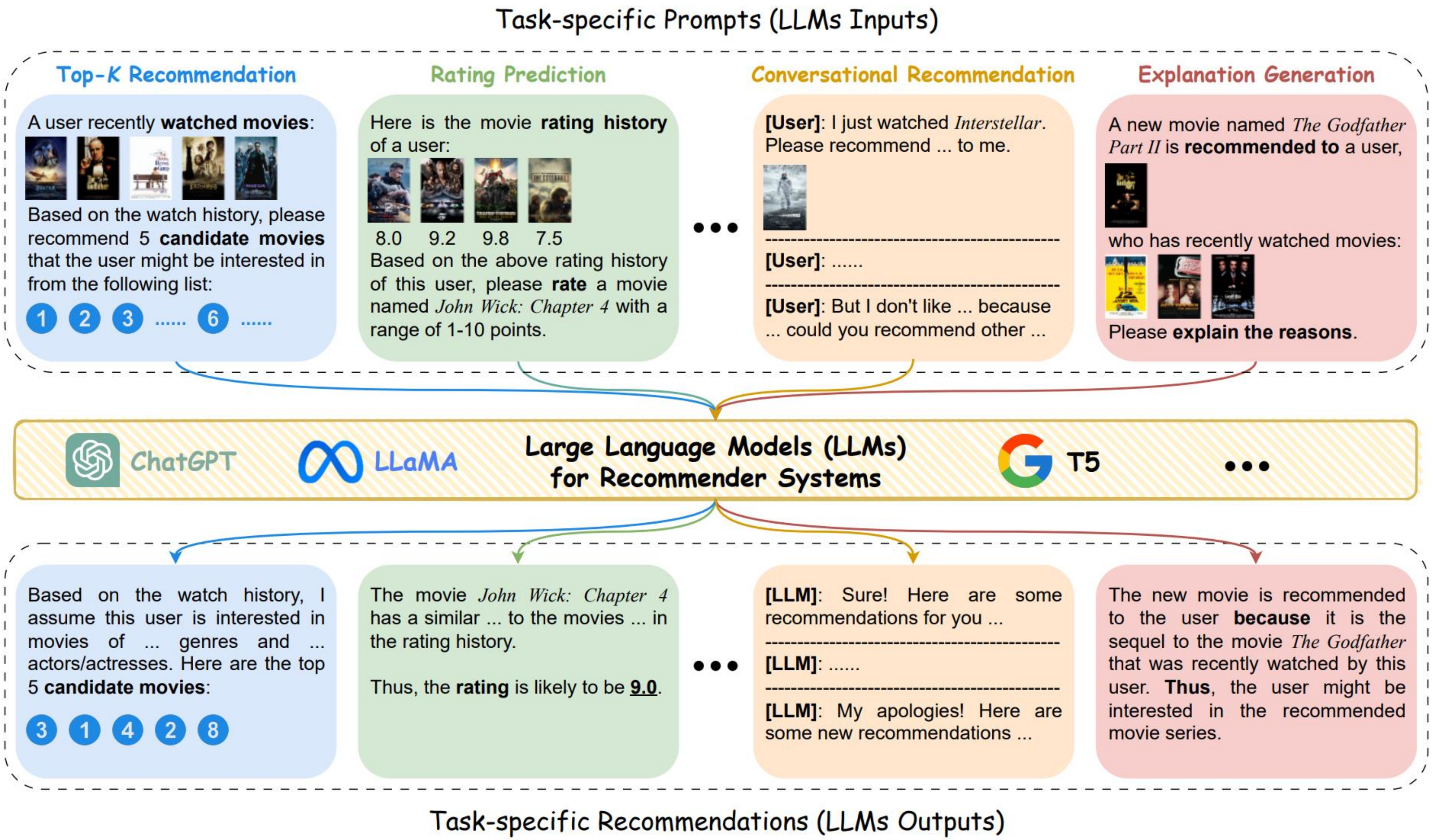
Large Language Models (LLMs)



Large Language Models (LLMs)

Large Language Models (LLMs)

A Promising Avenue: **LLM-empowered Recommender Systems**



* Zhao, Zihuai, et al. "Recommender systems in the era of large language models (llms)." IEEE Transactions on Knowledge and Data Engineering (2024).

Introduction

The seamless alignment of LLMs and RecSys is not a trivial task.

Challenge: How to Effectively Index User and Item IDs for LLM-based Recommendations?

Example:

I find the purchase history list of Peter:
Iphone 15 Pro Max, GPU, Apple Watch, …
I wonder what is the next item to recommend to the user. Can you help me decide?

LLM-based RecSys

The user might like MacBook 16 Pro Max.

Potential Problems

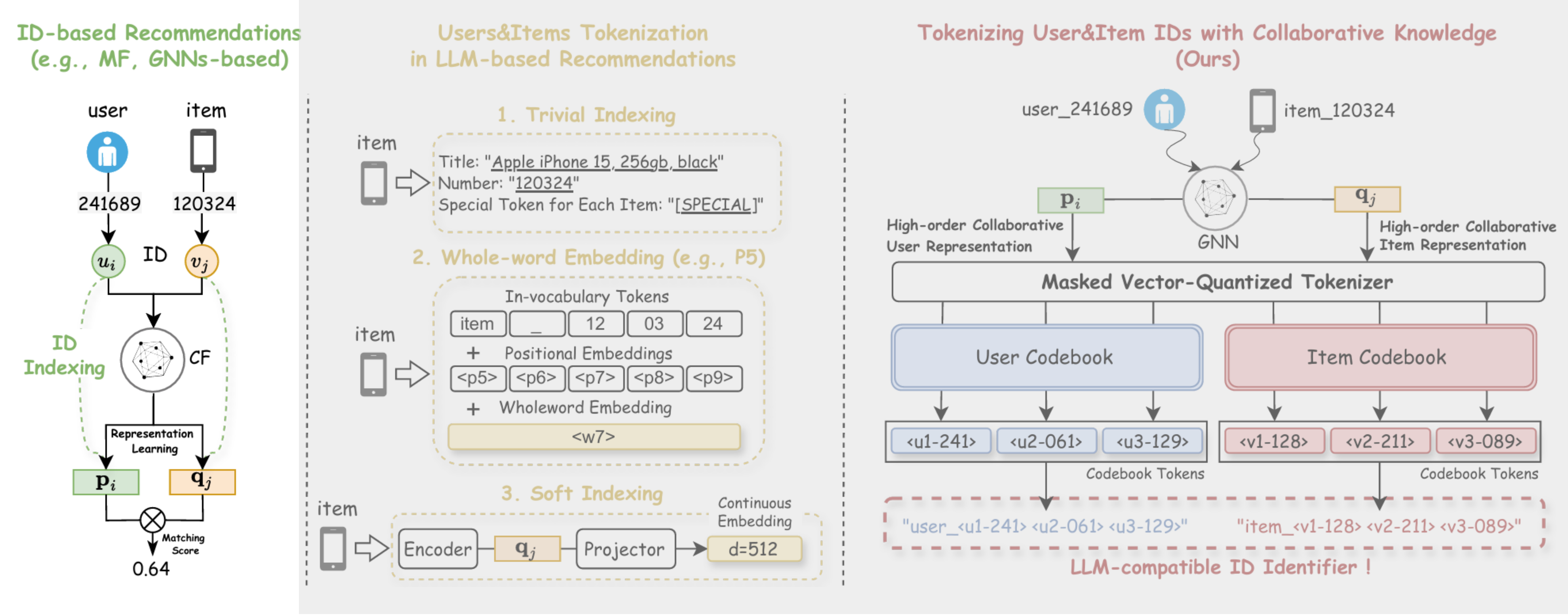
- **Ambiguity** (e.g., Peter, GPU): users and items need detailed information to identify themselves in LLM-based RecSys.
- **Over-Length**: In recommendation scenarios with a high volume of interactions, it is probable that the input length may exceed the token limit of the LLM.

- **Hallucination** (MacBook 16 Pro Max): The generated text may not even correspond to a real existing item in the item database.
- **Time-consuming Inference**: The auto-regressive decoding and beam search processes for generating items are laborious for existing LLM-based RecSys.
- ……

Hua, Wenyue, et al. "How to index item ids for recommendation foundation models." ACM SIGIR-AP, 2023.

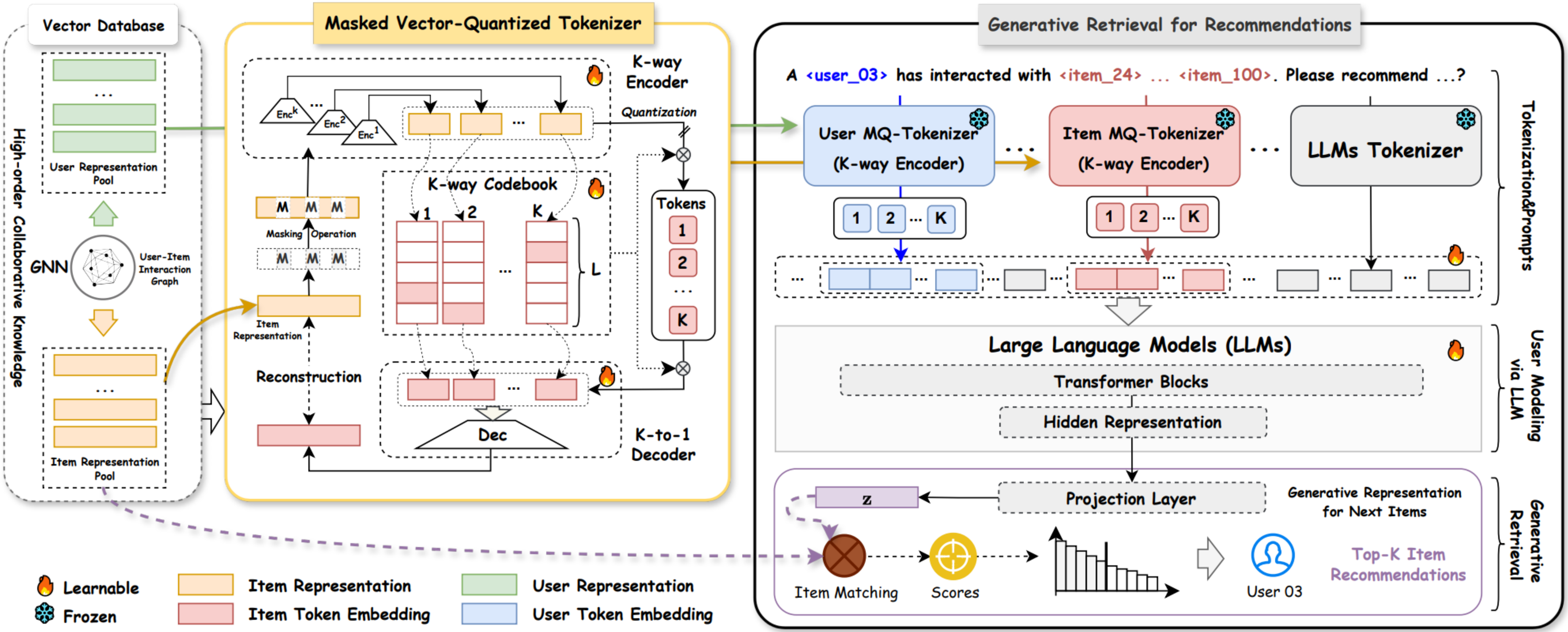
Users and Items Indexing in Collaborative Recommendations

**Tokenizing Users/Items with Collaborative Knowledge
into Discrete Tokens that are compatible for Natural Language**



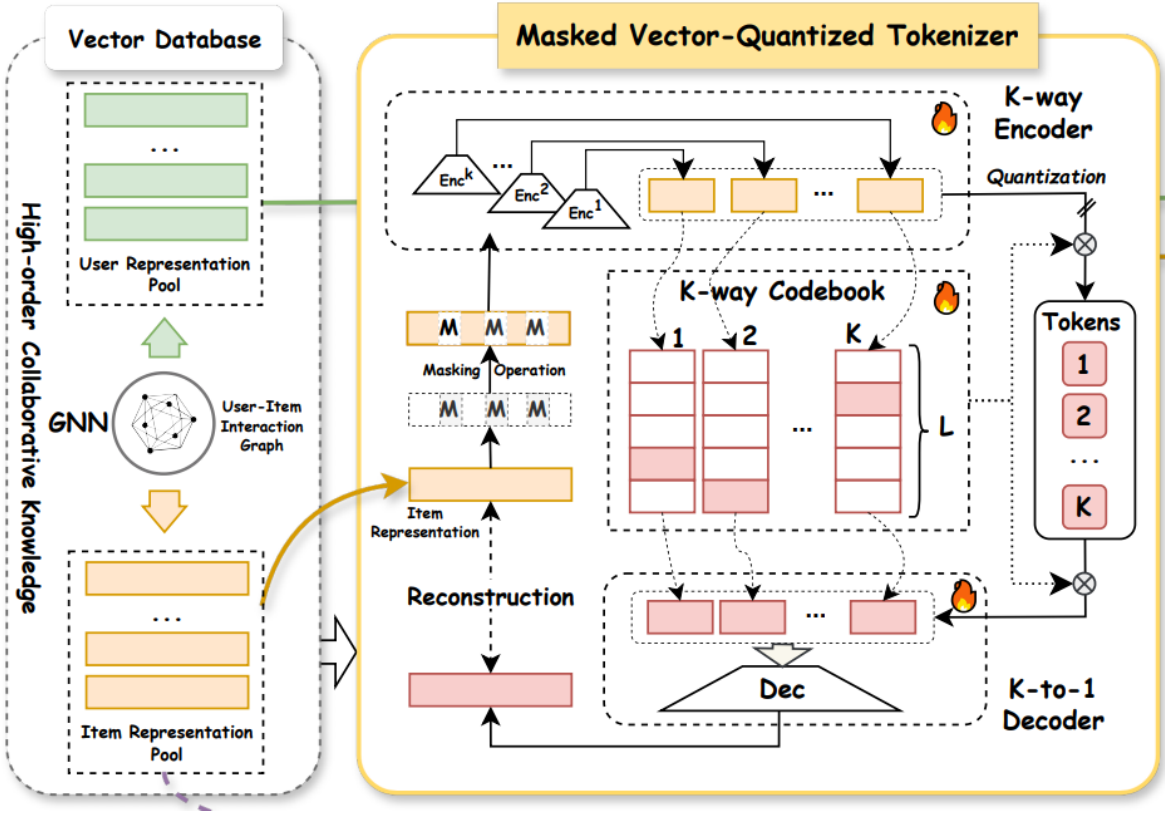
Methodology: Overview (Our TokenRec)

Learning to Tokenize ID for LLM-based Recommendations



Methodology: Masked Vector-Quantized Tokenizers (MQ-Tokenizer)

Encode Users&Items into Discrete Tokens.



Step 1: Masking (Generalizability)

$$\mathbf{p}_i' = \mathrm{Mask}(\mathbf{p}_i, \mathcal{E}),\ \ \mathbf{q}_j' = \mathrm{Mask}(\mathbf{q}_j, \mathcal{E}),\ \ \mathcal{E} \sim \mathrm{Bernoulli}(\rho),$$

Step 2: K-way Encoding (Generalizability)

$$\mathbf{a}_j^k = \mathrm{Enc}^k(\mathbf{q}_j') = \mathrm{MLP}^k(\mathbf{q}_j'),$$
$$w_j^k = \arg\min_l \|\mathbf{a}_j^k - \mathbf{c}_l^k\|^2,$$
$$\mathrm{Quantize}(\mathbf{a}_j^k) = \mathbf{c}_{w_j^k}^k,$$

item $v_j \rightarrow$ tokens: $\{w_j^1, w_j^2, ..., w_j^K\}$
$\rightarrow$ tokens' embeddings: $[\mathbf{c}_{w_j^1}^1, \mathbf{c}_{w_j^2}^2, ..., \mathbf{c}_{w_j^K}^K]$.

Step 3: K-to-1 Decoding (Self-supervised Training)

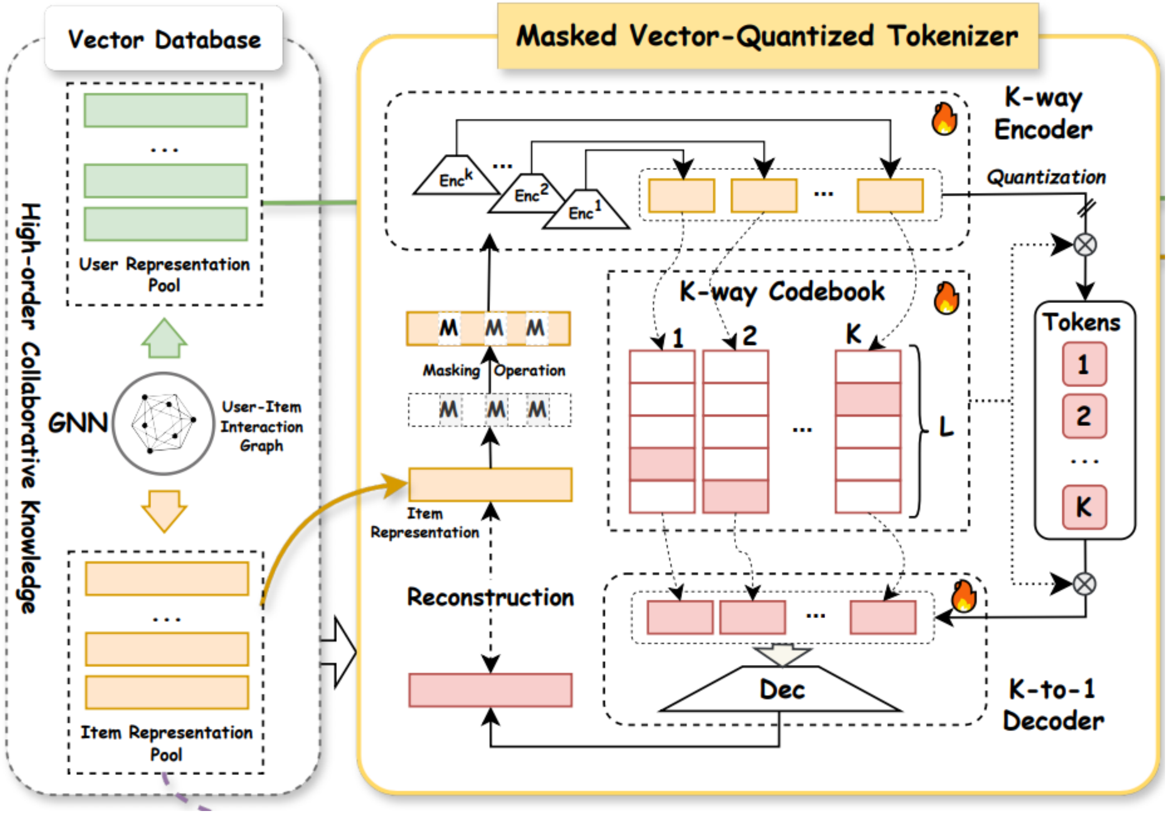
$$\mathbf{r}_j = \mathrm{Dec}(\{w_j^1, w_j^2, ..., w_j^K\}) = \mathrm{MLP}(\frac{1}{K}\sum_{k=1}^{K} \mathbf{c}_{w_j^k}^k).$$

Step 0:
Pre-training and Initialization (Collaborative Knowledge)

GNN: $\mathbf{p}_i \in \mathbb{R}^d$, $\mathbf{q}_j \in \mathbb{R}^d$ Codebook: $\mathbf{c}^k \in \mathbb{R}^{L \times d_c}$

Methodology: Masked Vector-Quantized Tokenizers (MQ-Tokenizer)

Encode Users&Items into Discrete Tokens



Learning Objective:

$$\mathcal{L}_{recon}^{Item} = \|\mathbf{q}_j - \mathbf{r}_j\|^2$$ Reconstruction

$$\mathcal{L}_{cb}^{Item} = \sum_{k=1}^{K} \|\text{sg}[\ \text{Enc}^k(\mathbf{q}'_j)\ ] - \mathbf{c}^k_{w^k_j}\|^2$$ Codebook

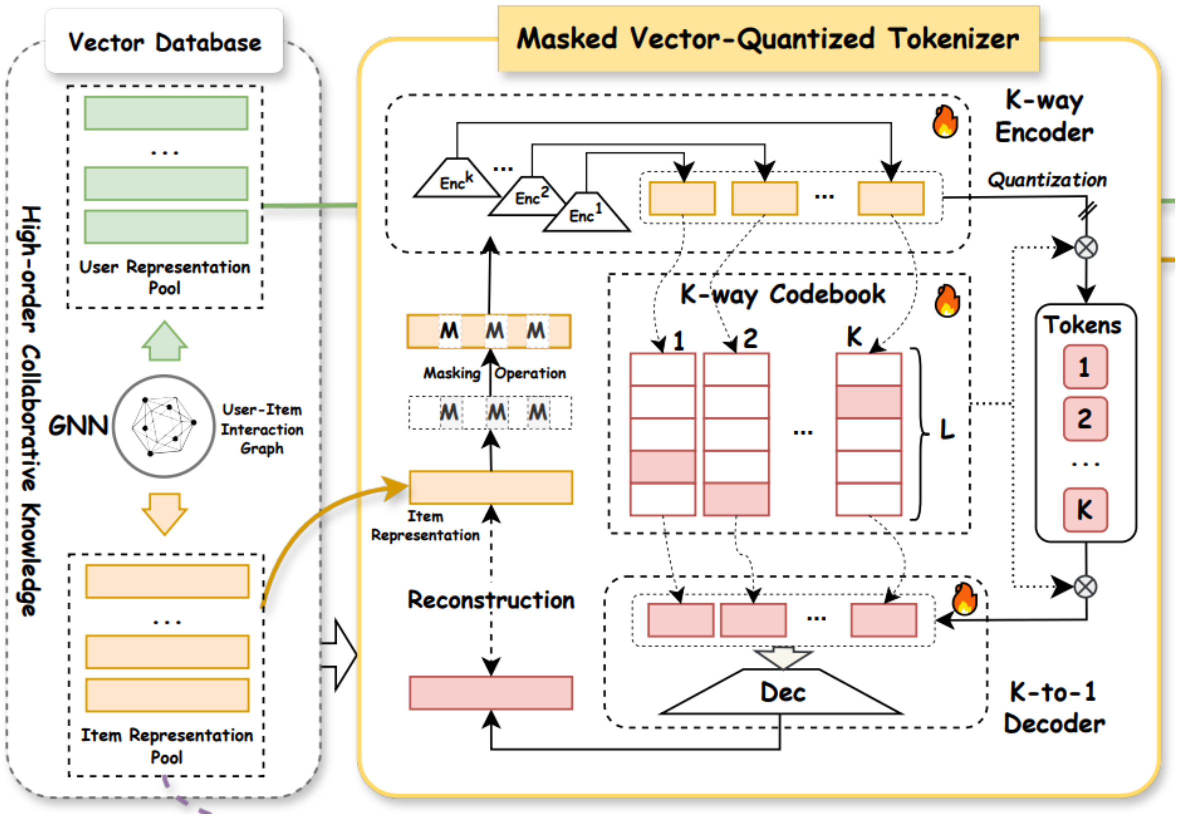
$$\mathcal{L}_{cm}^{Item} = \sum_{k=1}^{K} \|\ \text{Enc}^k(\mathbf{q}'_j)\ - \text{sg}[\mathbf{c}^k_{w^k_j}]\|^2$$ smooth gradient passing

MQ-Tokenizer:

$$\mathcal{L}_{MQ}^{Item} = \mathcal{L}_{recon}^{Item} + \mathcal{L}_{cb}^{Item} + \beta^{Item} \times \mathcal{L}_{cm}^{Item}$$

Methodology: MQ Tokenizer

Encode Users&Items into Discrete Tokens



By doing so, we can use only **1,536 (i.e., 3 × 512)** out-of-vocabulary (OOV) tokens to tokenize a total of **39,387 items** in the Amazon-Clothing dataset.

Prompt 1 (without user's historical interactions):
I wonder what the **user_03** will like. Can you help me decide?

$\Longrightarrow$ I wonder what the *user_* $\langle u1\text{-}128\rangle\langle u2\text{-}21\rangle\langle u3\text{-}35\rangle$ will like. Can you help me decide?

Prompt 2 (with user's historical interactions):
According to what items the **user_03** has interacted with: **item_08**, **item_24**, **item_63**. Can you describe the user's preferences?

$\Longrightarrow$ According to what items the *user_* $\langle u1\text{-}128\rangle\langle u2\text{-}21\rangle\langle u3\text{-}35\rangle$ has interacted with:
*item_* $\langle v1\text{-}42\rangle\langle v2\text{-}12\rangle\langle v3\text{-}98\rangle$,
*item_* $\langle v1\text{-}42\rangle\langle v2\text{-}12\rangle\langle v3\text{-}87\rangle$,
*item_* $\langle v1\text{-}42\rangle\langle v2\text{-}53\rangle\langle v3\text{-}128\rangle$.
Can you describe the user's preferences?

Methodology: Generative Retrieval

Time-consuming inference in decoding process

LLM-based RecSys encounter challenges with time-consuming inference because of the laborious auto-regressive decoding and beam search processes.

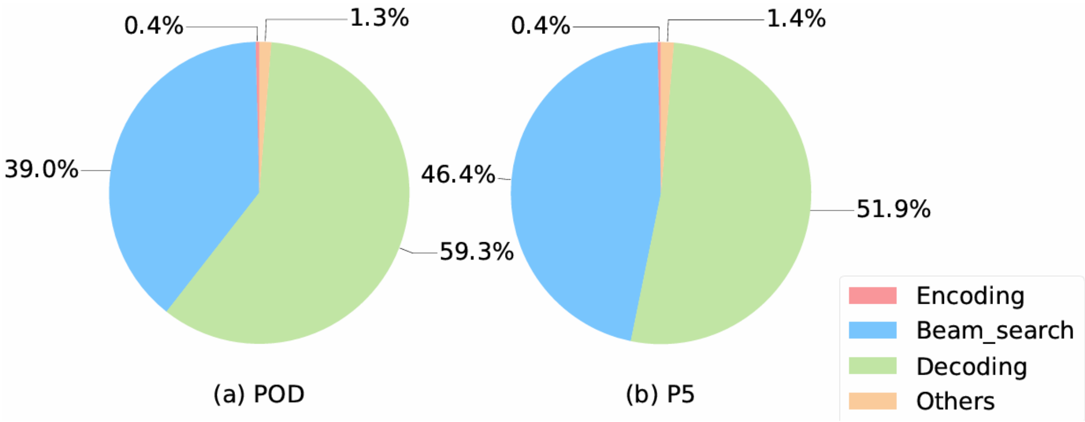


Image Credit: Wang H, Liu X, Fan W, et al. Rethinking large language model architectures for sequential recommendations[J]. arXiv preprint arXiv:2402.09543, 2024.

Hallucination issue
(invalid item identifiers)

For example, items' title "***iPhone SE, 256 GB, starlight***" "***iPhone 15, 256 GB, starlight***" share most tokens but are significantly different products - with "***~~iPhone 15, 256 GB, starlight~~***" being a non-factual product.

Unseen items in inference stage

Methodology: Generative Retrieval

The Proposed Pipeline:

Step 1. Constructing Query

$$\mathcal{X}_i \rightarrow (\mathcal{P}, \mathcal{T}^c_{u_i}) \ or \ (\mathcal{P}, \mathcal{T}^c_{u_i}, \{\mathcal{T}^c_{v_j} | v_j \in \mathcal{N}_{(u_i)}\}),$$

Step 2. User Modeling via LLMs

$$\mathbf{h}_i = \mathrm{LLM4Rec}(\mathcal{X}_i).$$

Step 3. Generating User Preference

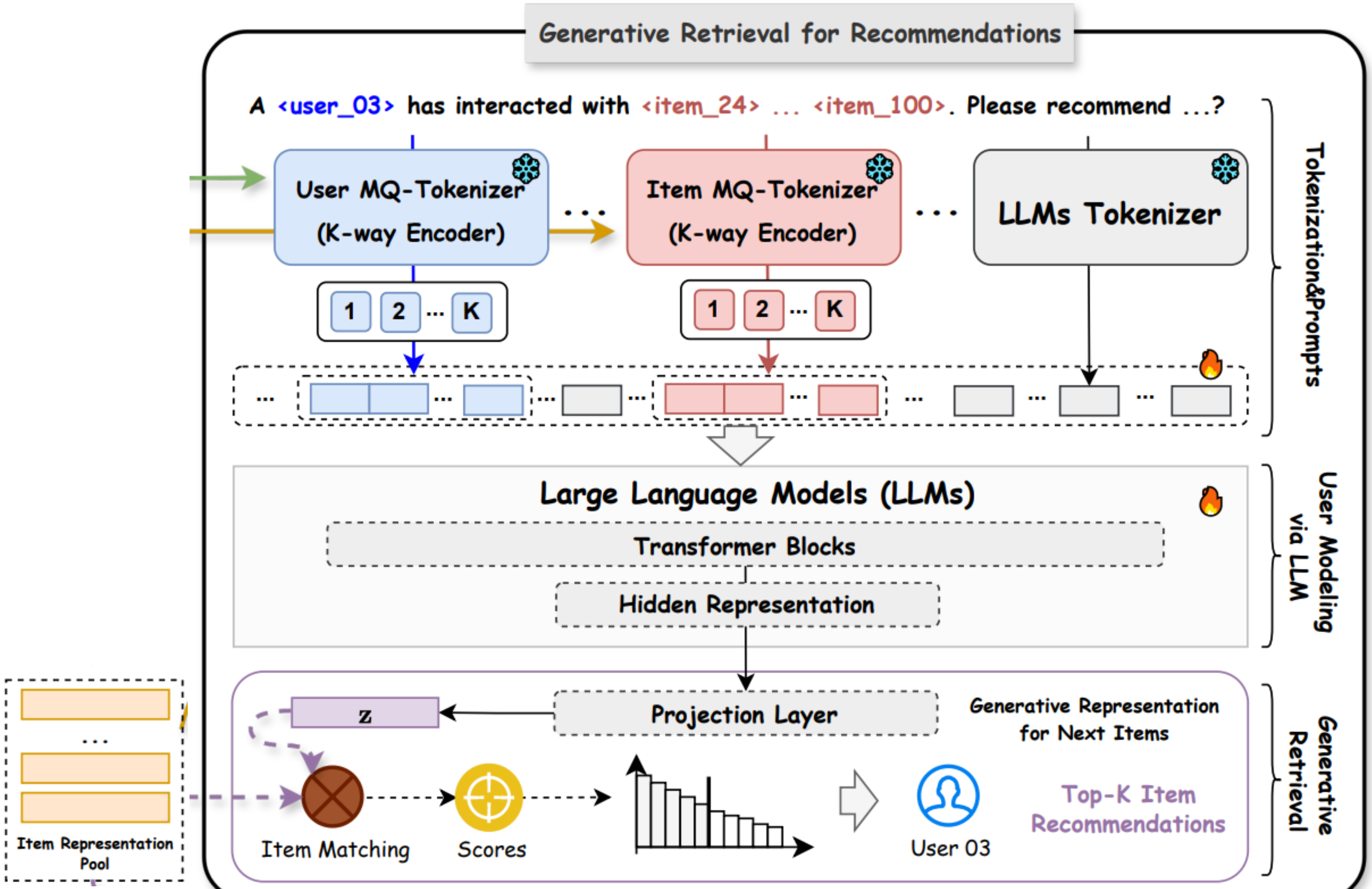
$$\mathbf{z}_i = \mathrm{Proj}(\mathbf{h}_i),$$

Step 4. Scoring

$$y_{ij} = \frac{\mathbf{z}_i \mathbf{q}_j}{\|\mathbf{z}_i\| \|\mathbf{q}_j\|}.$$

Step 5. Top-K Retrieval

Generate User Preferences for Top-K recommendations



Methodology: Generative Retrieval

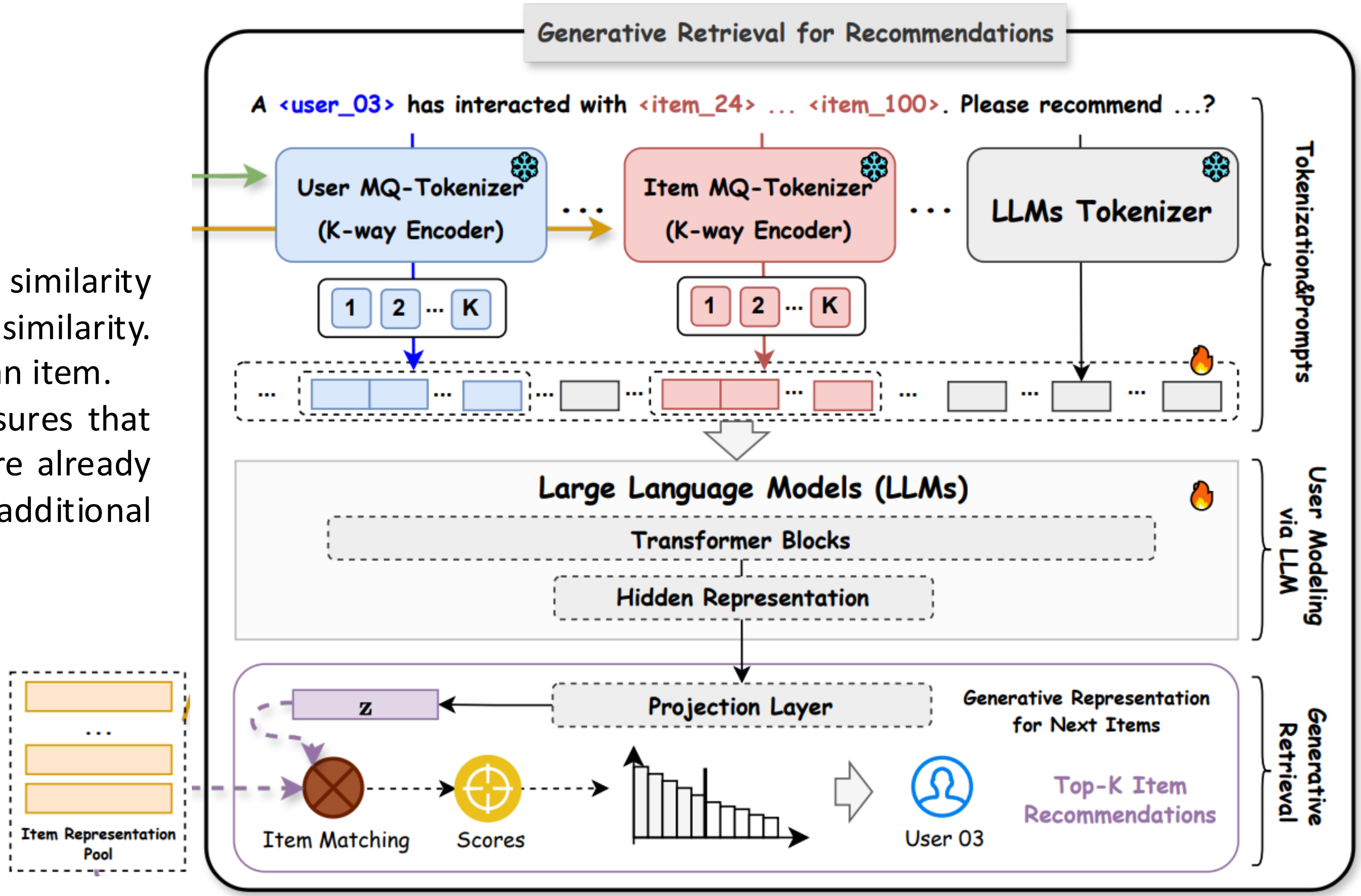
The Learning Objective:

A pairwise ranking loss

$$\mathcal{L}_{\text{LLM4Rec}} = \begin{cases} 1 - \text{sim}(\mathbf{z}_i, \mathbf{q}_j), & \text{if } \lambda = 1 \\ \max(0, \text{sim}(\mathbf{z}_i, \mathbf{q}_j) - \gamma), & \text{if } \lambda = -1 \end{cases}$$

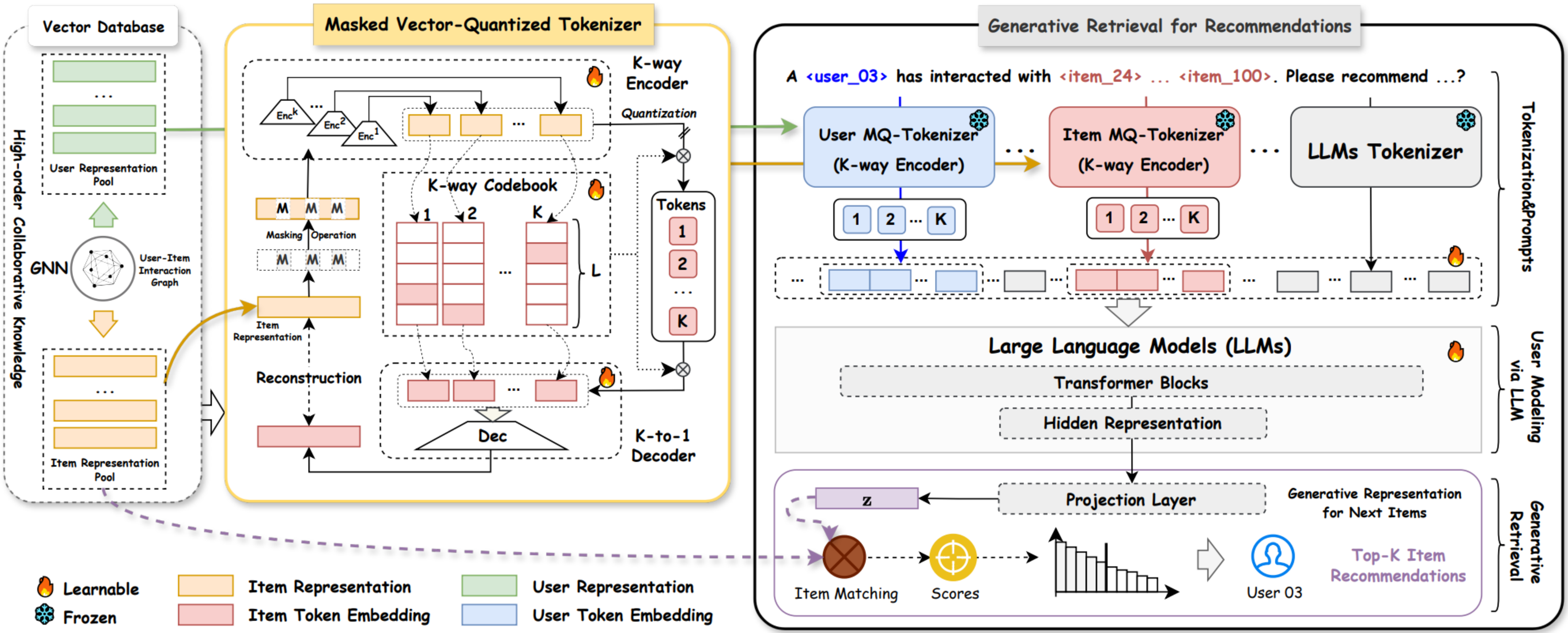
- sim(·, ·) is a metric function to measure the similarity between dense representations, such as cosine similarity.
- λ indicates whether a user has interacted with an item.
- γ is the margin value for negative pairs. It ensures that when the representations of a negative pair are already adequately distant, there is no need to expend additional effort in increasing the distance between them.

Generate User Preferences for Top-K recommendations



Methodology: Overview

Learning to Tokenize ID for LLM-based Recommendations



Methodology: Discussion

- **Efficient Recommendations**

TokenRec proposes a novel LLM-empowered collaborative recommendation framework in generative retrieval paradigms, **bypassing the time-consuming decoding process.**

- **Generalizability to New Users and Items**

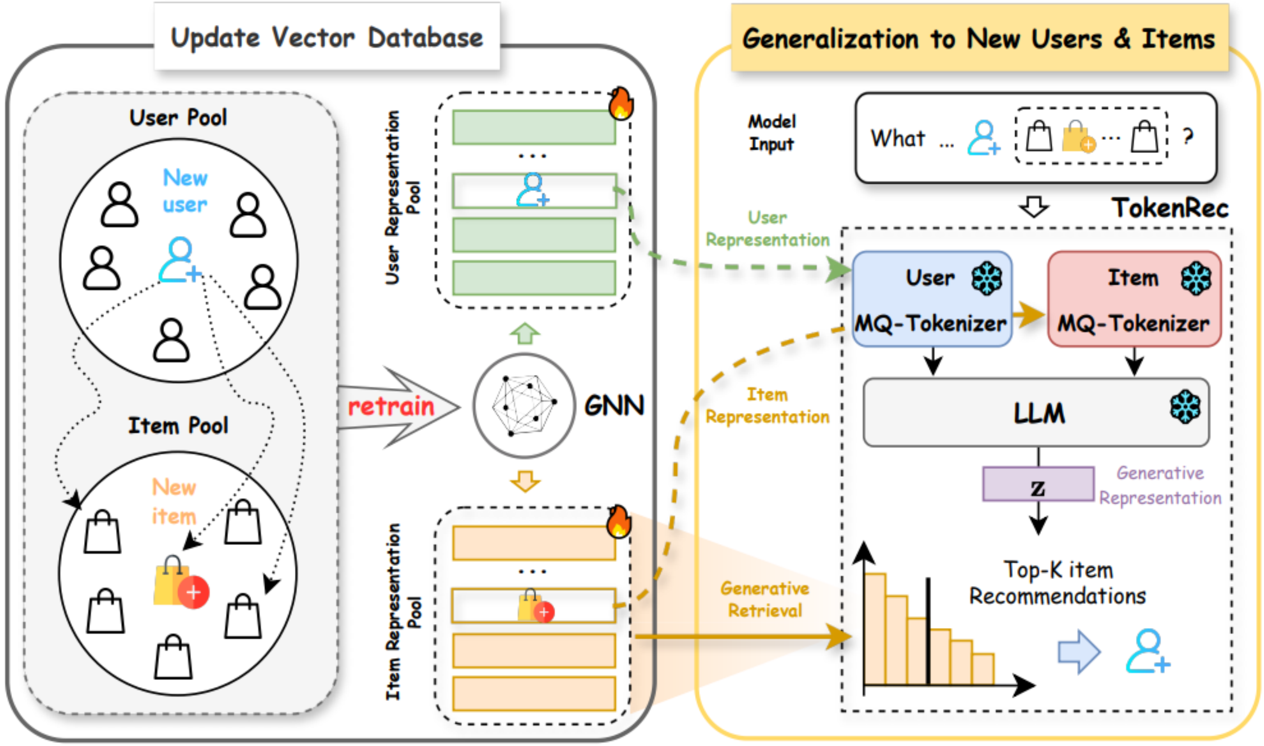
The proposed architecture can provide robust ID tokenization for unseen users and items **without fine-tuning the LLM4Rec component**.

- **Concise Prompts**

TokenRec provides an inference alternative that relies solely on **user ID tokens,**.

Prompt 1 (without user's historical interactions):
I wonder what the **user_03** will like. Can you help me decide?

$\Longrightarrow$ I wonder what the *user_* $\langle u1\text{-}128\rangle \langle u2\text{-}21\rangle \langle u3\text{-}35\rangle$ will like. Can you help me decide?



The proposed framework allows the generalization to new users&items by updating the external vector database instead of the LLM backbone and the Tokenizers.

Evaluation: Settings

1) Task

2) Datasets

TABLE I: Basic statistics of benchmark datasets.

Datasets	User-Item Interaction			
	#Users	#Items	#Interactions	Density (%)
LastFM	1,090	3,646	37,080	0.9330
ML1M	3,416	6,040	447,294	2.1679
Beauty	22,363	12,101	197,861	0.0731
Clothing	23,033	39,387	278,641	0.0307

represents the number of users, items, and interactions.

3) Metrics: Top-K Hit Ratio (HR@K)
Top-K Normalized Discounted Cumulative Gain (NDCG@K)

The higher the metrics, the better the performance.

Evaluation: Settings

3) Baselines

- **Collaborative Filtering (5):** MF, NeuCF , LightGCN, GTN, LTGNN.
- **Sequential Recommenders (3)**: SASRec, BERT4Rec, and S3Rec.
- **LLM-based Methods (4):**
  - **P5** is a pioneering work on LLM-based RecSys, which describes recommendation tasks in a text-to-text format and employs LLMs to capture deeper semantics for personalization and recommendation.
  - **CID** is a non-trivial indexing approach that considers the co-occurrence matrix of items to design numeric IDs so that items co-occur in user-item interactions will have similar numeric IDs.
  - **POD** encodes discrete prompts into continuous embeddings to reduce the excessive input length of LLMs based on P5 architecture.
  - **CoLLM** employs GNNs to provide continuous embeddings representing items and users for LLM-based recommendations.

Evaluation: Comparison Results

TABLE II: Performance comparison of recommendation algorithms on the LastFM and ML1M datasets.

Model	LastFM						ML1M					
	HR@10	HR@20	HR@30	NG@10	NG@20	NG@30	HR@10	HR@20	HR@30	NG@10	NG@20	NG@30
BERT4Rec	0.0319	0.0461	0.0640	0.0128	0.0234	0.0244	0.0779	0.1255	0.1736	0.0353	0.0486	0.0595
SASRec	0.0345	0.0484	0.0658	0.0142	0.0236	0.0248	0.0785	0.1293	0.1739	0.0367	0.052	0.0622
S^3Rec	0.0385	0.0490	0.0689	0.0177	0.0266	0.0266	0.0867	0.1270	0.1811	0.0361	0.0501	0.0601
MF	0.0239	0.0450	0.0569	0.0114	0.0166	0.0192	0.078	0.1272	0.1733	0.0357	0.0503	0.0591
NCF	0.0321	0.0462	0.0643	0.0141	0.0252	0.0254	0.0786	0.1273	0.1738	0.0363	0.0504	0.0601
LightGCN	0.0385	0.0661	0.0982	0.0199	0.0269	0.0336	0.0877	0.1288	0.1813	0.0374	0.0509	0.0604
GTN	0.0394	0.0688	0.0963	0.0199	0.0273	0.0331	0.0883	0.1307	0.1826	0.0378	0.0512	0.0677
LTGNN	0.0471	0.076	0.0925	0.0234	0.0318	0.0354	0.0915	0.1387	0.1817	0.0419	0.0570	0.0659
P5-RID	0.0312	0.0523	0.0706	0.0144	0.0199	0.0238	0.0867	0.1248	0.1811	0.0381	0.0486	0.0662
P5-SID	0.0375	0.0536	0.0851	0.0224	0.0255	0.0261	0.0892	0.1380	0.1784	0.0422	0.0550	0.0641
CID	0.0381	0.0552	0.0870	0.0229	0.0260	0.0277	0.0901	0.1294	0.1863	0.0379	0.0525	0.0706
POD	0.0367	0.0572	0.0747	0.0184	0.0220	0.0273	0.0886	0.1277	0.1846	0.0373	0.0487	0.0668
CoLLM	0.0483	0.0786	0.1017	0.0234	0.0319	0.0366	0.0923	0.1499	0.1998	0.0456	0.0620	0.0719
* (User ID Only)	0.0505	0.0881	0.1128	0.0251	0.0345	0.0397	0.0964	0.1546	0.2043	0.0493	0.0640	0.0745
* (Unseen Prompt)	0.0514	0.0917	**0.1294**	**0.0252**	0.0343	**0.0422**	**0.1012**	0.1672	0.2144	**0.0532**	**0.0698**	**0.0798**
TokenRec	**0.0532**	**0.0936**	0.1248	0.0247	**0.0348**	0.0415	0.1008	**0.1677**	**0.2149**	0.0528	0.0697	0.0797

* are the variants of **TokenRec**, namely the cases of using user ID tokens only for model inputs without considering item interaction history and using the unseen prompt during evaluation.

TokenRec significantly exceeds the strongest baselines by **19.08% on HR@20** and **9.09% on NCDG@20** in the LastFM dataset.

Evaluation: Comparison Results

TABLE III: Performance comparison of recommendation algorithms on the Beauty and Clothing datasets.

Model	Beauty						Clothing					
	HR@10	HR@20	HR@30	NG@10	NG@20	NG@30	HR@10	HR@20	HR@30	NG@10	NG@20	NG@30
BERT4Rec	0.0329	0.0464	0.0637	0.0162	0.0205	0.0255	0.0135	0.0217	0.0248	0.0061	0.0074	0.0079
SASRec	0.0338	0.0472	0.0637	0.0170	0.0213	0.0260	0.0136	0.0221	0.0256	0.0063	0.0076	0.0081
S^3Rec	0.0351	0.0471	0.0664	0.0169	0.0237	0.0278	0.0140	0.0213	0.0256	0.0069	0.0081	0.0086
MF	0.0127	0.0195	0.0245	0.0063	0.0081	0.0091	0.0116	0.0175	0.0234	0.0074	0.0088	0.0101
NCF	0.0315	0.0462	0.0623	0.0160	0.0196	0.0237	0.0119	0.0178	0.024	0.0072	0.0090	0.0103
LightGCN	0.0344	0.0498	0.0630	0.0194	0.0233	0.0261	0.0157	0.0226	0.0279	0.0085	0.0103	0.0114
GTN	0.0345	0.0502	0.0635	0.0198	0.0241	0.0268	0.0158	0.0226	0.0282	0.0084	0.0103	0.0111
LTGNN	0.0385	0.0564	0.0719	0.0207	0.0252	0.0285	0.0155	0.0218	0.0272	0.0082	0.0110	0.0116
P5-RID	0.0330	0.0511	0.0651	0.0146	0.0200	0.0144	0.0148	0.0225	0.0263	0.0071	0.0086	0.0095
P5-SID	0.0340	0.0516	0.0672	0.0154	0.0231	0.0176	0.0143	0.0222	0.0258	0.0070	0.0086	0.0091
CID	0.0341	0.0516	0.0673	0.0165	0.0236	0.0177	0.0146	0.0226	0.0276	0.0070	0.0087	0.0092
POD	0.0339	0.0498	0.0639	0.0185	0.0222	0.0221	0.0147	0.0225	0.0261	0.0074	0.0087	0.0091
CoLLM	0.0391	0.0606	0.0772	0.0200	0.0259	0.0303	0.0150	0.0218	0.0274	0.0079	0.0091	0.0117
* (User ID Only)	0.0396	0.0599	0.0763	0.0214	0.0265	0.0300	0.0160	0.0228	0.0282	0.0092	0.0109	0.0119
* (Unseen Prompt)	0.0402	**0.0622**	**0.0791**	0.0215	0.0270	**0.0306**	0.0164	0.0233	0.0286	0.0096	0.0111	0.0124
TokenRec	**0.0407**	0.0615	0.0782	**0.0222**	**0.0276**	0.0303	**0.0171**	**0.0240**	**0.0291**	**0.0108**	**0.0112**	**0.0130**

* are the variants of **TokenRec**, namely the cases of using user ID tokens only for model inputs without considering item interaction history and using the unseen prompt during evaluation.

LLM-empowered methods are **empirically superior** to conventional RecSys.

Evaluation: Generalizability, Efficiency, and Ablation Studies

TABLE IV: Performance comparison on seen and unseen users for generalizability evaluation.

Dataset	Model	Seen HR@20	Seen NG@20	Unseen HR@20	Unseen NG@20
LastFM	P5	0.0704	0.0320	0.0399	0.0137
	POD	0.0709	0.0323	0.0401	0.0138
	CID	0.0697	0.0314	0.0452	0.0196
	CoLLM	0.0812	0.0336	0.0574	0.0235
	TokenRec	**0.0973**	**0.0353**	**0.0773**	**0.0268**
Beauty	P5	0.0511	0.0236	0.0274	0.0130
	POD	0.0507	0.0225	0.0269	0.0123
	CID	0.5234	0.0240	0.3336	0.0146
	CoLLM	0.0612	0.0261	0.0477	0.0195
	TokenRec	**0.0629**	**0.0289**	**0.0591**	**0.0266**

TokenRec outperforms existing LLM-based methods in generalizability, thanks to the masking and K-way operations and the proposed generative retrieval framework.

Evaluation: Generalizability, Efficiency, and Ablation Studies

TABLE V: Average inference time (milliseconds) per user.

Inference Time	LastFM	ML1M	Beauty	Clothing
P5	96.04	99.75	86.39	93.38
POD	96.30	101.42	87.69	94.48
CID	94.96	99.42	84.87	92.02
TokenRec	6.92	8.43	5.76	6.00
Acceleration*	1284%	1089%	1398%	1455%

* The average improvement compared to the baselines.

TABLE VI: Results of Ablation Studies.

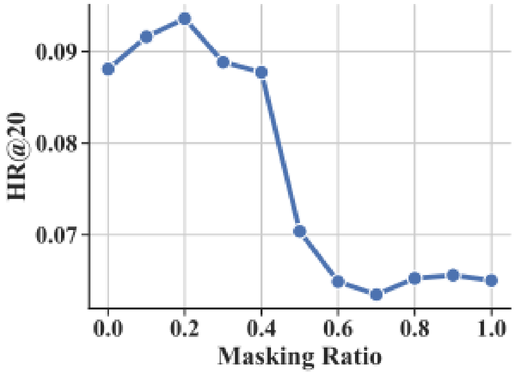
Module	LastFM		Beauty	
	HR@20	NG@20	HR@20	NG@20
Full*	0.0936	0.0348	0.0615	0.0276
w/o Masking	0.0848	0.0332	0.0573	0.0253
w/o K-way	0.0820	0.0309	0.0592	0.0250
w/o HOCK	0.0549	0.0172	0.0407	0.0149
s RQ-VAE	0.0831	0.0314	0.0596	0.0253
s VQ-VAE	0.0810	0.0308	0.0589	0.0247
s K-Means	0.0750	0.0281	0.0567	0.0237

* ”Full” denotes the complete version of TokenRec.
”s” denotes the substitution made to the MQ-Tokenizers.

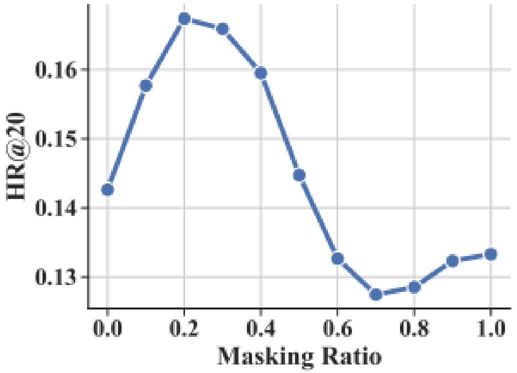
TokenRec are more efficient in the inference process compared to the representative LLM-based methods, because it bypasses the time-consuming auto-regressive generation and beam search processes of LLMs.

- **All the proposed components** contribute to the overall performance.
- The comparison with the three representative quantization/clustering methods illustrates **the effectiveness of our MQ-Tokenizers in encoding collaborative knowledge** for LLM-based recommendations.

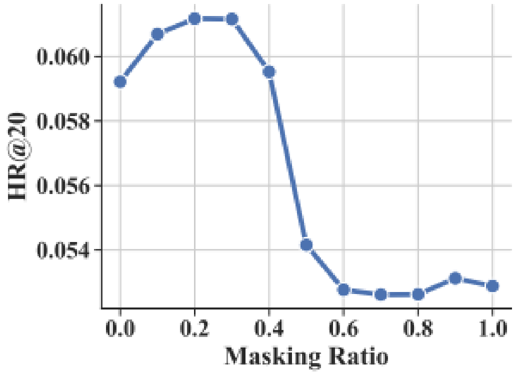
Evaluation: Hyper-Parameter Analysis



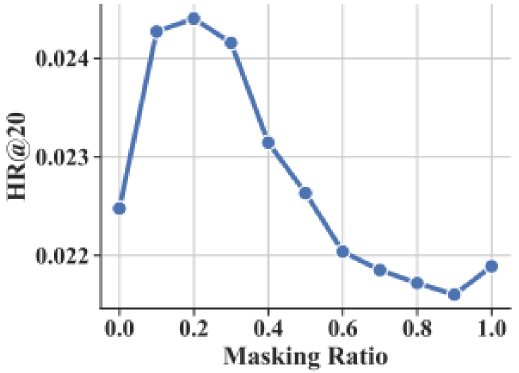
(a) LastFM - HR@20



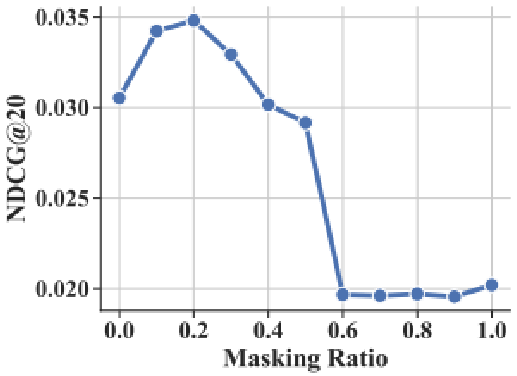
(b) ML1M - HR@20



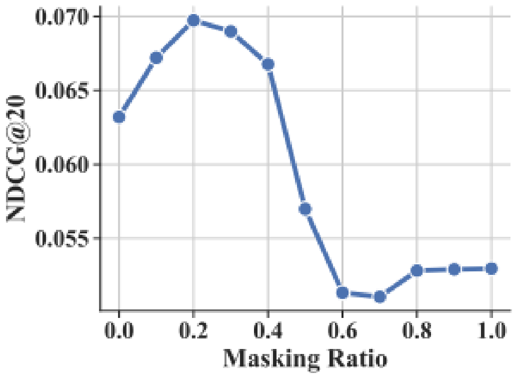
(c) Amazon-Beauty - HR@20



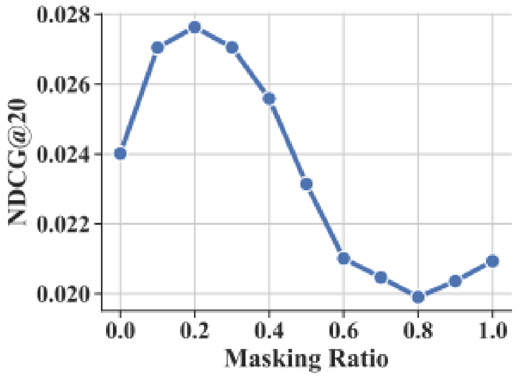
(d) Amazon-Clothing - HR@20



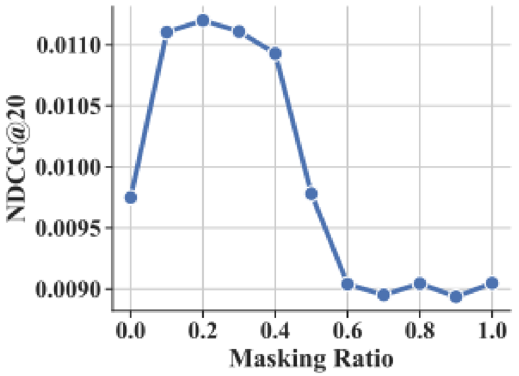
(e) LastFM - NDCG@20



(f) ML1M - NDCG@20



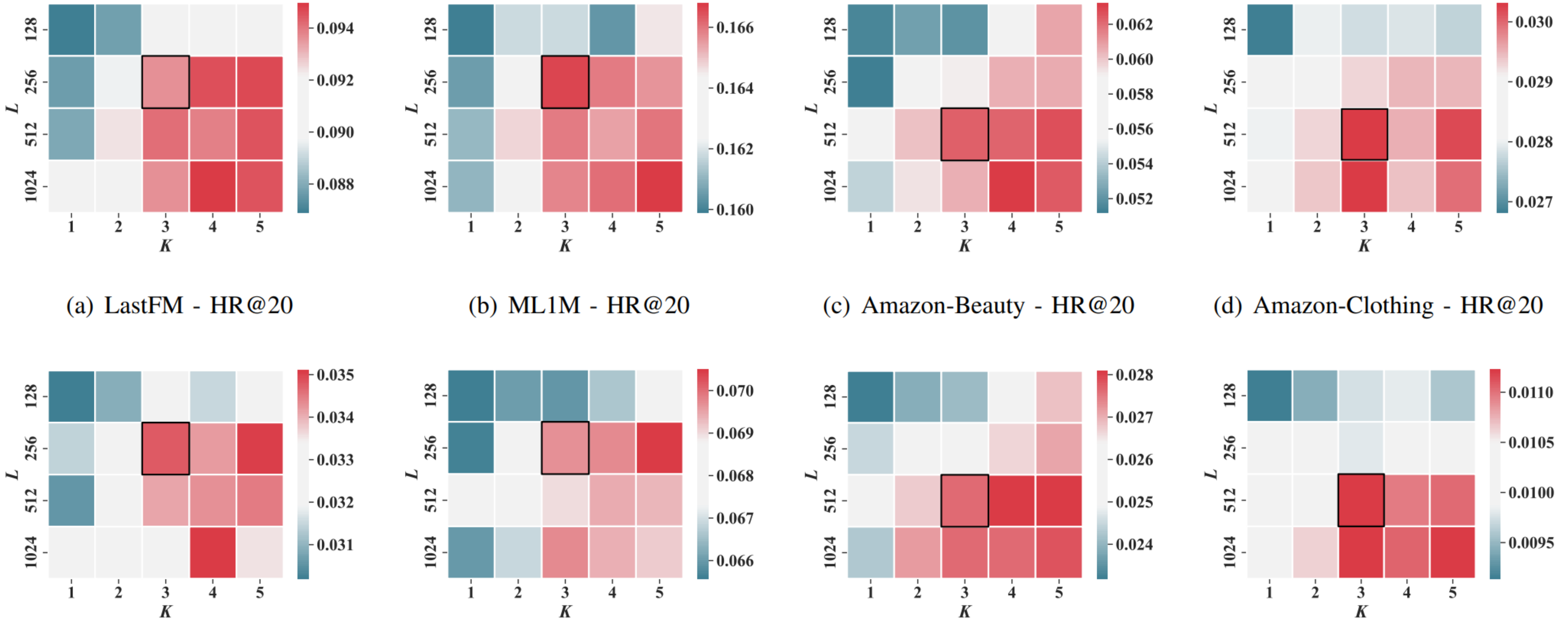
(g) Amazon-Beauty - NDCG@20



(h) Amazon-Clothing - NDCG@20

The suggested Masking Ratio is **0.2.**

Evaluation: Hyper-Parameter Analysis



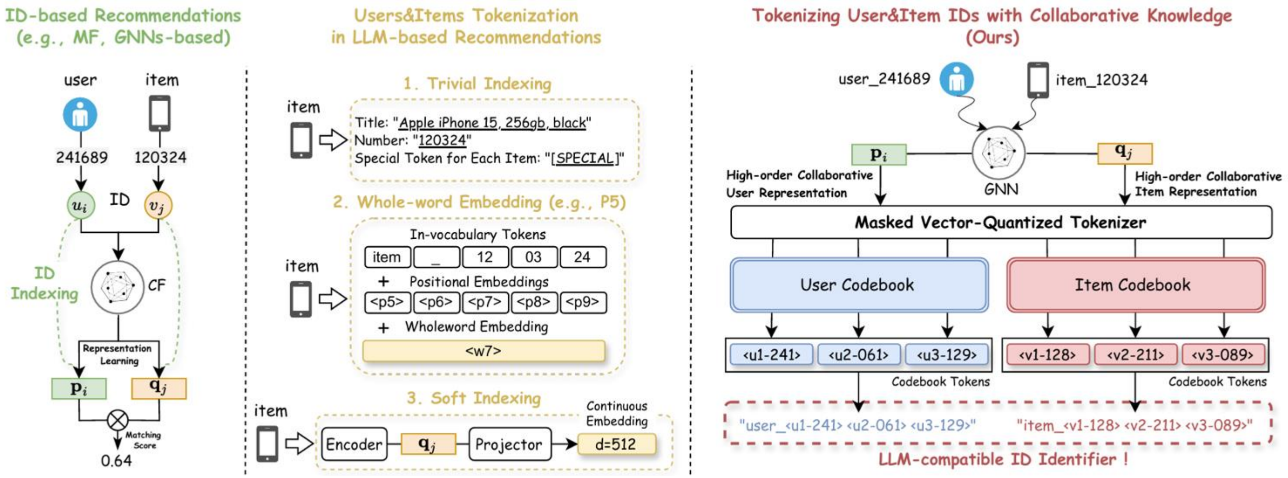
(a) LastFM - HR@20 (b) ML1M - HR@20 (c) Amazon-Beauty - HR@20 (d) Amazon-Clothing - HR@20

(e) LastFM - NDCG@20 (f) ML1M - NDCG@20 (g) Amazon-Beauty - NDCG@20 (h) Amazon-Clothing - NDCG@20

As the number of users and items grows, the associated values of K and L should increase accordingly.

Conclusion

- We introduce a principle strategy named **Masked Vector-Quantized Tokenizer** to tokenize users and items tailored to LLMs, which contributes to incorporating high-order collaborative knowledge in LLM-based recommendations.
- We propose a novel framework (**TokenRec**) for recommender systems in the era of LLMs, where a **Generative Retrieval** paradigm is designed to effectively and efficiently recommend top-K items for users rather than directly generating tokens in natural language.
- We conduct **extensive experiments** on four widely used real-world datasets to empirically demonstrate the effectiveness of our proposed TokenRec, including the superior recommendation performance and its generalization ability in predicting new and unseen users' preferences.



Haohao Qu, Wenqi Fan, **Zihuai Zhao**, Qing Li
TokenRec: Learning to Tokenize ID for LLM-based Generative Recommendation. arXiv:2406.10450, 2024.

THANK YOU

Email: wenqi.fan@polyu.edu.hk
Homepage: https://wenqifan03.github.io