ACL 2025 VIENNA
JULY 27 - AUGUST 1

GRAM: Generative Recommendation via Semantic-aware Multi-granular Late Fusion

Sunkyung Lee[1], Minjin Choi[2], Eunseong Choi[1], Hye-young Kim[1], Jongwuk Lee[1]

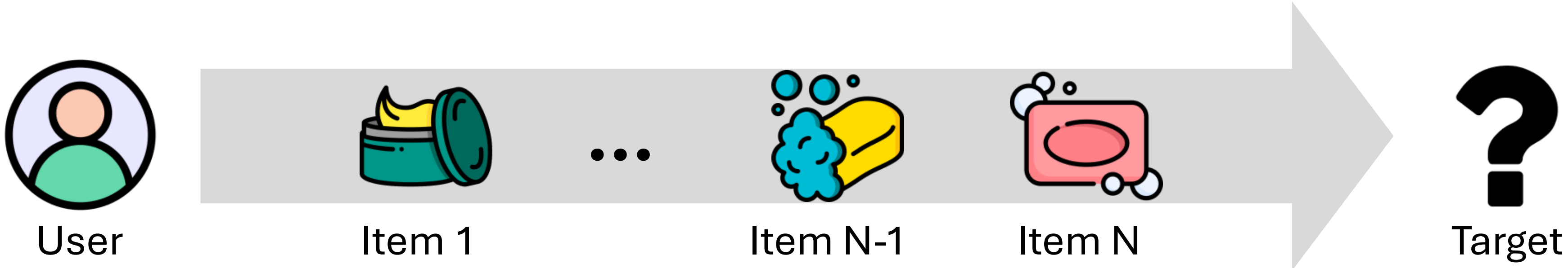
Sungkyunkwan University (SKKU), Republic of Korea[1],
Samsung Research, Republic of Korea[2]

Introduction

- Sequential Recommendation
- Generative Recommendation
- Limitations of Existing Methods
- Research Question
- Challenges
- Our Solution

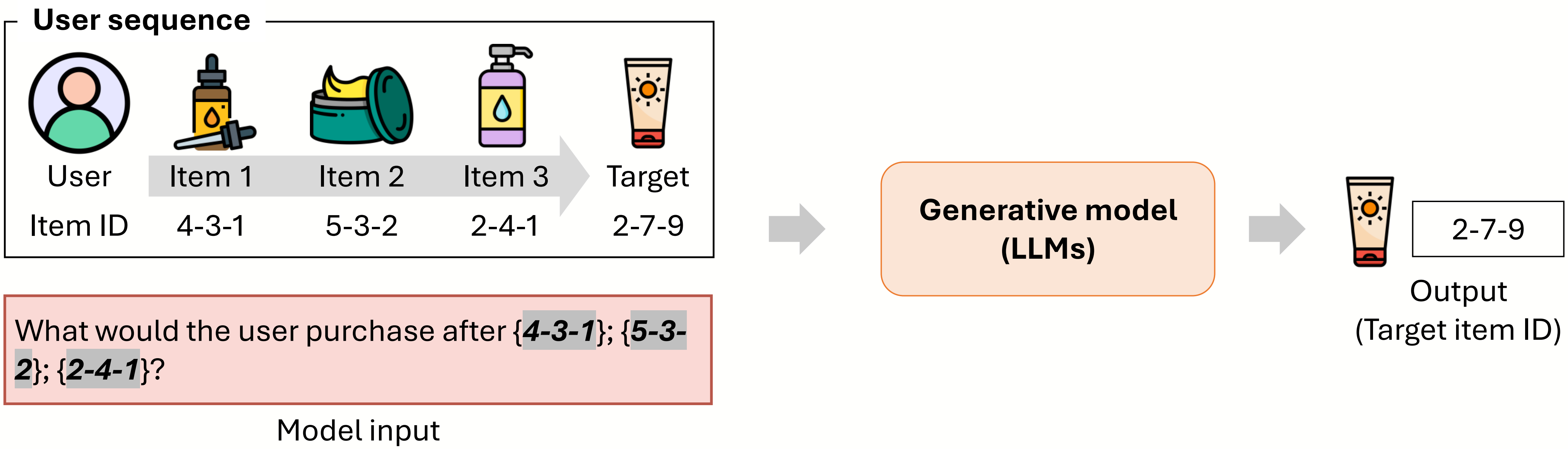
Task: Sequential Recommendation

- **Sequential recommendation predicts next actions from user behaviors over time.**
  - It captures user preferences through sequential interaction history.



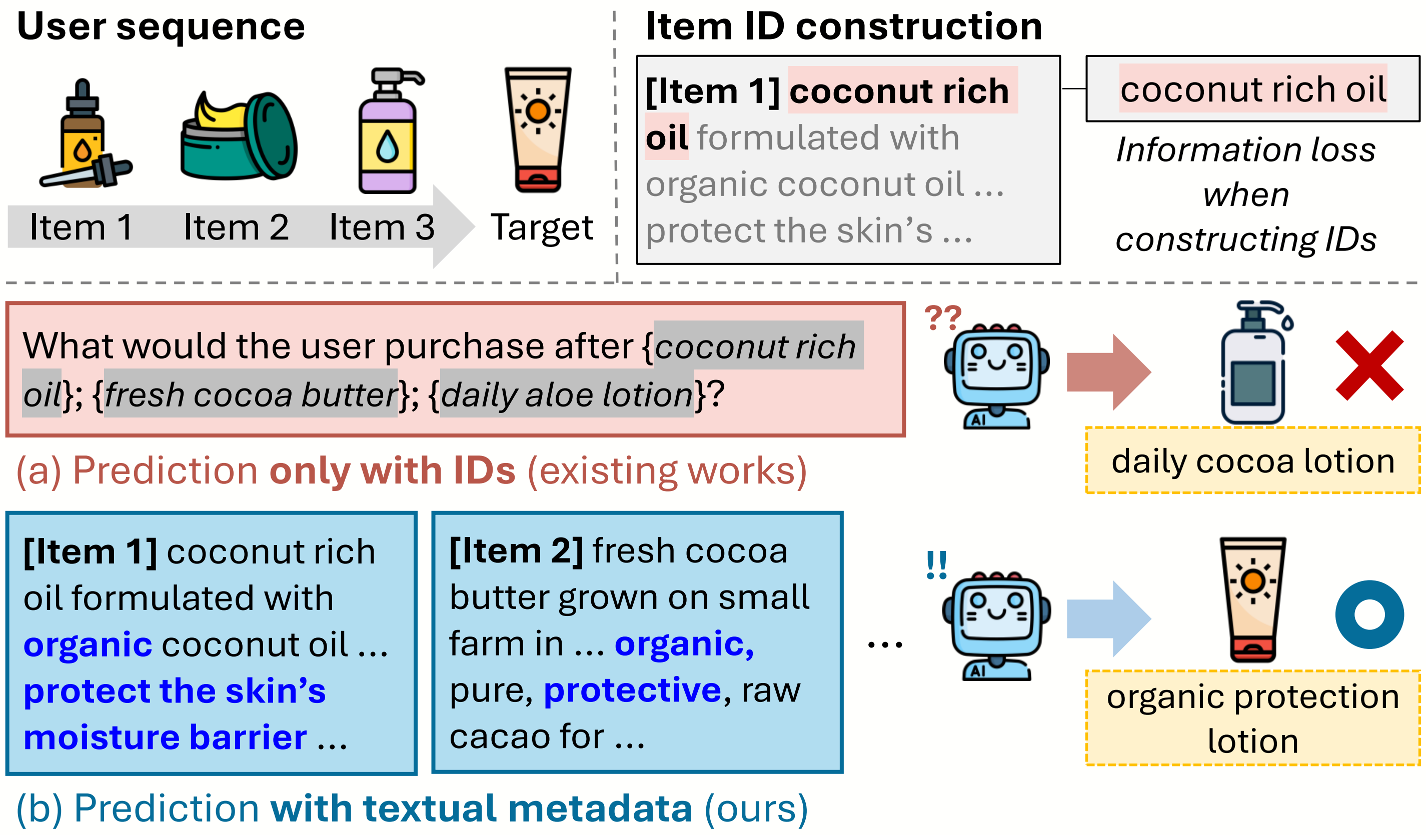
Generative Recommendation

- It aims to **directly generate a target item ID** based on the user's history.
- Typically, users are represented by **concatenating item IDs** into a sequence.



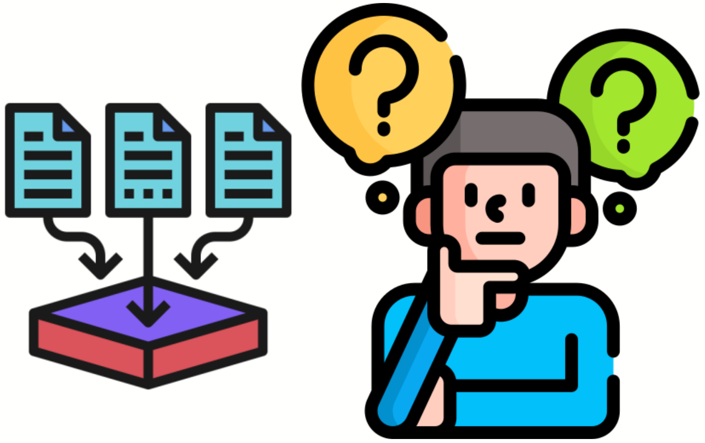
Limitations of Existing Methods

- **Existing works primarily use rich item metadata only for constructing short item IDs.**
  - It leads to a potential loss of valuable details of items during prediction.
- **It motivates us to utilize item information throughout the entire recommendation process.**



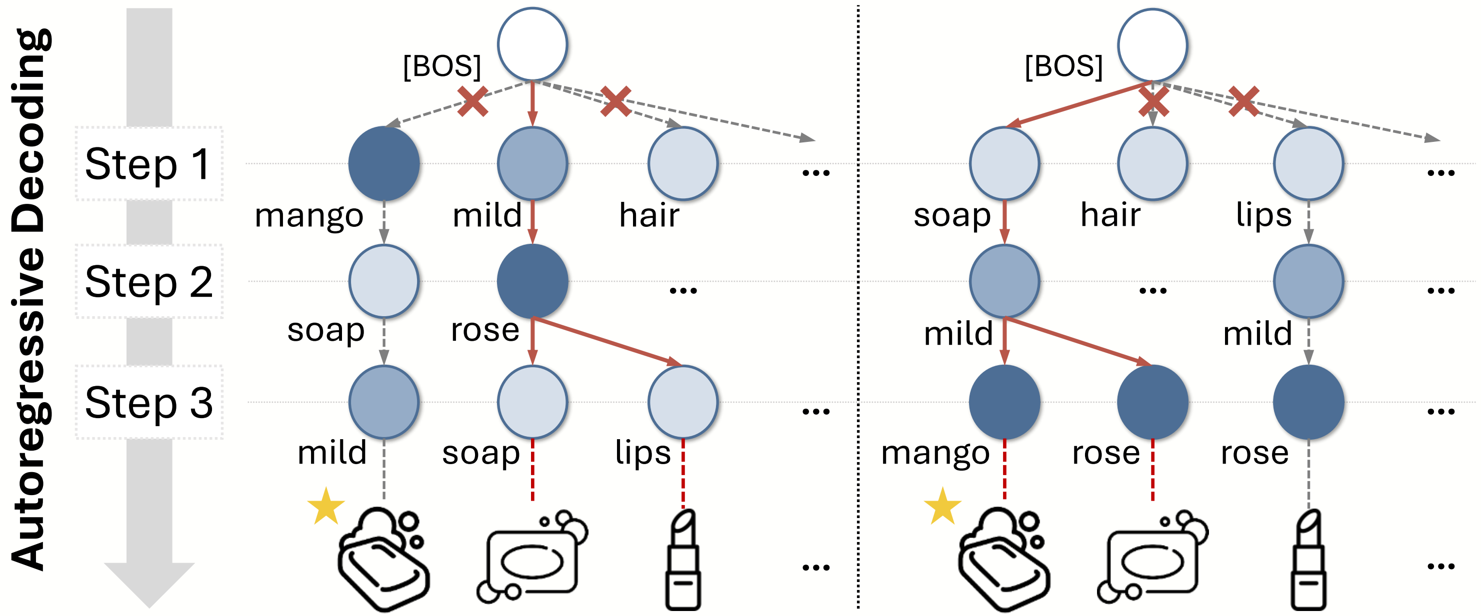
Research Question

How can LLMs effectively understand and utilize rich item information for recommendation?



Challenge 1: Capturing Item Relationships

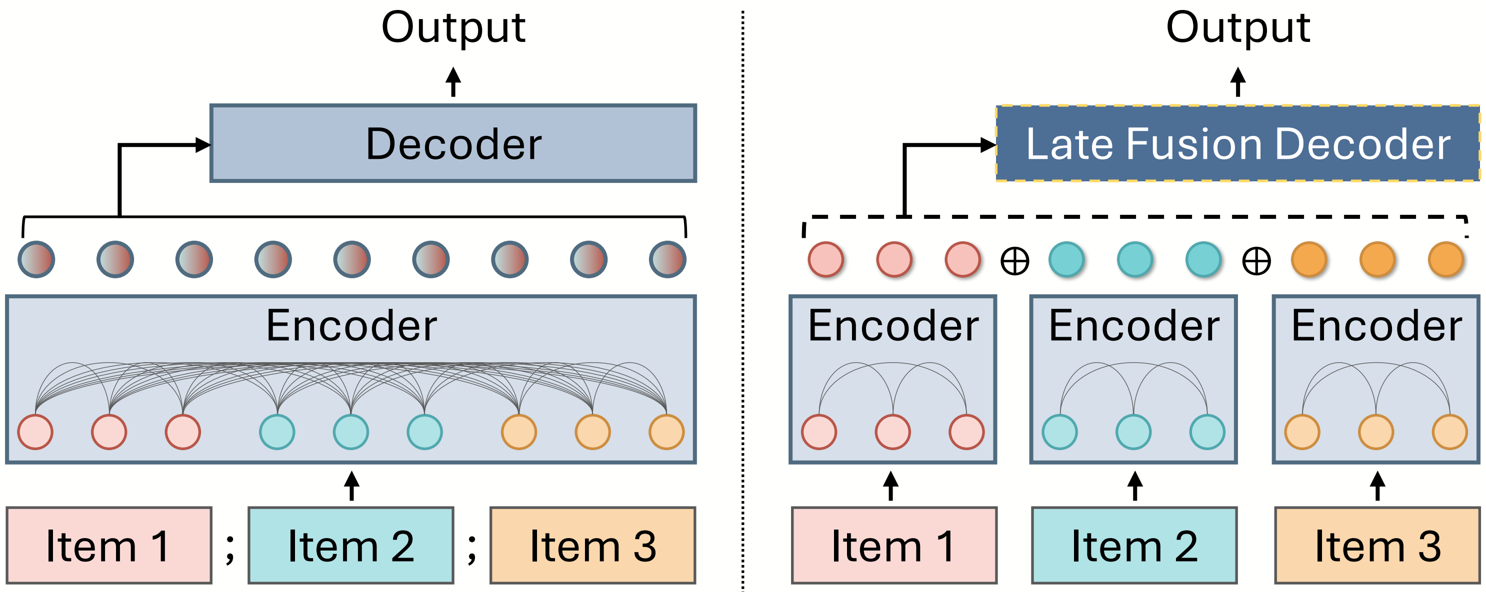
- **LLMs often struggle with recommendation-specific semantics.**
- **The implicit relationships with items breaks down into two problems:**
  - **Hierarchical relationship**: "lipstick" and "mascara" both belong to "cosmetics"
  - **Collaborative relationship:** users who bought item A also tend to buy item B



(a) Item IDs w/o hierarchy (b) Item IDs w/ hierarchy

Challenge 2: Handling Rich Item Information

- Items contain **rich yet lengthy metadata** (titles, categories, descriptions)
- Transformer's **quadratic complexity** leads to computational bottleneck.
- Using partial attributes or extracting keywords inevitably causes **information loss**.



(a) Early fusion (existing works)

(b) Late fusion (ours)

Our Solution

- **We propose Generative Recommender via Semantic-Aware Multi-granular Late Fusion (GRAM), which unlocks the capabilities of LLMs for recommendation.**

- **Innovation 1: Semantic-to-lexical translation**
  - Addresses Challenge 1: Encodes hierarchical & collaborative relationships into LLM's vocabulary

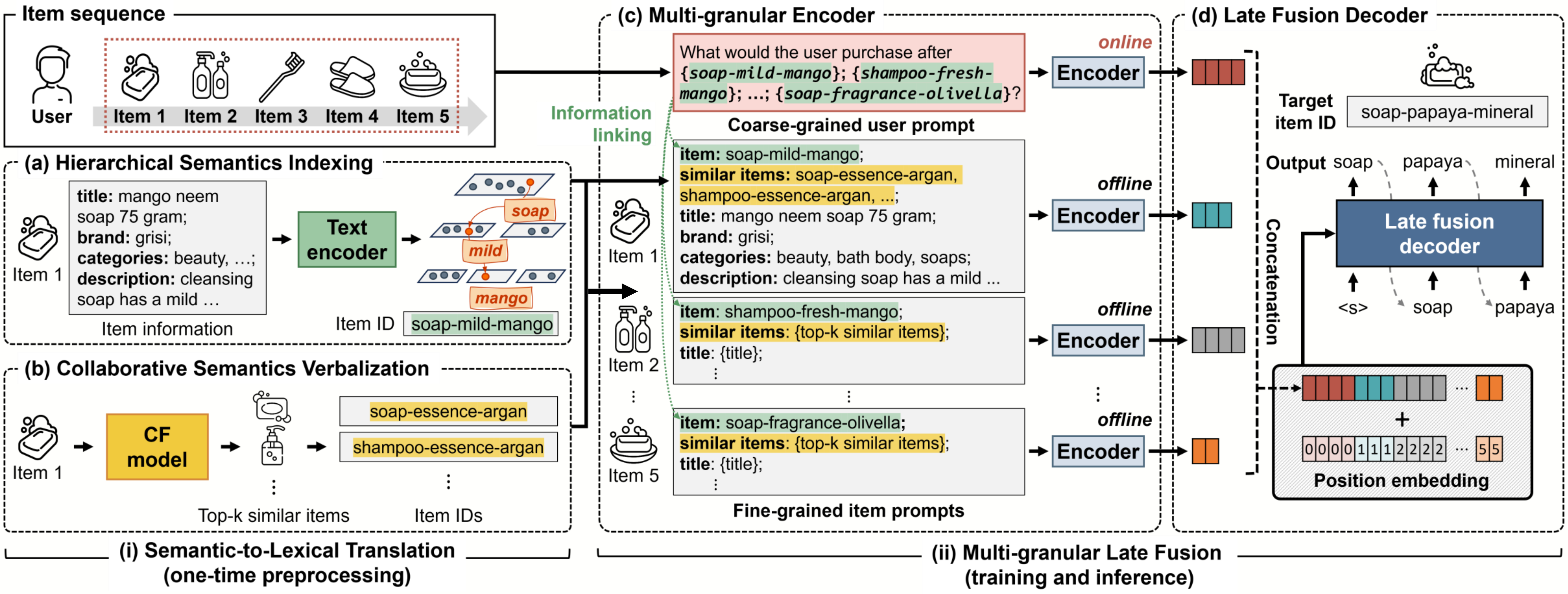
- **Innovation 2: Multi-granular late fusion**
  - Addresses Challenge 2: Processes rich item information efficiently with minimal information loss

Proposed Method

- Overview of GRAM
- Semantic-to-lexical Translation
- Multi-granular Late Fusion

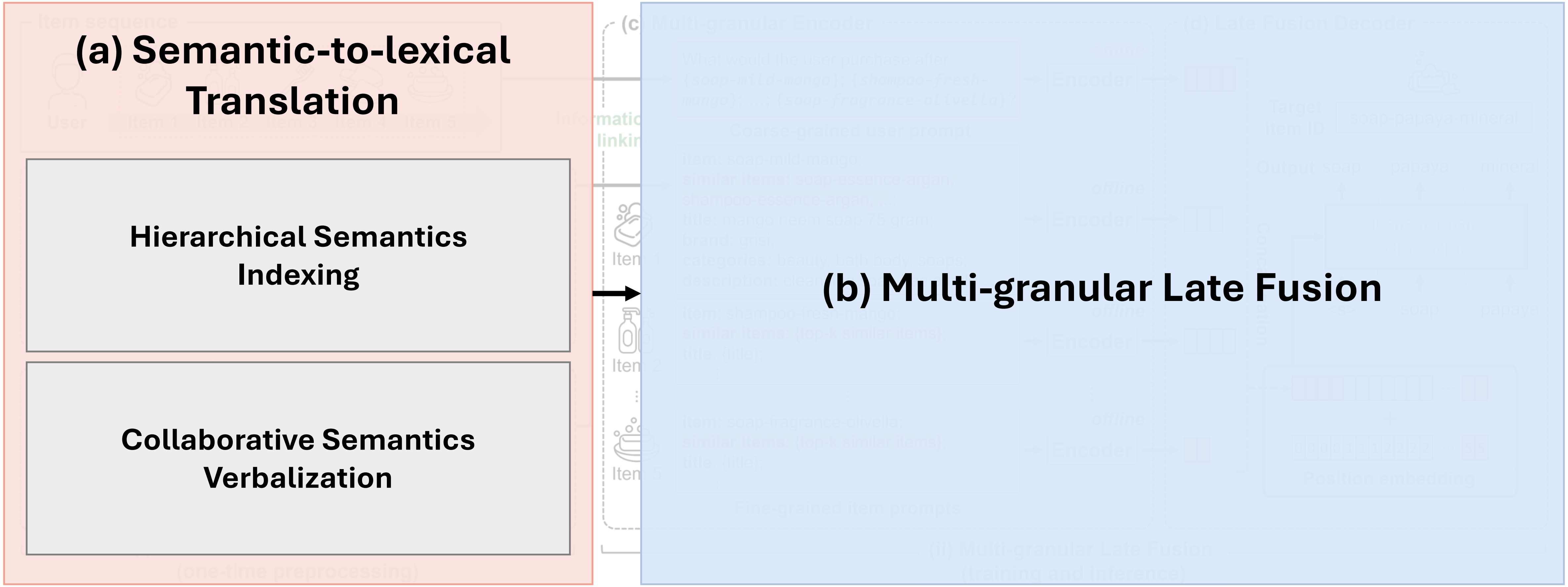
Overview of GRAM

- **GRAM (a) translates item relationships into textual forms and (b) processes user/item prompts separately via late fusion.**



Overview of GRAM

- **GRAM (a) translates item relationships into textual forms and (b) processes user/item prompts separately via late fusion.**



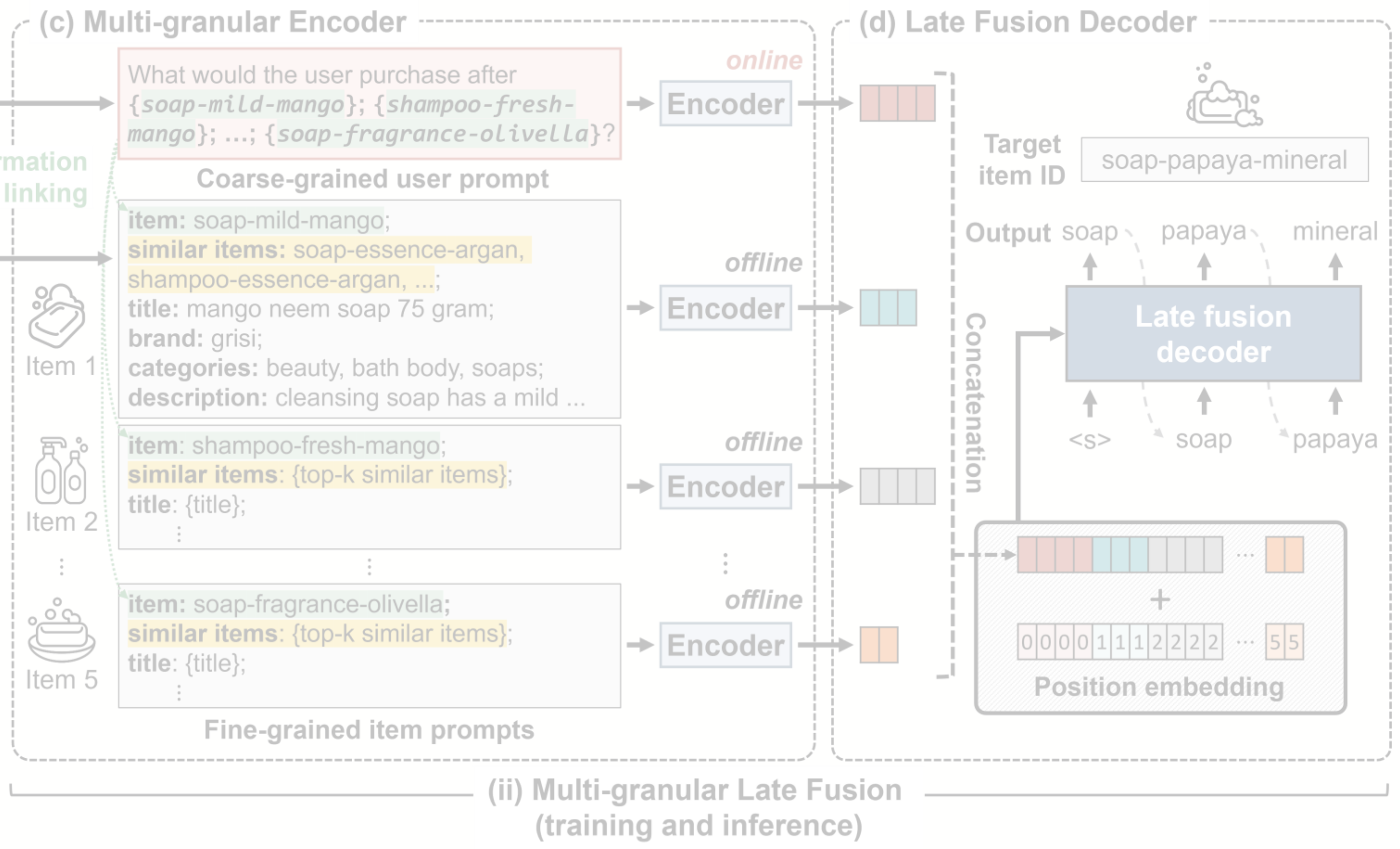
Semantic-to-Lexical Translation

- **Explicitly translates implicit relationships across items into LLM's vocabulary**

(a) Semantic-to-lexical Translation

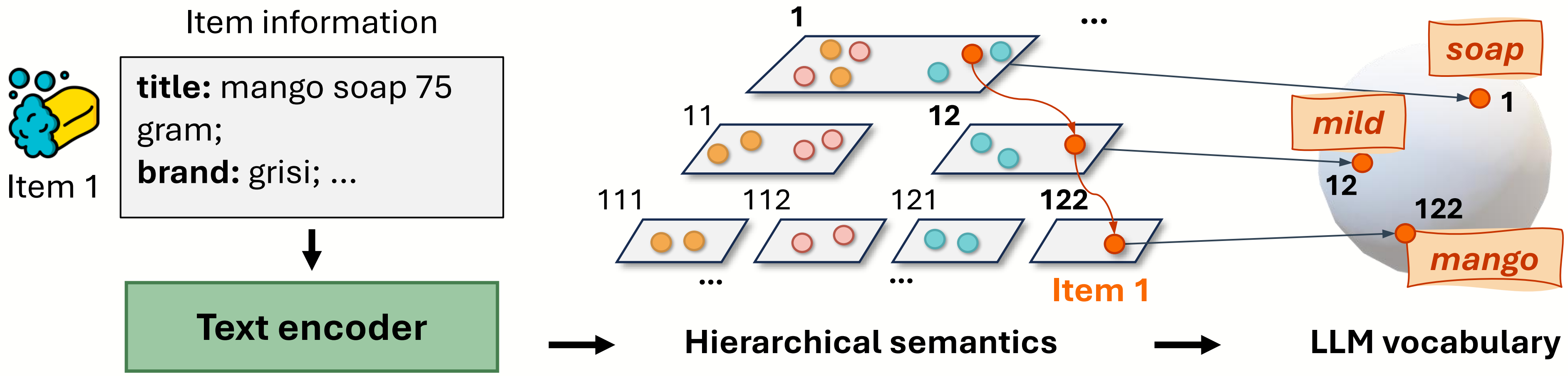
Hierarchical Semantics Indexing

Collaborative Semantics Verbalization



Semantic-to-Lexical Translation

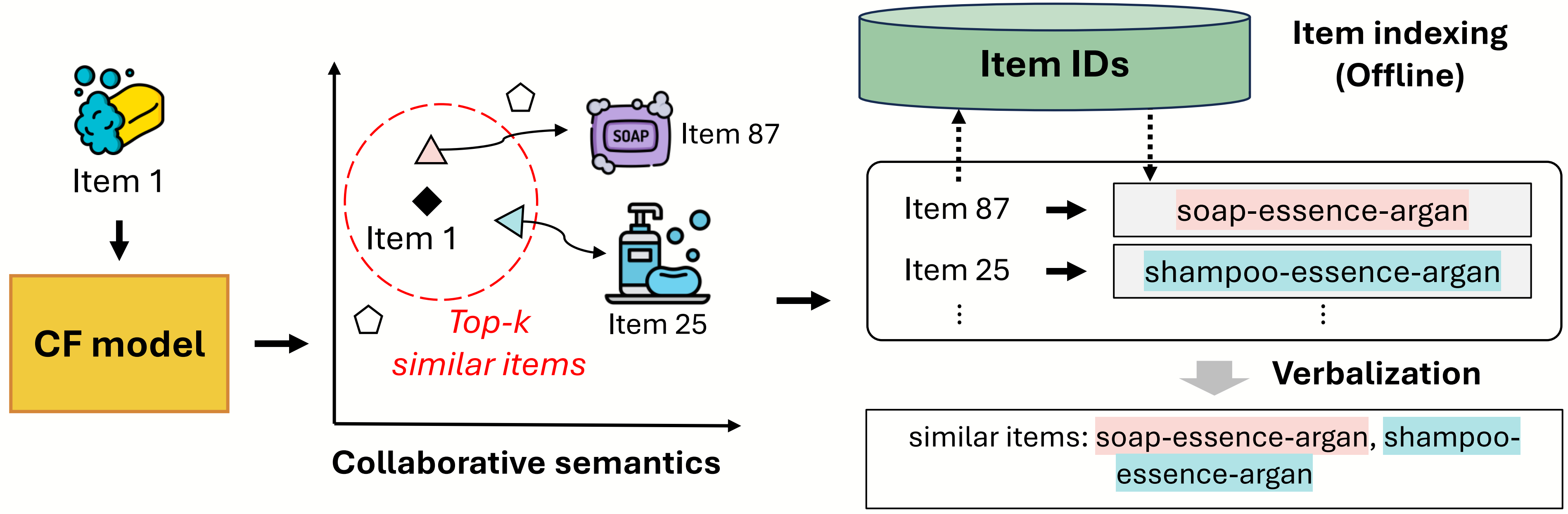
- **Hierarchical Semantics Indexing**
  - Goal: Transform item hierarchy into textual IDs where semantically similar items share identifier prefixes
  - Steps:
    ① Hierarchical clustering of item embeddings
    ② Map clusters to LLM vocabulary tokens
    ③ Create hierarchical IDs with shared prefixes



* We use NV-Embed for the text encoder.

Semantic-to-Lexical Translation

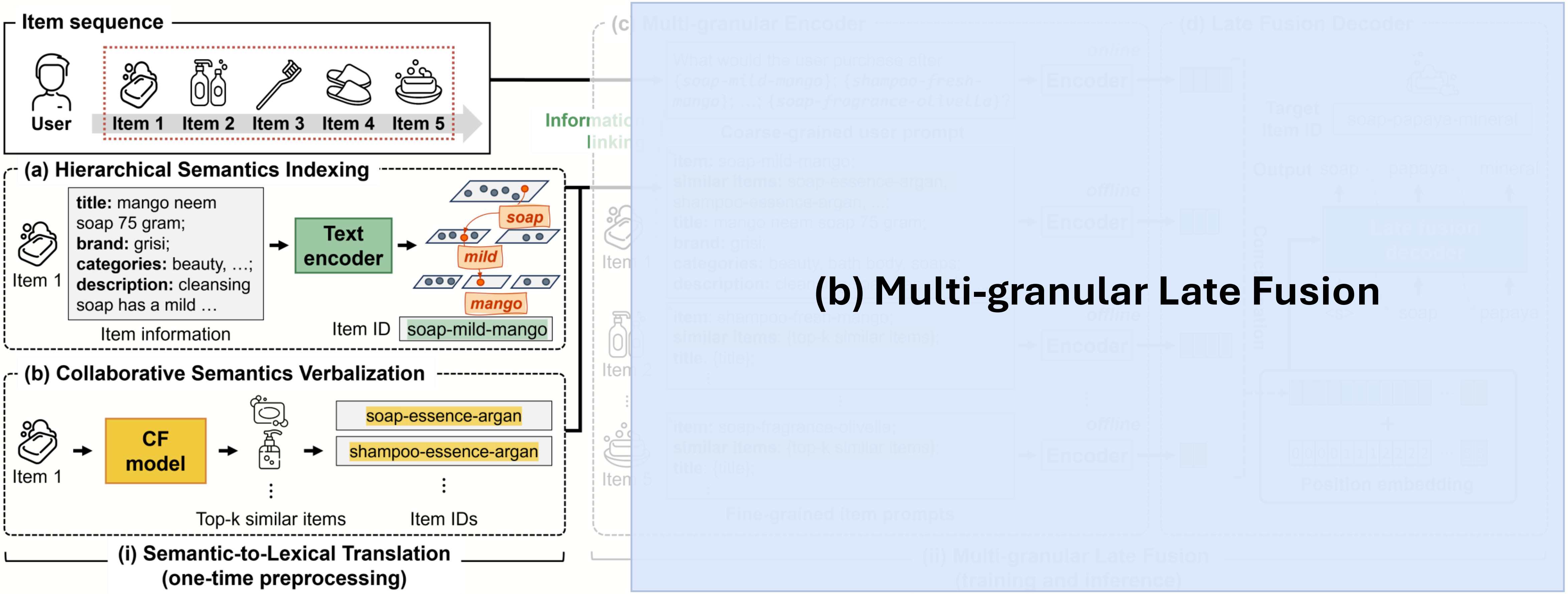
- **Collaborative Semantics Verbalization**
  - Goal: Enable LLMs to leverage collaborative patterns through text
  - Steps:
    - ① Extract collaborative signals using CF model
    - ② Identify top-k similar items for each item
    - ③ Express similar items as text using hierarchical IDs



* We use SASRec for the CF model.

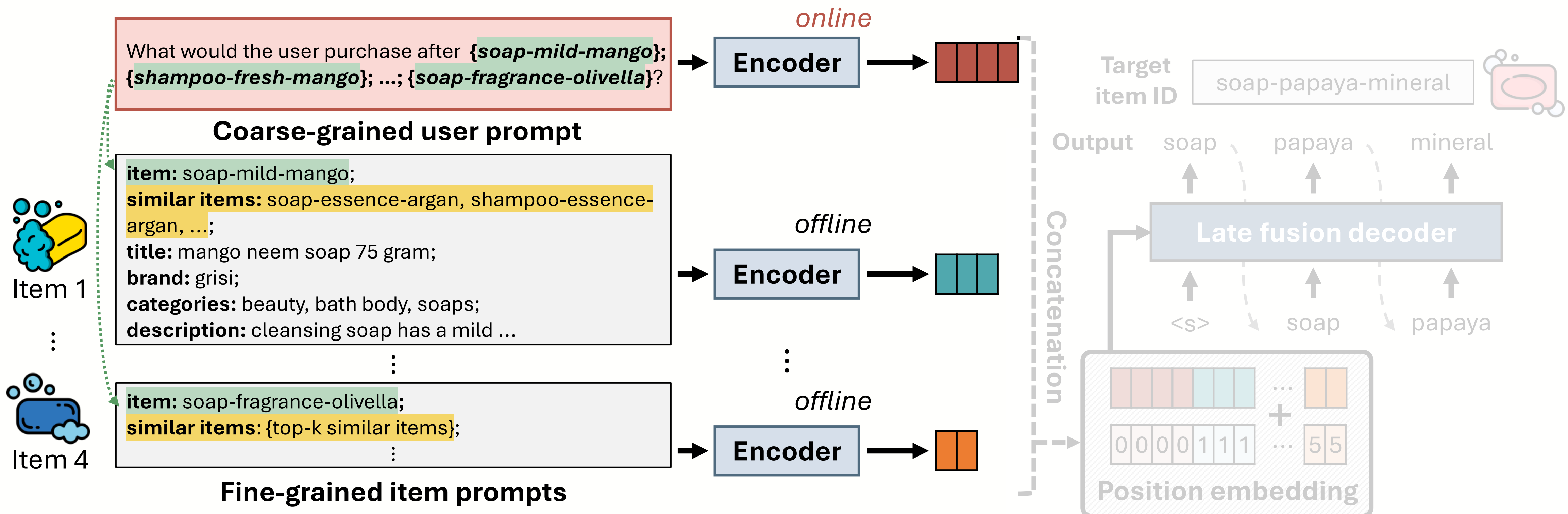
Multi-granular Late Fusion

- **Processes rich item information efficiently with minimal information loss**



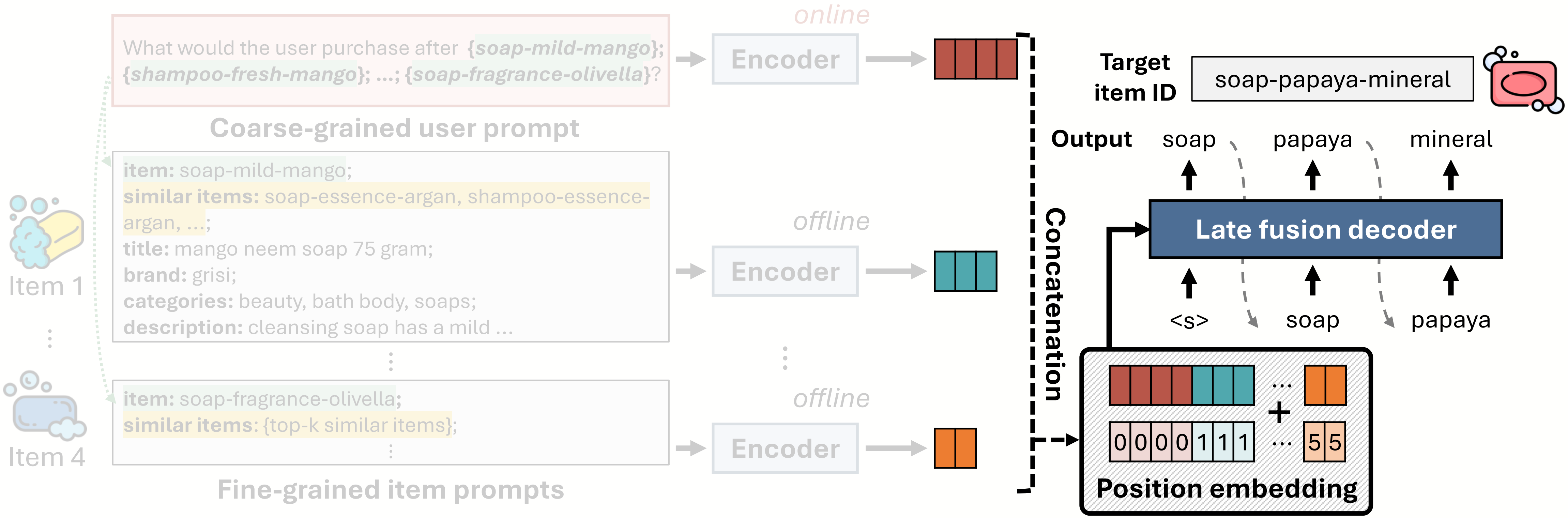
Multi-granular Late Fusion

- **Multi-granular Encoder:**
  - **Coarse-grained user prompt** captures overall user preferences.
  - **Fine-grained item prompt** represents detailed item attributes.
  - The prompts are encoded separately to avoid quadratic complexity.



Multi-granular Late Fusion

- **Late Fusion Decoder**
  - Integrates representations at decoding stage
  - Uses cross-attention to aggregate rich textual information
  - Generates target item ID considering both granularities



Experiments

- Main Results
- Ablation Study

Main Results

- **GRAM achieves state-of-the-art performance over existing methods in benchmark datasets.**

	Model	Beauty				Toys				Sports			
		R@5	N@5	R@10	N@10	R@5	N@5	R@10	N@10	R@5	N@5	R@10	N@10
Traditional	GRU4Rec	0.0429	0.0288	0.0643	0.0357	0.0371	0.0254	0.0549	0.0311	0.0237	0.0154	0.0373	0.0197
	HGN	0.0350	0.0217	0.0589	0.0294	0.0345	0.0212	0.0553	0.0279	0.0203	0.0127	0.0340	0.0171
	SASRec	0.0323	0.0200	0.0475	0.0249	0.0339	0.0208	0.0442	0.0241	0.0147	0.0089	0.0220	0.0113
	BERT4Rec	0.0267	0.0165	0.0450	0.0224	0.0210	0.0131	0.0355	0.0178	0.0136	0.0085	0.0233	0.0116
	FDSA	0.0570	0.0412	0.0777	0.0478	0.0619	0.0455	0.0805	0.0514	0.0283	0.0201	0.0399	0.0238
	S^3Rec	0.0377	0.0235	0.0627	0.0315	0.0365	0.0231	0.0592	0.0304	0.0229	0.0145	0.0370	0.0190
Generative	P5-SID	0.0465	0.0329	0.0638	0.0384	0.0216	0.0151	0.0325	0.0186	0.0295	0.0212	0.0403	0.0247
	P5-CID	0.0465	0.0325	0.0668	0.0391	0.0223	0.0143	0.0357	0.0186	0.0295	0.0214	0.0420	0.0254
	P5-SemID	0.0459	0.0327	0.0667	0.0394	0.0264	0.0178	0.0416	0.0270	0.0336	0.0243	0.0481	0.0290
	TIGER	0.0352	0.0236	0.0533	0.0294	0.0274	0.0174	0.0438	0.0227	0.0176	0.0143	0.0311	0.0146
	IDGenRec	0.0463	0.0328	0.0665	0.0393	0.0462	0.0323	0.0651	0.0383	0.0273	0.0186	0.0403	0.0228
	LETTER	0.0364	0.0243	0.0560	0.0306	0.0309	0.0296	0.0493	0.0262	0.0209	0.0136	0.0331	0.0176
	ELMRec	0.0372	0.0267	0.0506	0.0310	0.0148	0.0119	0.0193	0.0131	0.0241	0.0181	0.0307	0.0203
	LC-Rec	0.0503	0.0352	0.0715	0.0420	0.0543	0.0385	0.0753	0.0453	0.0259	0.0175	0.0384	0.0216
	GRAM	**0.0641**	**0.0451**	**0.0890**	**0.0531**	**0.0718**	**0.0516**	**0.0987**	**0.0603**	**0.0375**	**0.0256**	**0.0554**	**0.0314**
	Gain (%)	12.4*	9.5*	14.5*	11.0*	16.0*	13.6*	22.7*	17.1*	11.5*	5.3*	15.2*	8.3*

The best model is marked in **bold**, and the second-best model is underlined.
'*' indicates statistical significance ($p < 0.05$) by a paired t-test.
The result of the Yelp dataset is omitted for space limits.

Ablation Study

- **All components of GRAM contribute to performance, with collaborative semantics and item prompts showing the most significant improvements.**

Model	Beauty		Toys	
	R@5	N@5	R@5	N@5
GRAM	**0.0641**	**0.0451**	**0.0718**	**0.0516**
w/o hierarchy	0.0605	0.0438	0.0630	0.0466
w/o CF (a_{CF})	0.0567	0.0396	0.0589	0.0406
w/o user prompt (T_u)	0.0634	0.0443	0.0709	0.0510
w/o item prompt (T_i)	0.0582	0.0404	0.0574	0.0397
w/o linking (a_{ID})	0.0628	0.0441	0.0702	0.0507
w/o position (P)	0.0563	0.0395	0.0665	0.0465

Conclusion

Conclusion

- **We propose a novel generative recommender for leveraging LLMs with rich item semantics.**
  - GRAM: **G**enerative **R**ecommender via semantic-**A**ware **M**ulti-granular late fusion

- **GRAM exploits rich item semantics by:**
  - representing complex item relationships as textual identifiers via semantic-to-lexical translation
  - delaying the integration of multi-granular information until decoding via multi-granular late fusion

- **GRAM achieves the best performance among existing generative recommenders on the Amazon Beauty, Toys, Sports, and Yelp datasets, improving up to 16% in Recall@5.**

Thank you!
Any questions?

Email: sk1027@skku.edu

Code: https://github.com/skleee/GRAM

Paper

Code