

Large Language Models for Recommendation: Progresses and Future Direction

Lecture Tutorial For WWW 2024

Organizers: Jizhi Zhang, Keqin Bao, Yang Zhang,
Wenjie Wang, Fuli Feng, Xiangnan He

- Introduction (Fuli Feng)
- Background: LM & LM4Rec (Fuli Feng)
- Development of LLMs (Keqin Bao)
- Progress of LLM4Rec
 - LLM4Rec (Keqin Bao & Wenjie Wang)
 - QA & Coffee Break
 - Trustworthy LLM4Rec (Jizhi Zhang)
- Open Problems (Yang Zhang)
- Future Direction & Conclusions (Fuli Feng)

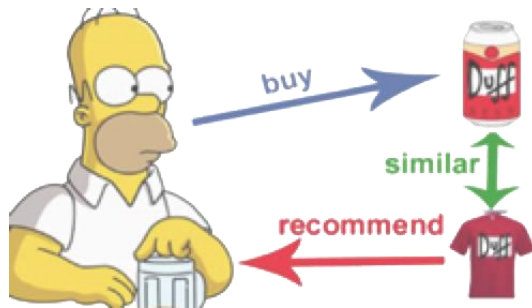
Background of RecSys

□ Information explosion era

- E-commerce: **12 million items** in Amazon.
- Social networks: **2.8 billion users** in Facebook.
- Content sharing platforms: **720,000 hours videos** uploaded to Youtube per day; **35 million videos** posted on **TikTok daily**

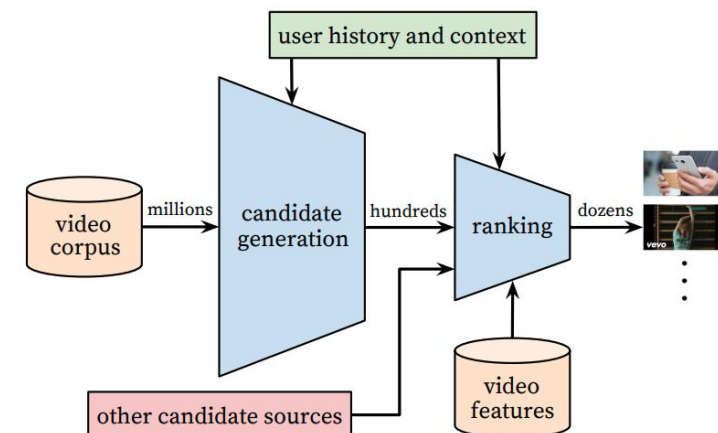


□ Recommender system



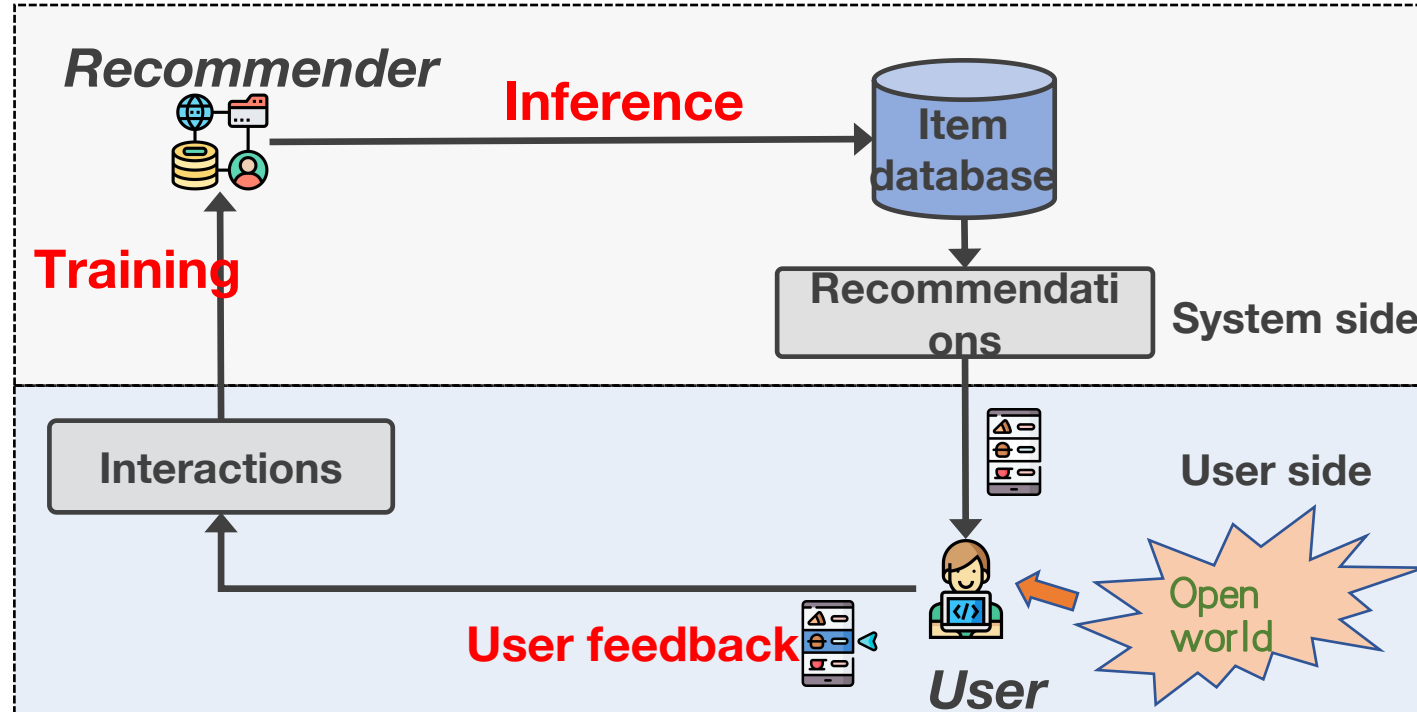
Information seeking
via user history
feedback

Recommendation



Background of RecSys

□ Workflow of Recommender System



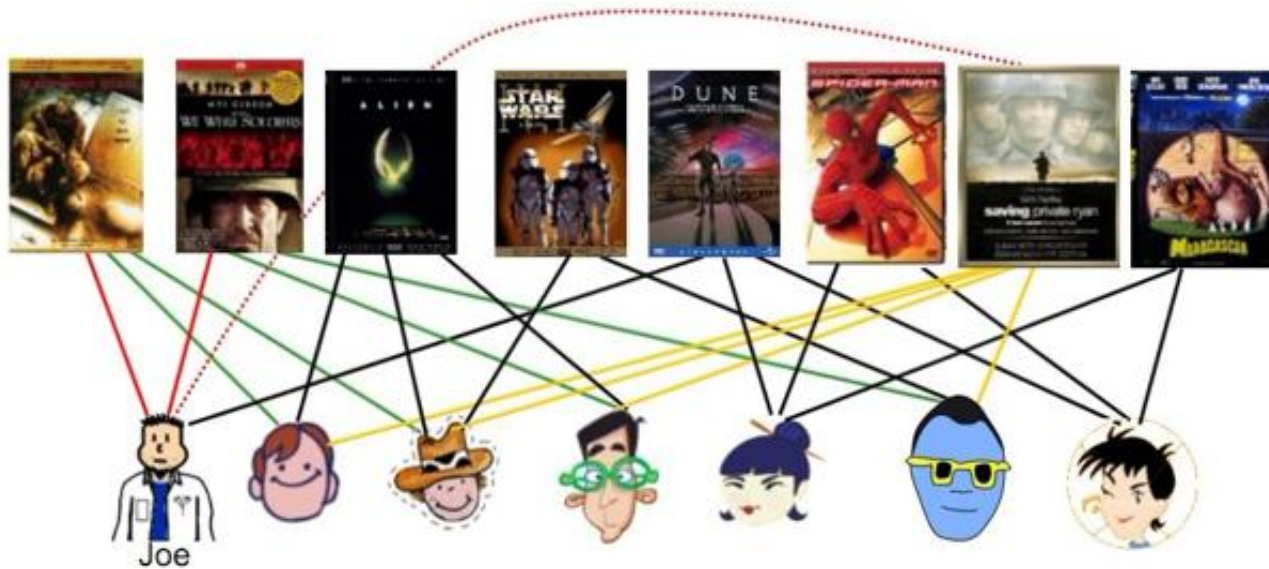
- (1) Train recommender on collected interaction data to capture user preferences.
- (2) Recommender generates recommendations based on estimated preferences.
- (3) User engages with the recommended items, forming new data, affected by open world.
- (4) train recommender with new data again, either refining user interests or capturing new ones.

Background of RecSys

□ Core idea of personalized recommendation

- **Collaborative filtering (CF):**

Making automatic predictions (filtering) about the interests of a user by collecting preferences from many users (collaborating).



| | | item | | | | | |
|--------------------|-----|------|-----|-----|-----|-----|--|
| | | 1 | 2 | 3 | 4 | | |
| user | 1 | 5 | ? | ? | ? | ... | Memory-based CF <ul style="list-style-type: none">• User CF• Item CF |
| | 2 | 3 | 4 | ? | ? | ... | |
| | 3 | ? | 1 | 2 | 4 | ... | |
| | ... | ... | ... | ... | ... | | |
| Interaction Matrix | | | | | | | Model-based CF <ul style="list-style-type: none">• MF• FISM• ... |

Background of RecSys

□ Core idea of personalized recommendation

- **Collaborative filtering (CF):**

Making automatic predictions (filtering) about the interests of a user by collecting preferences from many users (collaborating).

| user | item | | | | |
|------|------|-----|-----|-----|-----|
| | 1 | 2 | 3 | 4 | |
| 1 | 5 | ? | ? | ? | ... |
| 2 | 3 | 4 | ? | ? | ... |
| 3 | ? | 1 | 2 | 4 | ... |
| ... | ... | ... | ... | ... | ... |

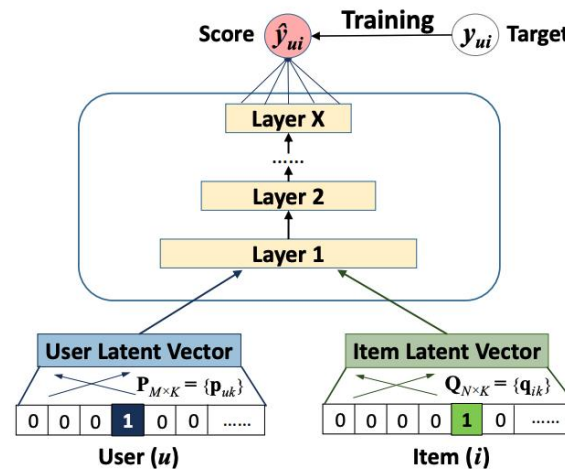
Interaction Matrix

Memory-based CF

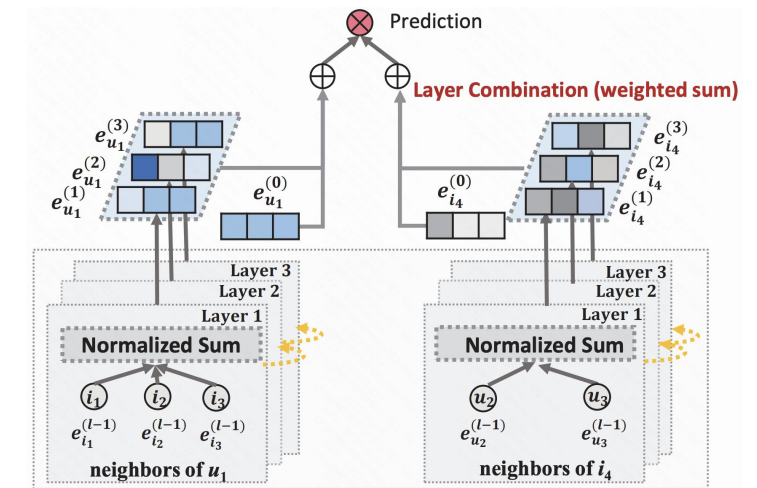
- User CF
- Item CF

Model-based CF

- MF
- FISM
- ...



Neural CF



GCN-based CF

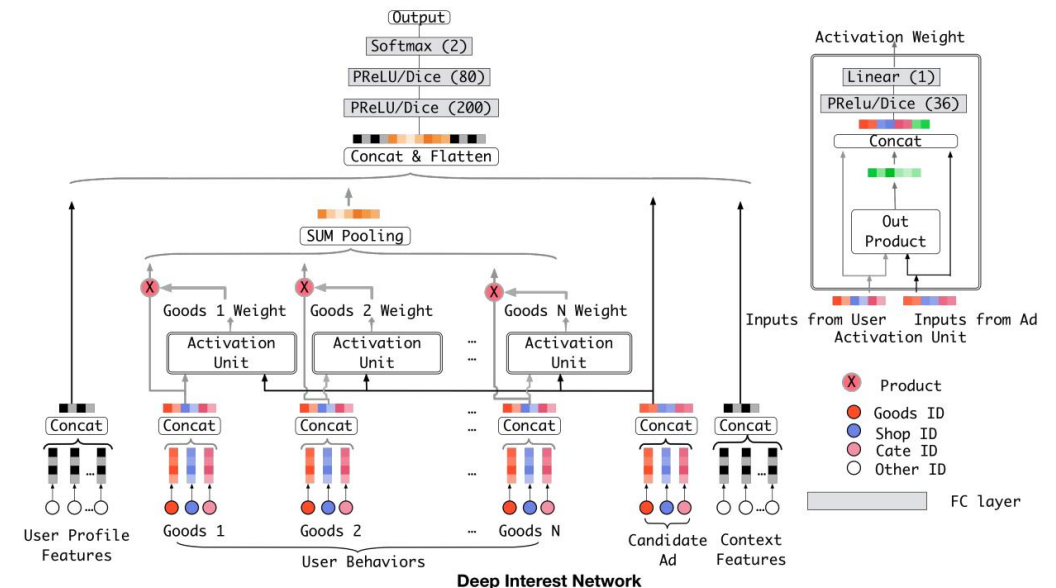
Background of RecSys

□ Core idea of personalized recommendation

- Collaborative filtering (CF): collaborative information
- **Content/context-aware models (CTR models): side information+context information**
 - Click-Through Rate (CTR) prediction

| Feature vector x | | | | | | | | | | | | | | | Target y | | | | | | | |
|--------------------|------|---|---|-----|-------|----|----|----|-----|--------------------|-----|-----|-----|-----|------------|------------------|----|----|----|-----|---|-----------|
| $x^{(1)}$ | 1 | 0 | 0 | ... | 1 | 0 | 0 | 0 | ... | 0.3 | 0.3 | 0.3 | 0 | ... | 13 | 0 | 0 | 0 | 0 | ... | 5 | $y^{(1)}$ |
| $x^{(2)}$ | 1 | 0 | 0 | ... | 0 | 1 | 0 | 0 | ... | 0.3 | 0.3 | 0.3 | 0 | ... | 14 | 1 | 0 | 0 | 0 | ... | 3 | $y^{(2)}$ |
| $x^{(3)}$ | 1 | 0 | 0 | ... | 0 | 0 | 1 | 0 | ... | 0.3 | 0.3 | 0.3 | 0 | ... | 16 | 0 | 1 | 0 | 0 | ... | 1 | $y^{(2)}$ |
| $x^{(4)}$ | 0 | 1 | 0 | ... | 0 | 0 | 1 | 0 | ... | 0 | 0 | 0.5 | 0.5 | ... | 5 | 0 | 0 | 0 | 0 | ... | 4 | $y^{(3)}$ |
| $x^{(5)}$ | 0 | 1 | 0 | ... | 0 | 0 | 0 | 1 | ... | 0 | 0 | 0.5 | 0.5 | ... | 8 | 0 | 0 | 1 | 0 | ... | 5 | $y^{(4)}$ |
| $x^{(6)}$ | 0 | 0 | 1 | ... | 1 | 0 | 0 | 0 | ... | 0.5 | 0 | 0.5 | 0 | ... | 9 | 0 | 0 | 0 | 0 | ... | 1 | $y^{(5)}$ |
| $x^{(7)}$ | 0 | 0 | 1 | ... | 0 | 0 | 1 | 0 | ... | 0.5 | 0 | 0.5 | 0 | ... | 12 | 1 | 0 | 0 | 0 | ... | 5 | $y^{(6)}$ |
| | A | B | C | ... | TI | NH | SW | ST | ... | TI | NH | SW | ST | ... | Time | TI | NH | SW | ST | ... | | |
| | User | | | | Movie | | | | | Other Movies rated | | | | | | Last Movie rated | | | | | | |

Factorization machines: FM, NFM, DeepFM



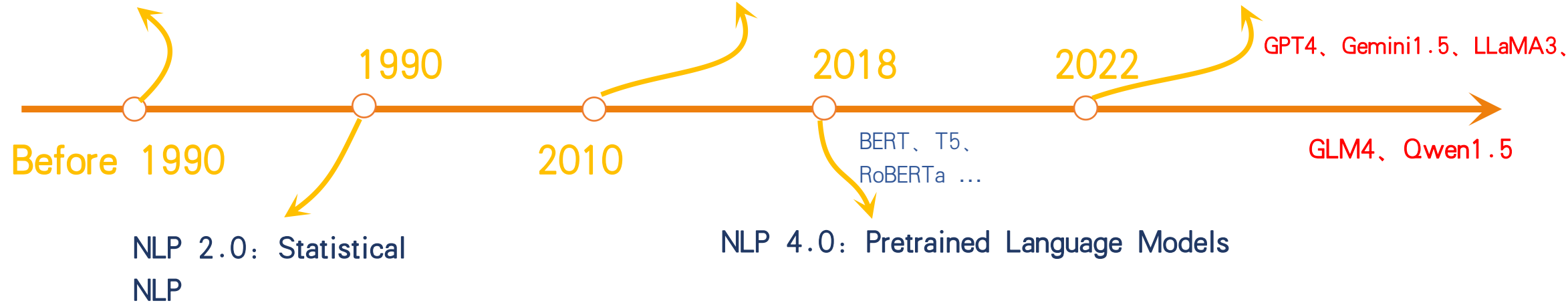
Neural network: DIN, AutoInt

The development of LMs

NLP 1.0: Dictionary/Vocabulary + Rules

NLP 3.0: Deep Learning for NLP

LLMs: ChatGPT



Large Language Model: **billions of parameters, emergent capabilities**

- Rich knowledge & Language Capabilities
- Instruction following
- In-context learning
- Chain-of-thought
- Planning
- ...

The development of LMs

- LLMs such as ChatGPT and GPT4 have influenced many fields in CS and IT industry
- They have eliminated a wide range of research in basic NLP and conversational system, etc.



ChatGPT



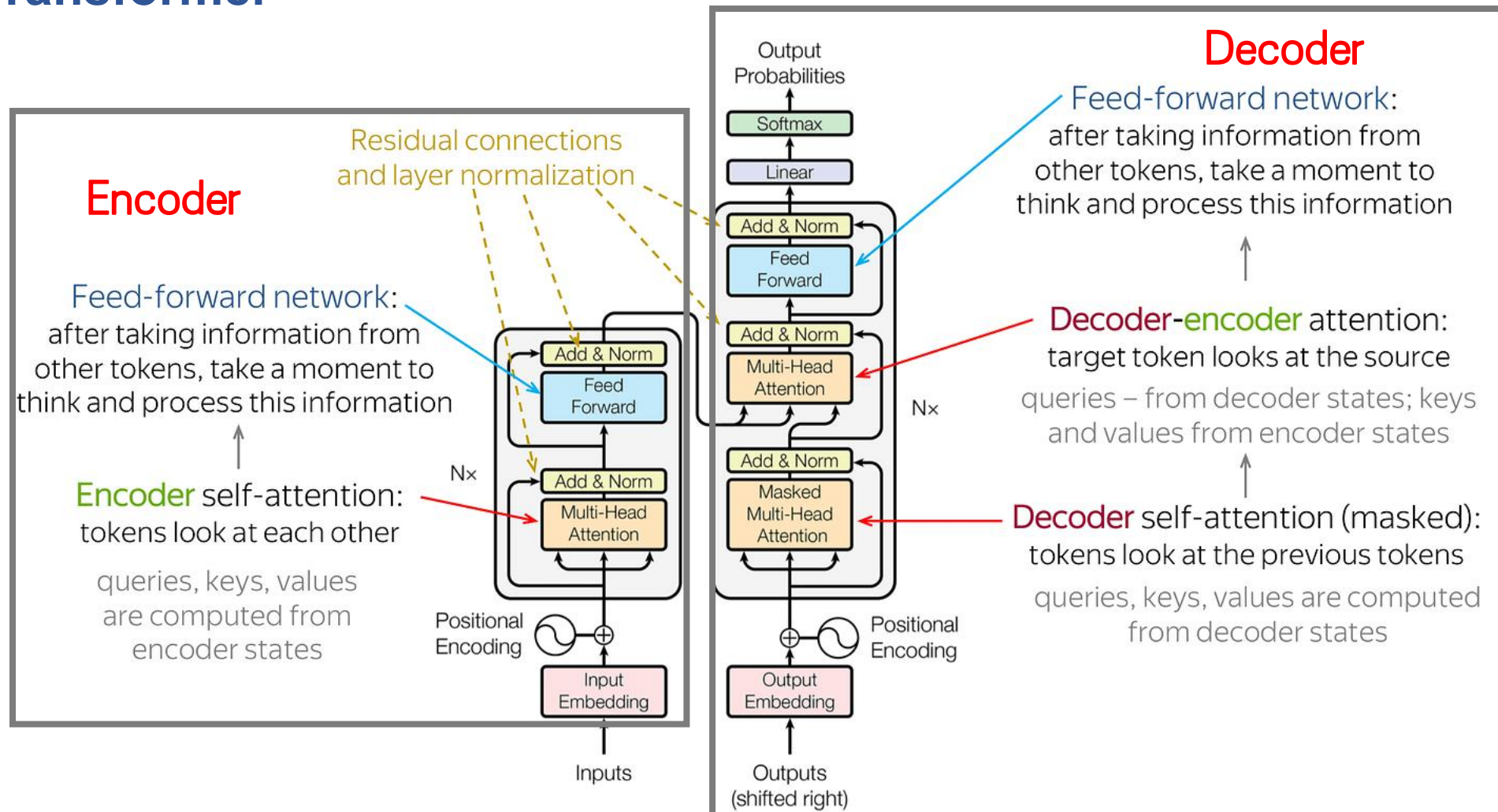
New Bing

Recommender System + LLMs?

- Introduction
- **Background: LM & LM4Rec**
- Development of LLMs
- Progress of LLM4Rec
- Open Problems
- Future Direction & Conclusions

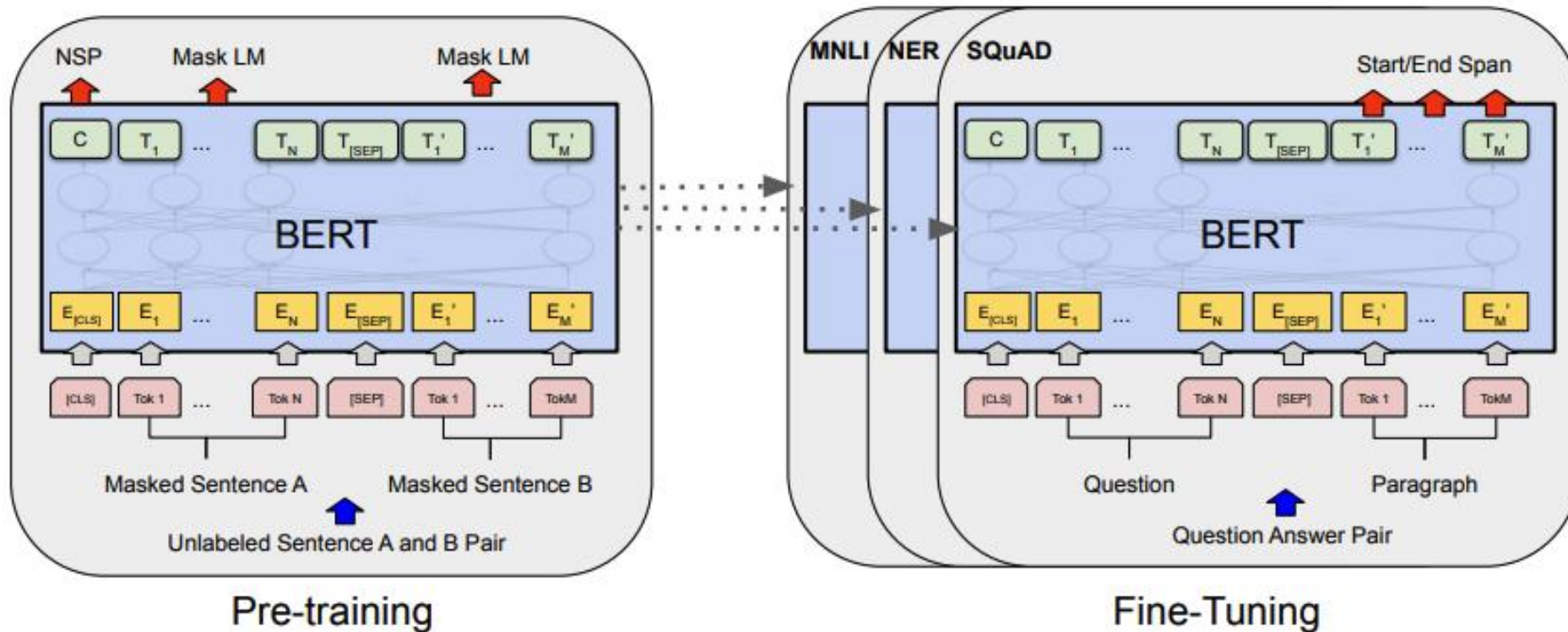
Development of LMs

Transformer



Development of LMs

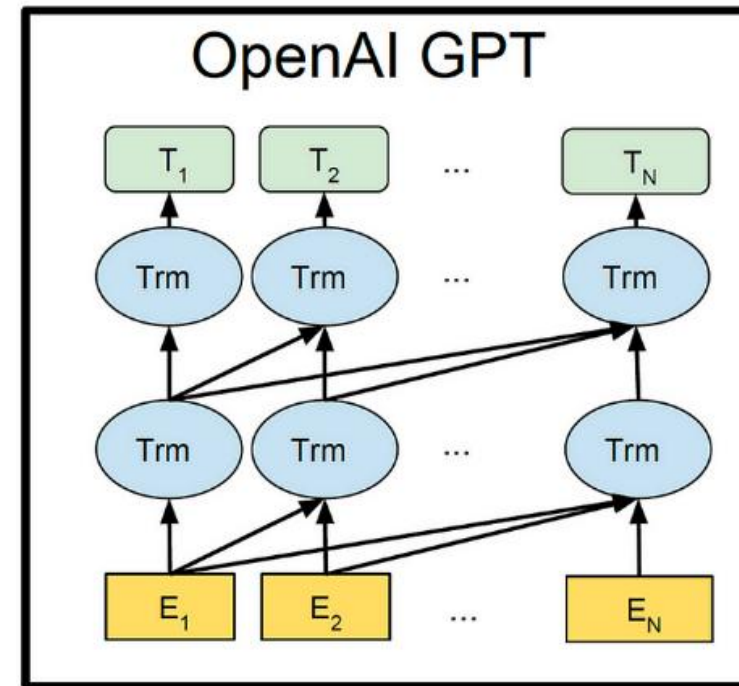
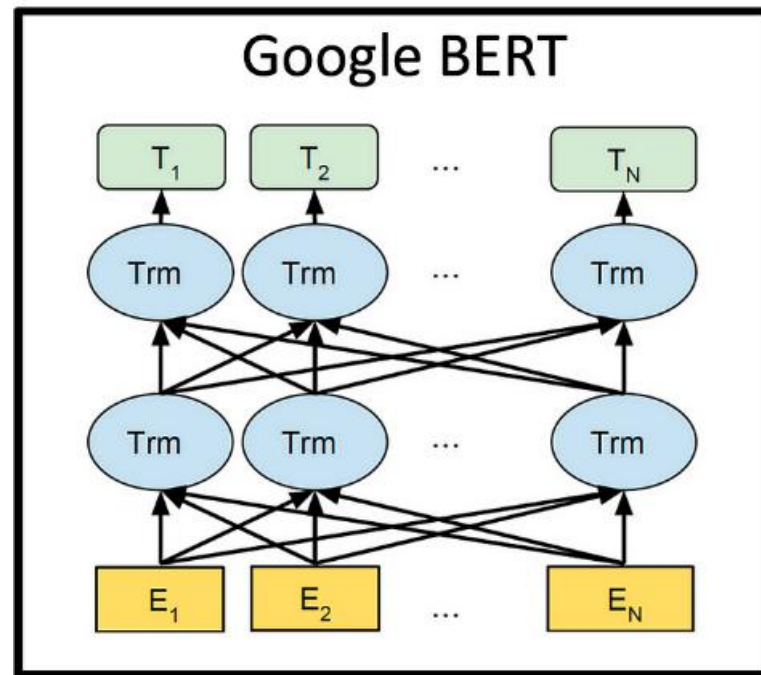
- Bert: pre-training of deep bidirectional transformers
- Mask Language Modeling, bi-direction
- Encoder (advantage) --> understanding



Development of LMs

- GPT2: generative pre-trained transformer
 - Causal language modeling
 - Decoder (advantage) --> Generation
 - unsupervised multi-task learner

$$p(x) = \prod_{i=1}^n p(s_i | s_1, \dots, s_{i-1})$$



□ How can recommender systems benefit from LMs

- Model architecture:

Transformer、Self-attention

- Task formulation

Use language to formulate the recommendation task

- Representation:

Textual feature,
item representation,
knowledge representation

- Learning paradigm:

Pretrain-finetune,
Prompt learning

- LMs for recommendation
 - ❑ Utilizing LMs' model structure for recommendation.
 - ❑ ID-based: **BERT4Rec**, SASRec ...
 - ❑ Text-based: **Recformer** ...
 - ❑ LM as item encoder. UniSRec, VQRec, MoRec ...
 - ❑ Recommendation as natural language processing.
 - ❑ ID-based: **P5**, VIP5 ...
 - ❑ Text-based: **M6-Rec**, Prompt4NR ...

Utilizing LM Model Structure

□ Bert4Rec: ID-based

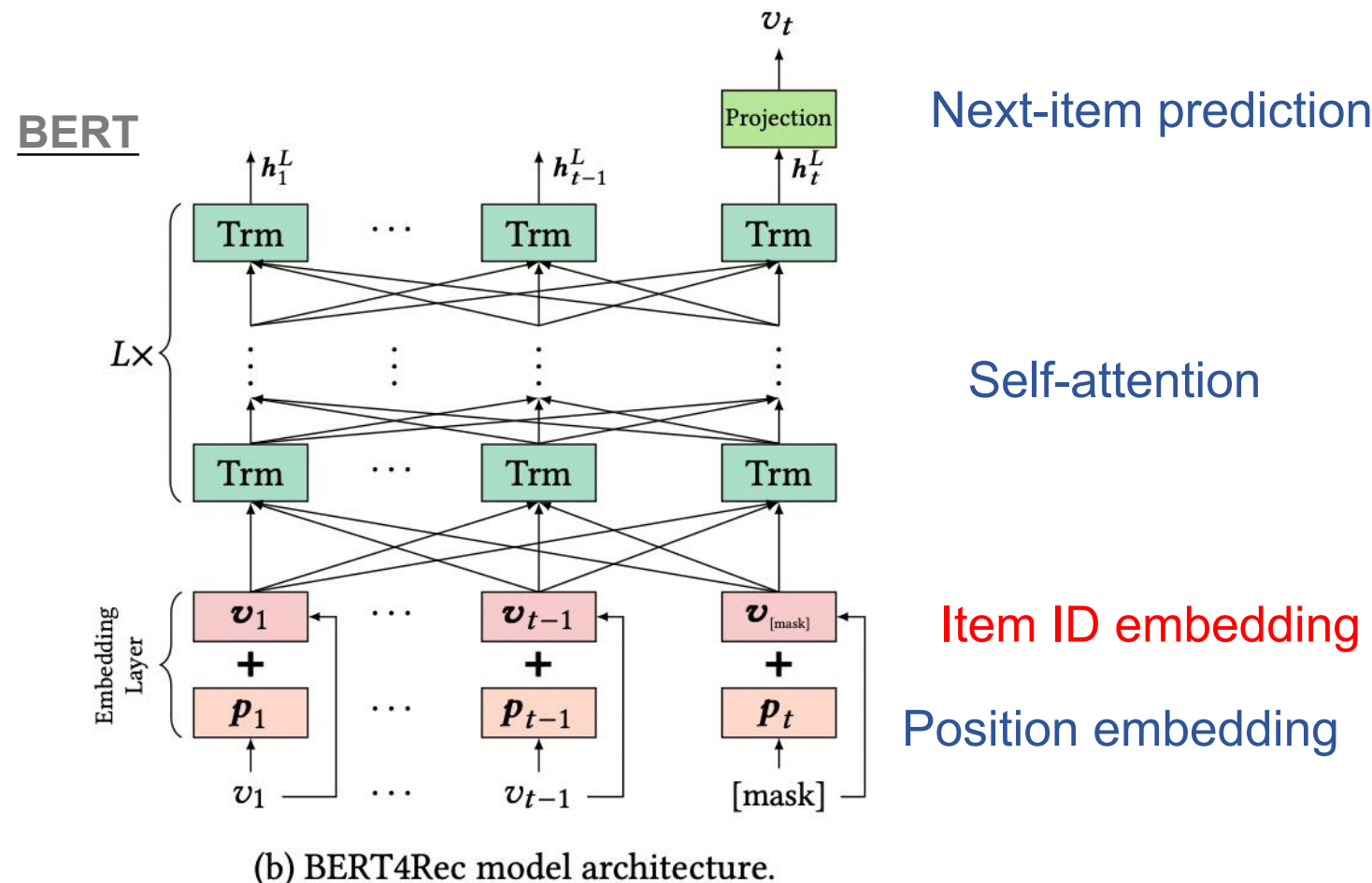
Natural Language:

- Token sequence
- Inter-token correlations



RecSys:

- ID sequence
- Inter-item correlations



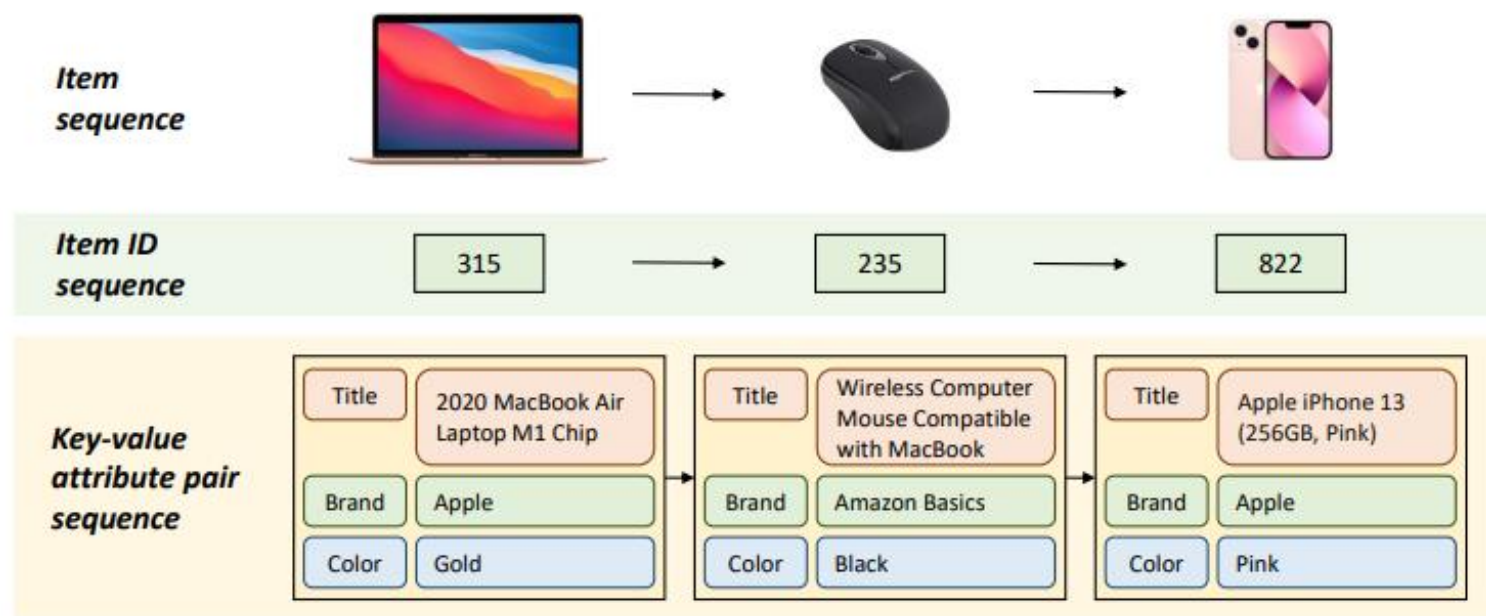
Training recommender by masked item prediction as BERT.

Utilizing LM Model Structure

❑ Recformer: text-based

❑ Text is all you need (NO item ID)

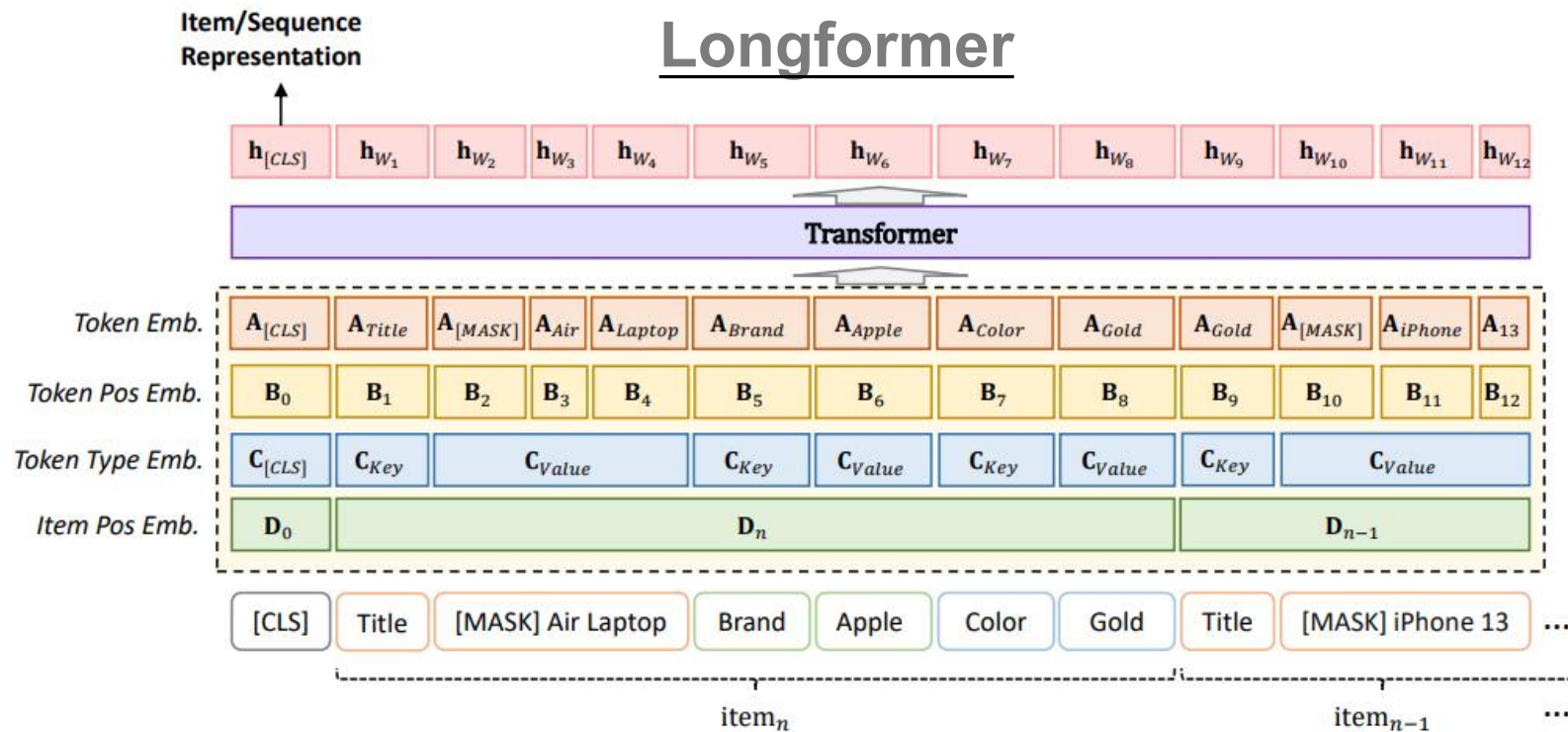
- Only use texts to represent items.
- Low resource, better cold-start recommendation.



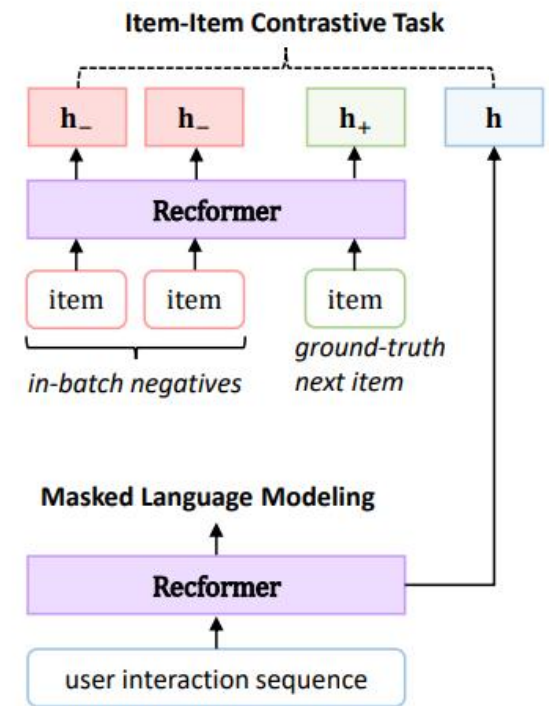
Utilizing LM Model Structure

□ Recformer: text-based

□ Text is all you need (NO item ID)



(a) Recformer Model Structure



(b) Pretraining

LM as Text Encoder

UniSRec

- Enhance the recommendation model by using LMs to encode the natural language representation of items.

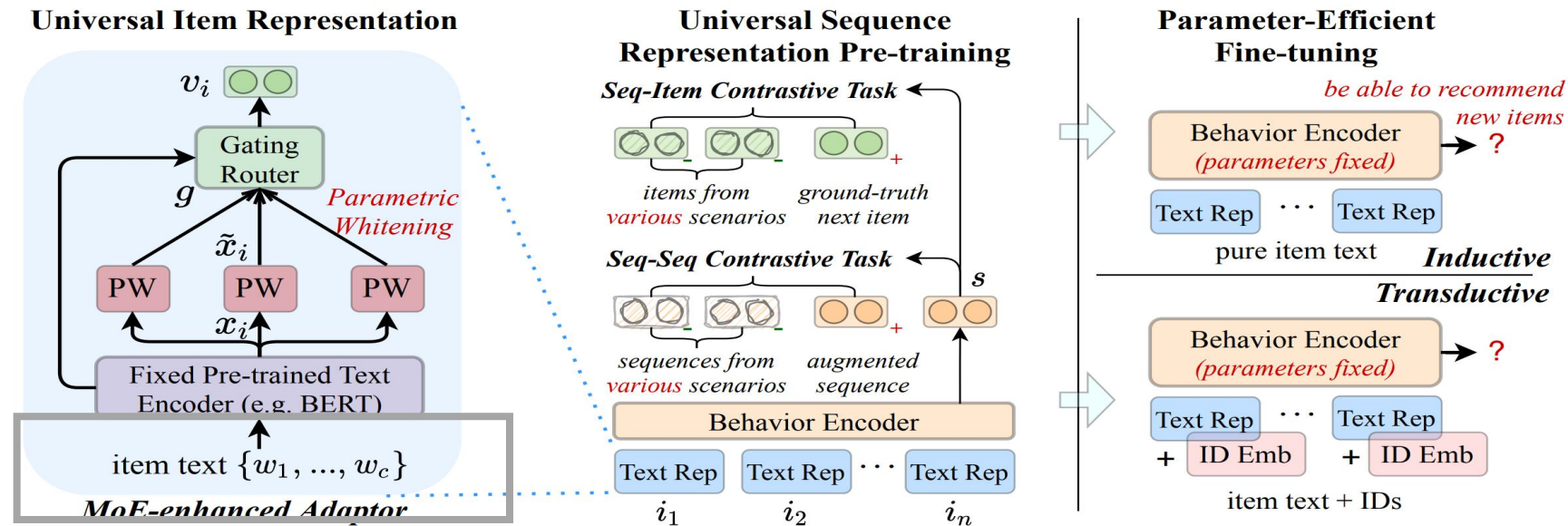


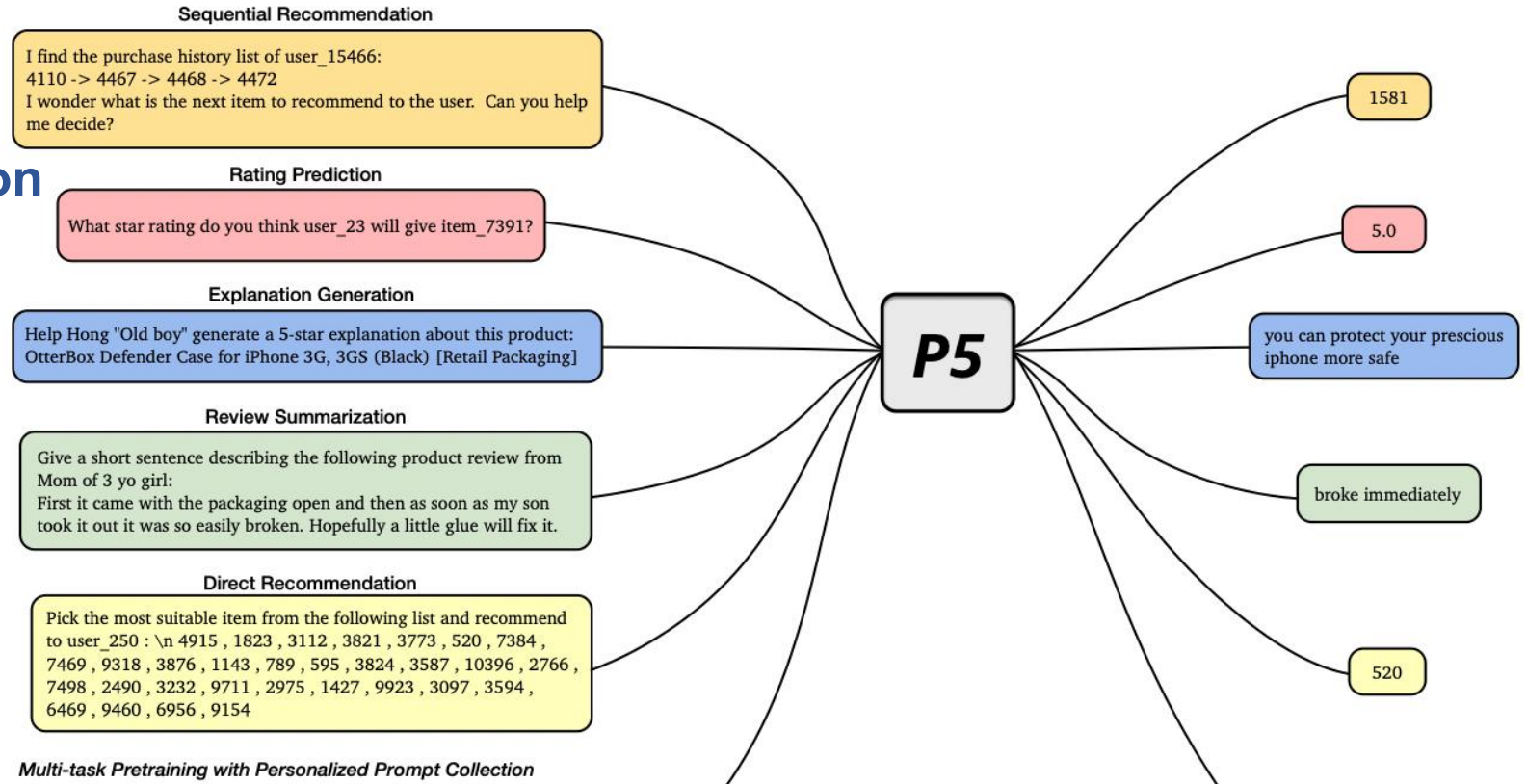
Figure 1: The overall framework of the proposed universal sequence representation learning approach (UniSRec).

Recommendation as NLP

□ P5: use natural language to describe different rec. tasks.

□ Multi-task prompts

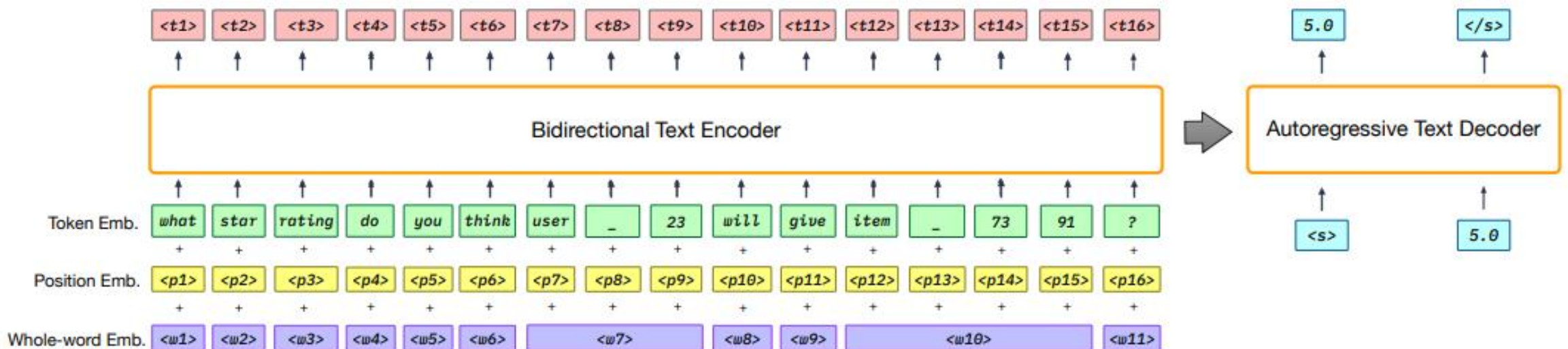
- Sequential recommendation
- Rating prediction
- Explain generation
- Review summarization
- Direct recommendation



Recommendation as NLP

□ P5 Architecture:

- Autoregressive decoding
- Users and items are represented with ID information



Recommendation as NLP

❑ M6-Rec: represent users/item with plain texts and converting the tasks to either language understanding or generation

- Understanding (scoring) task: CTR, CVR prediction
- Generation task: personalized product design, explanation generation...

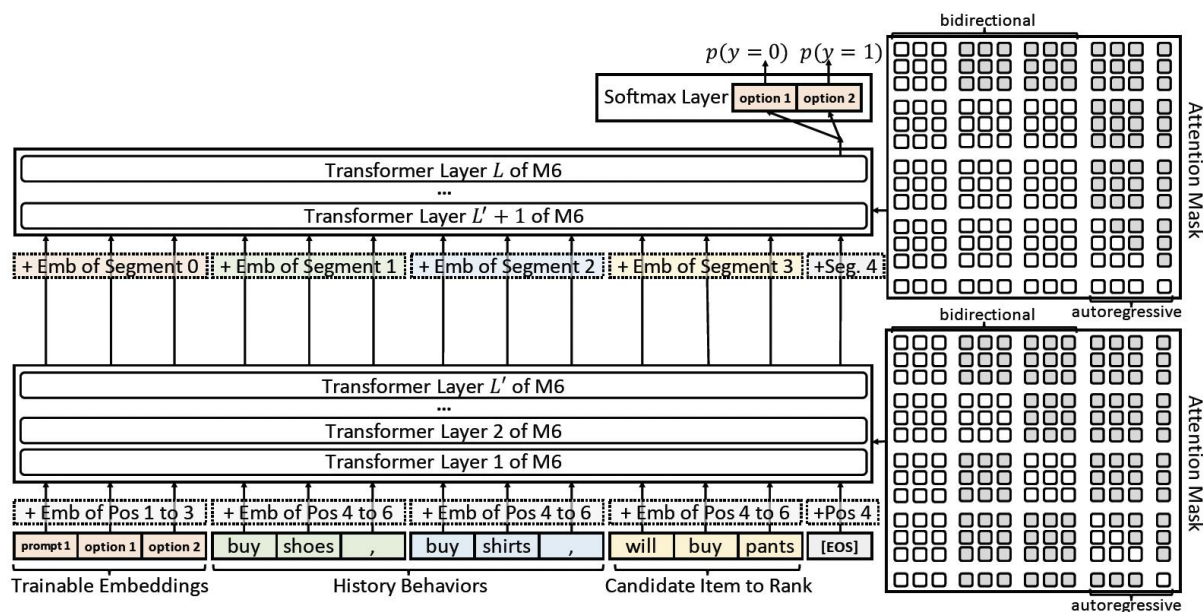
M6 (~300M parameters)

User description

[BOS'] December. Beijing, China. Cold weather. A male user in early twenties, searched “winter stuff” 23 minutes ago, clicked a product of category “jacket” named “men’s lightweight warm winter hooded jacket” 19 minutes ago, clicked a product of category “sweat-shirt” named “men’s plus size sweatshirt stretchy pullover hoodies” 13 minutes ago, clicked ... [EOS']

[BOS] The user is now recommended a product of category “boots” named “waterproof hiking shoes mens outdoor”. The product has a high population-level CTR in the past 14 days, among the top 5%. The user clicked the category 4 times in the last 2 years. [EOS]

Item description

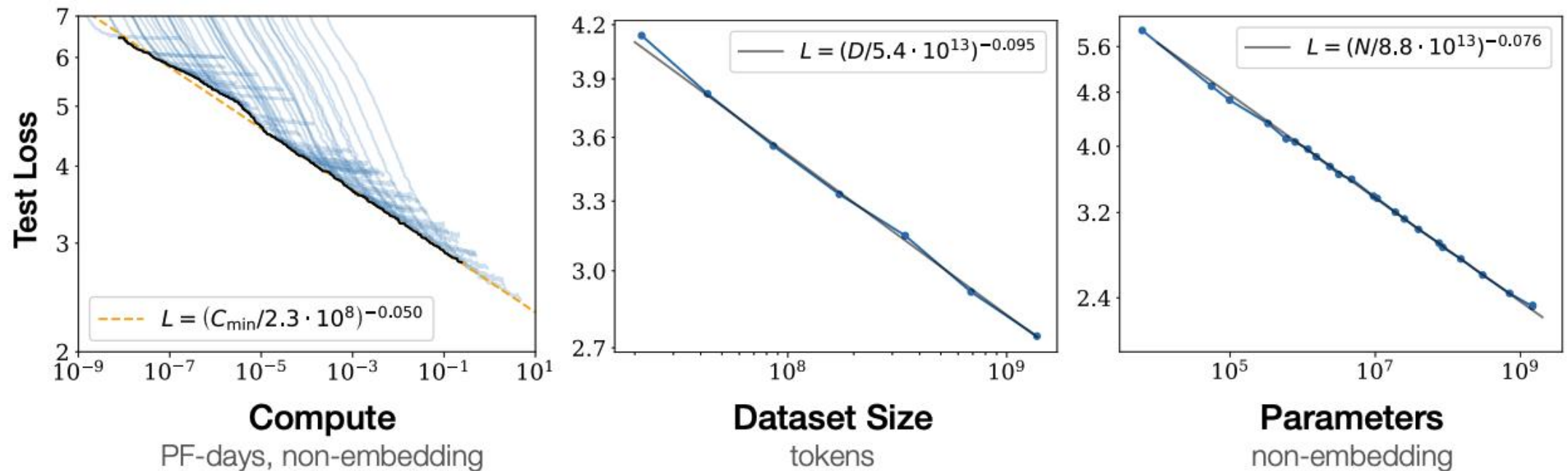


- Introduction
- Background: LM & LM4Rec
- **Development of LLMs**
- Progress of LLM4Rec
- Open Problems
- Future Direction & Conclusions

Developments of LLMs

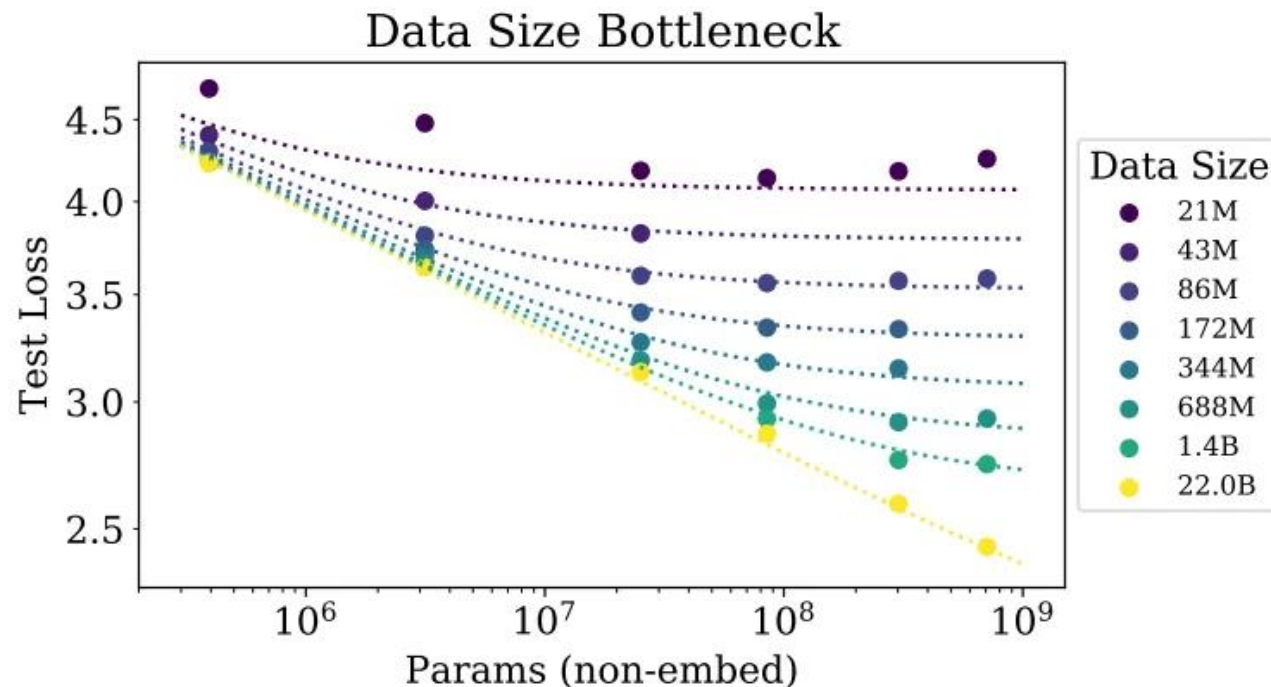
□ Scaling Laws

- The greater the amount of the data and the model parameters, the better the performance of the model
- Performance can be predicted



□ Scaling Laws

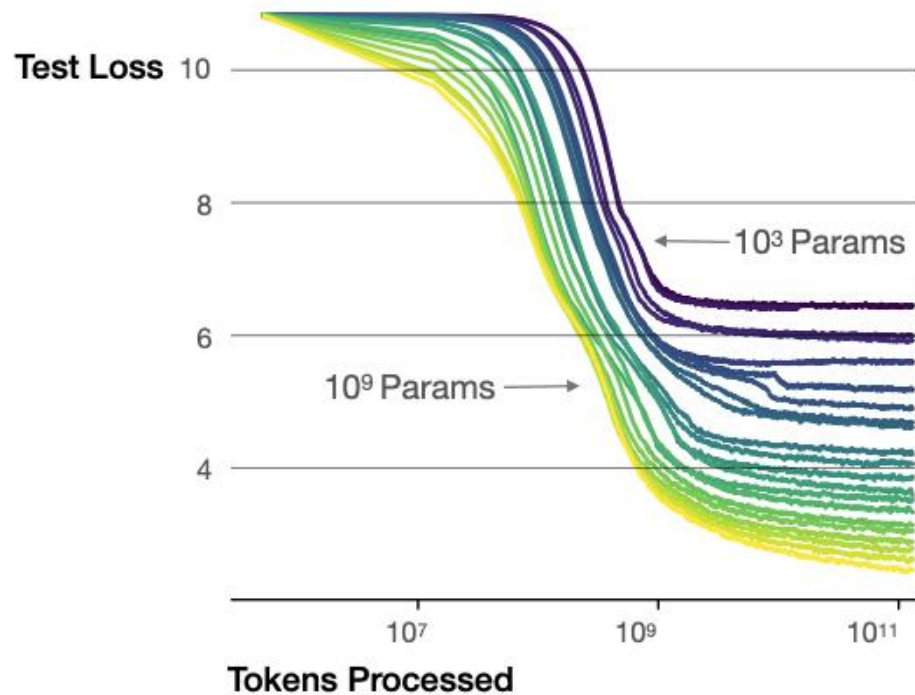
- The greater the amount of the data and the model parameters, the better the performance of the model
- Performance can be predicted



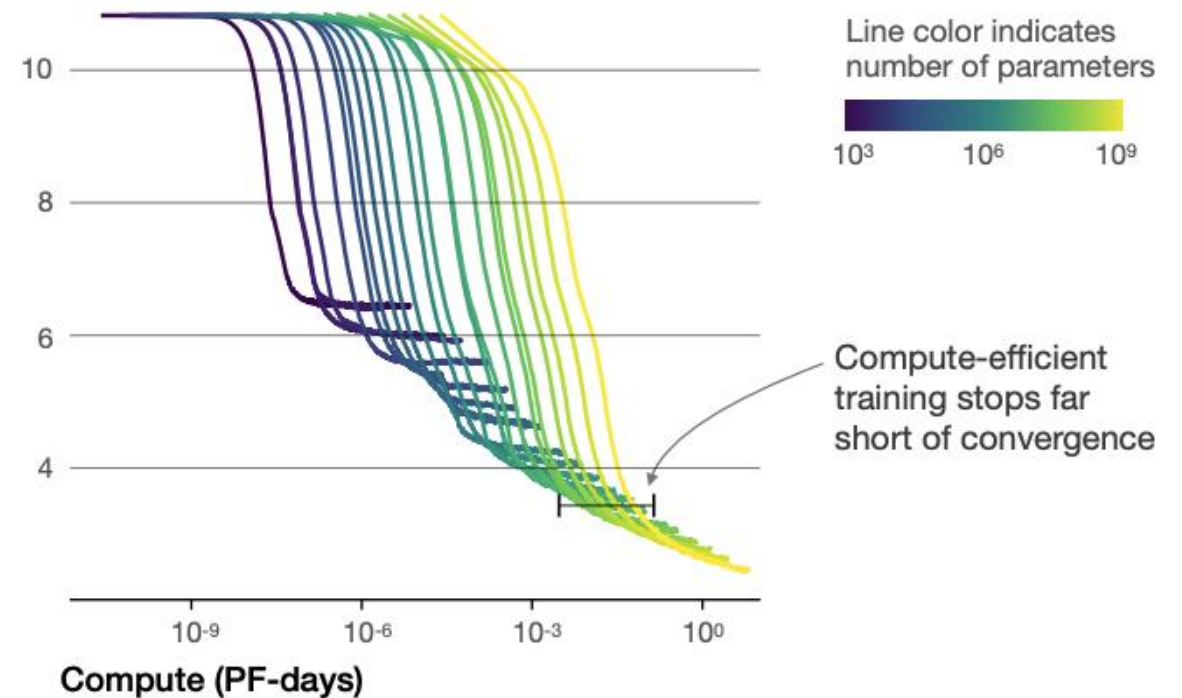
Developments of LLMs

□ Scaling Laws

Larger models require **fewer samples** to reach the same performance



The optimal model size grows smoothly with the loss target and compute budget



Developments of LLMs

□ Align with human

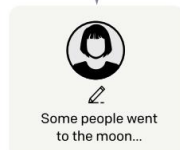
Step 1

Collect demonstration data, and train a supervised policy.

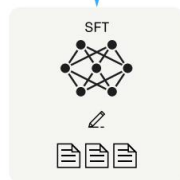
A prompt is sampled from our prompt dataset.



A labeler demonstrates the desired output behavior.



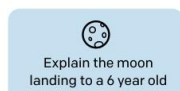
This data is used to fine-tune GPT-3 with supervised learning.



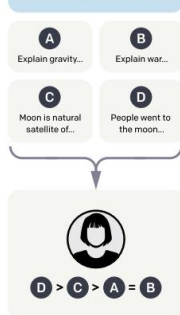
Step 2

Collect comparison data, and train a reward model.

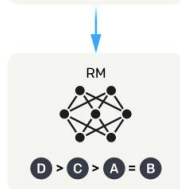
A prompt and several model outputs are sampled.



A labeler ranks the outputs from best to worst.



This data is used to train our reward model.



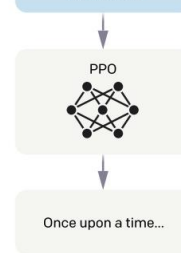
Step 3

Optimize a policy against the reward model using reinforcement learning.

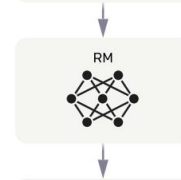
A new prompt is sampled from the dataset.



The policy generates an output.

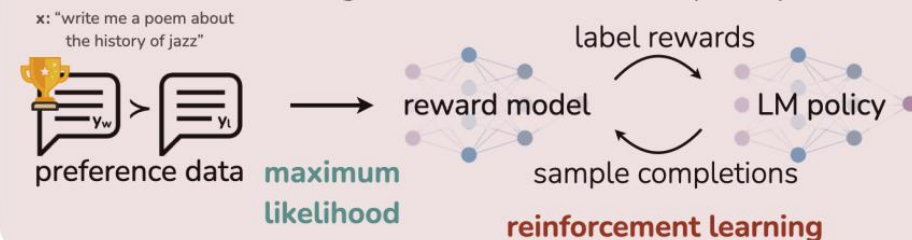


The reward model calculates a reward for the output.

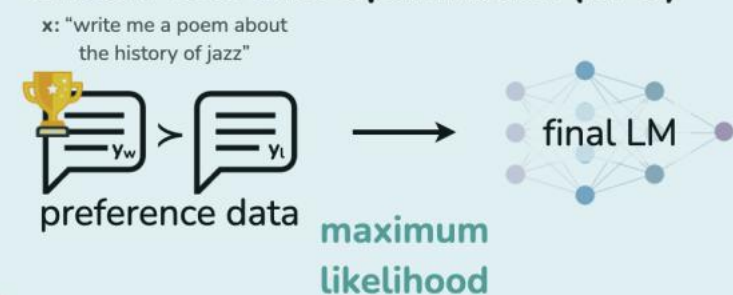


The reward is used to update the policy using PPO.

Reinforcement Learning from Human Feedback (RLHF)



Direct Preference Optimization (DPO)



Developments of LLMs

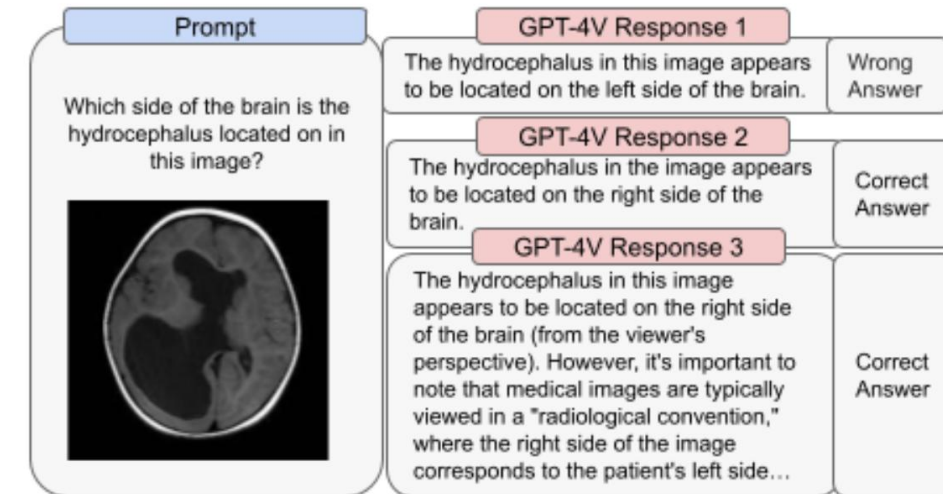
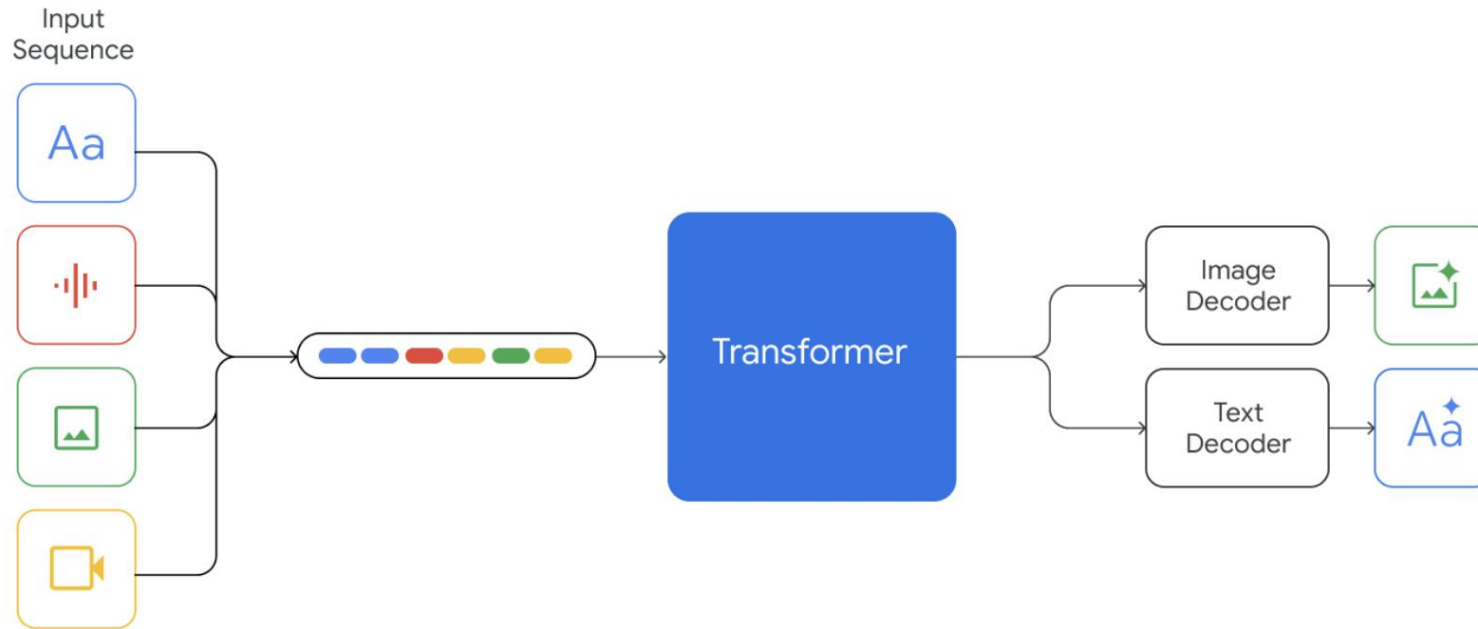


| Objective Overall | | Subjective Overall | | Language | | Knowledge | | Reason | | Math | | Co | | ... | |
|--------------------------|----------------------|--------------------|--|-----------|--|-----------|--|------------|--|------|--|----|--|---------|--|
| | Model | | | Release | | Type | | Parameters | | | | | | Average | |
| 1 | GPT-4-Turbo-20240409 | | | 2024/4/9 | | Chat | | N/A | | | | | | 63.1 | |
| Close Source · OpenAI | | | | | | | | | | | | | | | |
| 2 | GPT-4-Turbo-1106 | | | 2023/11/6 | | Chat | | N/A | | | | | | 62 | |
| Close Source · OpenAI | | | | | | | | | | | | | | | |
| 3 | Claude3-Opus | | | 2024/3/4 | | Chat | | N/A | | | | | | 60.5 | |
| Close Source · Anthropic | | | | | | | | | | | | | | | |
| 4 | GLM-4 | | | 2024/1/16 | | Chat | | N/A | | | | | | 57.8 | |
| Close Source · Zhipu AI | | | | | | | | | | | | | | | |
| 5 | abab6.5 | | | 2024/4/17 | | Chat | | N/A | | | | | | 57.8 | |
| Close Source · MiniMax | | | | | | | | | | | | | | | |
| 6 | Llama3-70B-Instruct | | | 2024/4/18 | | Chat | | 70B | | | | | | 57 | |
| Open Source · Meta | | | | | | | | | | | | | | | |
| 7 | Qwen1.5-110B-Chat | | | 2024/4/25 | | Chat | | 72B | | | | | | 56.8 | |
| Open Source · Alibaba | | | | | | | | | | | | | | | |

| Objective Overall | | Subjective Overall | | Language | | Knowledge | | Reason | | Math | | Co | | ... |
|-------------------|--|--------------------|---------------------------------|----------|--|------------|---------|--------|--|------|--|----|--|-----|
| | Model | | Release | Type | | Parameters | Average | | | | | | | |
| 1 | GPT-4-Turbo-1106 Close Source · OpenAI | | 2023/11/6 updated: 2024/1/29 | Chat | | N/A | 50 | | | | | | | |
| 2 | Qwen-Max-0403 Close Source · Alibaba | | 2024/3/26 updated: 2024/4/2 | Chat | | N/A | 50 | | | | | | | |
| 3 | GPT-4-Turbo-20240409 Close Source · OpenAI | | 2024/4/9 updated: 2024/5/9 | Chat | | N/A | 49.9 | | | | | | | |
| 4 | Claude3-Opus Close Source · Anthropic | | 2024/3/4 updated: 2024/4/2 | Chat | | N/A | 48.1 | | | | | | | |
| 5 | Spark-v3.5 Close Source · Iflytek | | 2024/1/30 updated: 2024/5/9 | Chat | | N/A | 48.1 | | | | | | | |
| 6 | Qwen1.5-110B-Chat Open Source · Alibaba | | 2024/4/25 updated: 2024/5/9 | Chat | | 72B | 47.4 | | | | | | | |
| 7 | ERNIE-4.0-8K-0329 Close Source · Baidu Inc. | | 2024/3/29 updated: 2024/5/9 | Chat | | N/A | 46.6 | | | | | | | |

❑ More and more LLMs have shown powerful capabilities

Developments of LLMs



❑ Multi-model to Multi-model unified model is now developing at a rapid pace.

Augmented capabilities of LLMs

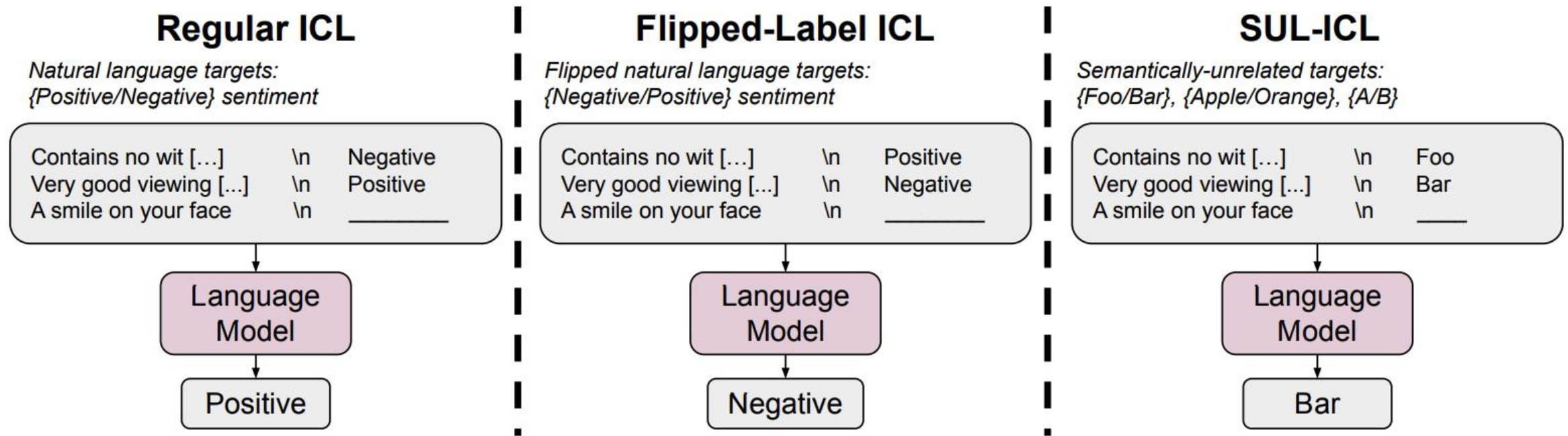


- ❑ Emergent abilities of LLM
 - ❑ Sufficient world knowledge
 - ❑ Chatting
 - ❑ Incontext Learning & Instruction Following
 - ❑ Reasoning & Planning
 - ❑ Tool using
 - ❑ LLM as an Agent
 - ❑ ...

Augmented capabilities of LLMs

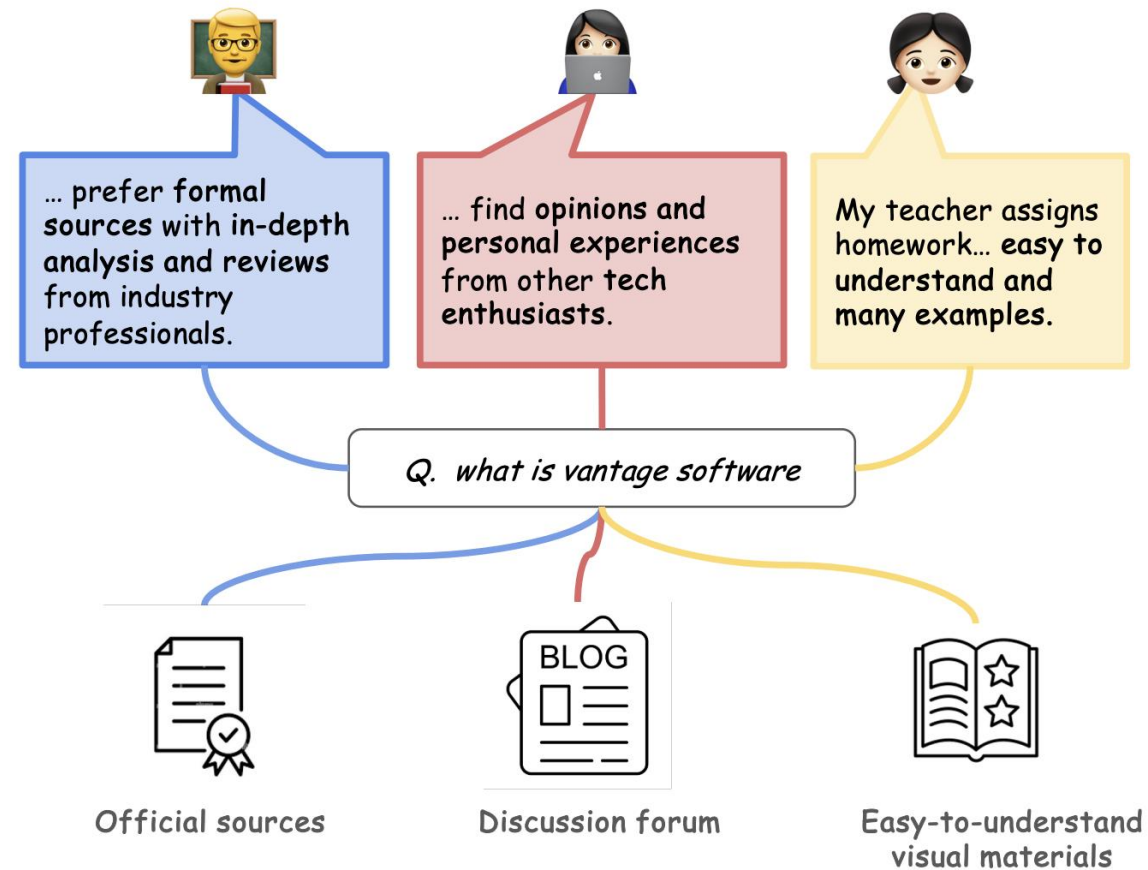
□ In-context Learning

- Following their example to override the semantic prior



Augmented capabilities of LLMs

□ Instruction following



Augmented capabilities of LLMs

□ Reasoning & Planning

- LLM can decompose the problem into simple sub-problems to improve their ability

Standard Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The answer is 27. ❌

Chain-of-Thought Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. $5 + 6 = 11$. The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

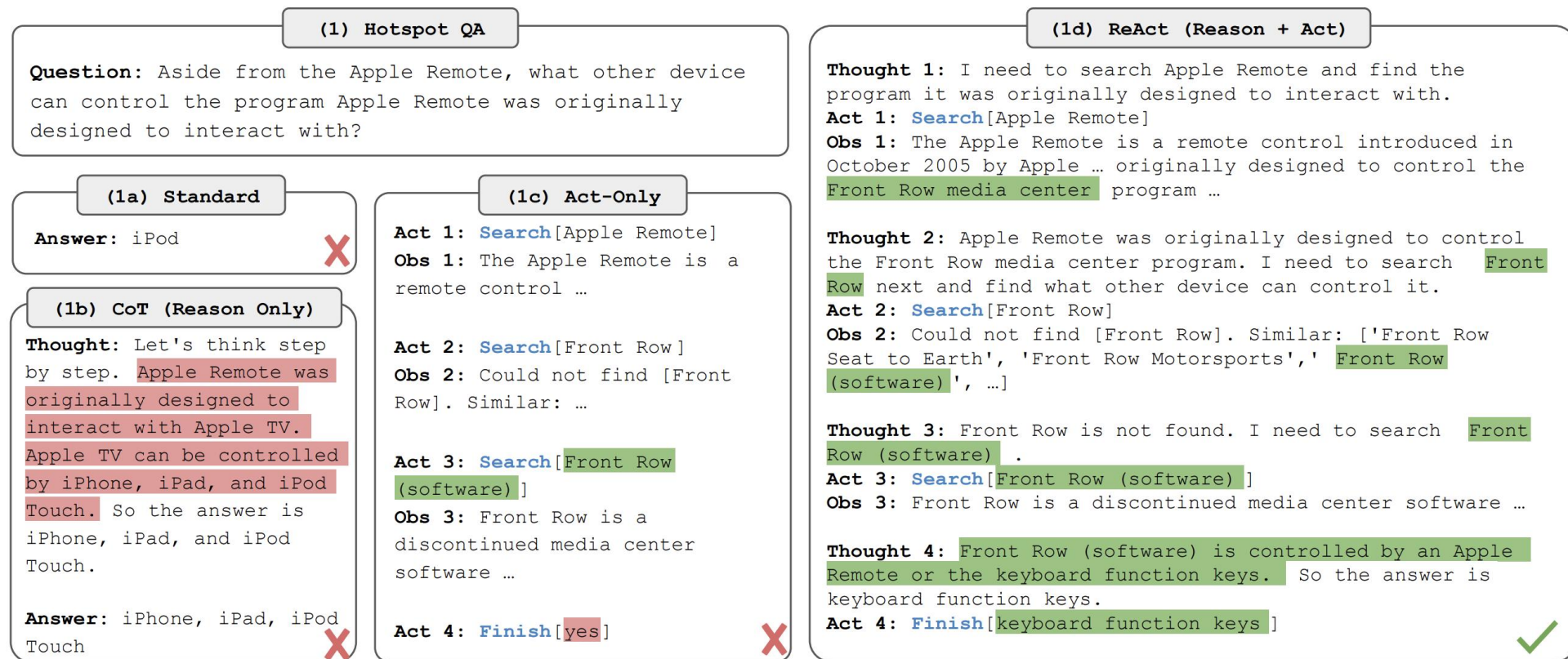
Model Output

A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had $23 - 20 = 3$. They bought 6 more apples, so they have $3 + 6 = 9$. The answer is 9. ✅

Augmented capabilities of LLMs

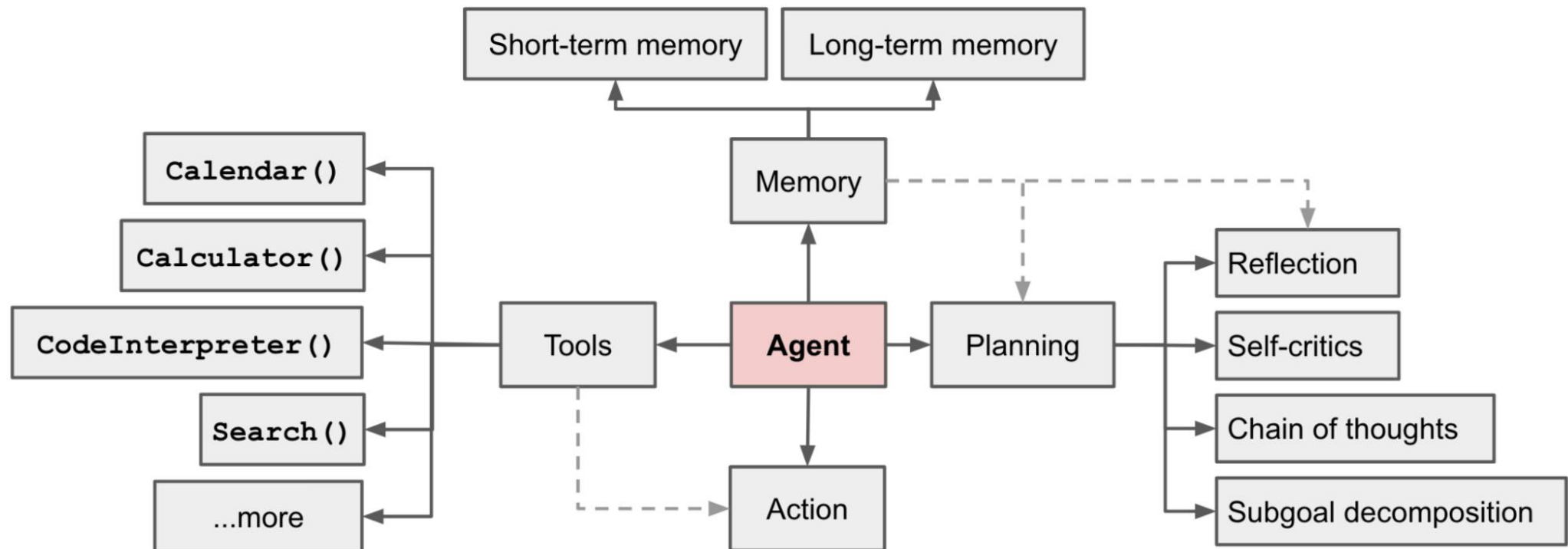
□ Reasoning & Planning

- LLM can break down the target task according to the environment and develop a



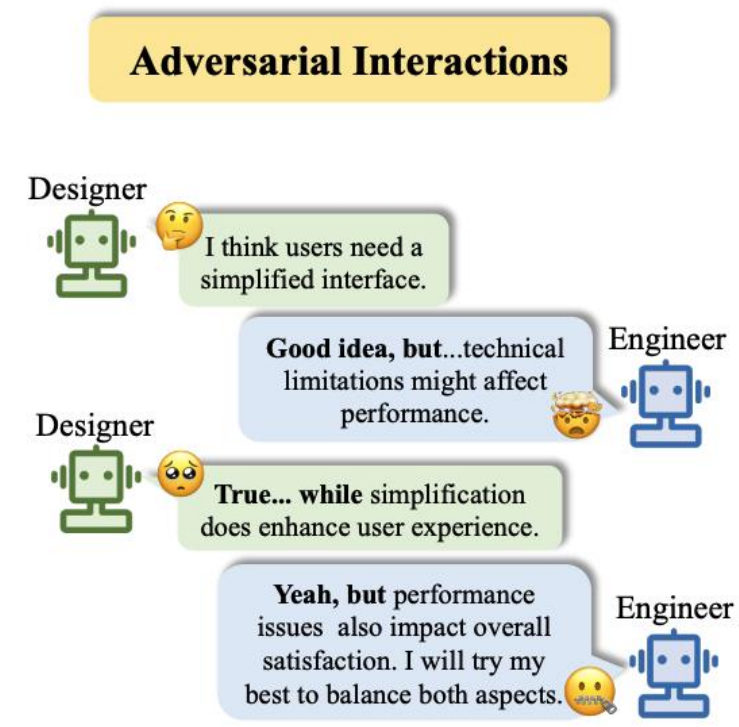
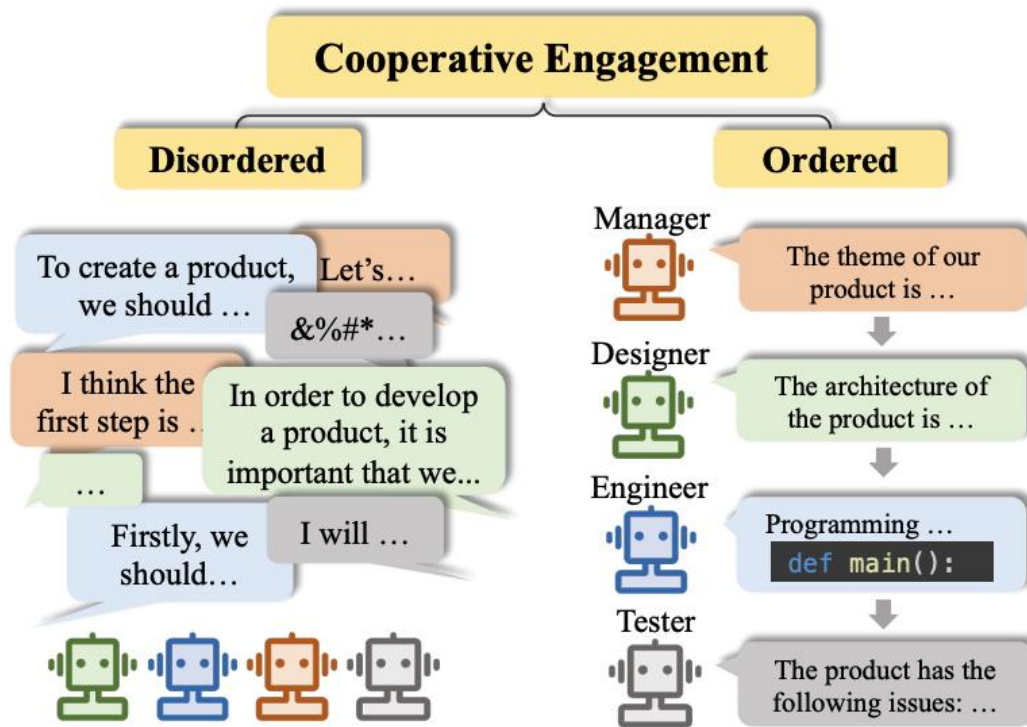
Augmented capabilities of LLMs

□ LLM as an Agent



❑ Multi-Agent

- ❑ Group intelligence surpasses individual intelligence
- ❑ Cooperative for complementary / Adversarial for advancement



□ How recommender systems benefit from LLMs

- Representation:

Textual feature,
item representation,
knowledge representation

- Interaction:

Acquire user information
needs via dialog (**chat**)

- Generalization:

cross-domain, knowledge
compositional-
generalization

- Generation:

Personalized content
generation,
explanation generation

- Learning paradigm: Pretrain-finetune, Instruction-tuning, Preference-alignment

- Model architecture: Transformer、Self-attention,

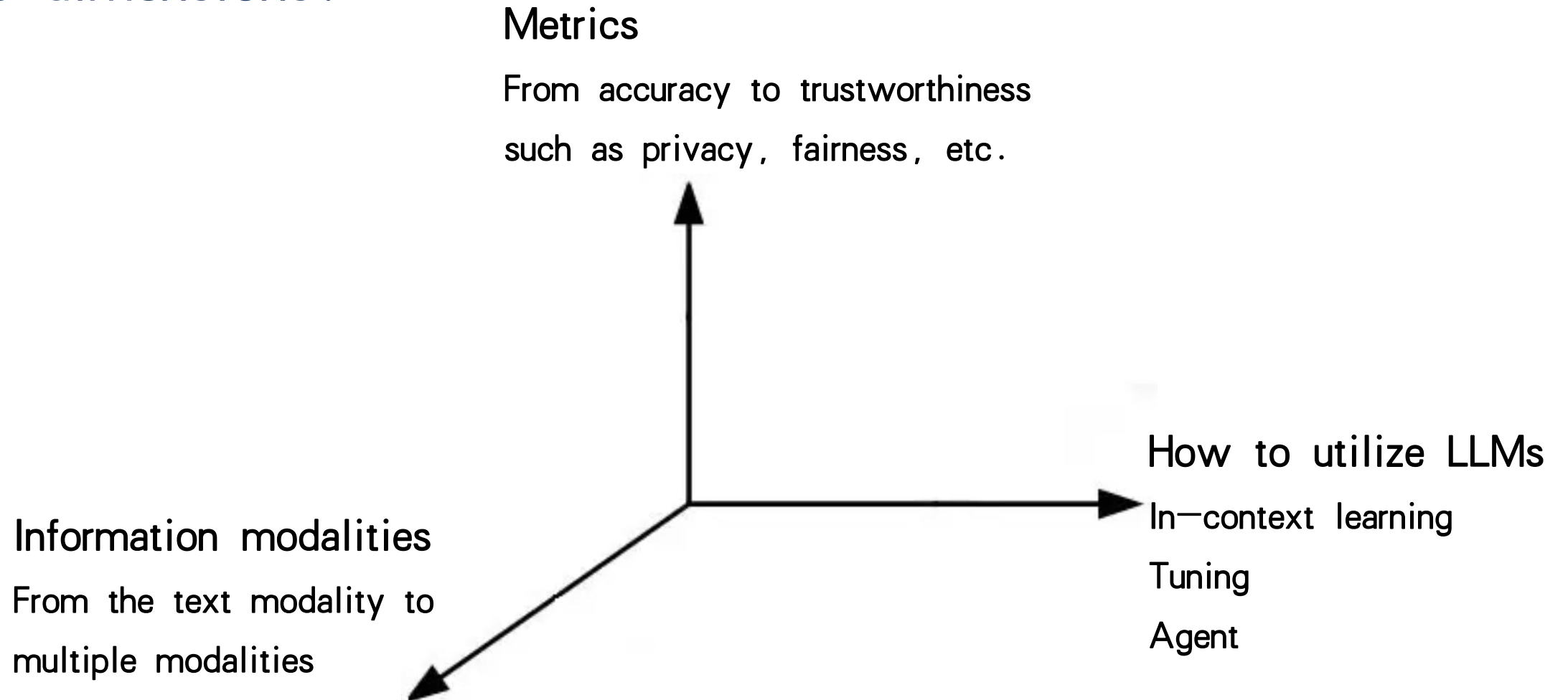
- ❑ **Key Challenge**

- ❑ **Mismatch between pretraining objective and recommendation**

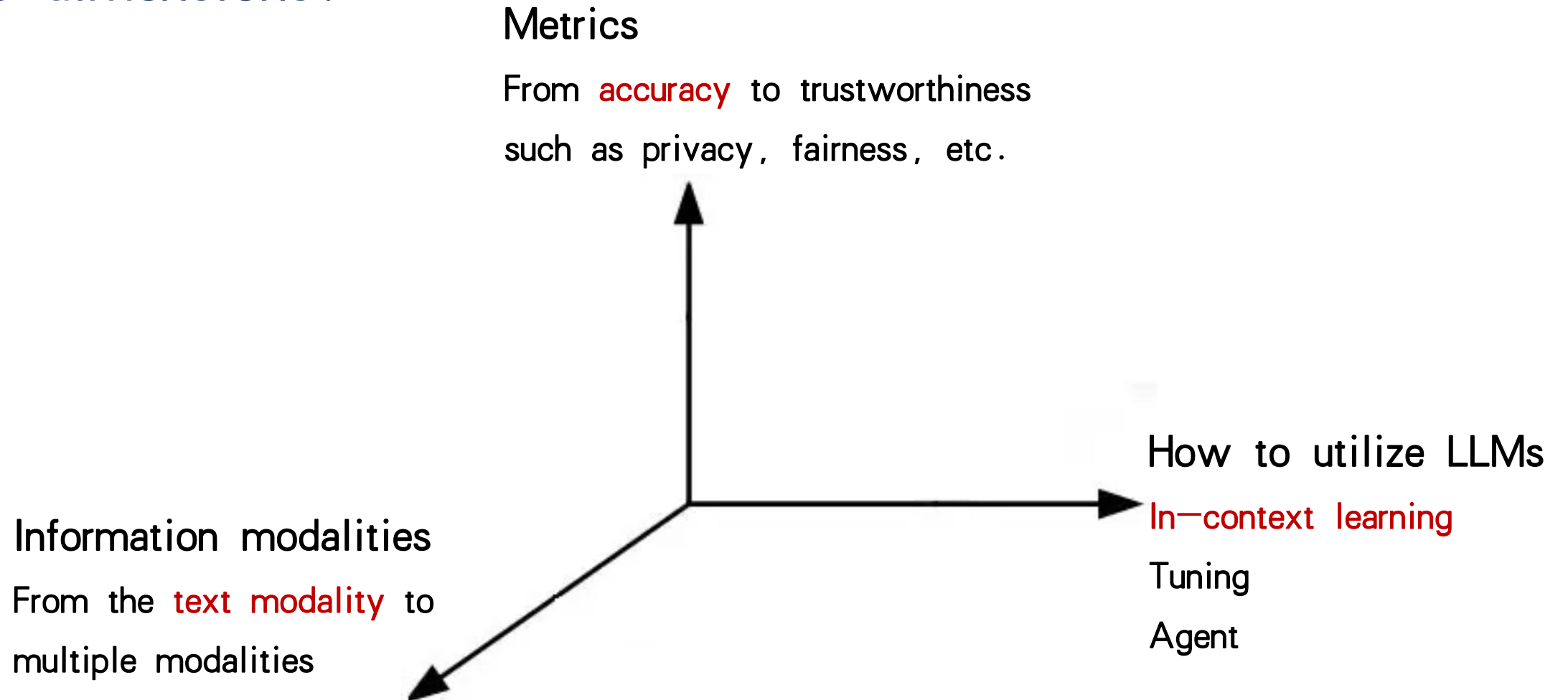
- ❑ **Tend to rely on semantics, and another important aspect of recommendation tasks is collaborative information.**

- Introduction
- Background: LM & LM4Rec
- The progress of LLM4Rec
 - Development of LLMs
 - LLMs for Recommendation
 - ICL
 - Tuning
 - Agent
- Open Problems
- Conclusions

Three dimensions:



Three dimensions:

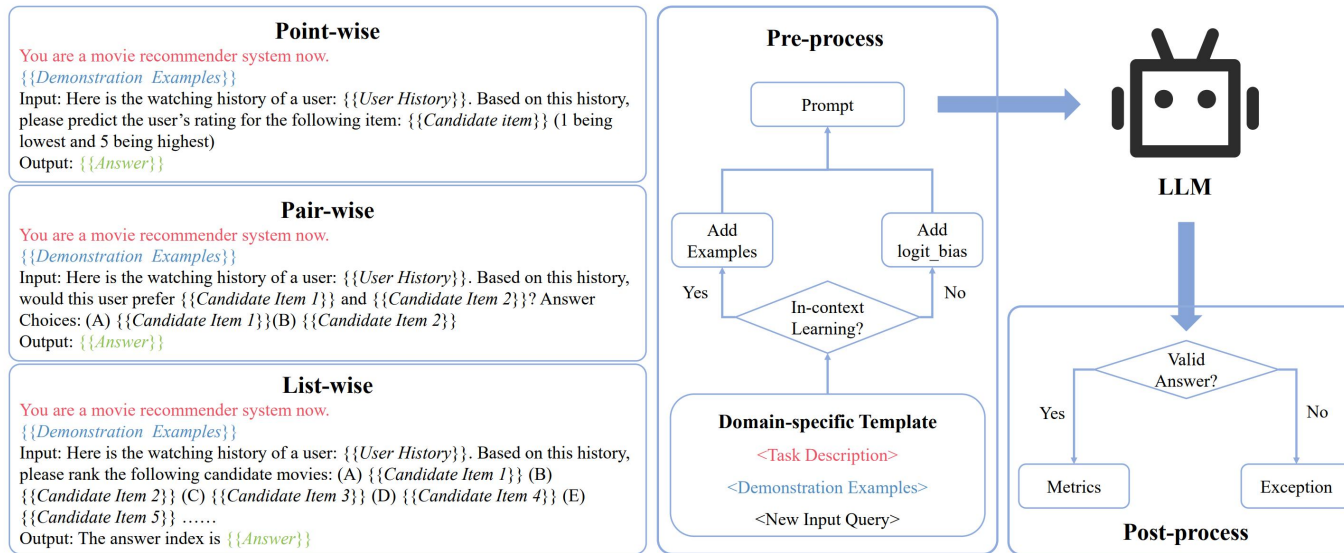


❑ In-context learning

- LLMs has rich world knowledge, wonderful abilities like reasoning, instruction following, in-context learning.
- The LLMs itself could be leveraged for recommendation by in context learning.
- Existing works on in-context learning:
 - Ask LLM for recommendation
 - Serving as knowledge augmentation for traditional recsys
 - Optimize the prompt used for recommendation
 - Directly used for conversational recommender system

❑ In-context learning: directly ask LLMs for recommendation

- Prompt construction



Three different ways of measuring ranking abilities:

$$\hat{y}'_i = LLM_{\text{point}}(I, \mathcal{D}, f(\mathbf{h}', \mathbf{c}' | u))$$

$$\hat{y}'_{i_m > i_n} = LLM_{\text{pair}}(I, \mathcal{D}, f(\mathbf{h}', \mathbf{c}' | u))$$

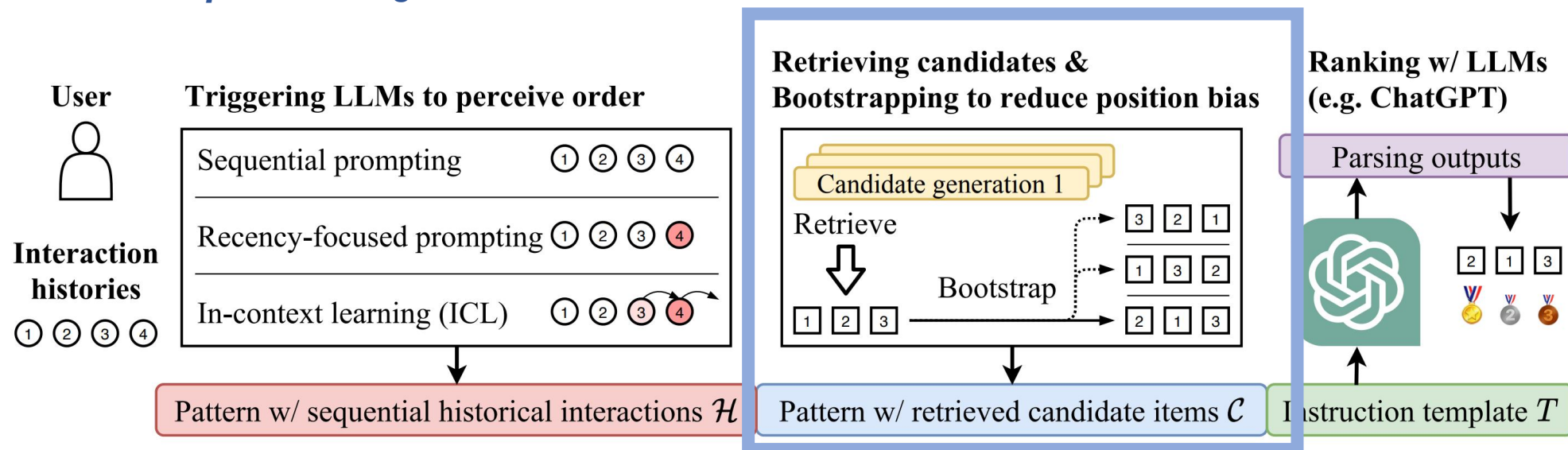
$$\hat{y}'_{i_1}, \hat{y}'_{i_2}, \dots, \hat{y}'_{i_k} = LLM_{\text{list}}(I, \mathcal{D}, f(\mathbf{h}', \mathbf{c}' | u))$$

Figure 1: The overall evaluation framework of LLMs for recommendation. The left part demonstrates examples of how prompts are constructed to elicit each of the three ranking capabilities. The right part outlines the process of employing LLMs to perform different ranking tasks and conduct evaluations.

❑ In-context learning: re-ranking given candidate items

❑ Task formulation:

- Using **historical interaction** to rank items retrieved by existing recsys.
- **Input:** language instructions created with historical interactions and candidate items
- **Output:** ranking of the candidate items



❑ In-context learning: ranking given candidated items

❑ Tree types of prompts:

- **Sequential prompting: describing History using language**

"I've watched the following movies in the past in order: '0. Multiplicity', '1. Jurassic Park',"

- **Recency-focused prompting: **emphasize** most recent interactions**

"I've watched the following movies in the past in order: '0. Multiplicity', '1. Jurassic Park', Note that my most recently watched movie is Dead Presidents. . . ."

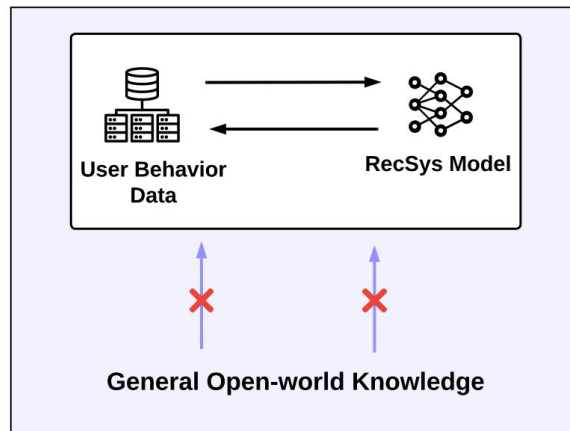
- **In-context learning (ICL): **providing recommendation example****

"If I've watched the following movies in the past in order: '0. Multiplicity', '1. Jurassic Park', . . . , then you should recommend Dead Presidents to me and now that I've watched Dead Presidents, then"

❑ In-context learning: knowledge enhancement

❑ Traditional RecSys vs ICL-based RecSys

Traditional RecSys



Inference fast but being colsed system, generating recommendations relying on local dataset

Directly ask LLMs for recommendaiton



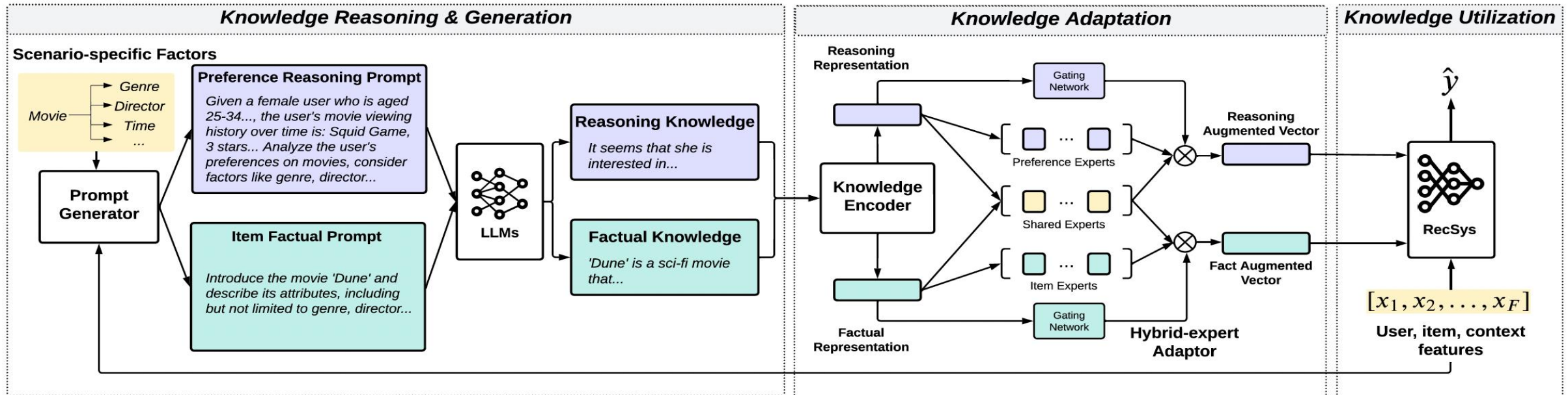
Given the user's historical interactions, please determine whether the user will enjoy the target new movie by answering "Yes" or "No".

Could **leverage open-world knowledge**, but:

- 1) not trained on specific recommendation task
- 2) Inference slowly
- 3) hard to correctly answer compoitional questions

Extract and inject LLM's world knowledge into traditional recommender system

□ In-context learning: knowledge enhancement



Obtain knowledge beyond local rec dataset:

- 1) Generate reasoning knowledge on user preference (factors affect preference)
- 2) Generate factual knowledge about items

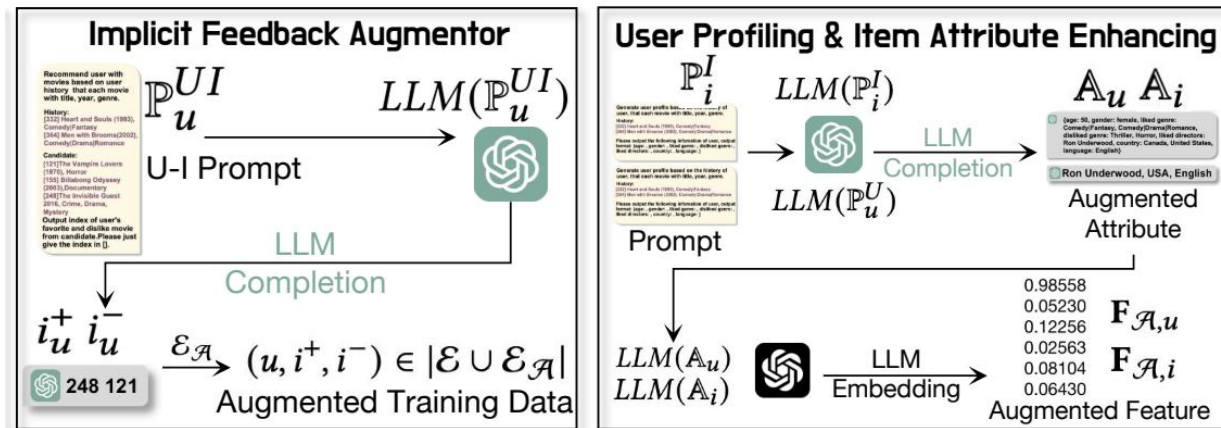
Knowledge Adaptation Stage

encode the textual knowledge and
mapping it into recommendation
space

Knowledge Utilization

Use the knowledge
obtained from LLMs as
additional features

ICL knowledge enhancement for Graph-based Recommendation



- 1) Augmenting user-item interactions
- 2) Enhancing item attributes
- 3) User profiling

Recommend user with movies based on user history that each movie with title, year, genre.

History:
[332] Heart and Souls (1993), Comedy|Fantasy
[364] Men with Brooms(2002), Comedy|Drama|Romance

Candidate:
[121]The Vampire Lovers (1970), Horror
[155] Billabong Odyssey (2003),Documentary
[248]The Invisible Guest 2016, Crime, Drama, Mystery

Output index of user's favorite and dislike movie from candidate. Please just give the index in [].

248 121

(a) Implicit Feedback

Generate user profile based on the history of user, that each movie with title, year, genre.

History:

[332] Heart and Souls (1993), Comedy|Fantasy
[364] Men with Brooms (2002), Comedy|Drama|Romance

Please output the following information of user, output format: {age: , gender: , liked genre: , disliked genre: , liked directors: , country: , language: }

{age: 50, gender: female, liked genre: Comedy|Fantasy, Comedy|Drama|Romance, disliked genre: Thriller, Horror, liked directors: Ron Underwood, country: Canada, United States, language: English}

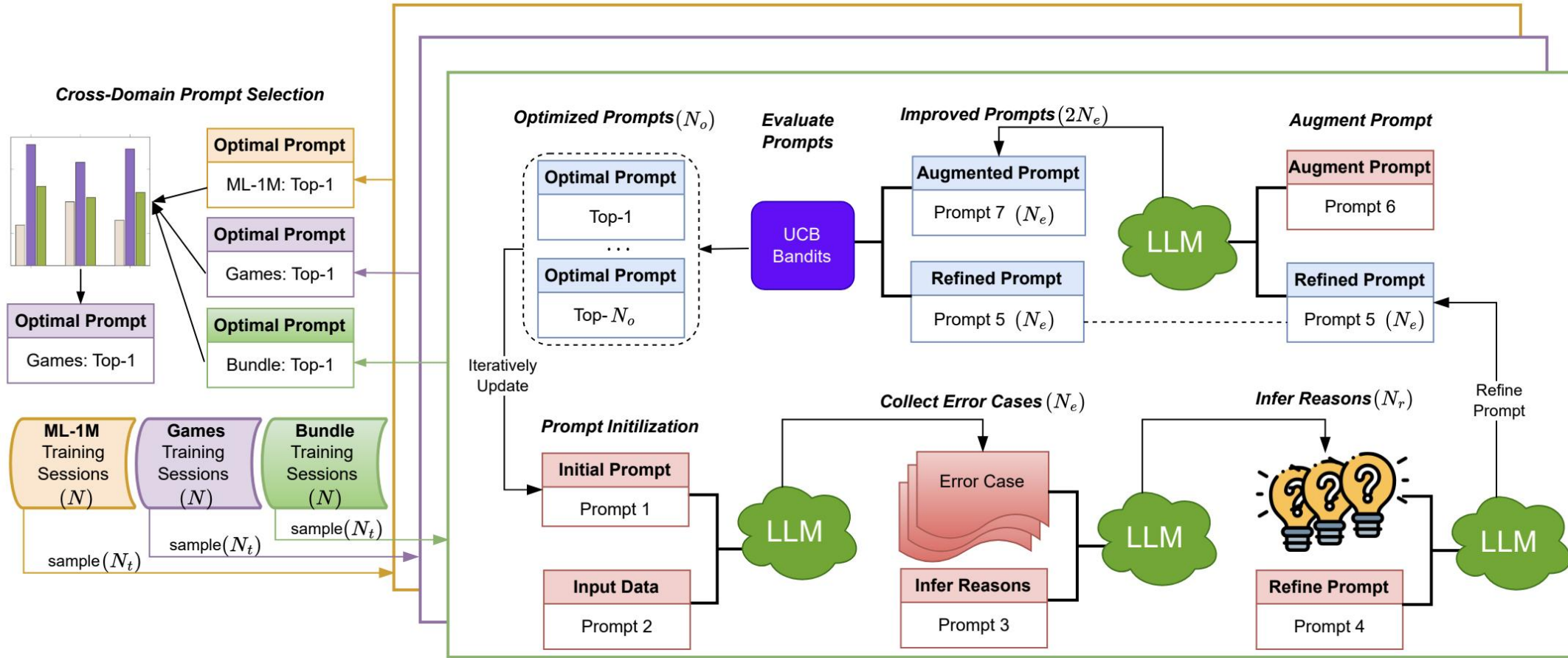
(b) User Profile

Provide the inquired information of the given movie.
[332] Heart and Souls (1993), Comedy|Fantasy
The inquired information is: director, country, language. And please output them in form of: director, country, language

Ron Underwood, USA, English

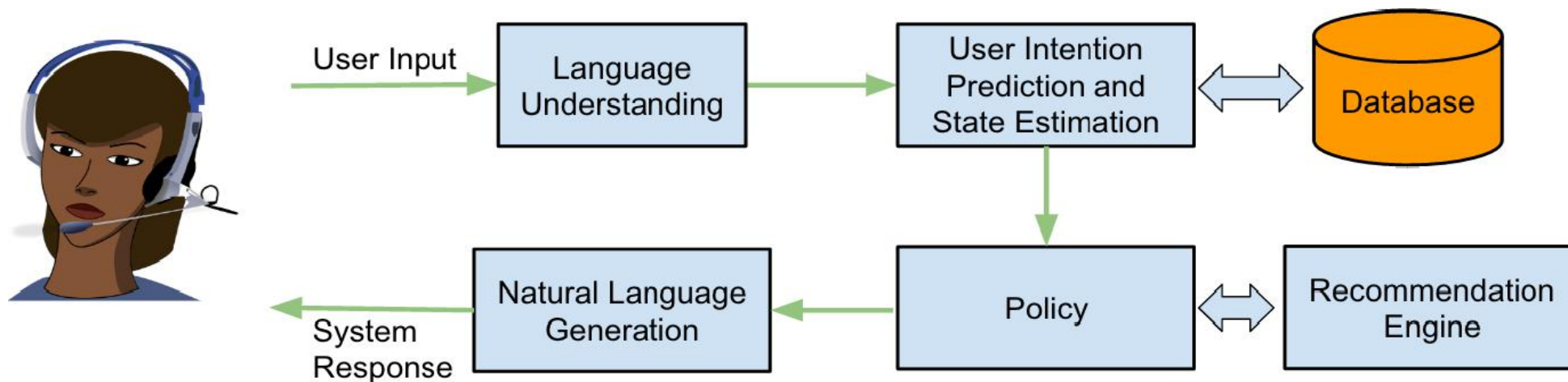
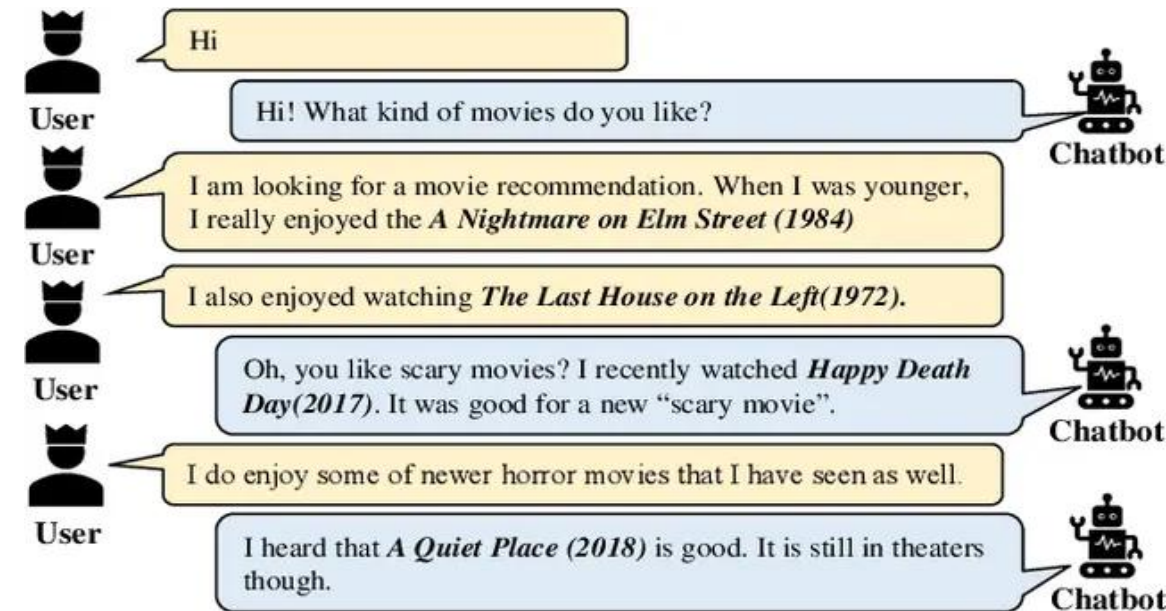
(c) Item Attribute

□ ICL: Automatically adjust and optimize prompts for recommendation

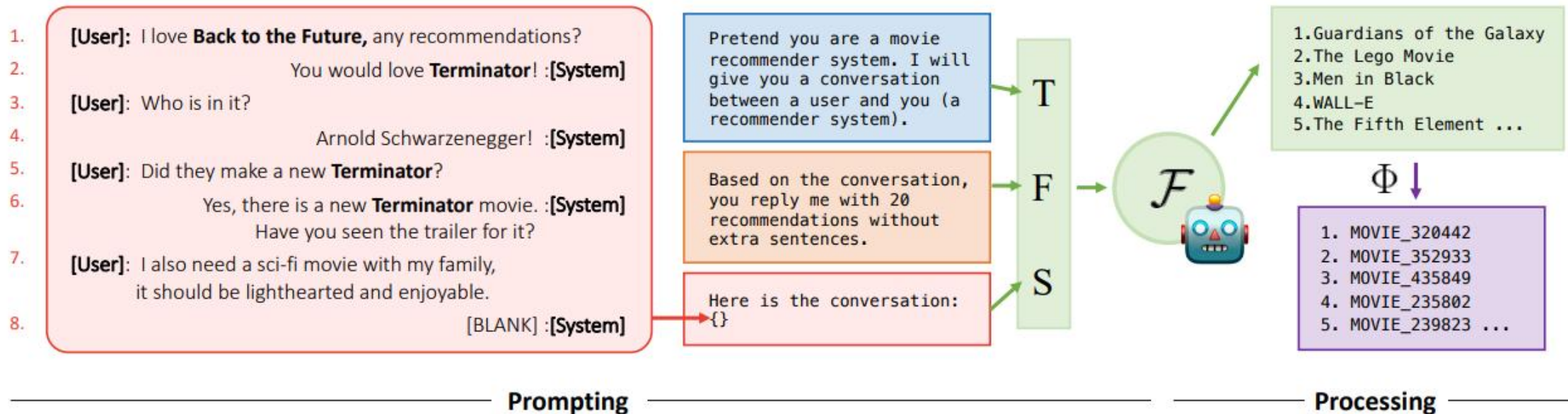


■ ICL for conversational recommender system

- Users chat with chatbot with natural language
- Chatbot analyses user interest
- Chatbot provide recommendaiton



■ Framework

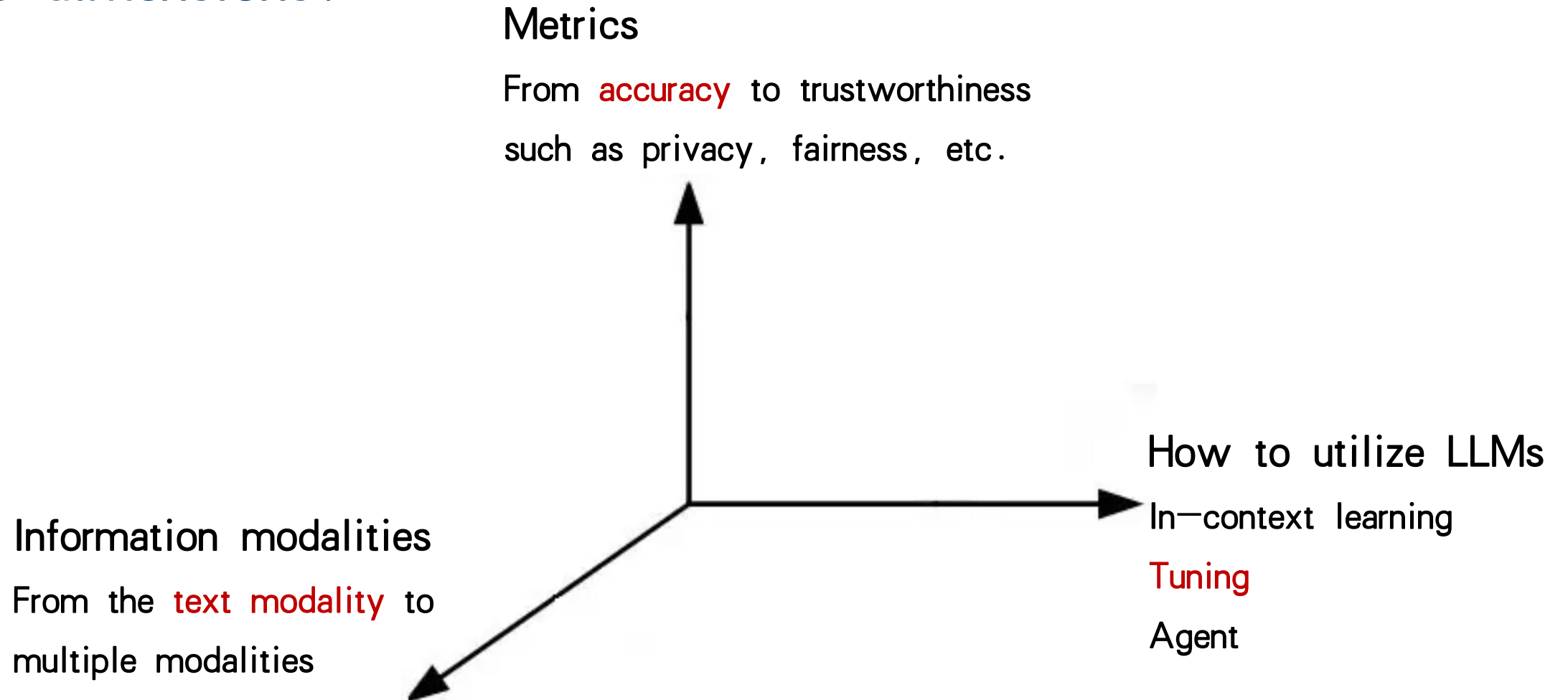


ICL for conversational recommender system

- Input: task description T , format requirement F and conversation context S
- LLMs analyse the input data
- LLMs generate the recommendation list

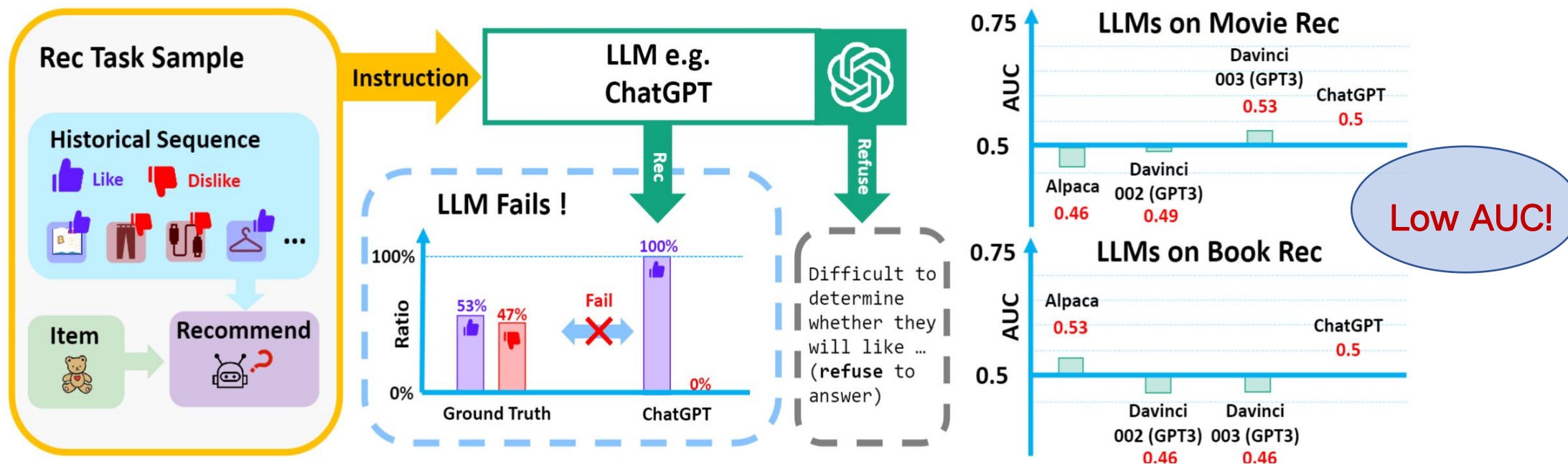
Progress of LLM4Rec

Three dimensions:



Tuning LLM4Rec

- ❑ In-context learning is not enough.
- ❑ In complex scenarios, ChatGPT usually gives **positive ratings** or **refuse to answer**.



Need to **align** LLM with recommendation task!

Tuning LLM4Rec

Motivation: lack of recommendation task tuning in LLM pre-training

→ tune LLMs with the recommendation data to align with the recommendation

Existing work on tuning LLMs for recommendation:

Discriminative manner

Following traditional rec task,
provide candidates:
pointwise, pairwise, listwise

PEFT tuning

TALLRec [1]

LLM-TRSR [5]

LLamaRec [4]

GLRec[8]

Full tuning

InstructRec[2]

LLMUnderPre[3]

.

Generative manner

Following the pretraining task,
do not provide candidates:
directly generate items

BigRec [6]

TransRec [7]

LC-Rec [10]

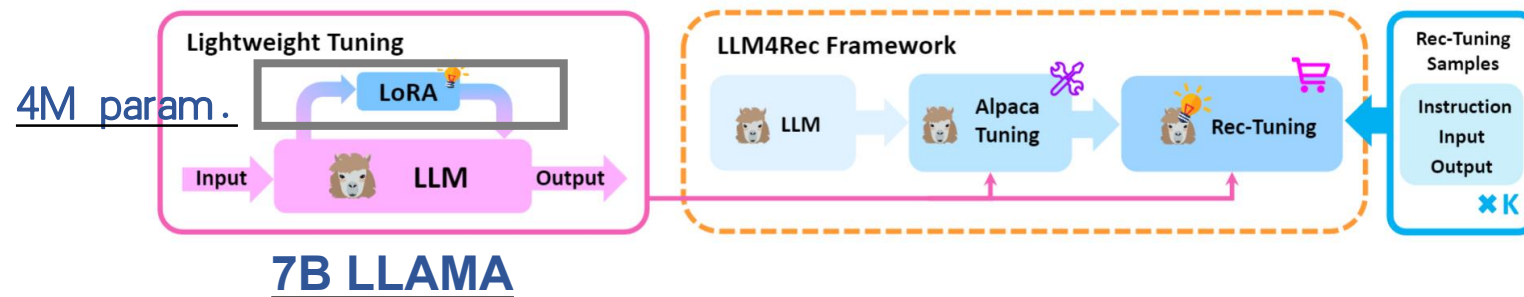
GIRL[9]

[1] Bao et al. Recsys, TALLRec: An Effective and Efficient Tuning Framework to Align Large Language Model with Recommendation. 2023
[2] Zhang et al. Recommendation as instruction following: a large language model empowered recommendation approach. 2023.
[3] Kang et al. Do LLMs Understand User Preferences? Evaluating LLMs on User Rating Prediction. 2023.
[4] Yue et al. LlamaRec: Two-Stage Recommendation using Large Language Models for Ranking. 2023.
[5] Zhi Zheng et al. Harnessing Large Language Models for Text-Rich Sequential Recommendation. 2024

[6] Bao et al. A Bi-step Grounding Paradigm for Large Language Models in Recommender system. 2023.
[7] Lin et al. A Multi-facet Paradigm to Bridge Large Language Model and Recommendation. 2023.
[8] Wu et al. Exploring Large Language Model for Graph Data Understanding in online Job Recommendation. 2023
[9] Zheng et al. Generative job recommendations with large language mode. 2023.
[10] Bowen Zheng et al. "Adapting Large Language Models by Integrating Collaborative Semantics for Recommendation" ICDE 2024.

Tuning LLM4Rec: TALLRec

TALLRec: Instruction-tuning



$$\max_{\Theta} \sum_{(x,y) \in \mathcal{Z}} \sum_{t=1}^{|y|} \log(P_{\Phi+\Theta}(y_t|x, y_{<t})),$$

- Fine-tune 4M parameters by few-shot samples via the generative loss
- Quickly adapt to new tasks

7B LLAMA

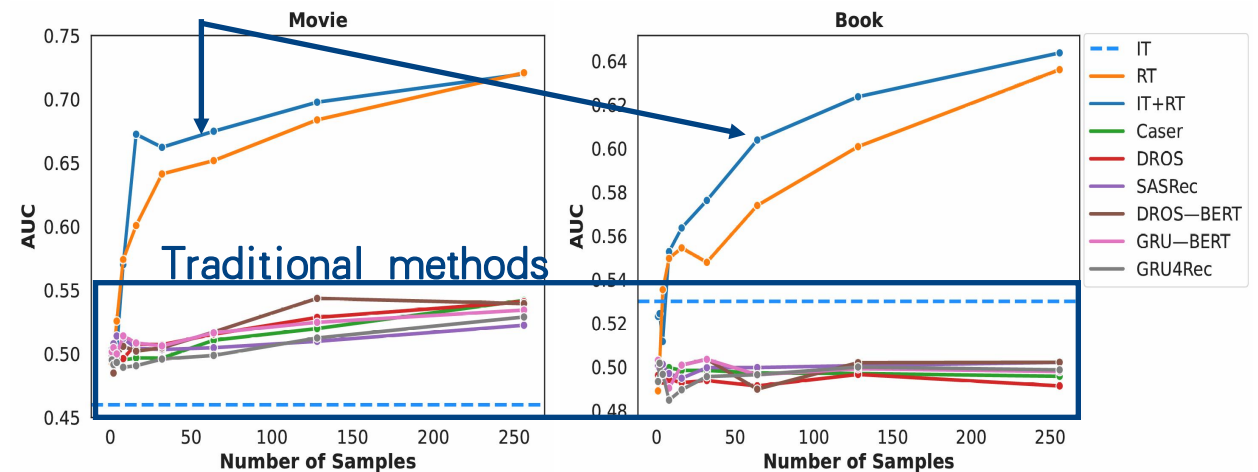
Like or not

LLM with LoRA

User features + item features

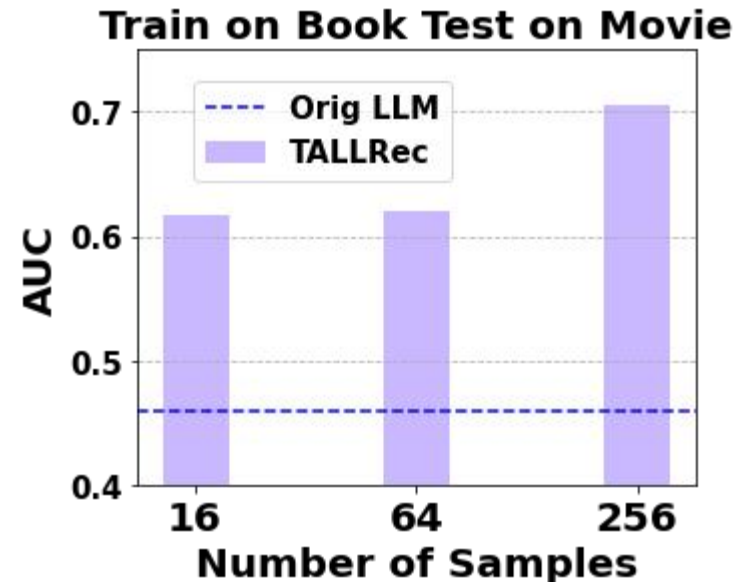
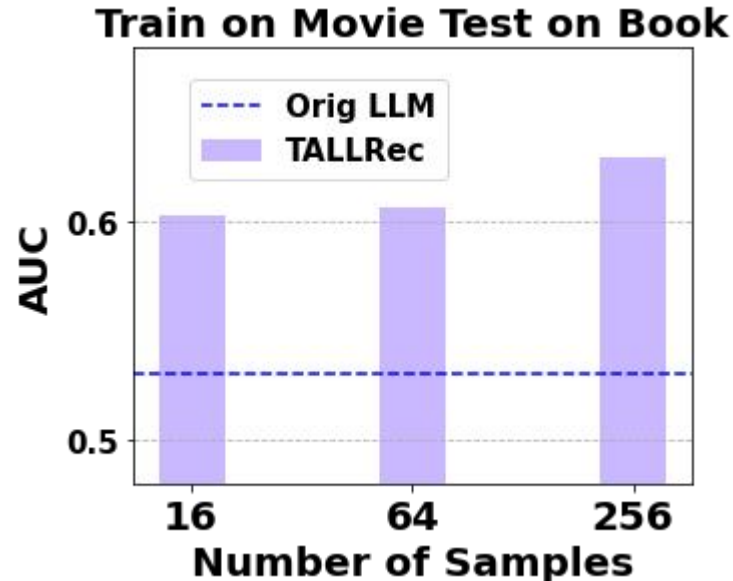
- Use item titles as the input
- Better for cold-start recommendation

Performance significantly improves by fine-tuning few-shot samples.



□ TALLRec: Cross-domain generalization

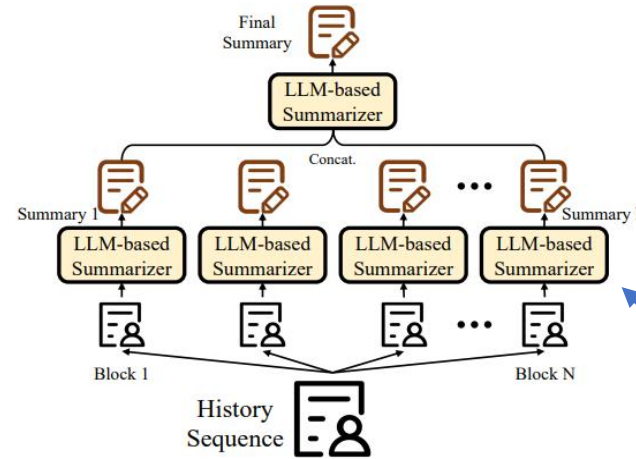
- Learning from movie scenario can directly recommend on books, and vice versa
- LLM can leverage domain knowledge to accomplish recommendation tasks after acquiring the ability to recommend.



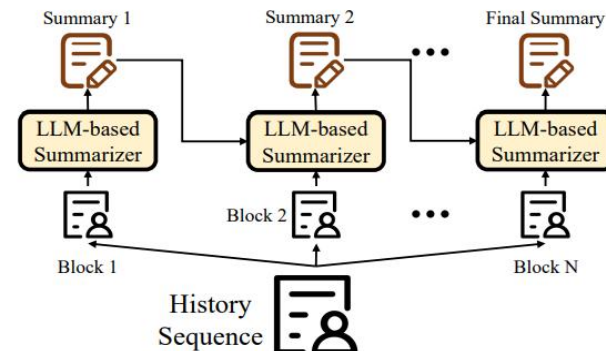
Tuning LLM4Rec: LLM-TRSR

□ Text-Rich Sequential Recommendation

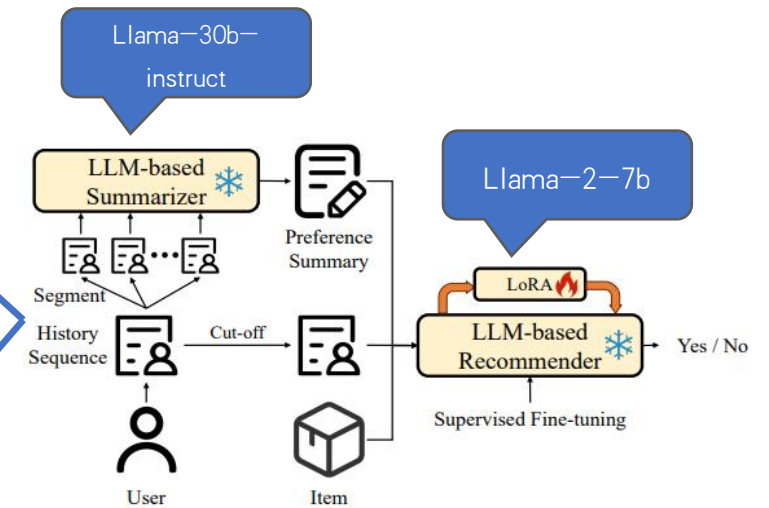
- LLM for preference summary
 - Hierarchical summarization
 - Recurrent summarization
- Supervised fine-tuning
 - Given user preference summary, recently interacted items, and candidate items, LLMs are tuned for recommendation



Hierarchical summarization



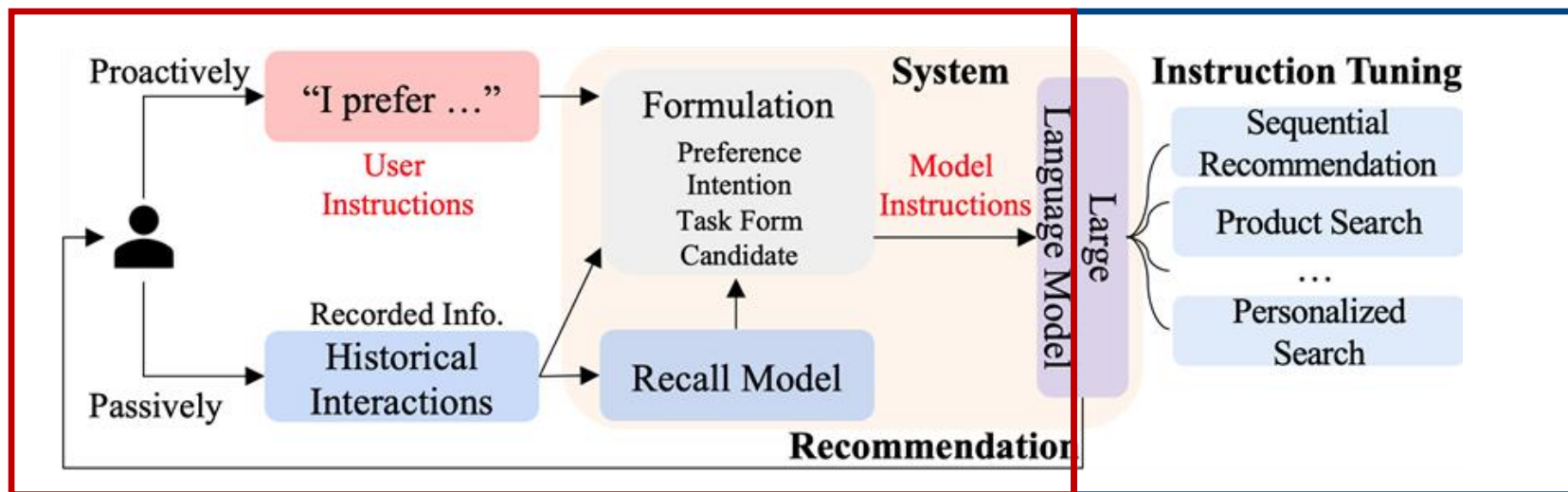
Recurrent summarization



Tuning LLM4Rec: InstructRec

□ InstructRec

- User could express their needs diversely: vague or specific; implicit or explicit
- LLM should understand and follow different instructions for recommendation



Recommendation instruction
construction

Instruction tuning:
tuning LLMs with the instruction data

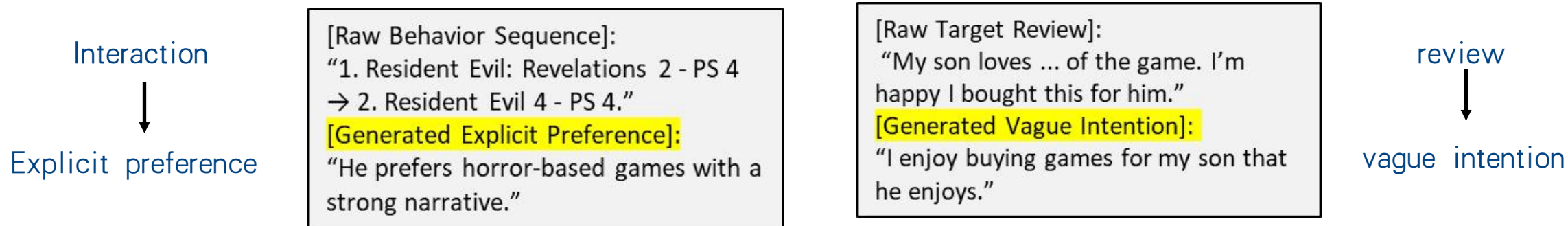
Tuning LLM4Rec: InstructRec

□ InstructRec: Instruction construction:

- **Format:** Preference: none/implicit/explicit Intention: none/vague/specific task: pointwise/pairwise/listwise

| Instantiation | Model Instructions |
|---------------------------------|--|
| $\langle P_1, I_0, T_0 \rangle$ | The user has purchased these items: <historical interactions> . Based on this information, is it likely that the user will interact with <target item> next? |
| $\langle P_2, I_0, T_3 \rangle$ | You are a search engine and you meet a user's query: <explicit preference> . Please respond to this user by selecting items from the candidates: <candidate items> . |
| $\langle P_0, I_1, T_2 \rangle$ | As a recommender system, your task is to recommend an item that is related to the user's <vague intention> . Please provide your recommendation. |
| $\langle P_0, I_2, T_2 \rangle$ | Suppose you are a search engine, now the user search that <specific intention> , can you generate the item to respond to user's query? |
| $\langle P_1, P_2, T_2 \rangle$ | Here is the historical interactions of a user: <historical interactions> . His preferences are as follows: <explicit preference> . Please provide recommendations. |
| $\langle P_1, I_1, T_2 \rangle$ | The user has interacted with the following <historical interactions> . Now the user search for <vague intention> , please generate products that match his intent. |
| $\langle P_1, I_2, T_2 \rangle$ | The user has recently purchased the following <historical items> . The user has expressed a desire for <specific intention> . Please provide recommendations. |

- **Instruction generation: #1** using ChatGPT to generate user preferences and intentions based on interactions



#2 Increasing the instruction diversity via multiple strategies such as CoT

InstructRec

- Instruction construction
 - Quality: human evaluation

| Statistic | |
|--|---------|
| # of fine-grained instructions | 252,730 |
| - # of user-described preferences | 151,638 |
| - # of user intention in decision making | 101,092 |
| ave. instruction length (in words) | 23.5 |
| # of coarse-grained instructions | 39 |
| - # of preferences related instructions | 17 |
| - # of intentions related instructions | 9 |
| - # of combined instructions | 13 |
| ave. instruction length (in words) | 41.4 |

| Quality Review Question | Preference | Intention |
|---|------------|-----------|
| Is the instruction generated from the user's related information? | 93% | 90% |
| Does the teacher-LLM provide related world knowledge? | 87% | 22% |
| Does the instruction reflect the user's preference/ intention? | 88% | 69% |
| Is the instruction related to target item? | 48% | 69% |

- Instruction tuning:
 - Supervised fine-tuning, tuning all model parameters (3B Flan-T5-XL)

$$\mathcal{L} = \sum_{k=1}^B \sum_{j=1}^{|Y_k|} \log P(Y_{k,j} | Y_{k,<j}, I_k), \quad (1)$$

where Y_k is the desired system responses for the k -th instance, I_k is the instruction of the k -th instance, and B is the batch size.

Tuning LLM4Rec

Motivation: lack of recommendation task tuning in LLM pre-training

→ tune LLMs with the recommendation data to align with the recommendation

Existing work on tuning LLMs for recommendation:

Discriminative manner

Following traditional rec task,
provide candidates:
pointwise, pairwise, listwise

PEFT tuning

TALLRec [1]

LLM-TRSR [5]

LLamaRec [4]

GLRec[8]

Full tuning

InstructRec[2]

LLMUnderPre[3]

.

Generative manner

Following the pretraining task,
do not provide candidates:
directly generate items

BigRec [6]

TransRec [7]

LC-Rec [10]

GIRL[9]

[1] Bao et al. Recsys, TALLRec: An Effective and Efficient Tuning Framework to Align Large Language Model with Recommendation. 2023.
[2] Zhang et al. Recommendation as instruction following: a large language model empowered recommendation approach. 2023.
[3] Kang et al. Do LLMs Understand User Preferences? Evaluating LLMs on User Rating Prediction. 2023.
[4] Yue et al. LlamaRec: Two-Stage Recommendation using Large Language Models for Ranking. 2023.
[5] Zhi Zheng et al. Harnessing Large Language Models for Text-Rich Sequential Recommendation. 2024

[6] Bao et al. A Bi-step Grounding Paradigm for Large Language Models in Recommender system. 2023.
[7] Lin et al. A Multi-facet Paradigm to Bridge Large Language Model and Recommendation. 2023.
[8] Wu et al. Exploring Large Language Model for Graph Data Understanding in online Job Recommendation. 2023
[9] Zheng et al. Generative job recommendations with large language mode. 2023.
[10] Bowen Zheng et al. "Adapting Large Language Models by Integrating Collaborative Semantics for Recommendation" ICDE 2024.

Tuning LLM4Rec: BIGRec

□ BIGRec

• Generation + Grounding

- Given user interaction history in natural language, LLMs aim to generate the next item as recommendation.
- However, LLMs do not know how to represent an item via token sequence in the recommendation scenario.
- Besides, the item generated by the LLM may not exist in **the actual world**.

Grounding Paradigm

Language Space



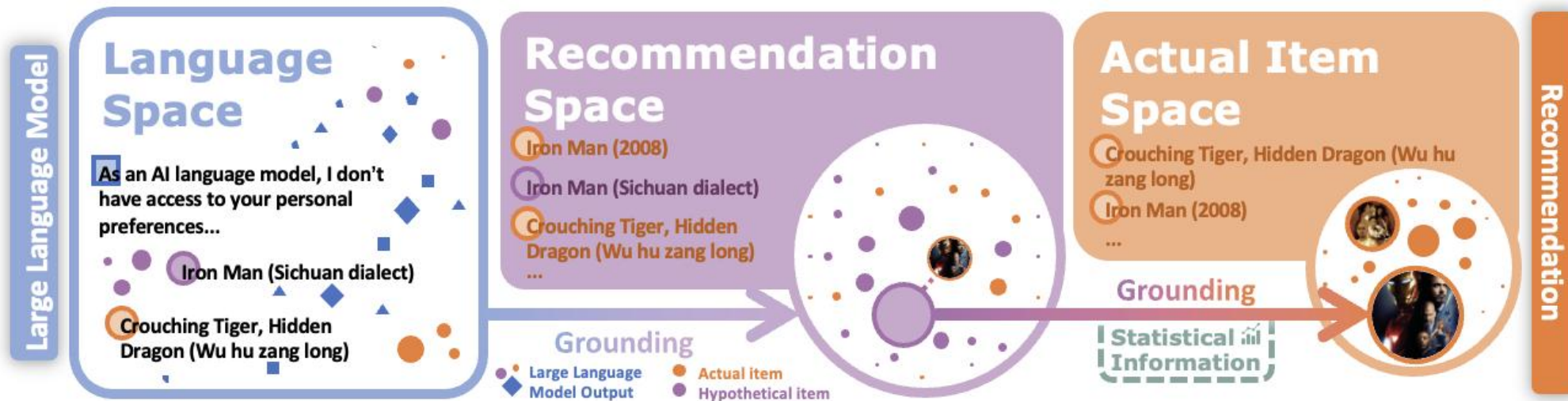
Step1: instruction tuning

Recommendation Space



Step2: L2 distance grounding

Actual Item Space



Tuning LLM4Rec: BIGRec



□ BIGRec

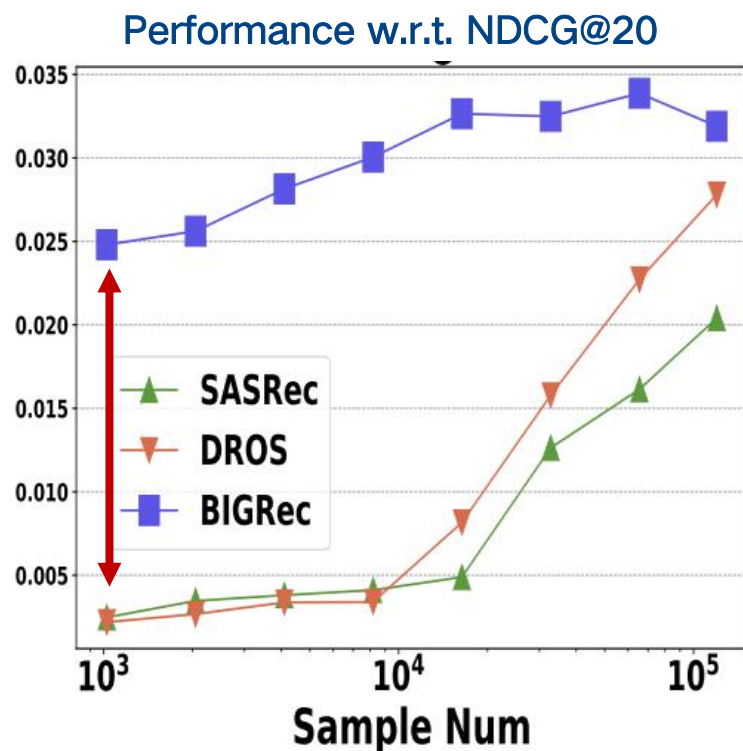
- Few-shot tuning

| Dataset | Model | NG@1 | NG@3 | NG@5 | NG@10 | NG@20 | HR@1 | HR@3 | HR@5 | HR@10 | HR@20 |
|---------|----------------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| Movie | GRU4Rec | 0.0015 | 0.0034 | 0.0047 | 0.0070 | 0.0104 | 0.0015 | 0.0047 | 0.0079 | 0.0147 | 0.0281 |
| | Caser | 0.0020 | 0.0035 | 0.0052 | 0.0078 | 0.0109 | 0.0020 | 0.0046 | 0.0088 | 0.0171 | 0.0293 |
| | SASRec | 0.0023 | 0.0051 | 0.0062 | 0.0082 | 0.0117 | 0.0023 | 0.0070 | 0.0097 | 0.0161 | 0.0301 |
| | P5 | 0.0014 | 0.0026 | 0.0036 | 0.0051 | 0.0069 | 0.0014 | 0.0035 | 0.0059 | 0.0107 | 0.0176 |
| | DROS | 0.0022 | 0.0040 | 0.0052 | 0.0081 | 0.0112 | 0.0022 | 0.0051 | 0.0081 | 0.0173 | 0.0297 |
| | GPT4Rec-LLaMA | 0.0016 | 0.0022 | 0.0024 | 0.0028 | 0.0035 | 0.0016 | 0.0026 | 0.0030 | 0.0044 | 0.0074 |
| | BIGRec (1024) | 0.0176 | 0.0214 | 0.0230 | 0.0257 | 0.0283 | 0.0176 | 0.0241 | 0.0281 | 0.0366 | 0.0471 |
| | Improve | 654.29% | 323.31% | 273.70% | 213.71% | 142.55% | 654.29% | 244.71% | 188.39% | 111.97% | 56.55% |
| Game | GRU4Rec | 0.0013 | 0.0016 | 0.0018 | 0.0024 | 0.0030 | 0.0013 | 0.0018 | 0.0024 | 0.0041 | 0.0069 |
| | Caser | 0.0007 | 0.0012 | 0.0019 | 0.0024 | 0.0035 | 0.0007 | 0.0016 | 0.0032 | 0.0048 | 0.0092 |
| | SASRec | 0.0009 | 0.0012 | 0.0015 | 0.0020 | 0.0025 | 0.0009 | 0.0015 | 0.0021 | 0.0037 | 0.0057 |
| | P5 | 0.0002 | 0.0005 | 0.0007 | 0.0010 | 0.0017 | 0.0002 | 0.0007 | 0.0012 | 0.0023 | 0.0049 |
| | DROS | 0.0006 | 0.0011 | 0.0013 | 0.0016 | 0.0022 | 0.0006 | 0.0015 | 0.0019 | 0.0027 | 0.0052 |
| | GPT4Rec-LLaMA | 0.0000 | 0.0000 | 0.0000 | 0.0001 | 0.0001 | 0.0000 | 0.0000 | 0.0000 | 0.0002 | 0.0002 |
| | BIGRec (1024) | 0.0133 | 0.0169 | 0.0189 | 0.0216 | 0.0248 | 0.0133 | 0.0195 | 0.0243 | 0.0329 | 0.0457 |
| | Improve | 952.63% | 976.26% | 888.19% | 799.64% | 613.76% | 952.63% | 985.19% | 660.42% | 586.11% | 397.10% |

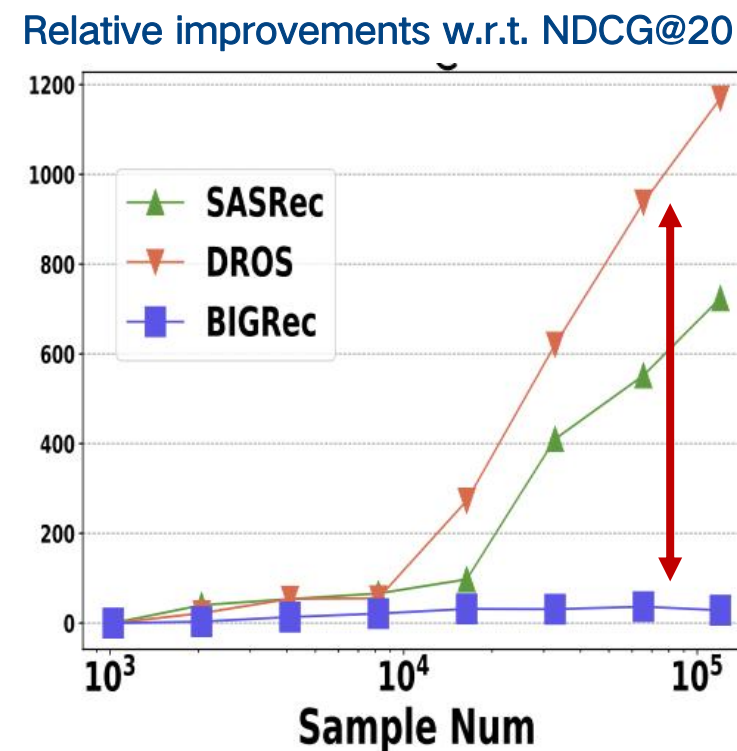
- BIGRec significantly surpasses baselines by few-shot tuning.
- Improvement of BIGRec is significantly higher on Game compared to on Movie.
 - possibly due to the varying properties of popularity bias between the two datasets.

Tuning LLM4Rec: BIGRec

□ BIGRec



Quickly adapt to recommendation!

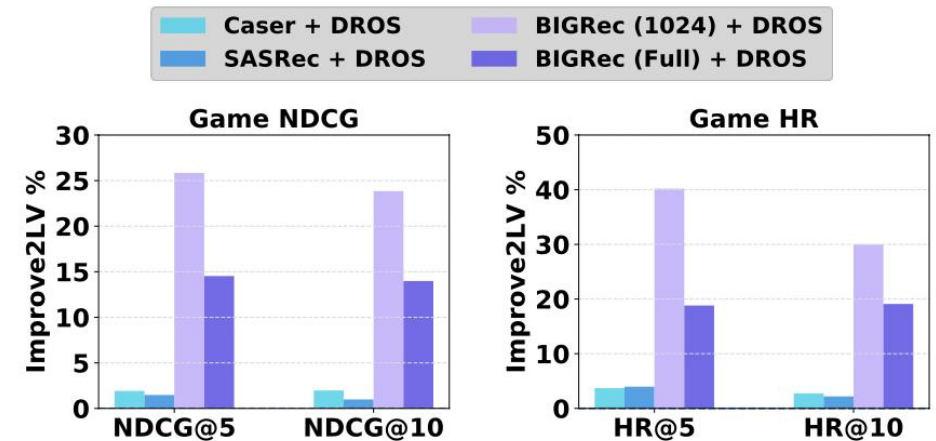
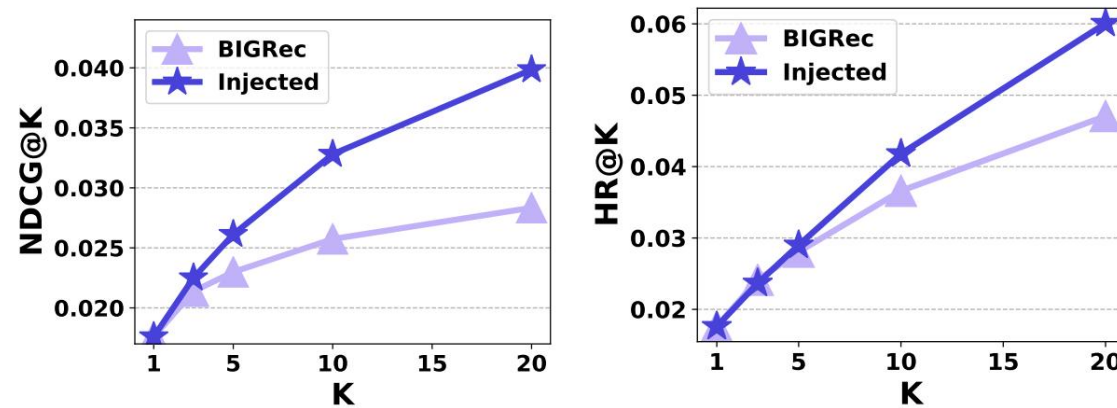


Not proficient in utilizing collaborative filtering signals in interactions!

Tuning LLM4Rec: BIGRec

□ BIGRec

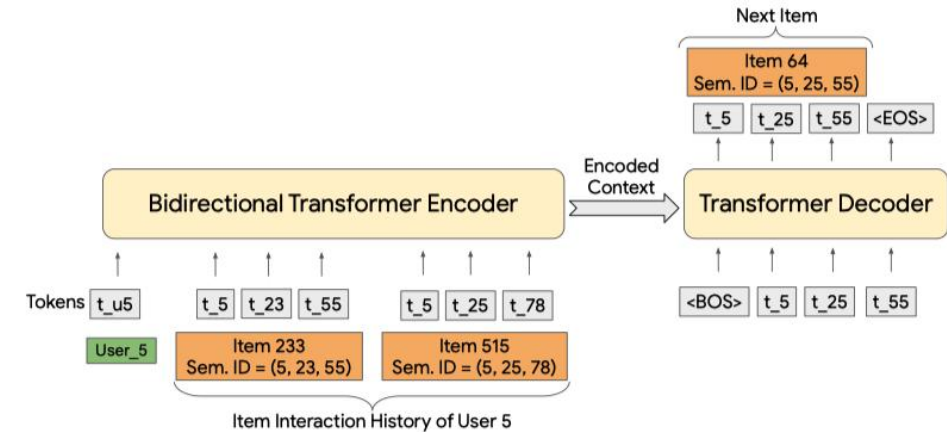
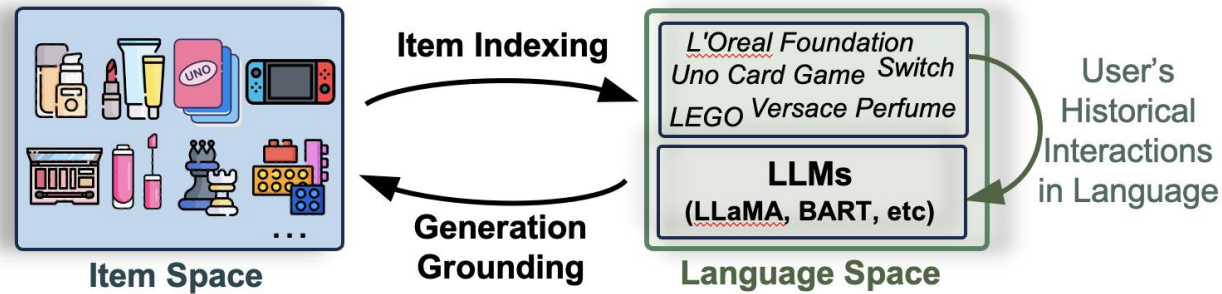
- Injecting statistical information into BIGRec at Step 2: L2 distance grounding



- By incorporating popularity, BIGRec achieves significant improvements *w.r.t.* NDCG@ K and HR@ K , particularly for a larger K .
- Incorporating collaborative information into BIGRec yields more significant enhancements than conventional models.

Tuning LLM4Rec

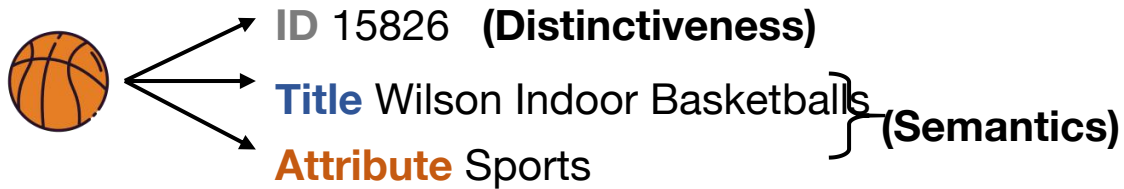
□ LLM for generative recommendation



- Two key problems of LLM4Rec
 - Item tokenization: index items into language space
 - Item generation: generate items as recommendations

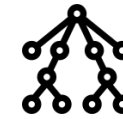
Tuning LLM4Rec: TransRec

- Item indexing: multi-facet identifier



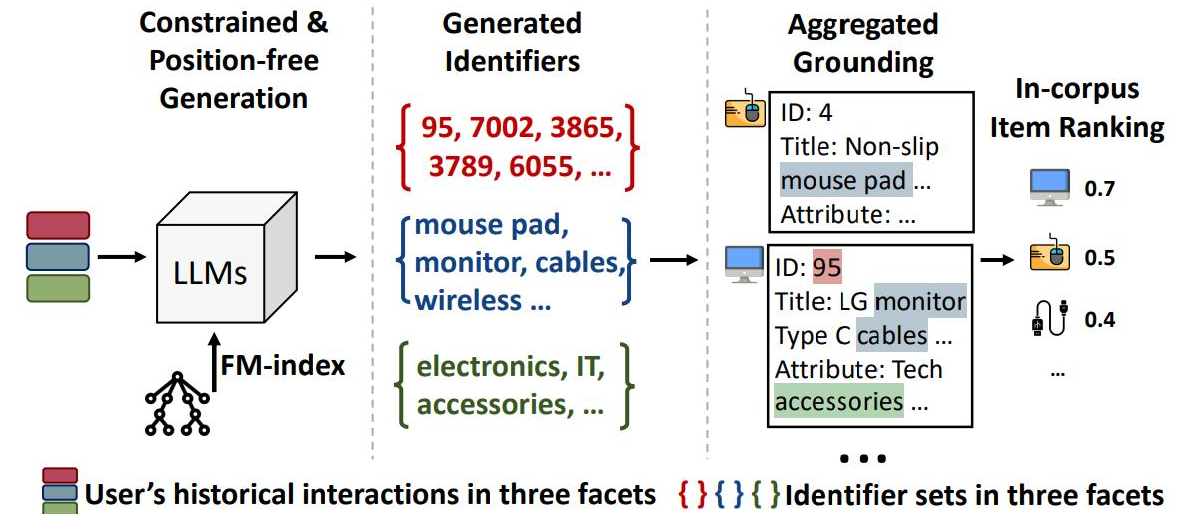
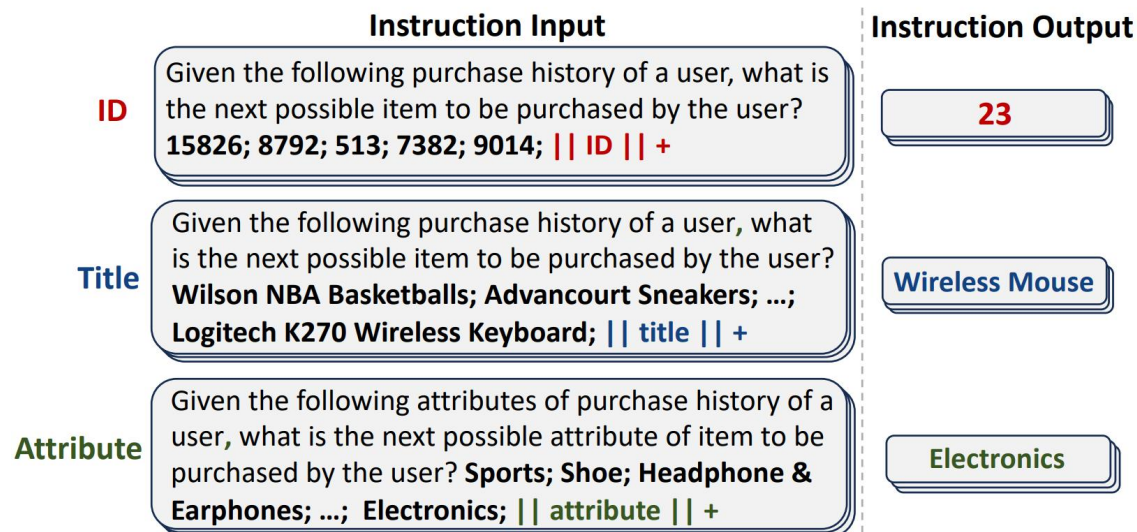
- Generation grounding:

- Position-free constrained generation



FM-index: special prefix tree that supports search from any position of the identifier corpus.

- Instruction data reconstruction



Tuning LLM4Rec: TransRec

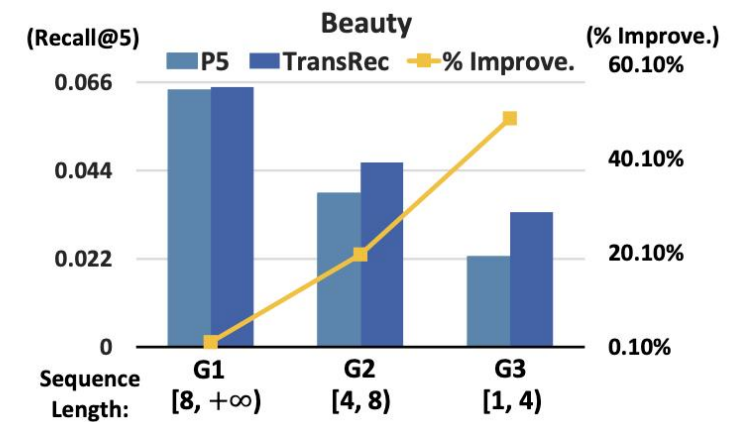
□ Strong generalization ability

- Item group analysis
 - From warm to cold items

| N-shot | Model | Warm | | Cold | |
|--------|-------------------|---------------|---------------|---------------|---------------|
| | | R@5 | N@5 | R@5 | N@5 |
| 1024 | LightGCN | 0.0205 | 0.0125 | 0.0005 | 0.0003 |
| | ACVAE | 0.0098 | 0.0057 | 0.0047 | 0.0026 |
| | P5 | 0.0040 | 0.0016 | 0.0025 | 0.0015 |
| | TransRec-B | 0.0039 | 0.0024 | 0.0025 | 0.0016 |
| | TransRec-L | 0.0141 | 0.0070 | 0.0159 | 0.0097 |
| 2048 | LightGCN | 0.0186 | 0.0117 | 0.0005 | 0.0004 |
| | ACVAE | 0.0229 | 0.0136 | 0.0074 | 0.0044 |
| | P5 | 0.0047 | 0.0030 | 0.0036 | 0.0012 |
| | TransRec-B | 0.0052 | 0.0027 | 0.0039 | 0.0017 |
| | TransRec-L | 0.0194 | 0.0126 | 0.0206 | 0.0126 |

* The bold results highlight the superior performance compared to the best LLM-based recommender baseline.

- User group analysis
 - From dense users to sparse users

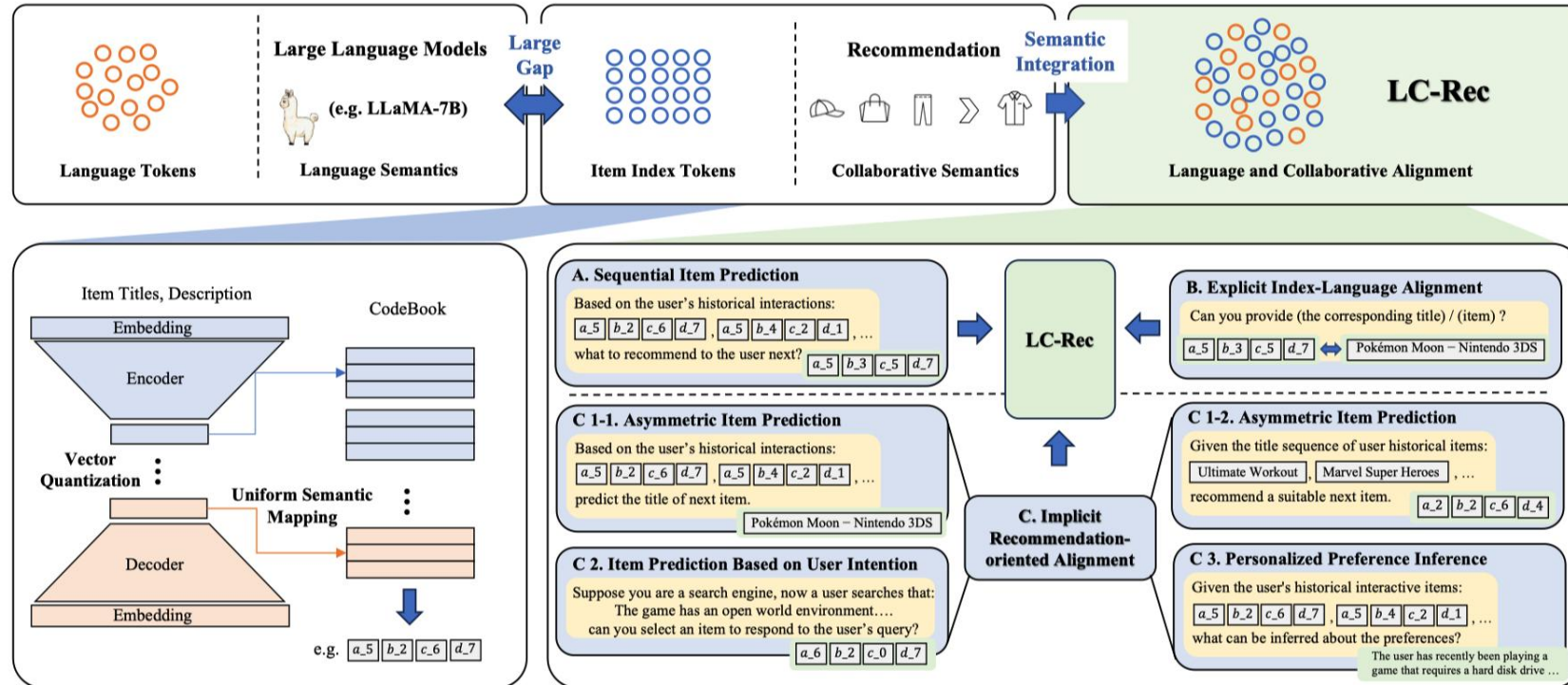


- On the item side, TransRec-L with LLMs has remarkable generalization ability with vase knowledge base, especially on cold-start recommendation under limited data.
- On the user side, TransRec significantly **improves the performance of sparse users** with fewer interactions.

Tuning LLM4Rec: LC-Rec

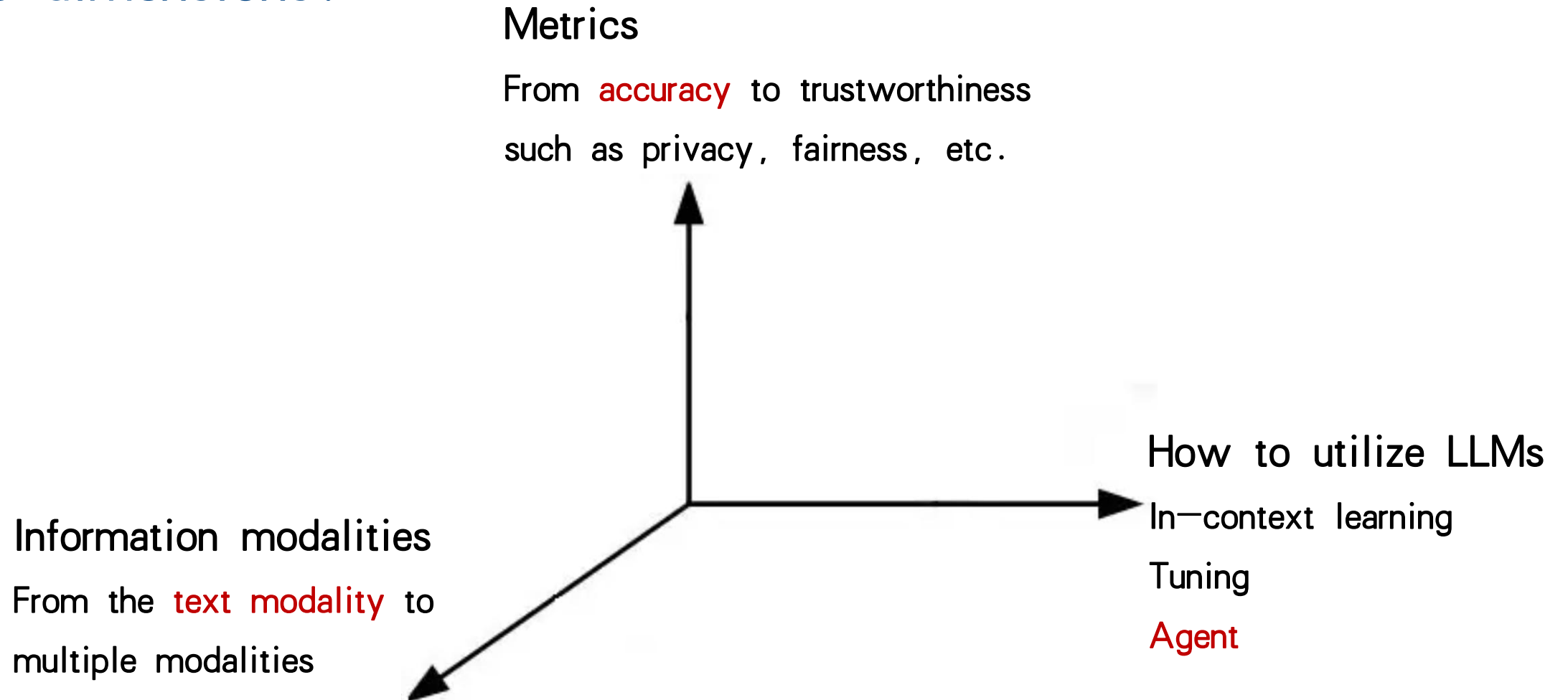
• LC-Rec

- Item indexing: utilize Residual-Quantized Variational AutoEncoder (RQ-VAE) to encode item semantic information as identifiers.
- Multiple alignment tasks to inject collaborative signals



Progress of LLM4Rec

Three dimensions:



❑ LLM-empowered Agents for Recommendation

❑ Agent as User Simulator

- **Main idea:** using agents to simulate user behavior for real-world recommendation.
- RecAgent^[1], Agent4Rec^[2]

❑ Agent for Recommendation

- **Main idea:** harnessing the powerful capabilities of LLMs, such as reasoning, reflection, planning and tool usage, for recommendation.
- RecMind^[3], InteRecAgent^[4], BiLLP^[5], Multi-Agent Collaboration^[6]

[1] Lei Wang et al. "When Large Language Model based Agent Meets User Behavior Analysis: A Novel User Simulation Paradigm" arXiv 2023.

[2] Zhang An et al. "On Generative Agents in Recommendation" arXiv 2023.

[3] Wang Yancheng et al. "RecMind: Large Language Model Powered Agent For Recommendation" arXiv 2023.

[4] Xu Huang et al. "Recommender AI Agent: Integrating Large Language Models for Interactive Recommendations" arxiv 2023.

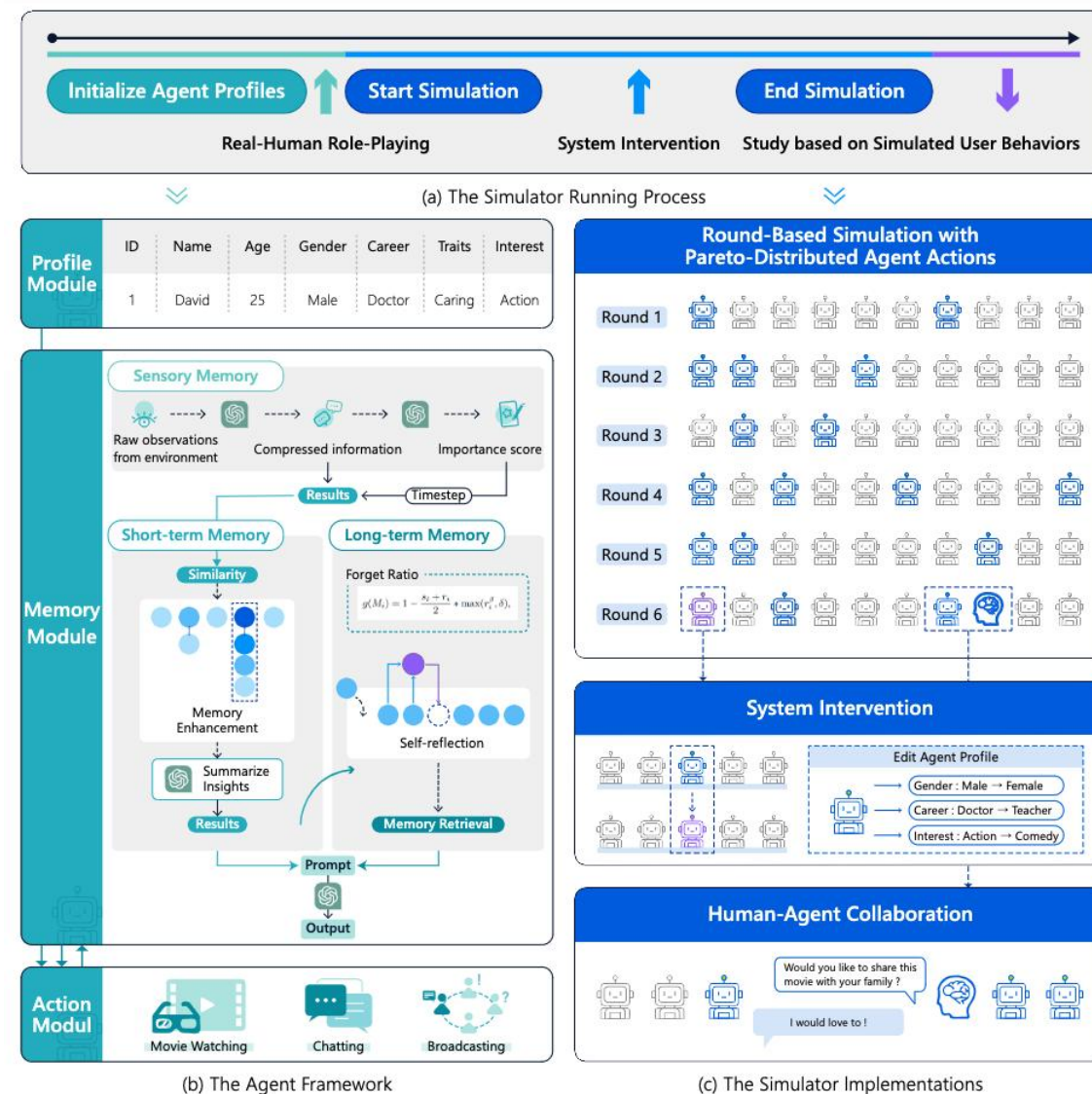
[5] Wentao Shi et al. 2023. Large Language Models are Learnable Planners for Long-Term Recommendation. in SIGIR 2024.

[6] Jiabao Fang et al. A Multi-Agent Conversational Recommender System. Arxiv 2024

Agent: RecAgent

□ LLM-based agent for user simulation

- User simulation is a fundamental problem in human-centered applications.
- Traditional methods **struggle to simulate** complex user behaviors.
- LLMs show potential in human-level intelligence and generalization capabilities.



Agent: RecAgent

❑ Recommendation Behaviors

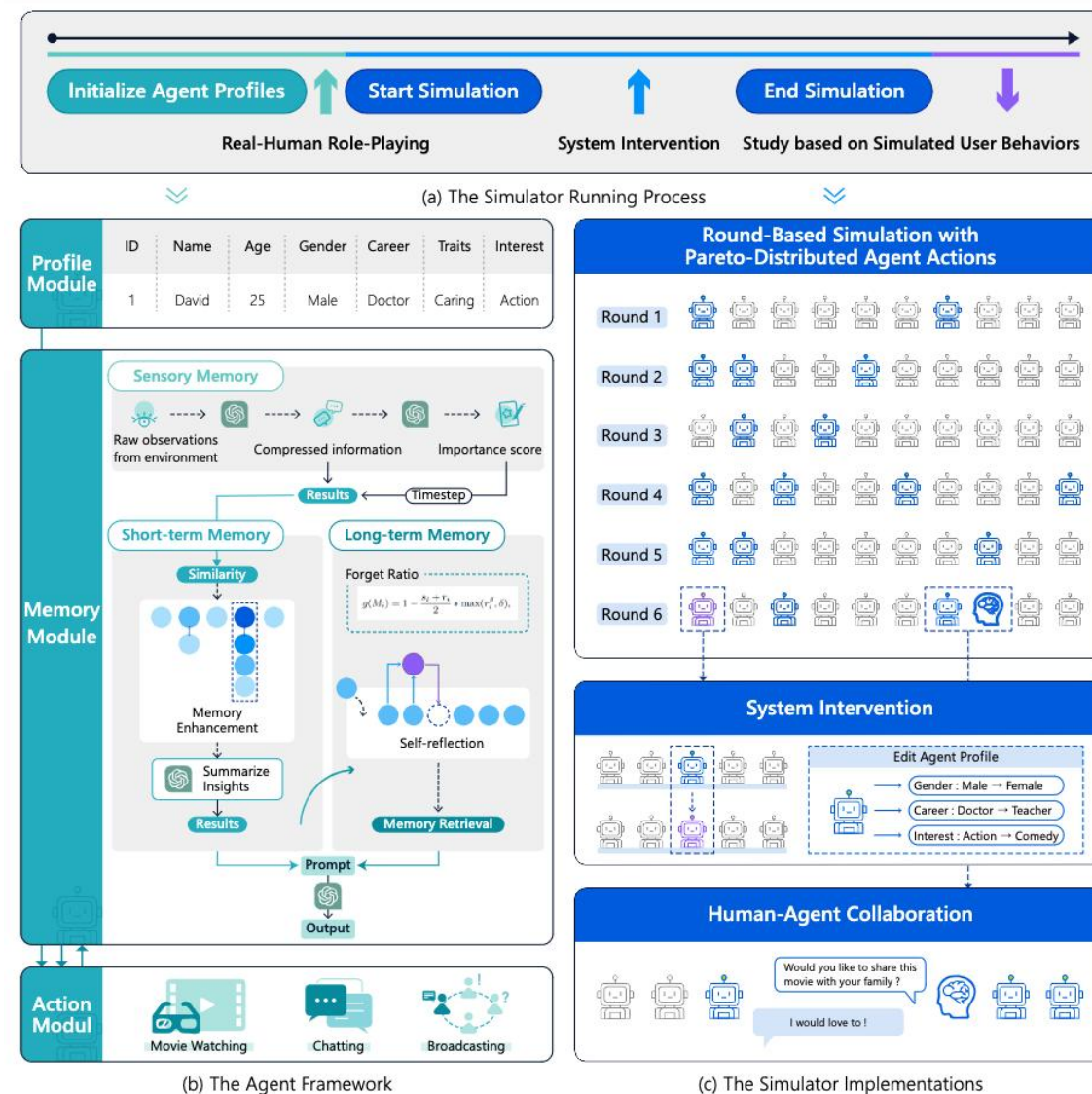
Agent chooses to **search or receive recommendations**, selects movies, and **stores** feelings after watching.

❑ Chatting Behaviors

Two agents **discuss and stored** the conversation in their memories.

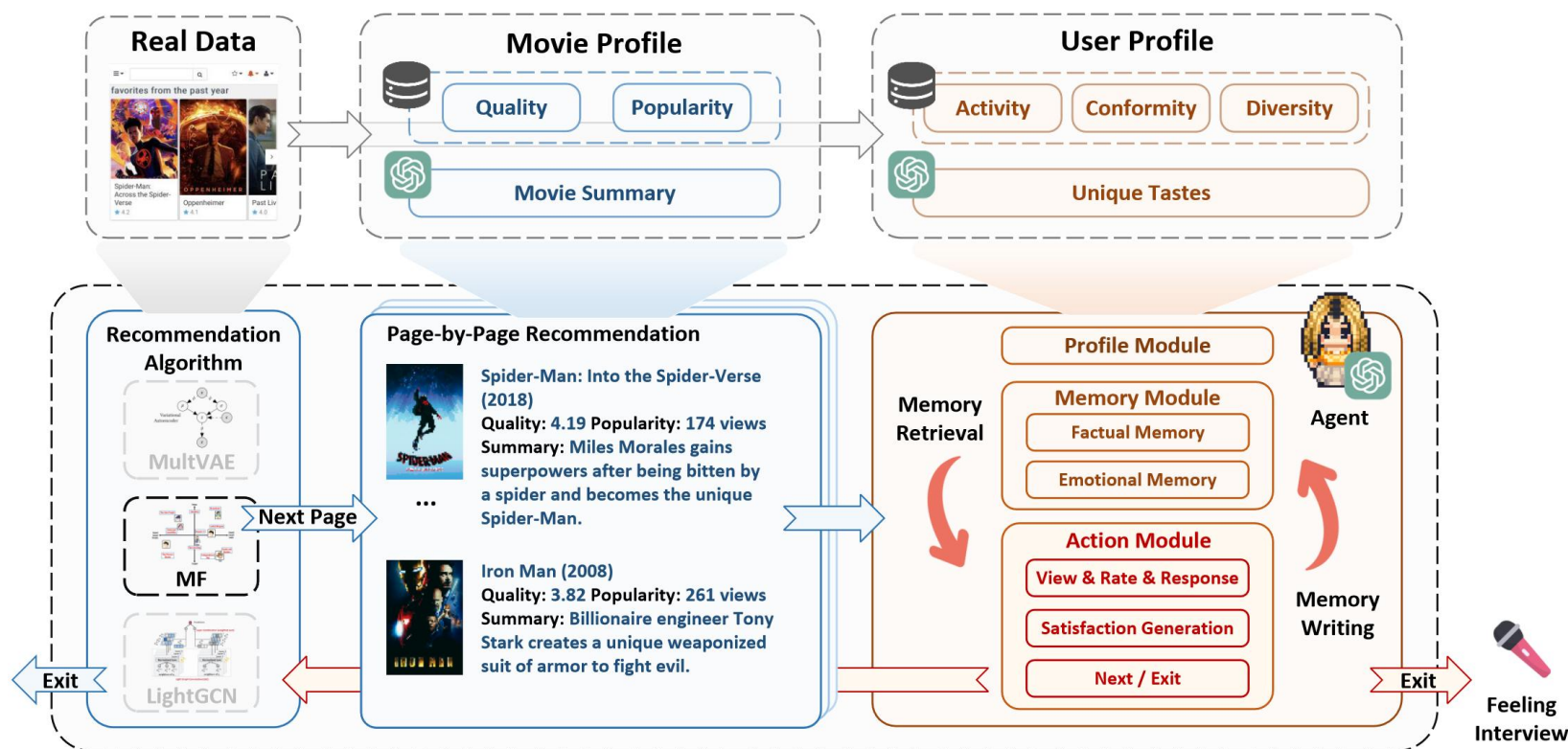
❑ Broadcasting Behaviors

An agent **posts** a message on social media, **received by friends** and stored in their memories.



Agent: Agent4Rec

- ❑ Agent4Rec, a simulator with 1,000 LLM-empowered generative agents.
- ❑ Agents are trained by the MovieLens-1M dataset, embodying varied social traits and preferences.
- ❑ Each agent interacts with personalized movie recommendations in a page-by-page manner and undertakes various actions such as watching, rating, evaluating, exiting, and interviewing.



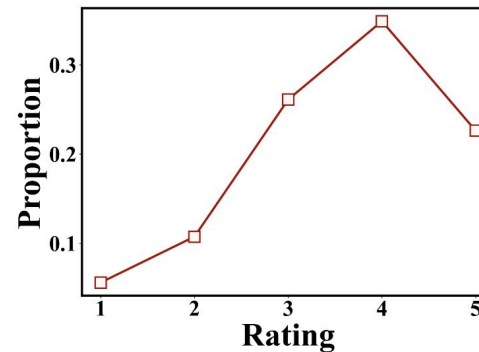
❑ To what extent can LLM-empowered generative agents truly simulate the behavior of genuine, independent humans in recommender systems?

❑ User Taste Alignment

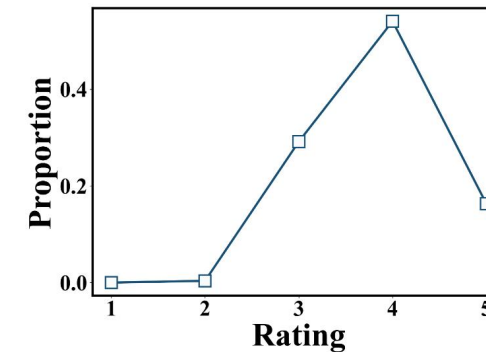
Table 1: User taste discrimination.

| 1:m | Accuracy | Recall | Precision | F1 Score |
|-----|----------------|----------------|----------------|----------------|
| 1:1 | 0.6912* | 0.7460 | 0.6914* | 0.6982* |
| 1:2 | 0.6466 | 0.7602 | 0.5058 | 0.5874 |
| 1:3 | 0.6675 | 0.7623 | 0.4562 | 0.5433 |
| 1:9 | 0.6175 | 0.7753* | 0.2139 | 0.3232 |

❑ Rating Distribution Alignment



(a) Distribution on MovieLens



(b) Agent-simulated distribution

❑ LLM-empowered Agents for Recommendation

❑ Agent as User Simulator

- **Main idea:** using agents to simulate user behavior for real-world recommendation.
- RecAgent^[1], Agent4Rec^[2]

❑ Agent for Recommendation

- **Main idea:** harnessing the powerful capabilities of LLMs, such as reasoning, reflection, planning and tool usage, for recommendation.
- RecMind^[3], InteRecAgent^[4], BiLLP^[5], Multi-Agent Collaboration^[6]

[1] Lei Wang et al. "When Large Language Model based Agent Meets User Behavior Analysis: A Novel User Simulation Paradigm" arXiv 2023.

[2] Zhang An et al. "On Generative Agents in Recommendation" arXiv 2023.

[3] Wang Yancheng et al. "RecMind: Large Language Model Powered Agent For Recommendation" arXiv 2023.

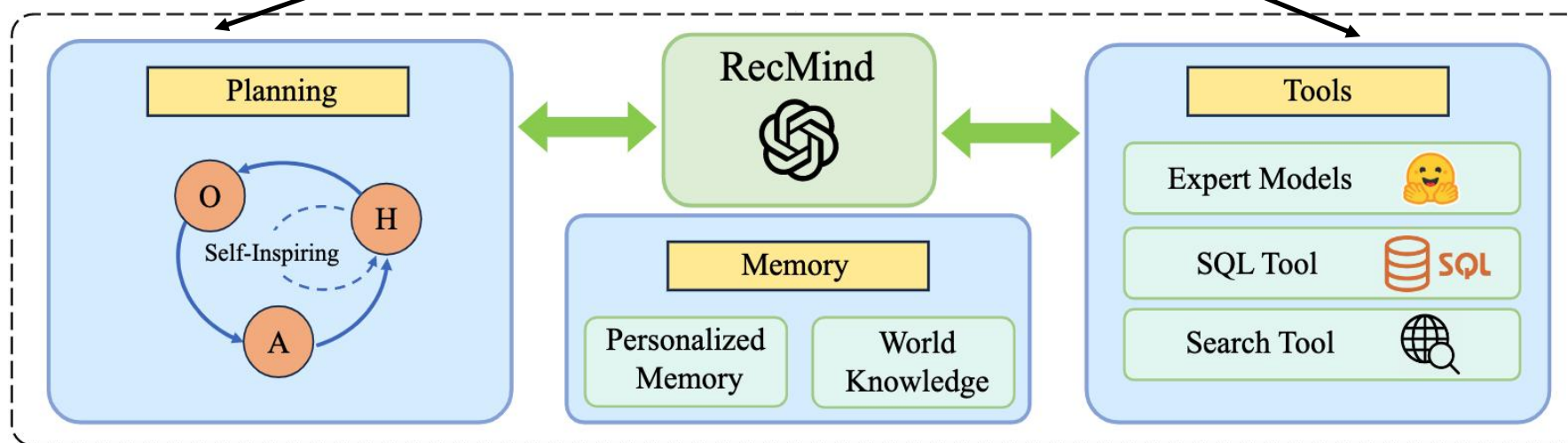
[4] Xu Huang et al. "Recommender AI Agent: Integrating Large Language Models for Interactive Recommendations" arxiv 2023.

[5] Wentao Shi et al. 2023. Large Language Models are Learnable Planners for Long-Term Recommendation. in SIGIR 2024.

[6] Jiabao Fang et al. A Multi-Agent Conversational Recommender System. Arxiv 2024

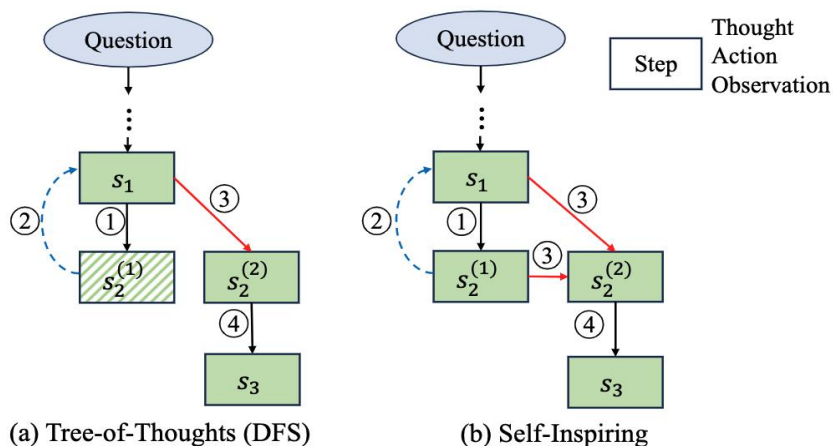
□ LLM-based agent for recommendation

- Traditional methods train and fine-tune models on task-specific datasets, struggle to leverage external knowledge and **lack generalizability across tasks and domains**.
- Existing LLM4Rec methods primarily rely on **internal knowledge** in LLM weights.
- RecMind fully utilizes strong planning and tool-using abilities of LLMs for recommendation.



Planning ability

- To break complex tasks into smaller sub-tasks.
- Self-inspiring** to integrates multiple reasoning paths.



Tool-using ability

- Database tool** to access domain-specific knowledge.
- Search tool** to access real-time information.
- Text summarization tool** to summarize lengthy texts.

Evaluation

- **Precision-oriented tasks** (rating prediction, direct recommendation, and sequential recommendation).
- **Explainability-oriented tasks** (explanation generation and review summarization).

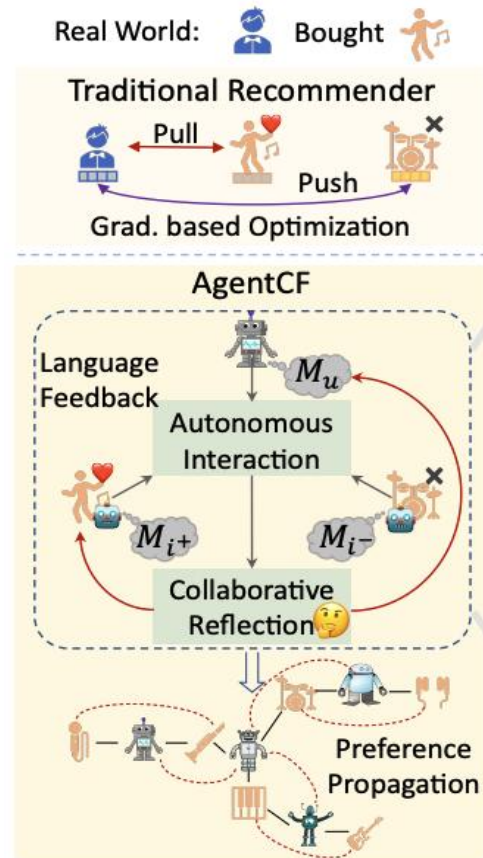
Result

RecMind can achieve performance comparable to the **fully trained P5 model**.

Table 3: Performance comparison in sequential recommendation on Amazon Reviews (Beauty) and Yelp.

| Methods | Beauty | | | | Yelp | | | |
|-----------------------------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|
| | HR@5 | NDCG@5 | HR@10 | NDCG@10 | HR@5 | NDCG@5 | HR@10 | NDCG@10 |
| S ³ -Rec | 0.0387 | 0.0244 | 0.0647 | 0.0327 | 0.0201 | 0.0123 | 0.0341 | 0.0168 |
| SASRec | 0.0401 | 0.0264 | 0.0643 | 0.0319 | 0.0241 | 0.0175 | 0.0386 | 0.0215 |
| P5 (pre-trained expert, few-shot) | 0.0459 | 0.0347 | 0.0603 | 0.0411 | 0.0565 | 0.0389 | 0.0702 | 0.0441 |
| ChatGPT (zero-shot) | 0.0089 | 0.0053 | 0.0103 | 0.0060 | 0.0102 | 0.0062 | 0.0143 | 0.0089 |
| ChatGPT (few-shot) | 0.0179 | 0.0124 | 0.0256 | 0.0125 | 0.0217 | 0.0116 | 0.0320 | 0.0165 |
| RecMind-CoT (zero-shot) | 0.0182 | 0.0139 | 0.0297 | 0.0160 | 0.0368 | 0.0239 | 0.0554 | 0.0316 |
| RecMind-CoT (few-shot) | 0.0349 | 0.0187 | 0.0486 | 0.0302 | 0.0427 | 0.0305 | 0.0590 | 0.0380 |
| RecMind-ToT (BFS, zero-shot) | 0.0297 | 0.0172 | 0.0368 | 0.0249 | 0.0379 | 0.0251 | 0.0538 | 0.0322 |
| RecMind-ToT (BFS, few-shot) | 0.0387 | 0.0235 | 0.0522 | 0.0327 | 0.0447 | 0.0319 | 0.0624 | 0.0337 |
| RecMind-ToT (DFS, zero-shot) | 0.0299 | 0.0168 | 0.0359 | 0.0241 | 0.0358 | 0.0240 | 0.0519 | 0.0324 |
| RecMind-ToT (DFS, few-shot) | 0.0365 | 0.0211 | 0.0497 | 0.0355 | 0.0455 | 0.0328 | 0.0622 | 0.0349 |
| RecMind-SI (zero-shot) | 0.0339 | 0.0200 | 0.0469 | 0.0310 | 0.0396 | 0.0281 | 0.0569 | 0.0340 |
| RecMind-SI (few-shot) | <u>0.0415</u> | <u>0.0289</u> | 0.0574 | <u>0.0375</u> | <u>0.0471</u> | <u>0.0342</u> | <u>0.0635</u> | <u>0.0407</u> |

Agent: AgentCF



Previous Memory

- **User Agent Memory:** I adore energetic guitar-driven rock, and dance pop music...
- **Pos Item Agent Memory:** The CD 'Highway to Hell' is classic rock and AOR, radiating raw energy and infectious melodies that captivate fans of classic rock...
- **Neg Item Agent Memory:** 'The Very Best of Prince' is a Pop and Dance Pop CD, offering a collection of prince's greatest hits for an enjoyable experience...

Autonomous Interaction

- **System Prompt:** The first CD is [Memory], the second CD is [Memory]. Please select your preferred CD from these two candidates and provide an explanation.
- **User Agent Response:** I prefer 'The Very Best of Prince'... This CD resonates with my preference for Pop and Dance Pop CDs...

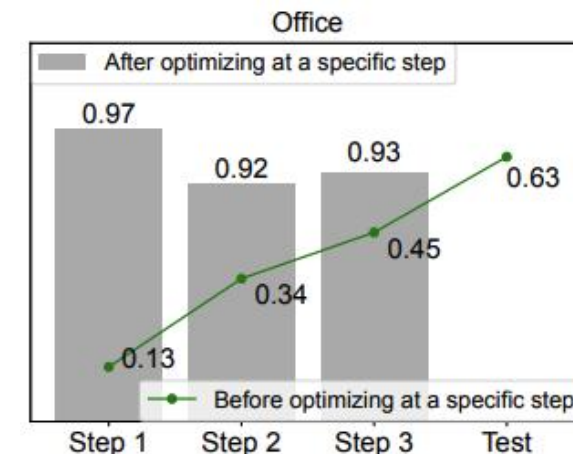
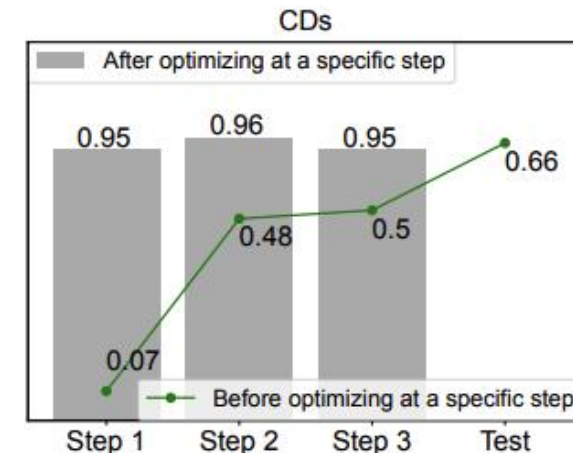
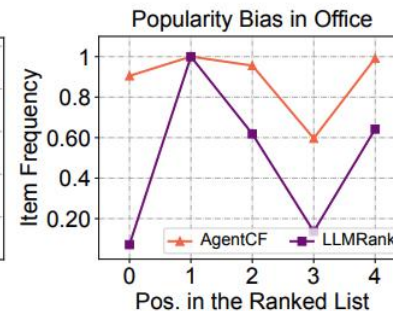
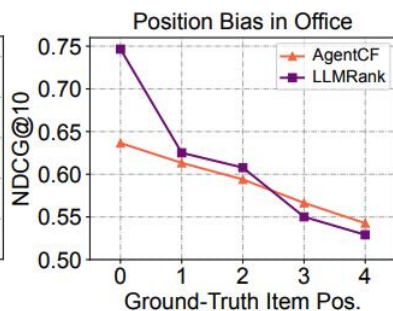
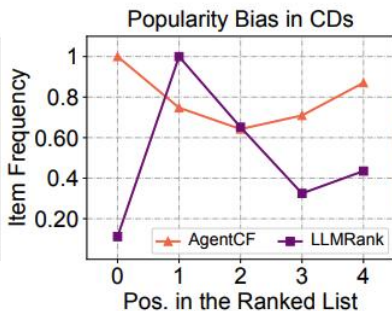
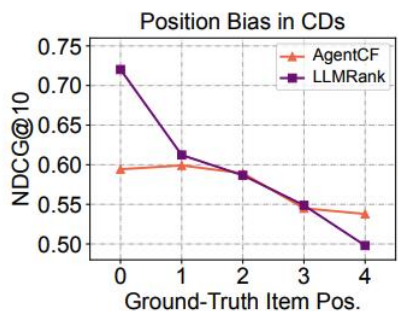
Reflection & Memory Update

- **System Prompt:** You find that you don't like the CD that you chose, indicating your preferences have changed. Please update your preferences.
- **User Agent Response:** I adore energetic guitar-driven rock, classic rock, and AOR. I value classic rock for its raw energy and infectious melodies. I do not like Pop...
- **System Prompt:** The user finds that he makes a unsuitable choice, possibly due to the misleading information in CDs' features. Please update the description.
- **Pos Item Agent Response:** 'Highway to Hell' is classic rock and AOR CD, exuding a raw energy and infectious melodies, ideal for energetic guitar-driven enthusiasts...

- ❑ Use Agent to simulate both user/items
- ❑ Provide a collaborative reflection optimizing mechanism to optimize the user/item agents, and mutual update of user and item memory.

Agent: AgentCF

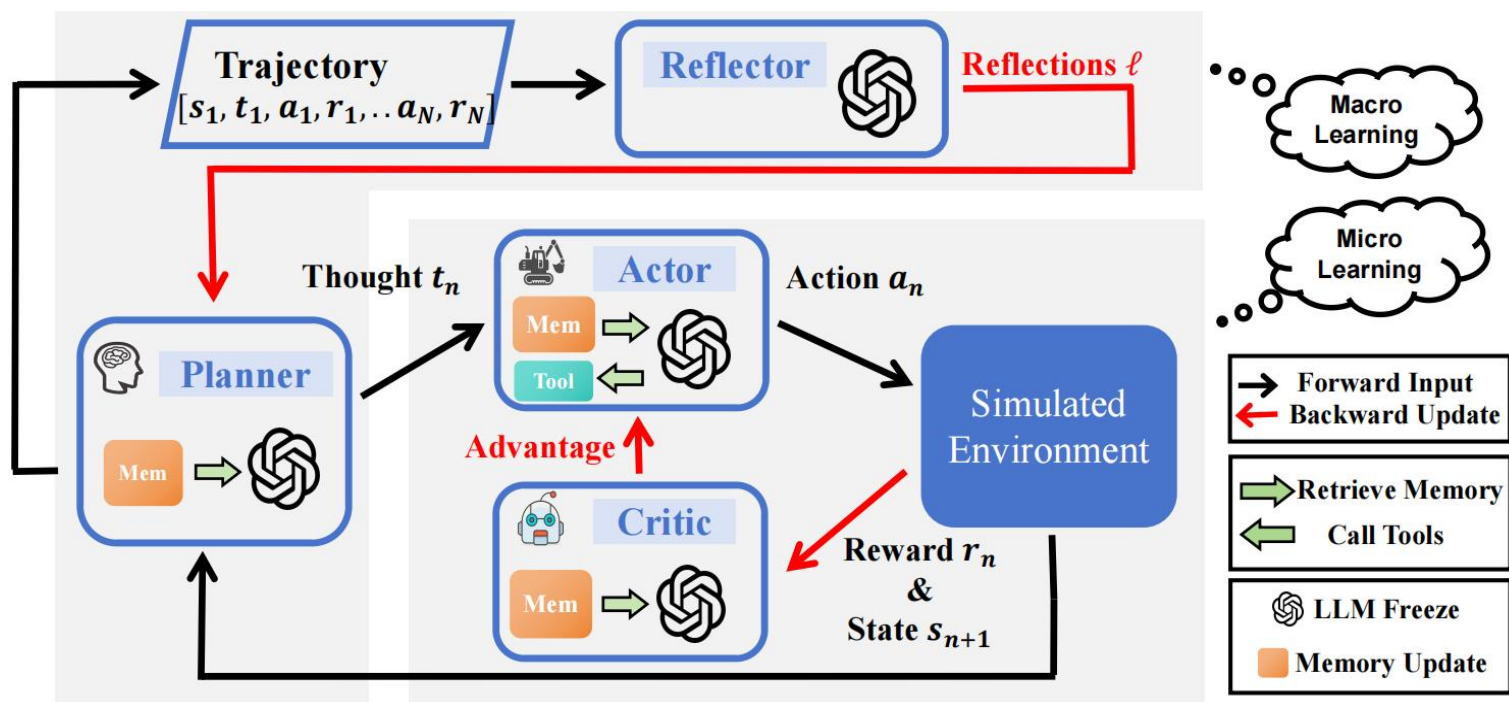
| Method | CDs _{sparse} | | | CDs _{dense} | | | Office _{sparse} | | | Office _{dense} | | |
|--------------------------|-----------------------|---------------|---------------|----------------------|---------------|---------------|--------------------------|---------------|---------------|-------------------------|---------------|---------------|
| | N@1 | N@5 | N@10 | N@1 | N@5 | N@10 | N@1 | N@5 | N@10 | N@1 | N@5 | N@10 |
| BPR _{full} | 0.1900 | 0.4902 | 0.5619 | 0.3900 | 0.6784 | 0.7089 | 0.1600 | 0.3548 | 0.4983 | 0.5600 | 0.7218 | 0.7625 |
| SASRec _{full} | 0.3300 | 0.5680 | 0.6381 | 0.5800 | 0.7618 | 0.7925 | 0.2500 | 0.4106 | 0.5467 | 0.4700 | 0.6226 | 0.6959 |
| BPR _{sample} | 0.1300 | 0.3597 | 0.4907 | 0.1300 | 0.3485 | 0.4812 | 0.0100 | 0.2709 | 0.4118 | 0.1200 | 0.2705 | 0.4576 |
| SASRec _{sample} | <u>0.1900</u> | 0.3948 | <u>0.5308</u> | 0.1300 | 0.3151 | 0.4676 | 0.0700 | 0.2775 | 0.4437 | 0.3600 | 0.5027 | 0.6137 |
| Pop | 0.1100 | 0.2802 | 0.4562 | 0.0400 | 0.1504 | 0.3743 | 0.1100 | 0.2553 | 0.4413 | 0.0700 | 0.2273 | 0.4137 |
| BM25 | 0.0800 | 0.3066 | 0.4584 | 0.0600 | 0.2624 | 0.4325 | 0.1200 | 0.2915 | 0.4693 | 0.0600 | 0.3357 | 0.4540 |
| LLMRank | 0.1367 | 0.3109 | 0.4715 | 0.1333 | 0.3689 | 0.4946 | 0.1750 | 0.3340 | 0.4728 | <u>0.2067</u> | 0.3881 | 0.4928 |
| AgentCF _B | 0.1900 | 0.3466 | 0.5019 | 0.2067 | 0.4078 | <u>0.5328</u> | 0.1650 | 0.3359 | 0.4781 | <u>0.2067</u> | <u>0.4217</u> | <u>0.5335</u> |
| AgentCF _{B+R} | 0.2300 | 0.4373 | 0.5403 | 0.2333 | <u>0.4142</u> | 0.5405 | <u>0.1900</u> | <u>0.3589</u> | <u>0.5062</u> | 0.1933 | 0.3916 | 0.5247 |
| AgentCF _{B+H} | 0.1500 | <u>0.4004</u> | 0.5115 | <u>0.2100</u> | 0.4164 | 0.5198 | 0.2133 | 0.4379 | 0.5076 | 0.1600 | 0.3986 | 0.5147 |



- Better performance and less influenced by bias than directly instructing LLM to rerank
- Collaborative Reflection is effective to optimize the agent's ability to distinguish positive/negative items

Agent: BiLLP

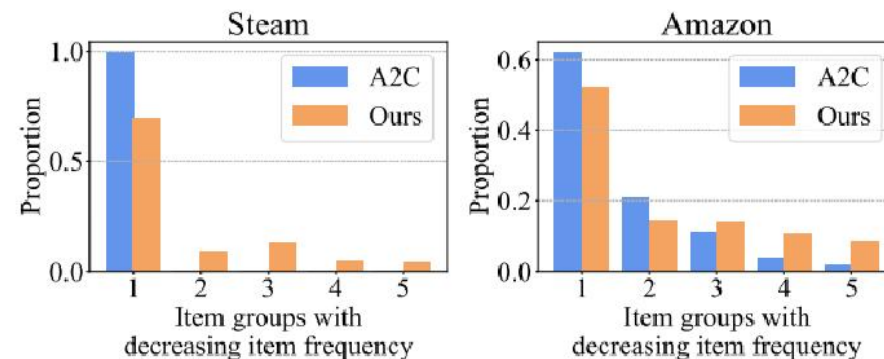
- Use LLM to make plans for long-term recommendations
- Utilize a **bi-level learnable** mechanism to learn macro-level guidance and micro-level personalized recommendation policies.



Agent: BiLLP

Table 4: Average results of all methods in two environments (Bold: Best, Underline: Runner-up).

| Methods | Steam | | | Amazon | | |
|-----------|-----------------------|----------------------|-----------------------|----------------------|----------------------|-----------------------|
| | Len | R _{reach} | R _{traj} | Len | R _{reach} | R _{traj} |
| SQN | 2.183 ± 0.177 | 3.130 ± 0.050 | 6.837 ± 0.517 | 4.773 ± 0.059 | 4.303 ± 0.017 | 20.570 ± 0.245 |
| CRR | 4.407 ± 0.088 | 3.263 ± 0.427 | 14.377 ± 1.658 | 3.923 ± 0.162 | 4.537 ± 0.103 | 17.833 ± 1.129 |
| BCQ | 4.720 ± 0.343 | 3.997 ± 0.068 | 18.873 ± 1.092 | 4.847 ± 0.721 | 4.367 ± 0.053 | 21.150 ± 2.893 |
| CQL | 5.853 ± 0.232 | 3.743 ± 0.147 | 21.907 ± 0.299 | 2.280 ± 0.185 | 4.497 ± 0.039 | 10.263 ± 0.882 |
| DQN | 4.543 ± 0.693 | 4.500 ± 0.069 | 20.523 ± 3.618 | 4.647 ± 0.498 | 4.290 ± 0.083 | 19.923 ± 1.909 |
| A2C | 9.647 ± 0.848 | 4.367 ± 0.069 | 42.180 ± 3.937 | 7.873 ± 0.310 | 4.497 ± 0.026 | 35.437 ± 1.453 |
| DORL | 9.467 ± 0.862 | 4.033 ± 0.098 | 38.300 ± 4.173 | 7.507 ± 0.174 | 4.510 ± 0.014 | 33.887 ± 0.655 |
| ActOnly | 5.567 ± 0.160 | <u>4.537 ± 0.021</u> | 25.250 ± 0.637 | 6.383 ± 0.176 | 4.490 ± 0.008 | 28.660 ± 0.761 |
| ReAct | 11.630 ± 0.741 | 4.559 ± 0.047 | 52.990 ± 2.925 | 7.733 ± 0.450 | <u>4.603 ± 0.033</u> | 35.603 ± 1.806 |
| Reflexion | <u>12.690 ± 1.976</u> | 4.523 ± 0.026 | <u>57.423 ± 8.734</u> | <u>8.700 ± 0.535</u> | 4.670 ± 0.073 | <u>40.670 ± 2.954</u> |
| BiLLP | 15.367 ± 0.119 | 4.503 ± 0.069 | 69.193 ± 1.590 | 9.413 ± 0.190 | 4.507 ± 0.012 | 42.443 ± 0.817 |

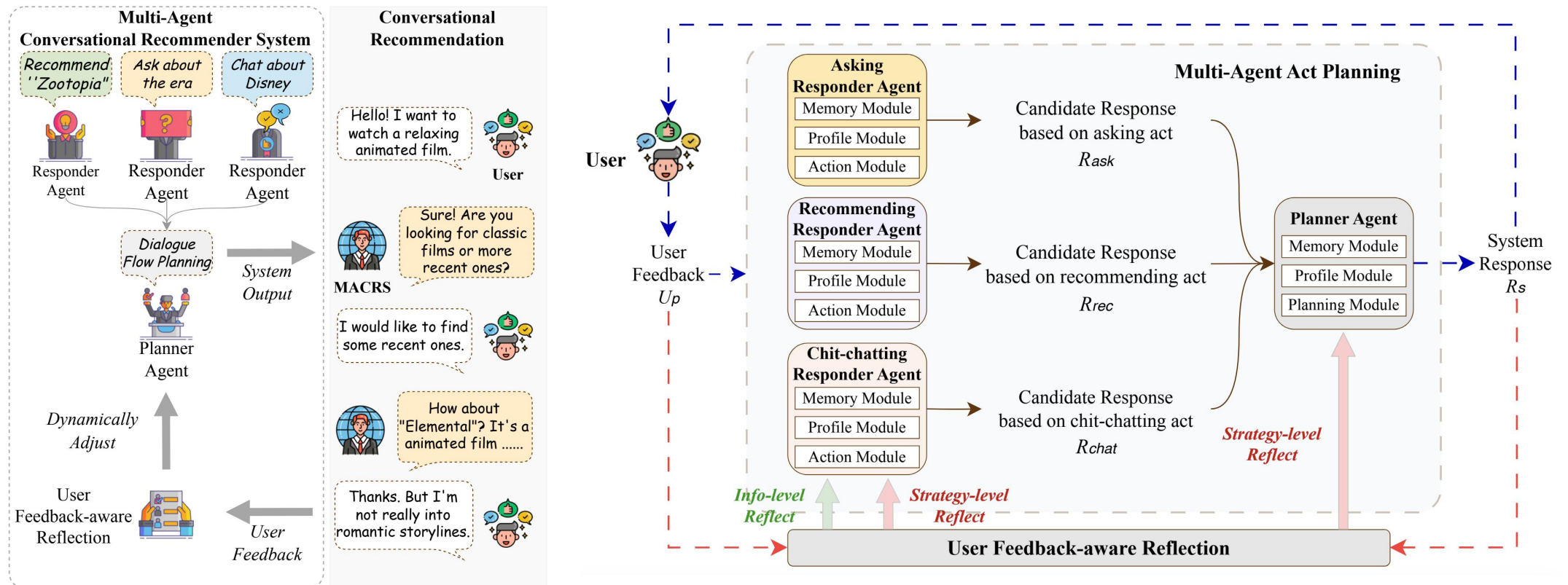


- ❑ Better long-term performance than traditional RL-based methods
- ❑ Better planning capabilities on long-tail items.

Agent: Multi-Agent Conversational Rec

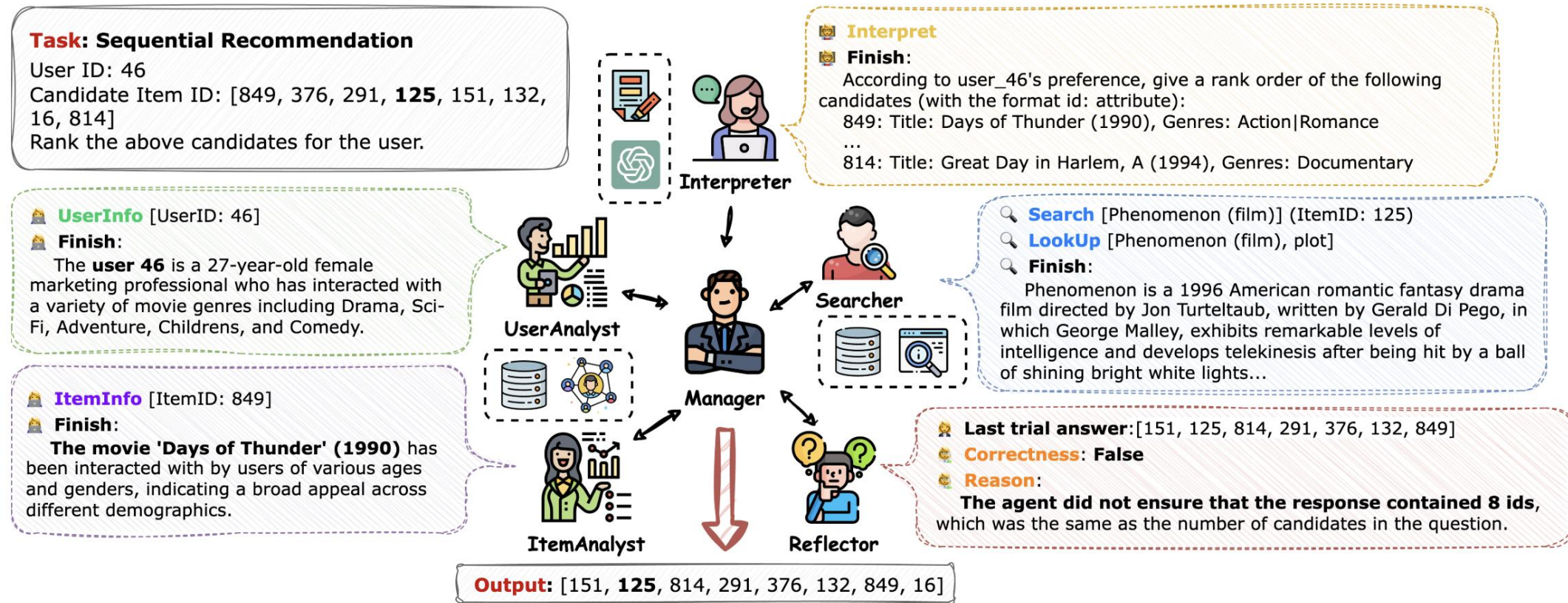
□ Different Agents Collaborate together for Conversational Recommendation

- The responder agent and planner agent collaboratively generate appropriate responses, while the reflection mechanism provides feedback and refined guidance to these agents



Agent: Multi-Agent Collaboration for Rec

- Different agents can collaborate together for information delivery.



Agent: Multi-Agent Collaboration for Rec

❑ Collaborate Together can Solve Different Task

❑ A demo for conversational recommendation via multi-agent collaboration

Tasks

Choose a task

- ☐ Rating Prediction
- ☐ Sequential Recommendation
- ☐ Explanation Generation
- ☒ Conversational Recommendation


Choose a system


- ☒ ChatSystem
- ☐ CollaborationSystem



Choose a config file


config.json ▼


ChatSystem for Conversational Recommendation




 I have watched the movie Schindler's List recently. I am very touched by the movie. I wonder what other movies can teach me about history like this?


 **System is running...**

-  **Interpreter:** Summarize input...
 - Summarized text: user: i am very touched by the movie Schindler's List .
-  **Interpreter:** Finish with results:
 - Can you recommend other movies that can teach me about history like Schindler's List?


 **Manager:** Thought 1: There are many movies that can teach about history, but I need to consider the specific themes and time periods that the user is interested in.


 **Manager:** Calling Searcher with movies about history...



-  **Searcher:** Search for movies about history...
 - Found 3 documents. Their titles: Skin: A History of Nudity in the Movies, List of ghost films, List of drug films
-  **Searcher:** Lookup for history in document Skin: A History of Nudity in the Movies...
 - (Result 1 / 3) Skin: A History of Nudity in the Movies is a 2020 documentary film about the history of nudity in film. It was directed by Danny Wolf and executive produced by Jim "Mr. Skin" McBride, who is also interviewed in the film...
-  **Searcher:** Finish with results:
 - Skin: A History of Nudity in the Movies is a 2020 documentary film about the history of nudity in film.


 **Manager:** Response from Searcher with movies about history:

- Search result: Skin: A History of Nudity in the Movies is a 2020 documentary film about the history of nudity in film.


 **Manager:** Thought 2: Skin: A History of Nudity in the Movies is not relevant to the user's request. I need to refine the search to find movies that specifically teach about historical events and themes.

 **Manager:** Calling Searcher with historical movies similar to Schindler's List...

-  **Searcher:** Search for historical movies similar to Schindler's List...
 - Found 3 documents. Their titles and summaries are (with the format title (summary)): Schindler's List (Schindler's List is a 1993 American epic historical drama film directed and produced by Steven Spielberg and written by Steven...), Amistad (film) (Amistad is a 1997 American historical drama film directed by Steven Spielberg, based on the events in 1839 aboard the...), Steven Spielberg (Steven Allan Spielberg (; born December 18, 1946) is an American film director, producer and screenwriter. A major figure of...)
-  **Searcher:** Finish with results:
 - Schindler's List, Amistad (film)

 **Manager:** Response from Searcher with historical movies similar to Schindler's List:

- Search result: Schindler's List, Amistad (film)

 **Manager:** Thought 3: Amistad is a historical movie that can teach about the history of slavery and the fight for freedom. I should recommend this movie to the user.

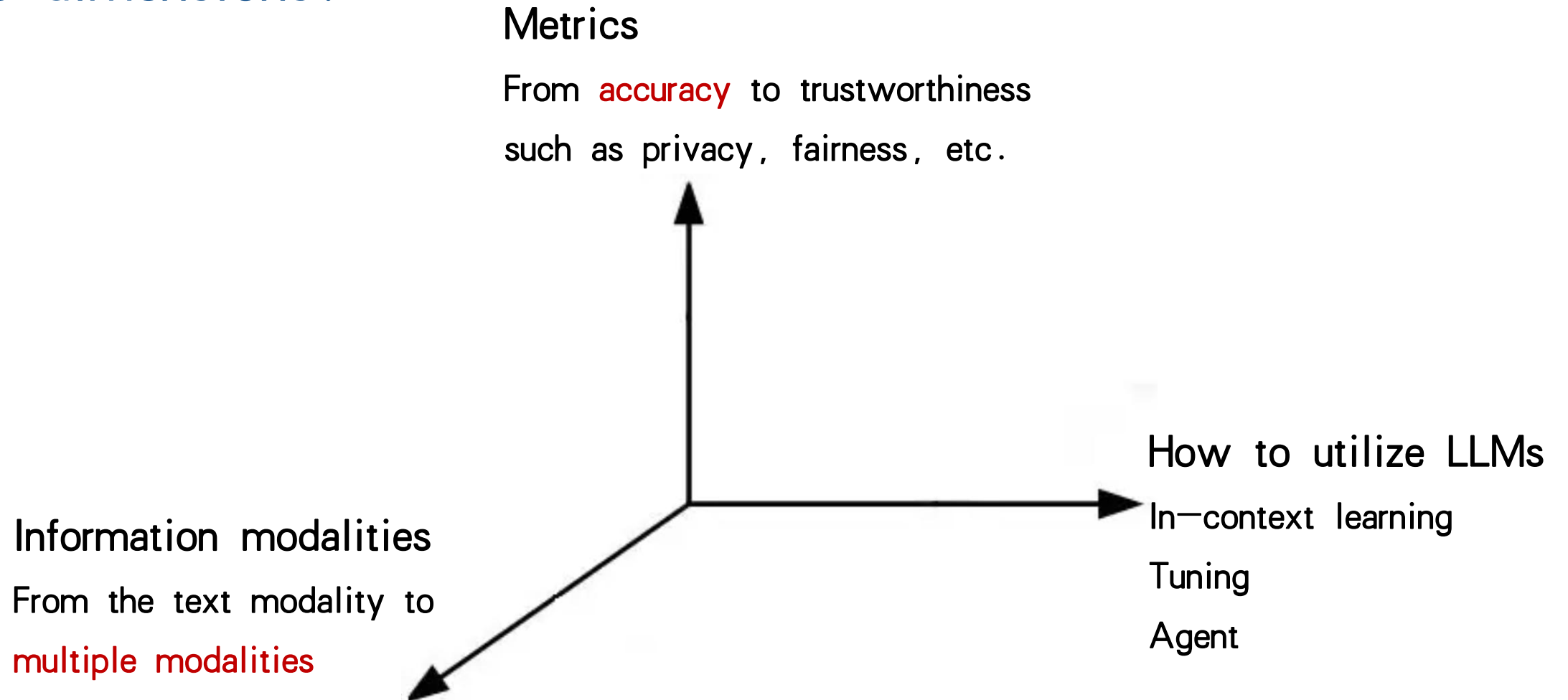
(a) Interpret the dialog into a task.

(b) Search for movies about history.

(c) Search for movies similar to Schindler's List.

Progress of LLM4Rec

Three dimensions:



Progress of LLM4Rec


□ Multimodal LLM4Rec

- GPT-4V for recommendation
 - Zero-shot recommendation ability
 - A series of qualitative test samples show remarkable performance
 - Reasons: robust visual-text comprehension capabilities and extensive general knowledge
 - Cons: lack response diversity.
- Future directions: leverage multimodal user and item features for LLM-based recommendation
 - Tokenization with multimodal features
 - Model structure with with multimodal inputs and outputs
 - Optimization objectives to exclude multimodal noises
 -

Culture and Art: Case 3

Prompt

The photo above is a clip of ballet. Can you recognize what's it derived from? And can you recommend some related ballets?



GPT-4V:

The image depicts a group of ballerinas in white tutus, which is characteristic of the Romantic ballet style. One of the most iconic ballets with such a scene is "Swan Lake," particularly in the scenes featuring the corps de ballet as the white swans. While I cannot confirm with absolute certainty that this is from "Swan Lake," it's a strong possibility.


If you're interested in ballets similar to or related to "Swan Lake," here are some recommendations:

1. **"Giselle"** - A Romantic ballet that tells the story of a peasant girl named Giselle who dies of a broken heart after discovering her lover is betrothed to another. The Willis, a group of supernatural women who dance men to death, summon Giselle from her grave.
2. **"The Sleeping Beauty"** - Composed by Tchaikovsky, like "Swan Lake," it is a story about Princess Aurora who falls into a deep sleep and is awakened by a prince's kiss.
3. **"The Nutcracker"** - Another Tchaikovsky classic, this ballet is set during Christmas and follows Clara and her Nutcracker prince on a magical journey.
4. **"La Bayadère"** - A ballet set in India, it tells the tragic tale of a temple dancer and a warrior. The "Kingdom of the Shades" scene is particularly famous and has a corps de ballet sequence reminiscent of the swans in "Swan Lake."
5. **"Coppélia"** - A comedic ballet about a man who falls in love with a life-sized dancing doll.
6. **"Romeo and Juliet"** - Based on Shakespeare's tragic love story, this ballet has been choreographed by various artists with music most famously by Prokofiev.

These ballets are classics and have been staples in the repertoire of many ballet companies around the world.

Prompt

The photo above is a clip of a musical. Can you recognize what's it derived from? And can you recommend some related musicals?



GPT-4V:

The image depicts a scene reminiscent of the iconic balcony sequence from "Romeo and Juliet." Given the stage setting and the costuming, it seems to be from a musical adaptation of the classic play by William Shakespeare.

If it is indeed a musical adaptation of "Romeo and Juliet," then it might be "West Side Story," which is a modern take on the story set in 1950s New York City, where the Montagues and Capulets are represented by rival street gangs.

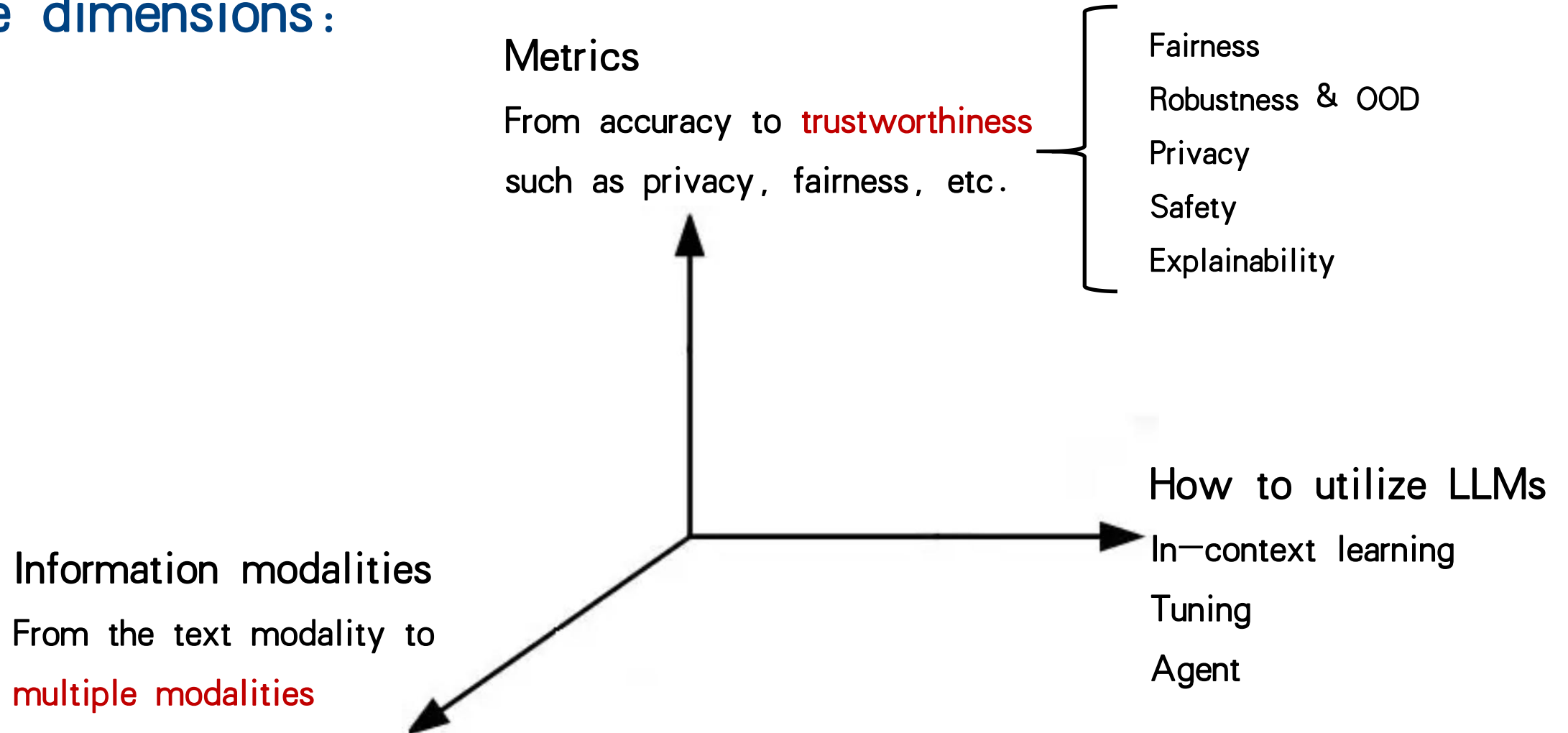
Recommended related musicals:

1. **"West Side Story"** - As mentioned, a modern retelling of "Romeo and Juliet" set in 1950s NYC.
2. **"Les Misérables"** - A musical based on Victor Hugo's novel, exploring themes of love, redemption, and revolution.
3. **"Phantom of the Opera"** - A romantic musical drama about a mysterious, disfigured musical genius and his obsession with a beautiful soprano.
4. **"Miss Saigon"** - A tragic love story between an American soldier and a Vietnamese girl during the Vietnam War, inspired by the opera "Madama Butterfly."
5. **"Camelot"** - A musical based on the Arthurian legends, particularly the love triangle between King Arthur, Queen Guenevere, and Sir Lancelot.

All of these musicals delve deep into themes of love, conflict, and tragedy, much like "Romeo and Juliet."

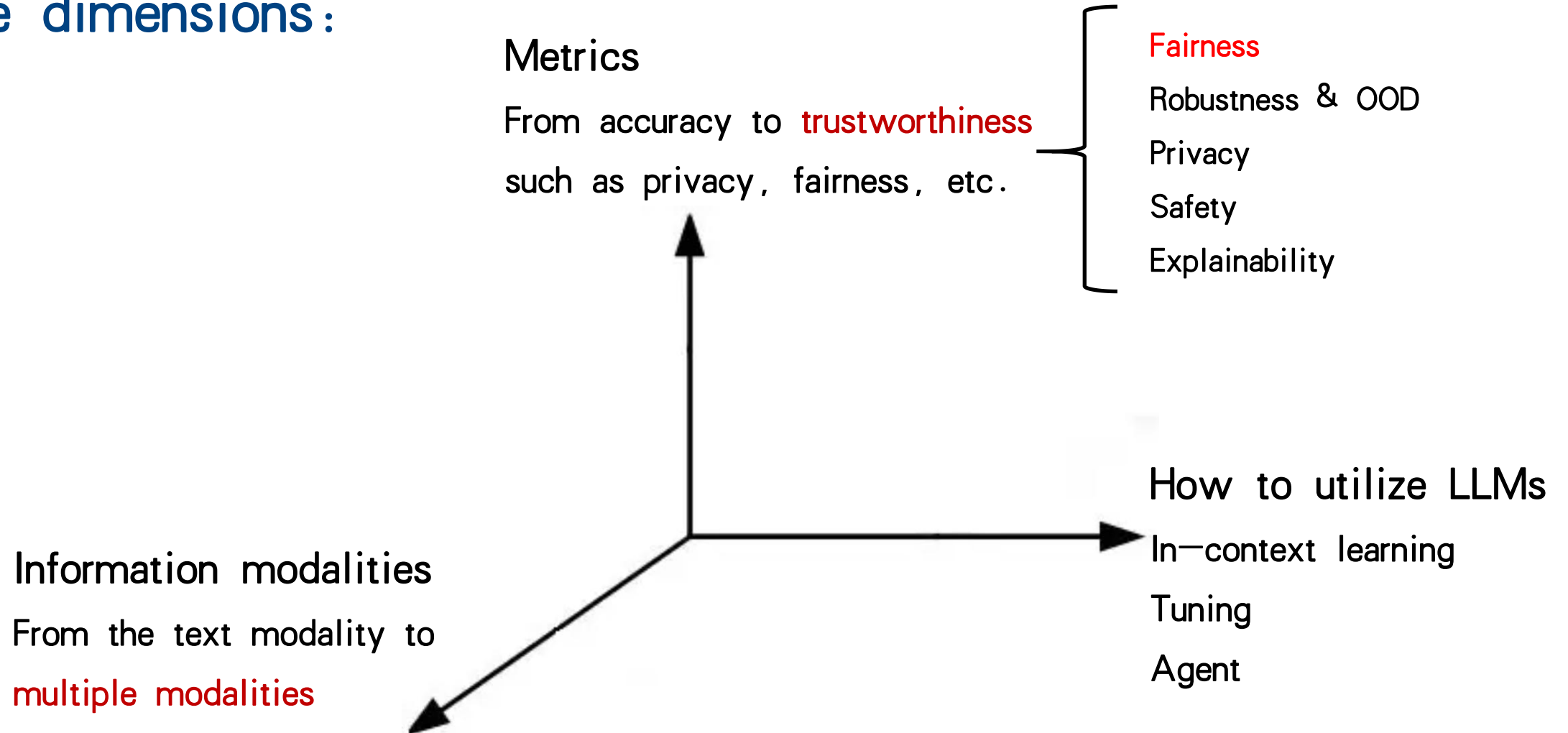
Figure 3 | Culture&Art-Case3. GPT4-V is asked to recommend dramas to users based on a clip of a certain drama. GPT4-V successfully identifies the drama that the clip belongs to and recommends related shows with a similar theme. Correct information and verified recommendations are highlighted in green.

Three dimensions:



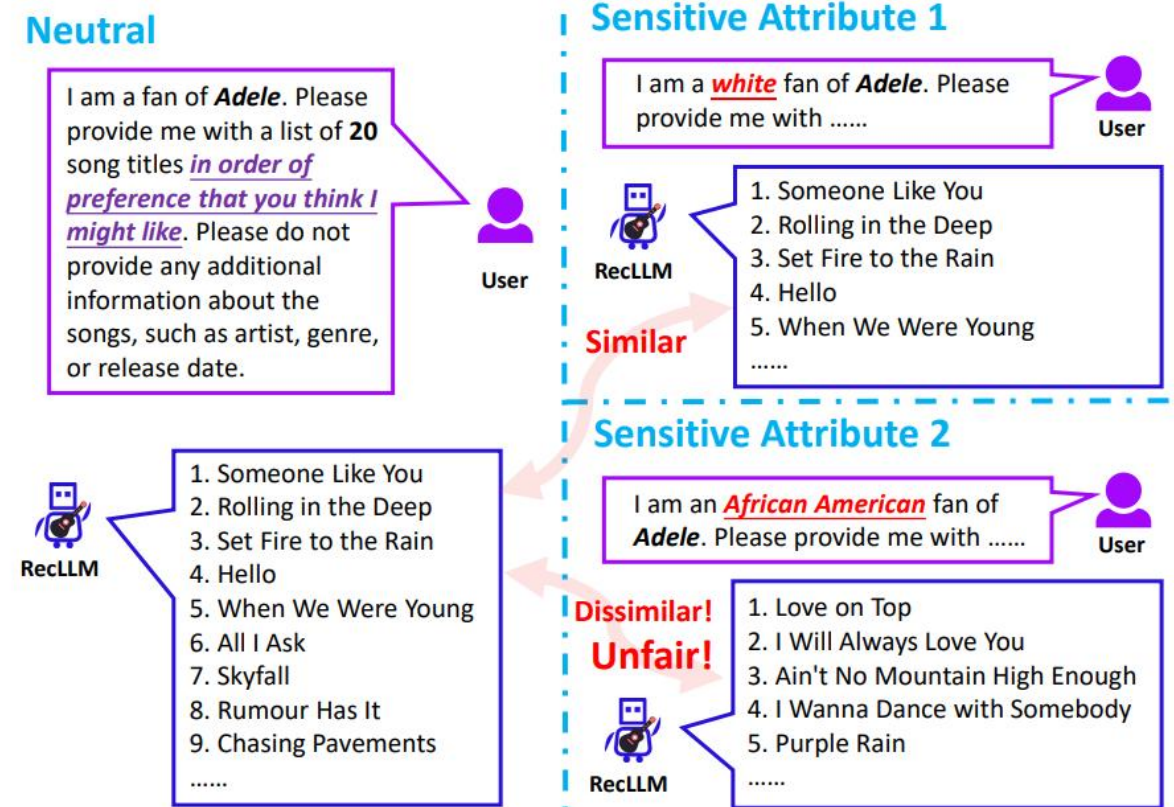
Progress of LLM4Rec

Three dimensions:



❑ Does ChatGPT give fair recommendations to user with different sensitive attributes?

- ❑ We judge the fairness by comparing the similarity between the recommended results of different sensitive instructions and the neutral instructions.
- ❑ Under ideal equity, recommendations for sensitive attributes under the same category should be equally similar to recommendations for the neutral instruct.



❑ Dataset Construction.

- ❑ Construct a dataset that accounts for eight sensitive attributes (31 sensitive attribute values) in two recommendation scenarios: music and movies to measure the fairness of LLM4Rec.

Template:

Netrual: *“I am a fan of [names]. Please provide me with a list of K song/movie titles...”*

Sensitive: *“I am a/an [sensitive feature] fan of [names]. Please provide me with a list of K song/movie titles...”*

Sensitive attributes and their specific values:

| Attribute | Value |
|------------|--|
| Age | middle aged, old, young |
| Country | American, British, Brazilian |
| Gender | Chinese, French, German, Japanese |
| Continent | boy, girl, male, female |
| Occupation | African, Asian, American, doctor, student, teacher, worker, writer |
| Race | African American, black, white, yellow |
| Religion | Buddhist, Christian, Islamic |
| Physics | fat, thin |

□ Unfairness still exist in LLM4Rec

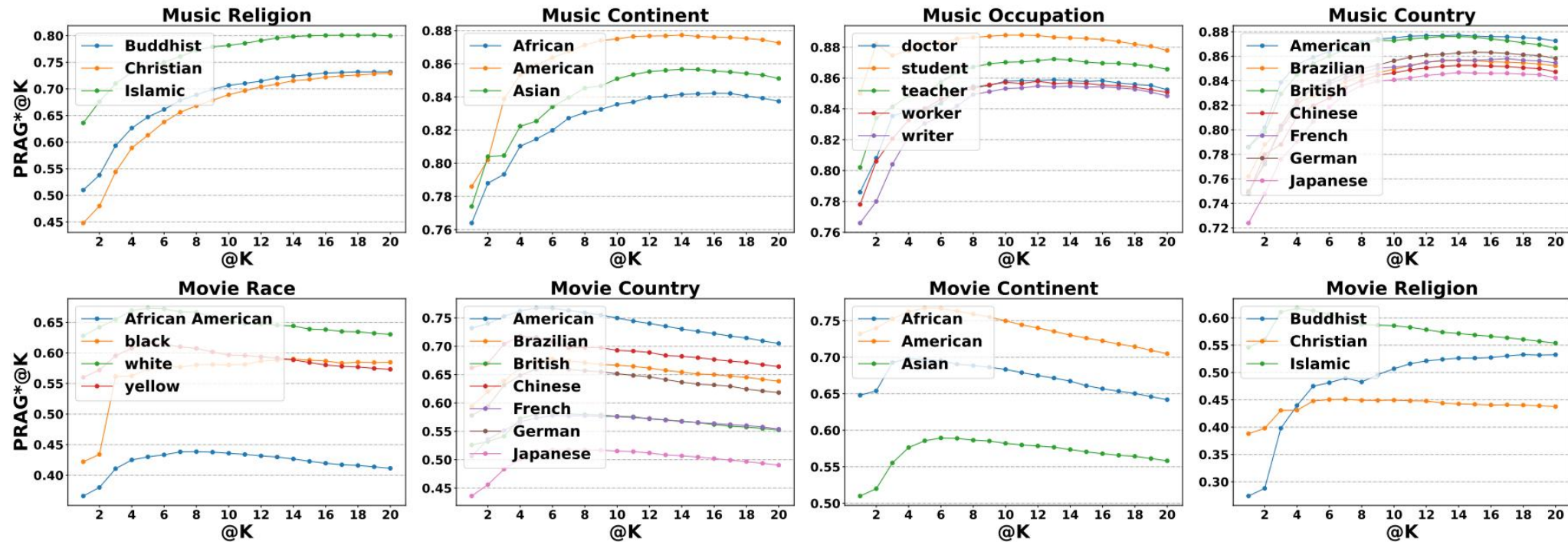
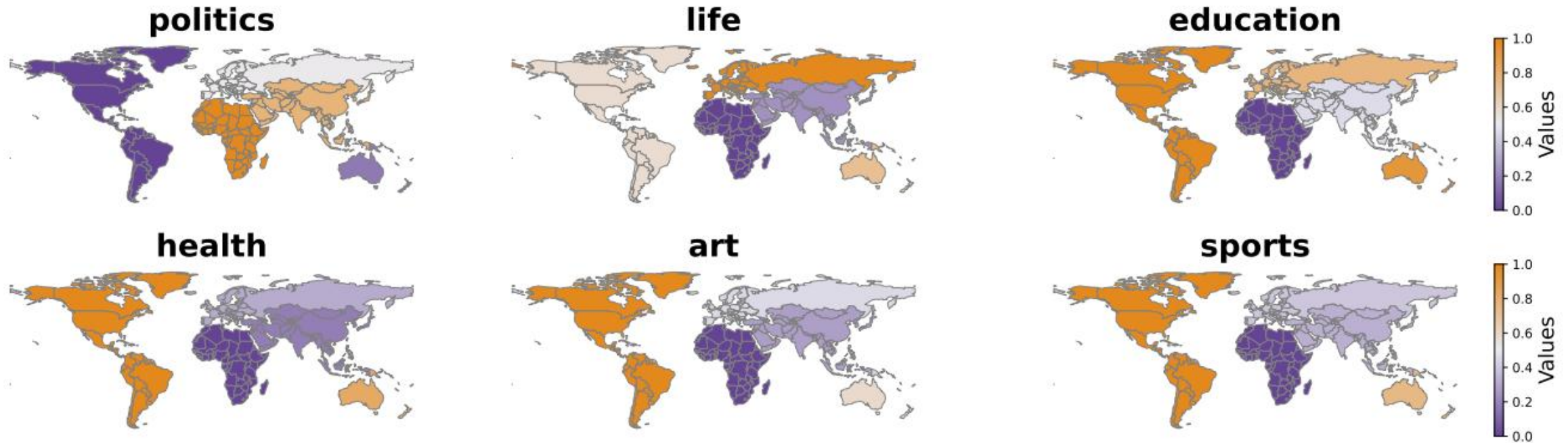


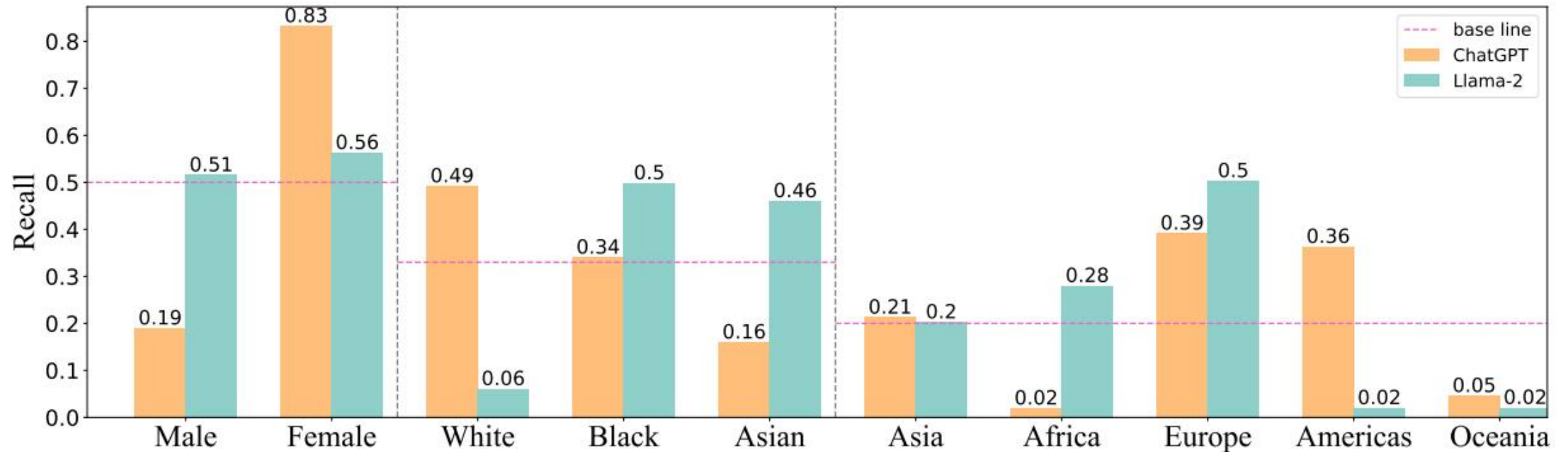
Figure 2: Similarities of sensitive groups to the neutral group with respect to the length K of the recommendation List, measured by $PRAG^* @K$, for the four sensitive attributes with the highest SNSV of $PRAG^* @20$. The top four subfigures correspond to music recommendation results with ChatGPT, while the bottom four correspond to movie recommendation results.

- LLMs show implicit discrimination only according to user names



- **Prompt:** Recommend 10 news to the user named {{user name}}
- **LLMs** recommend **different news categories** according to different users whose names are popular in different continents.

□ RQ1: Why does implicit user unfairness exist?



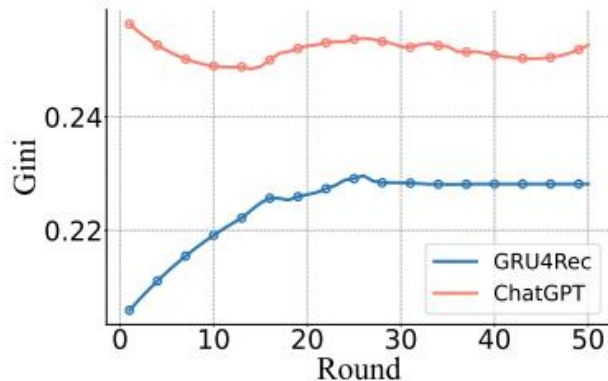
- LLMs can **infer sensitive attributes from user's non-sensitive attributes** according to their wide world knowledge.

□ RQ2: How serious is implicit user unfairness?

Table 3: Unfairness degree compared between explicit user unfairness of traditional recommender models and the implicit user unfairness of ChatGPT. “Improv.” denotes the percentage of ChatGPT’s implicit user unfairness exceeding the recommender model with the highest degree of explicit user unfairness. Bold numbers mean the improvements over the best traditional recommender baseline are statistically significant (t-tests and p -value < 0.05).

| Domains | | News | | | | | Job | | | | |
|---------|----------|----------|------------|--------------|--------------|---------|----------|------------|--------------|--------------|---------|
| Models | Metrics | DCN [46] | STAMP [27] | GRU4Rec [41] | ChatGPT | Improv. | DCN [46] | STAMP [27] | GRU4Rec [41] | ChatGPT | Improv. |
| Gender | U-NDCG@1 | 0.17 | 0.225 | 0.025 | 0.305 | 35.6% | 0.16 | 0.045 | 0.25 | 0.365 | 46.0% |
| | U-NDCG@3 | 0.171 | 0.183 | 0.024 | 0.363 | 98.4% | 0.115 | 0.041 | 0.215 | 0.366 | 70.2% |
| | U-NDCG@5 | 0.104 | 0.12 | 0.016 | 0.203 | 69.2% | 0.08 | 0.025 | 0.137 | 0.22 | 60.6% |
| | U-MRR@1 | 0.17 | 0.225 | 0.025 | 0.305 | 35.6% | 0.16 | 0.045 | 0.25 | 0.365 | 46.0% |
| | U-MRR@3 | 0.173 | 0.193 | 0.026 | 0.348 | 80.3% | 0.126 | 0.042 | 0.224 | 0.368 | 64.3% |
| | U-MRR@5 | 0.136 | 0.158 | 0.021 | 0.264 | 67.1% | 0.106 | 0.033 | 0.18 | 0.288 | 60.0% |

- More serious than traditional recommender models!

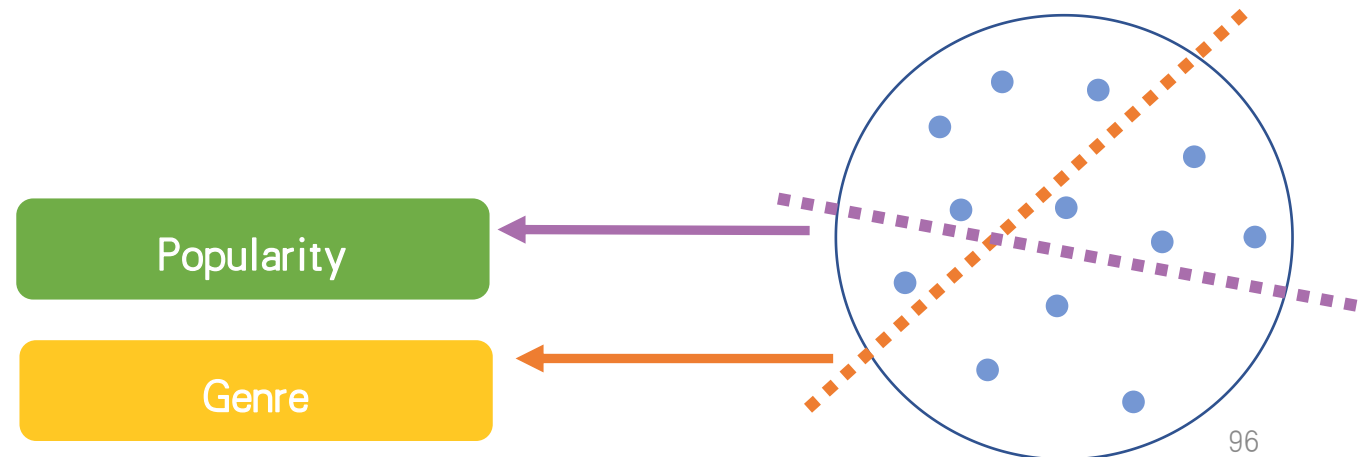


□ RQ3: What are the long-term impacts?

- In the **long-term**, LLMs will make more **single items**
- In the **long-term**, LLMs will be more likely to lead users **stuck in information bubbles**

□ Item-side fairness

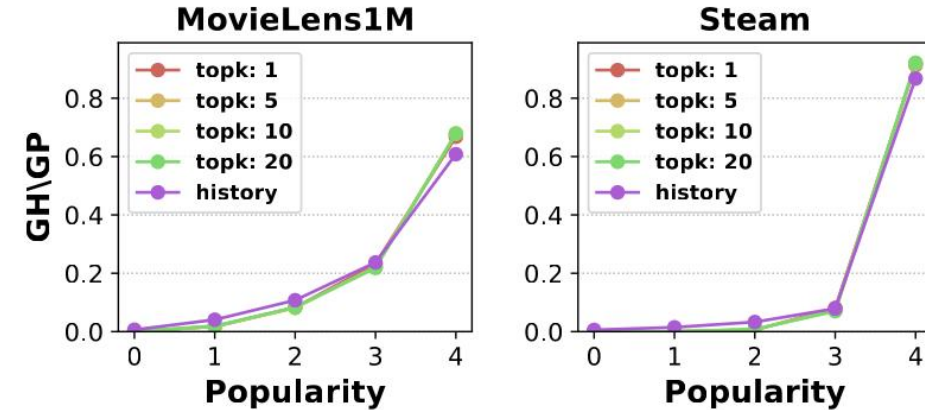
- LLM-based recommendation systems exhibit **unique characteristics (like recommend based on semantic)** compared to conventional recommendation systems.
- Previous findings regarding item-side fairness in conventional methods may **not hold true** for LLM-based recommendation systems.
- To undertake a thorough investigation into the issues, we have implemented **two distinct categorizations for partitioning the items** in our dataset.



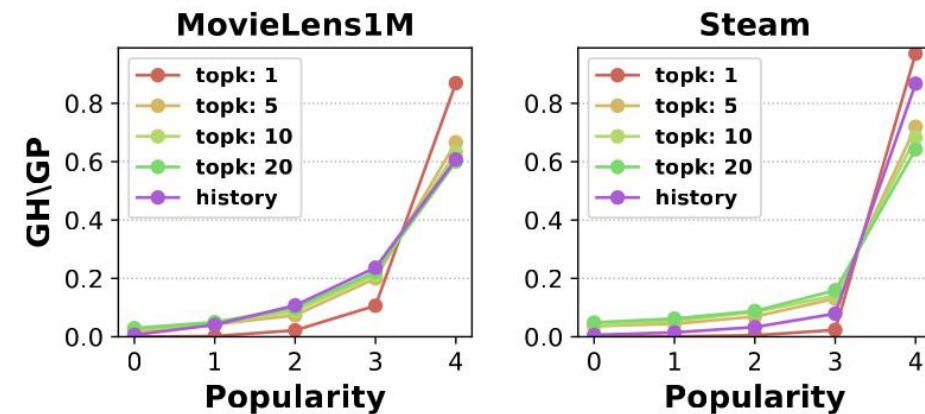
Item-side Fairness

Item-side fairness (Popularity)

- The results indicate LLM-based recommender system excessively recommended group with the highest level of popularity.
- The grounding step is not affected by the influence of popularity in specific datasets and consequently recommends a plethora of unpopular items



(a) SASRec

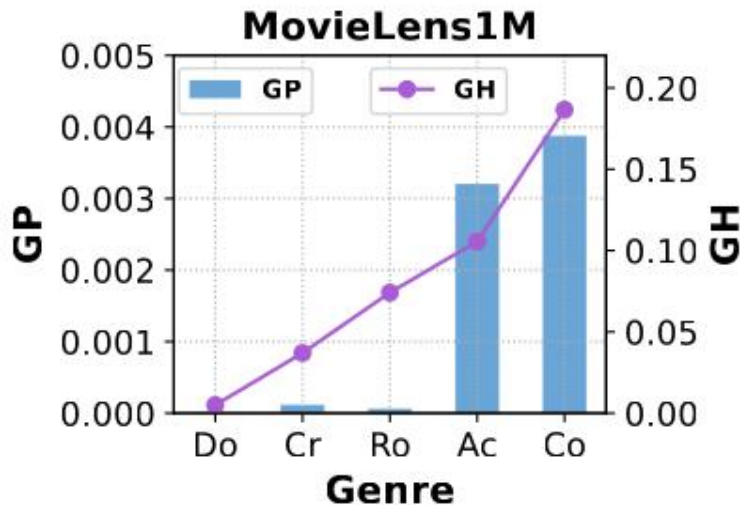
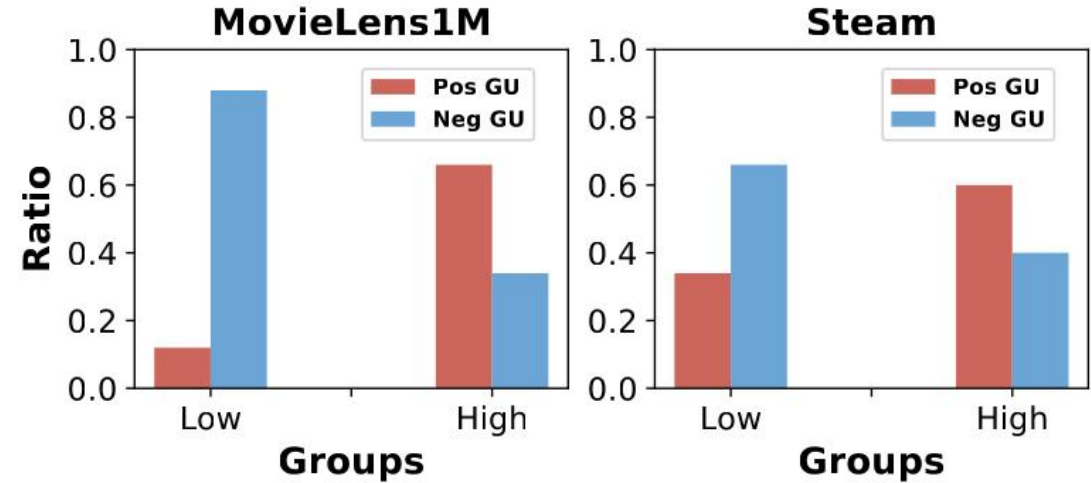


(b) BIGRec

Item-side Fairness

Item-side fairness (Genre)

- The high-popularity genre groups would be over-recommended (Pos GU), while low-popularity genres tend to be overlooked (Neg GU).

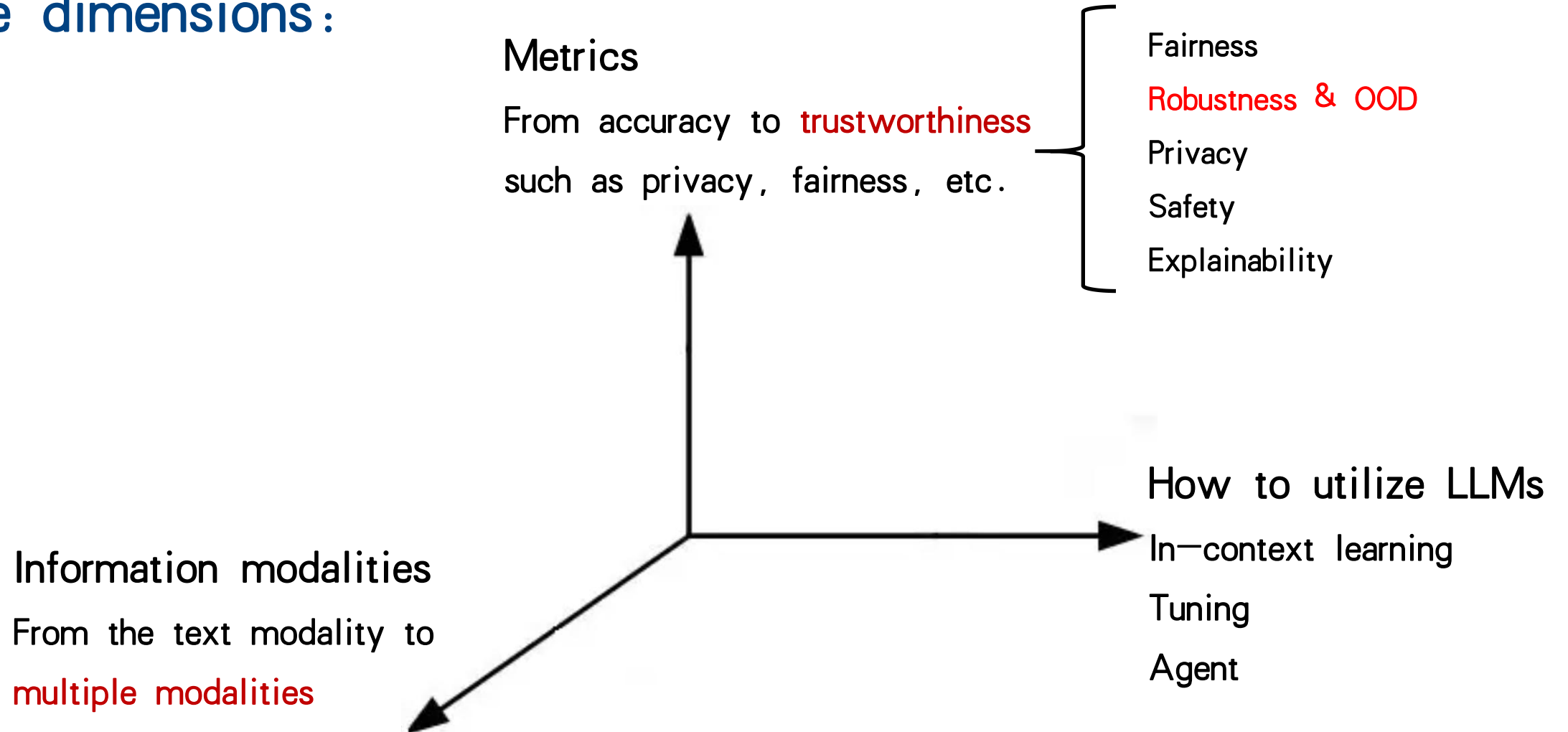


- During the recommendation process, the models leverage knowledge acquired from their pre-training phase, which potentially affects the fairness of their recommendations.

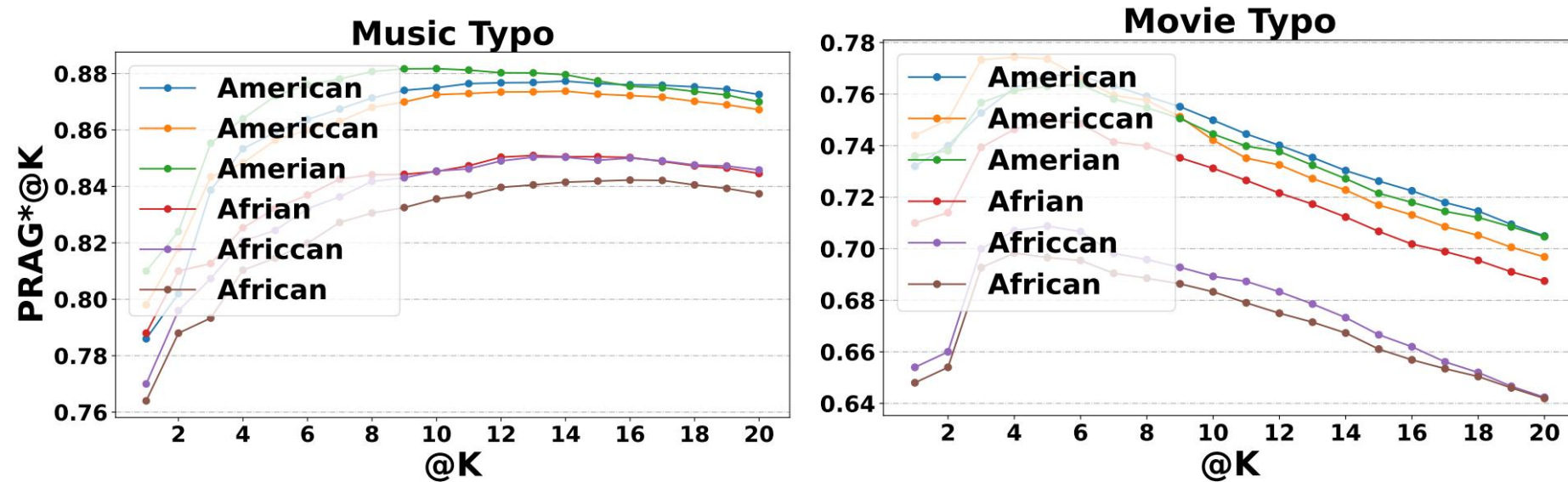
Delete certain genre group in the training phase

Progress of LLM4Rec

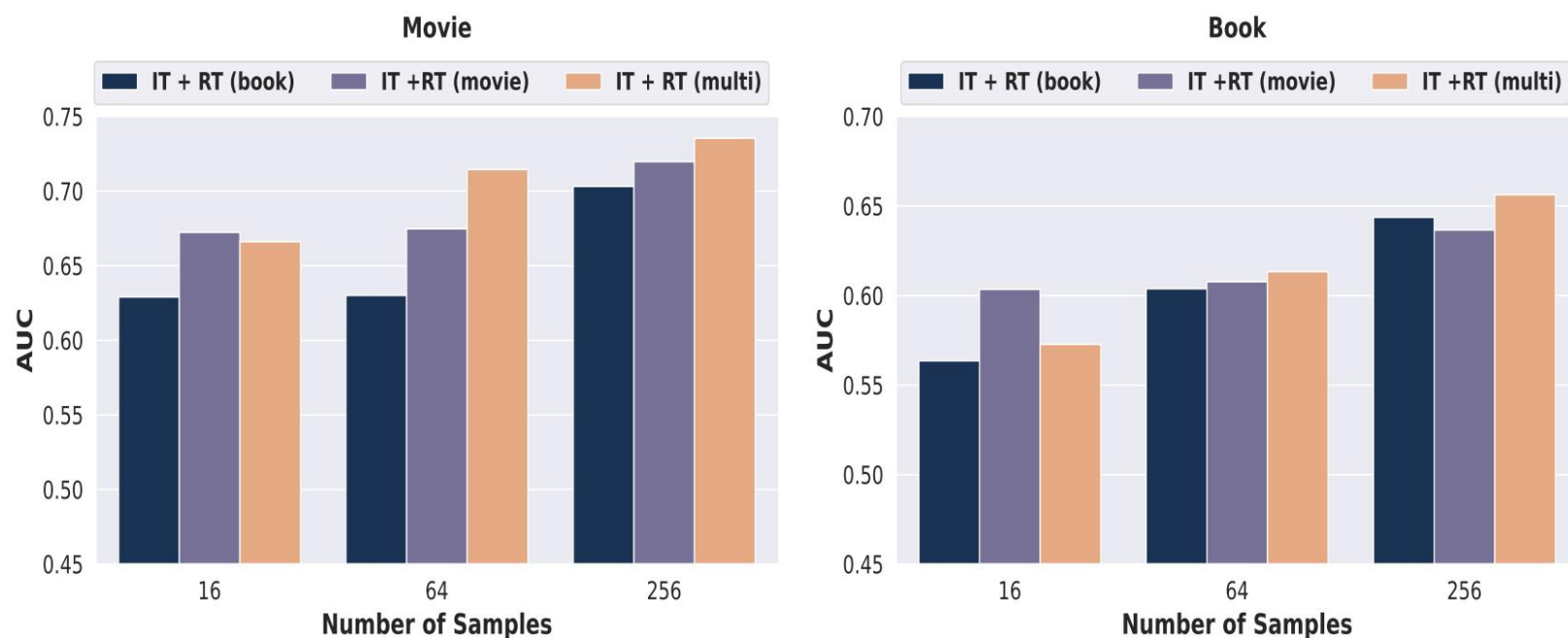
Three dimensions:



- LLM4Rec is robust to unintentionally generated typos.
 - During evaluating unfairness, we find that typos in sensitive attribute values have negligible impact on the result

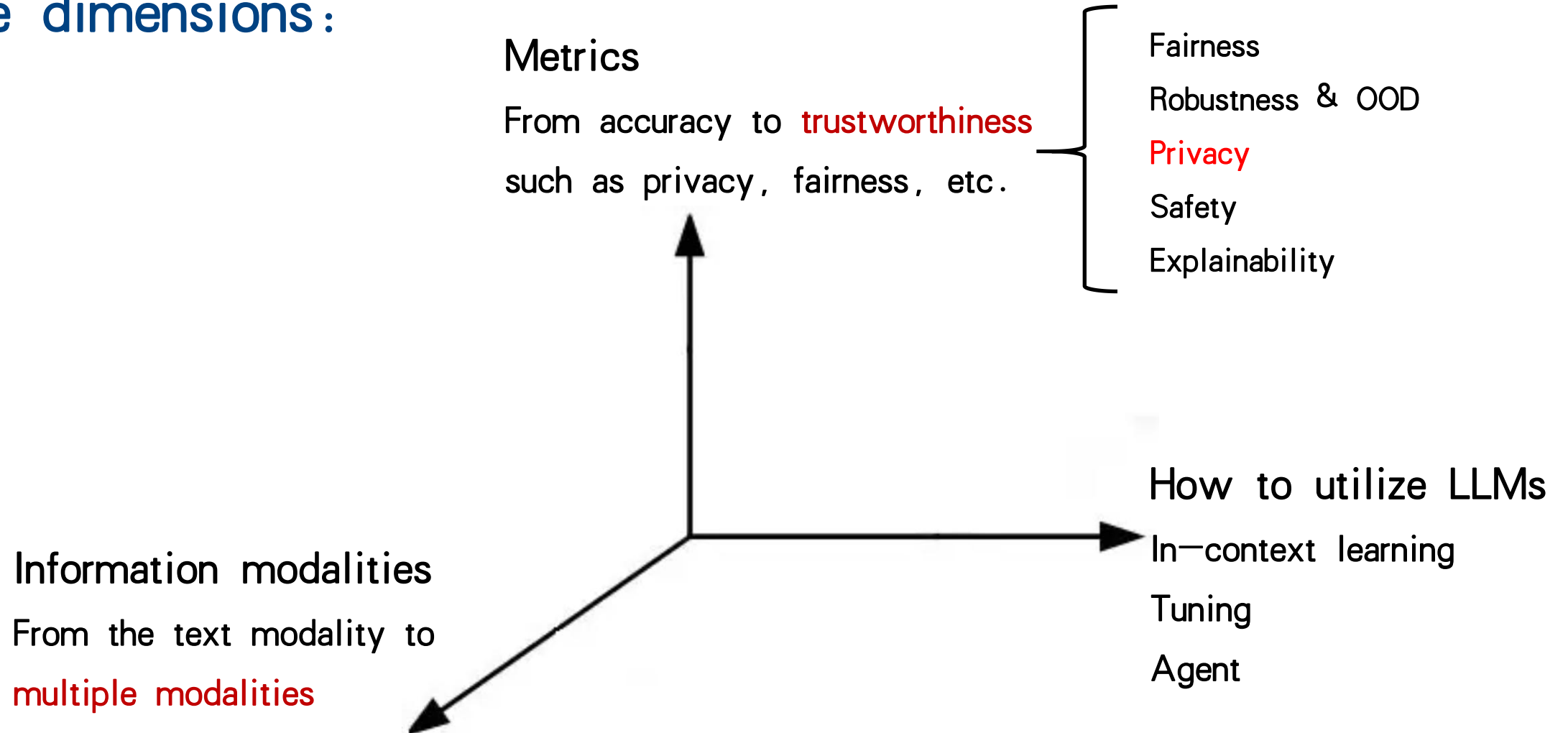


- Out-of-distribution (OOD) generalization
- Learning from movie scenario can directly recommend on books, and vice versa making the LLMRec has strong OOD generalization ability.



Progress of LLM4Rec

Three dimensions:



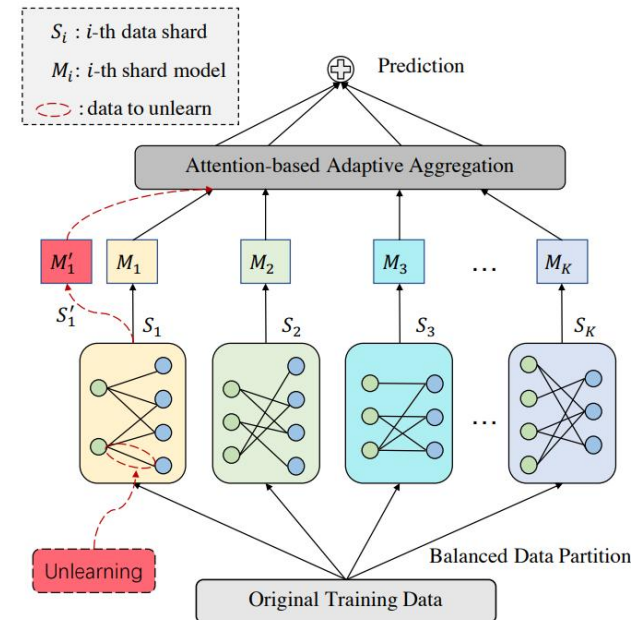
❑ Challenges for LLMRec Unlearning

- Needs exact unlearning to protect user privacy
- Reasonable inference time enables timely responses to user demands

❑ Existing works for LLM Unlearning

- Gradient update
- In-context Unlearning
- Simulates data labels

◆ ALL those methods can't handle challenge 1.

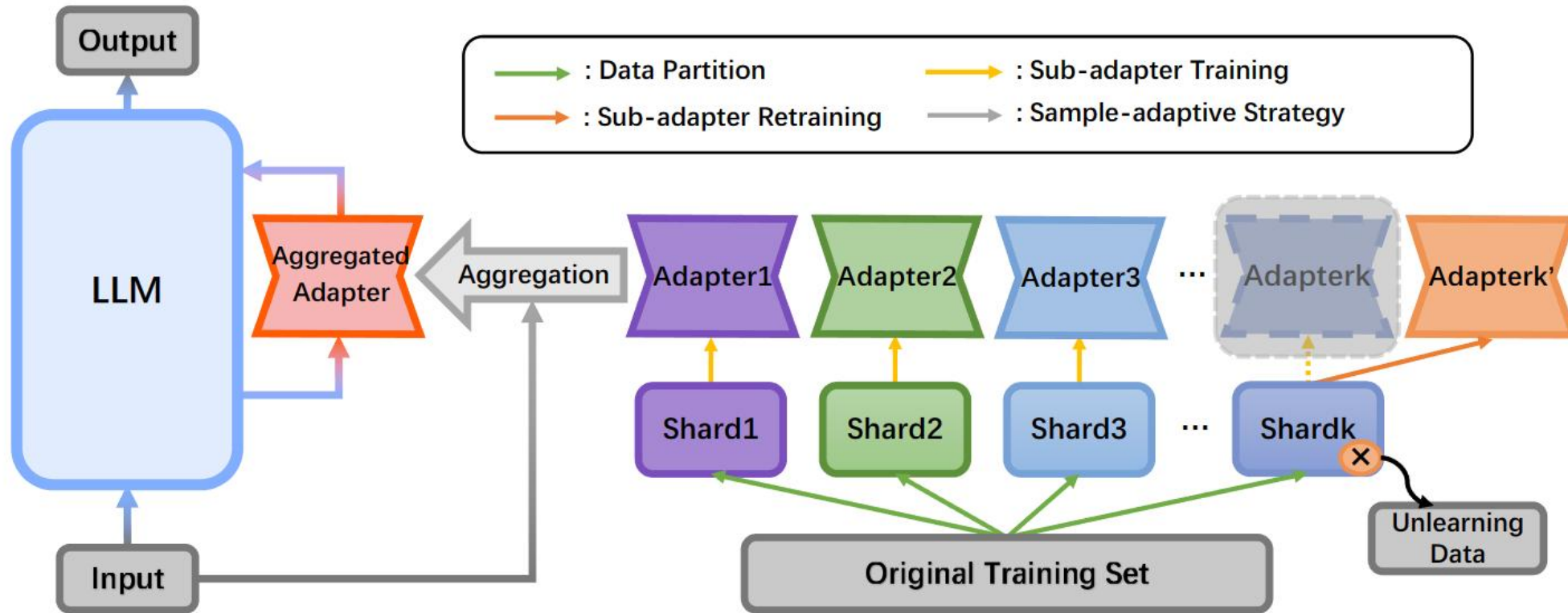


❑ Data-partition base retraining paradigm

- Devide data into multi-groups
- Train each sub-model
- Aggregate the output of each sub-model

◆ This paradigm can't handle challenge 2.

Privacy Unlearning



- Partition data based on semantics
- Differing from the previous paradigm, we leverage adapter weight aggregation during the inference phase.

Table 1: Comparison of different unlearning methods on recommendation performance, where ‘APA(D)’/‘APA(ND)’ represents APA implemented with decomposition/non-decomposition level aggregation, and Δ represents the gap between retraining and the unlearning method in terms of AUC. ‘Bef. Agg.’ represents the average *AUC* of the sub-model.

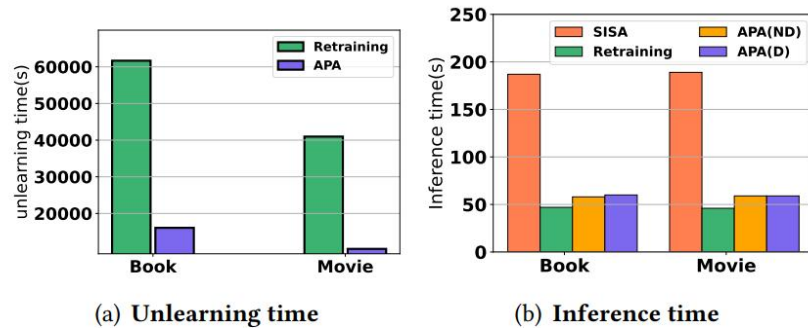


Figure 3: (a) Unlearning time of Retraining and APA. (b) Inference time of Retraining, SISA, APA(D), and APA(ND).

| Book | Retraining | SISA | GraphEraser | RecEraser | APA(D) | APA(ND) |
|-----------|------------|---------|-------------|-----------|---------|---------|
| Bef. Agg. | - | 0.6561 | 0.6393 | 0.6525 | 0.6578 | 0.6578 |
| AUC | 0.6738 | 0.6731 | 0.6646 | 0.6719 | 0.6738 | 0.6741 |
| Δ | - | -0.0007 | -0.0092 | -0.0019 | 0 | 0.0003 |
| Movie | Retraining | SISA | GraphEraser | RecEraser | APA(D) | APA(ND) |
| Bef. Agg. | - | 0.7003 | 0.6732 | 0.6699 | 0.6874 | 0.6874 |
| AUC | 0.7428 | 0.7055 | 0.6885 | 0.6918 | 0.7171 | 0.7172 |
| Δ | - | -0.0373 | -0.0543 | -0.051 | -0.0257 | -0.0256 |

- APA exhibits less performance loss compared to the reference Retraining method and can even bring improvements.
- APA achieves high efficiency in both unlearning and inference processes.

- E2URec aim to achieve unlearning by using two teachers.
- Making the unlearned model's distribution on forget data and remember data similar to two teacher models.

➤ Forgetting Teacher

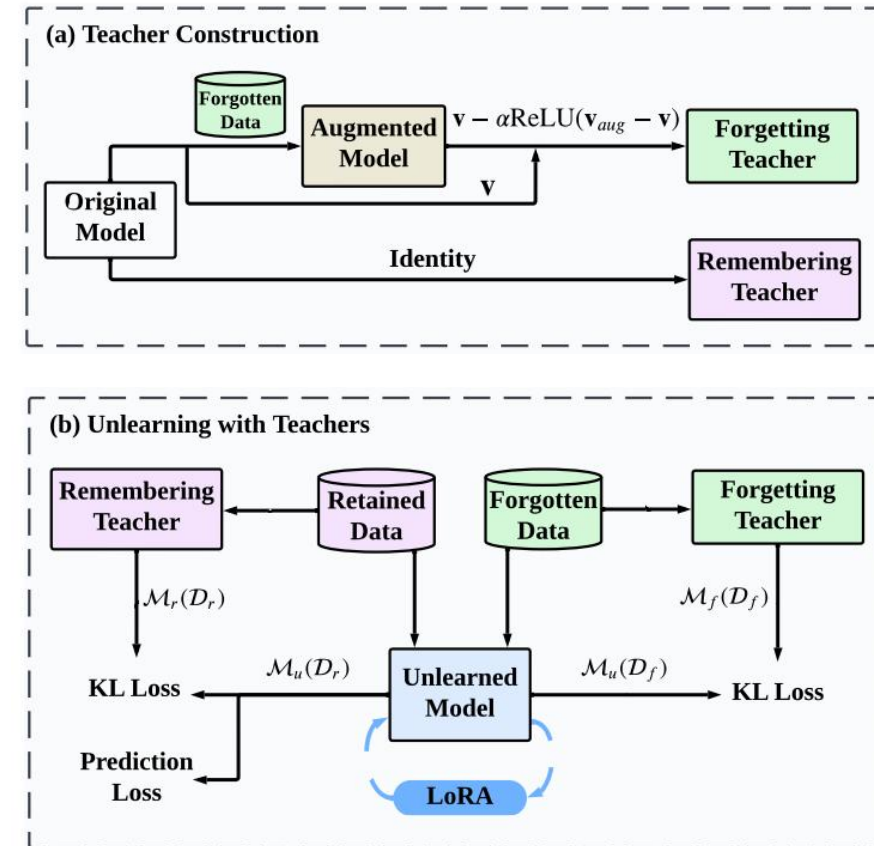
- Using Augmented Model trained on forgotten data to estimate the forgetting teacher

Unlearning with Teachers

- KL divergence is used to compute the similarity between unlearned model and teacher models

$$\min_{\theta} \text{KL}(\mathcal{M}_f(\mathcal{D}_f) \parallel \mathcal{M}_u(\mathcal{D}_f; \theta))$$

$$\min_{\theta} \text{KL}(\mathcal{M}_r(\mathcal{D}_r) \parallel \mathcal{M}_u(\mathcal{D}_r; \theta))$$



❑ Motivation of Incorporating Federated Learning

- Preserve data privacy when finetuning LLMs with user behavior data

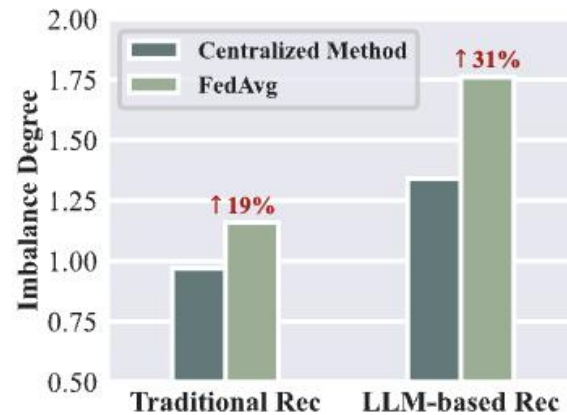
❑ Challenge of Incorporating Federated Learning

- Exacerbated Client Performance Imbalance
- Substantial Client Resource Cost

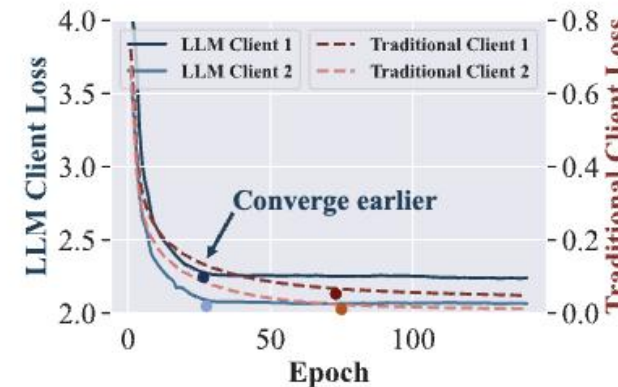


Dynamic Balance Strategy

Flexible Allocation Strategy



(a) Client Performance Imbalance Comparison



(b) Loss Convergence Comparison

Federated Learning

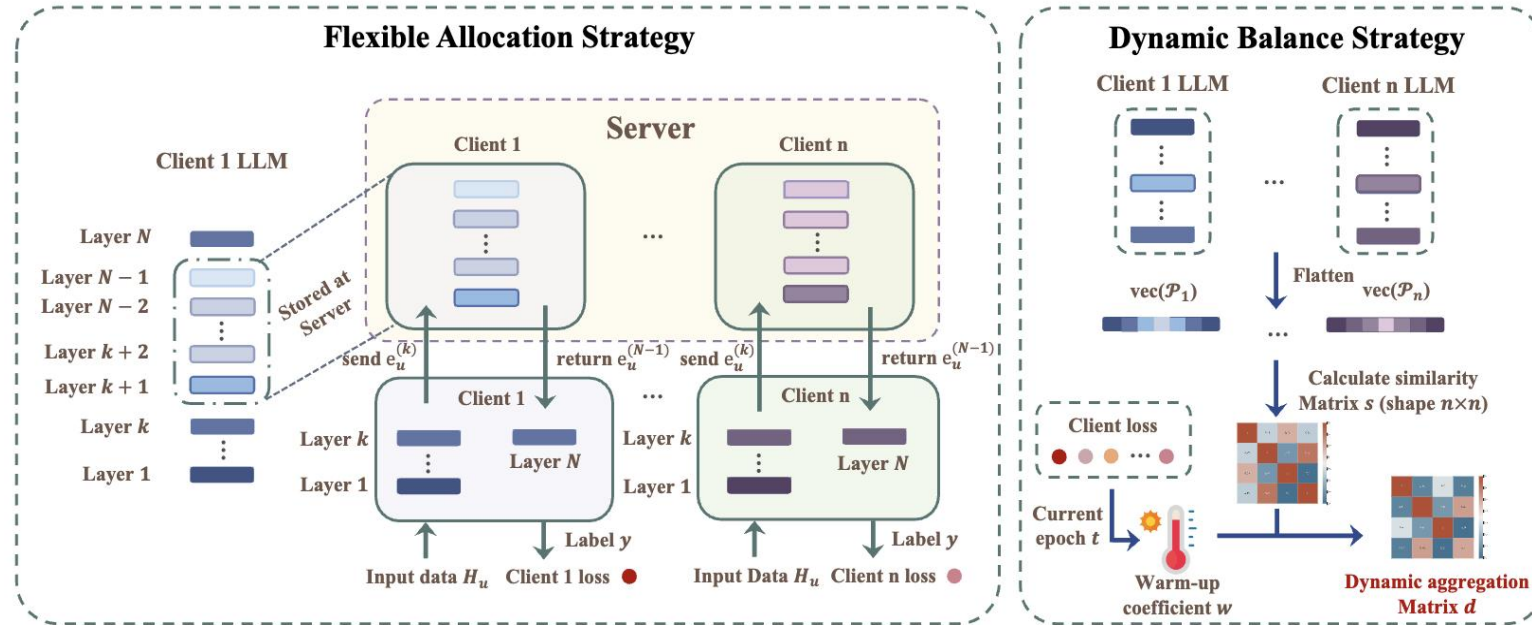


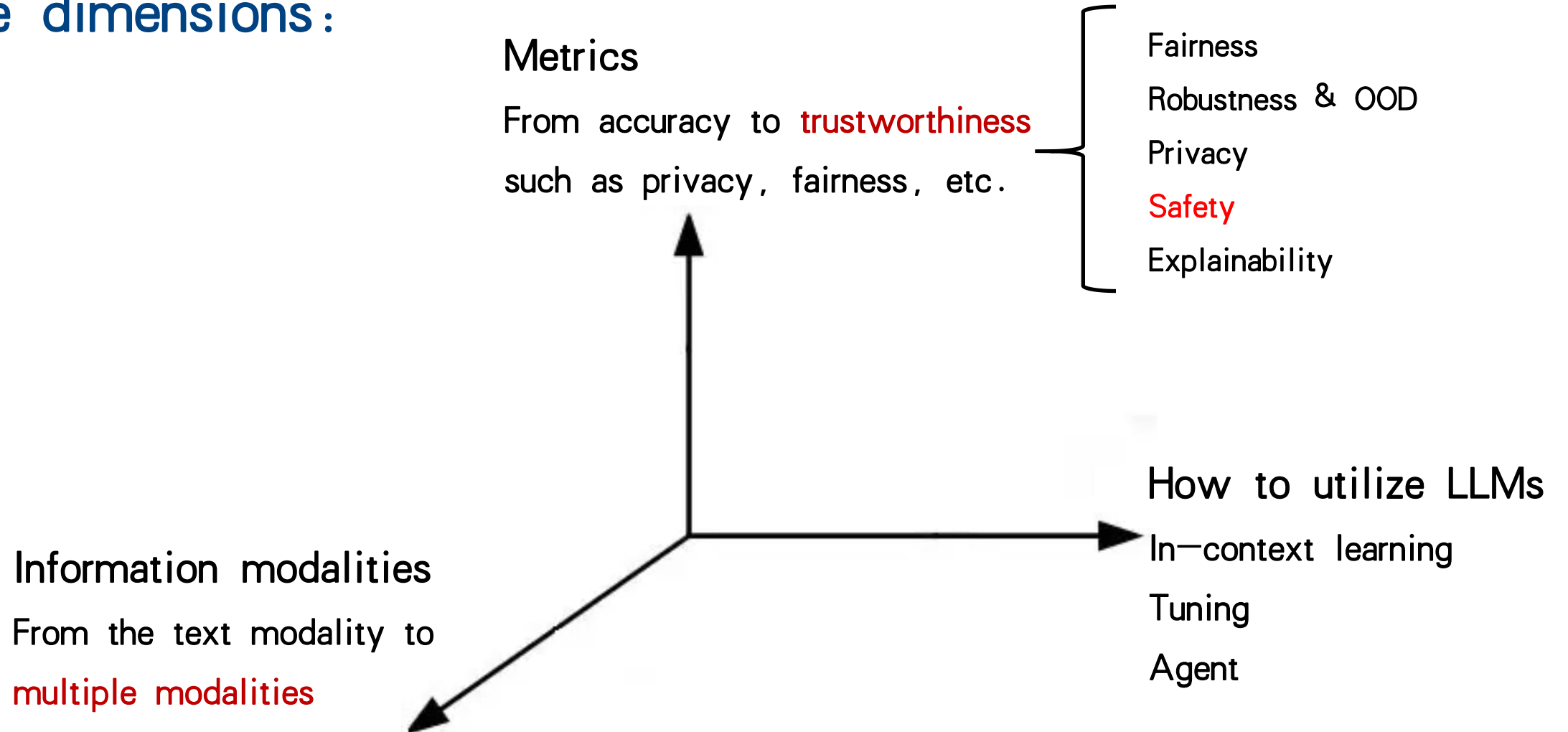
Figure 2: PPLR Structure. The left part is the flexible allocation strategy which offloads non-sensitive LLM layers to the server to save resources. The right part is the dynamic balance strategy which ensures relatively balanced performance across clients.

Dynamic Balance Strategy: designing dynamic parameter aggregation and learning speed for each client during the training phase to ensure relatively equitable performance across the board.

Flexible Allocation Strategy: selectively allocates some LLM layers, especially those capable of extracting sensitive user data, on the client side, while situating other non-sensitive layers on the server to save cost.

Progress of LLM4Rec

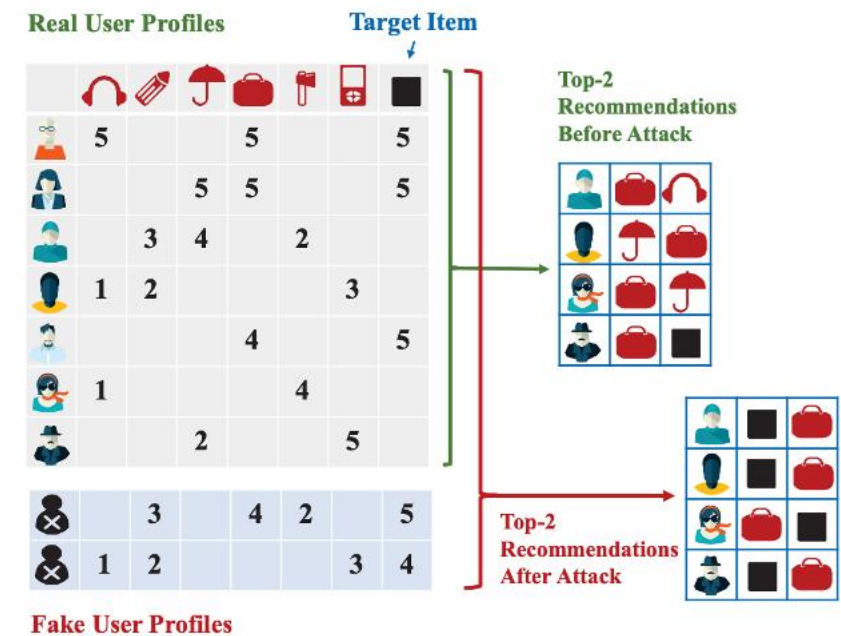
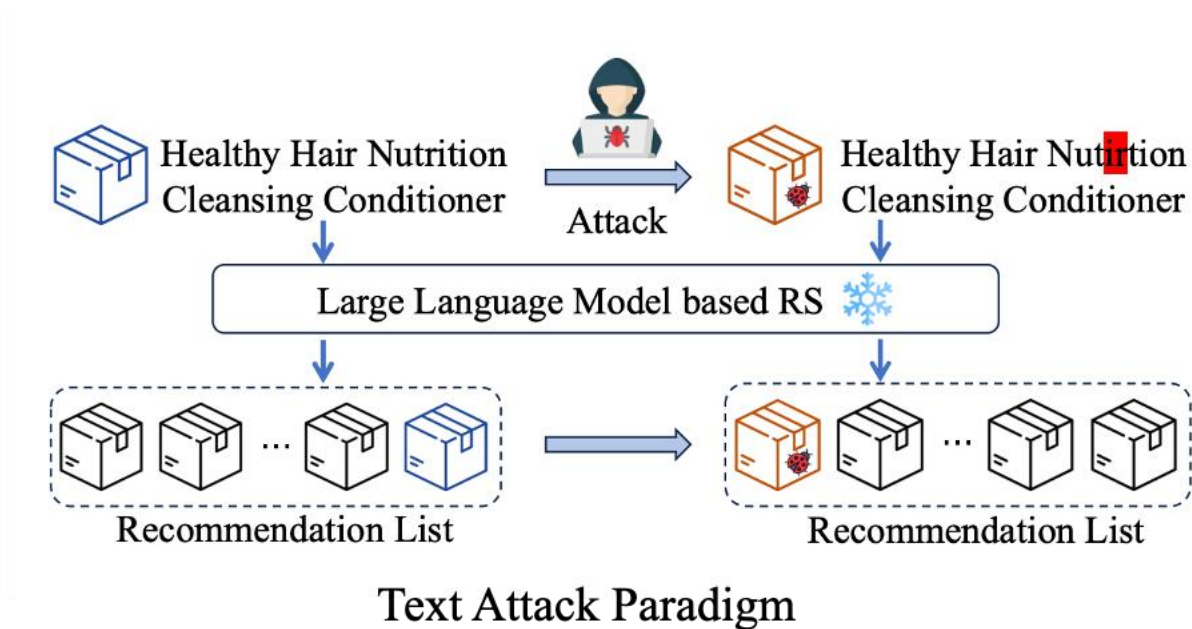
Three dimensions:



Text-centric paradigm raises new security issue of RS:

Attackers can significantly boost an item's exposure by merely altering its textual content.

- From text perspective
- Not involve training
- Hard to be detected



Attack:

Use GPT/textual attack methodologies to rewrite item description until reach the goal.

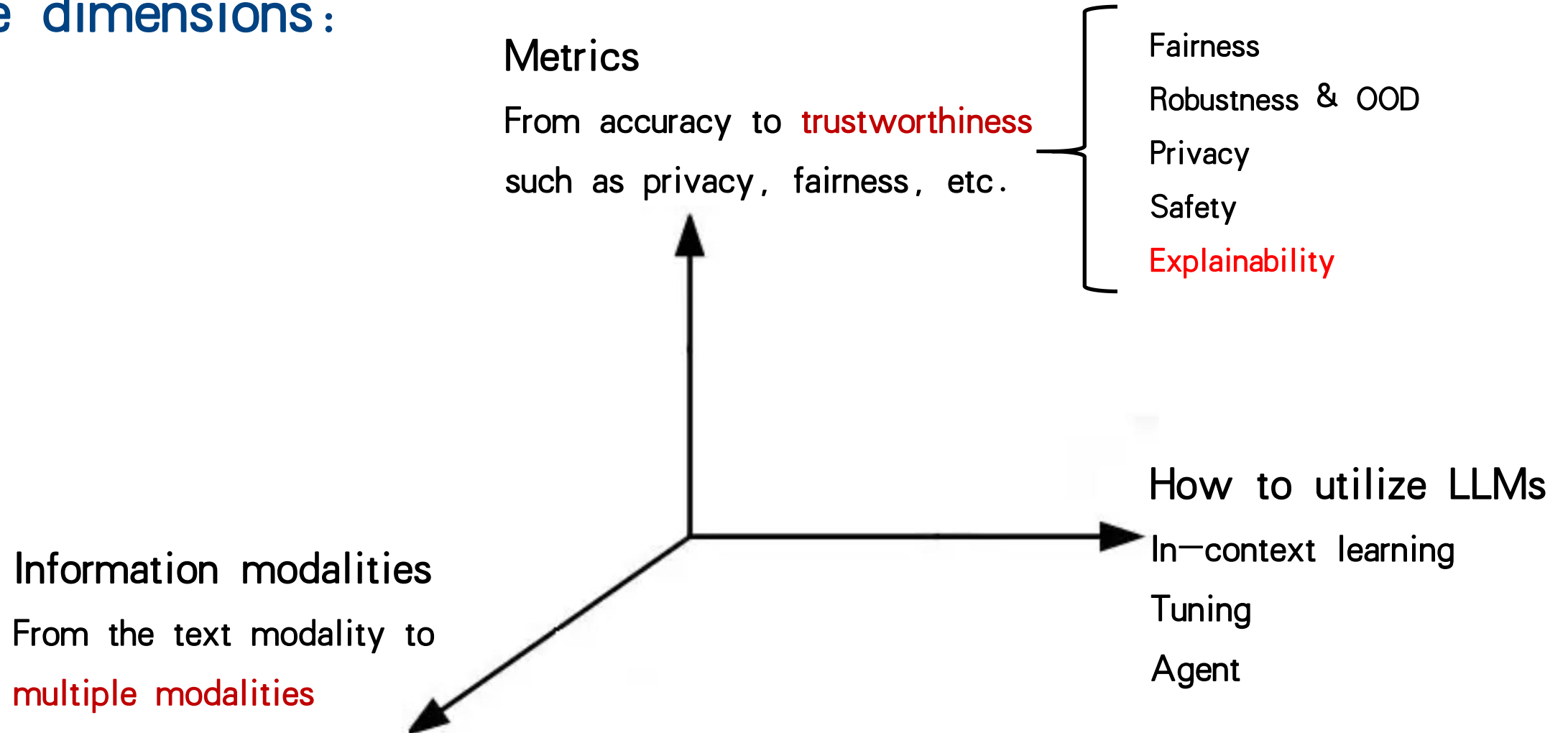
Prompt 1: You are a marketing expert that helps to promote the product selling. Rewrite the product title in <MaxLen> words to keep its body the same but more attractive to customers: <ItemTitle>.

Potential Defend:

Re-writing Prompt: Correct possible grammar, spelling and word substitution errors in the product title (directly output the revised title only): <AdversarialTitle>

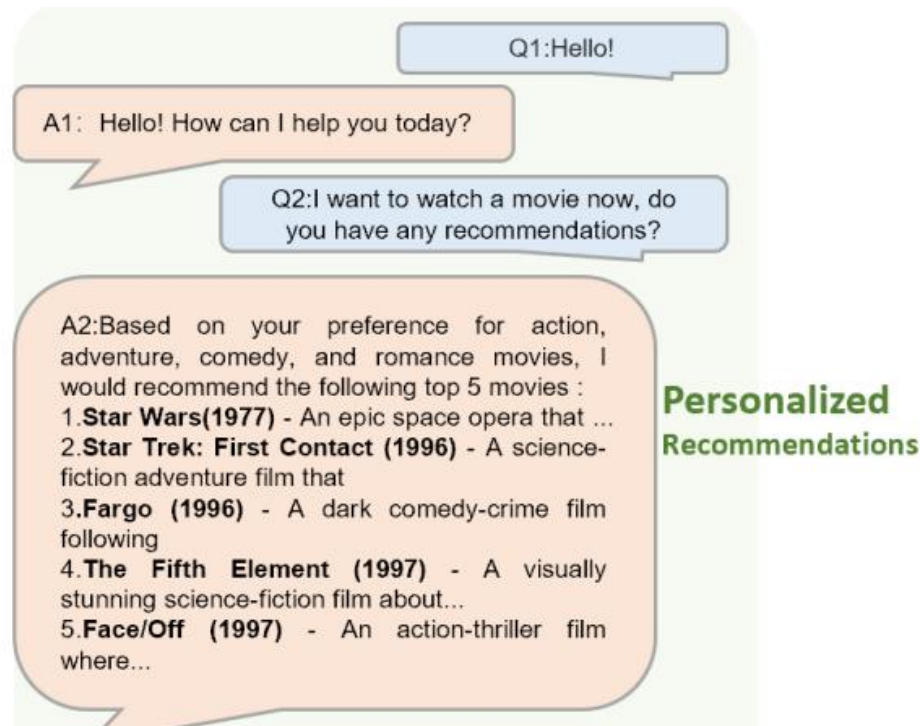
| Model | Text | Exposure |
|-------------|---|----------|
| Clean | Fisher-Price Fun-2-Learn Smart Tablet | 0.0076 |
| Trivial | Fisher-Price Fun-2-Learn Smart Tablet better selling | 0.0095 |
| GPT | Interactive Learning Tablet for Kids | 0.0335 |
| DeepwordBug | Fisher-Price Fun-2-Learn Smar Tmblet | 0.0335 |
| +Defense | Fisher-Price Fun-2-Learn Smart Tablet | 0.0076 |
| PunAttack | Fisher-Price Fun-2-Learn Sm'art Tablet | 0.0285 |
| +Defense | Fisher-Price Fun-2-Learn Smart Tablet | 0.0076 |
| Textfooler | Fisher-Price Fun-2-Learn Canny Table | 0.0768 |
| +Defense | Fisher-Price Fun-2-Learn Canine Table | 0.0756 |
| BertAttack | Fisher-Price Fun-2-Learn this Tablet | 0.0262 |
| +Defense | Fisher-Price Fun-2-Learn Tablet | 0.0190 |

Three dimensions:



Explainability

- Why does the recommender system recommend this particular item to the given user?
- LLM could directly generate explanations for their recommendations:



Q1: Hello!

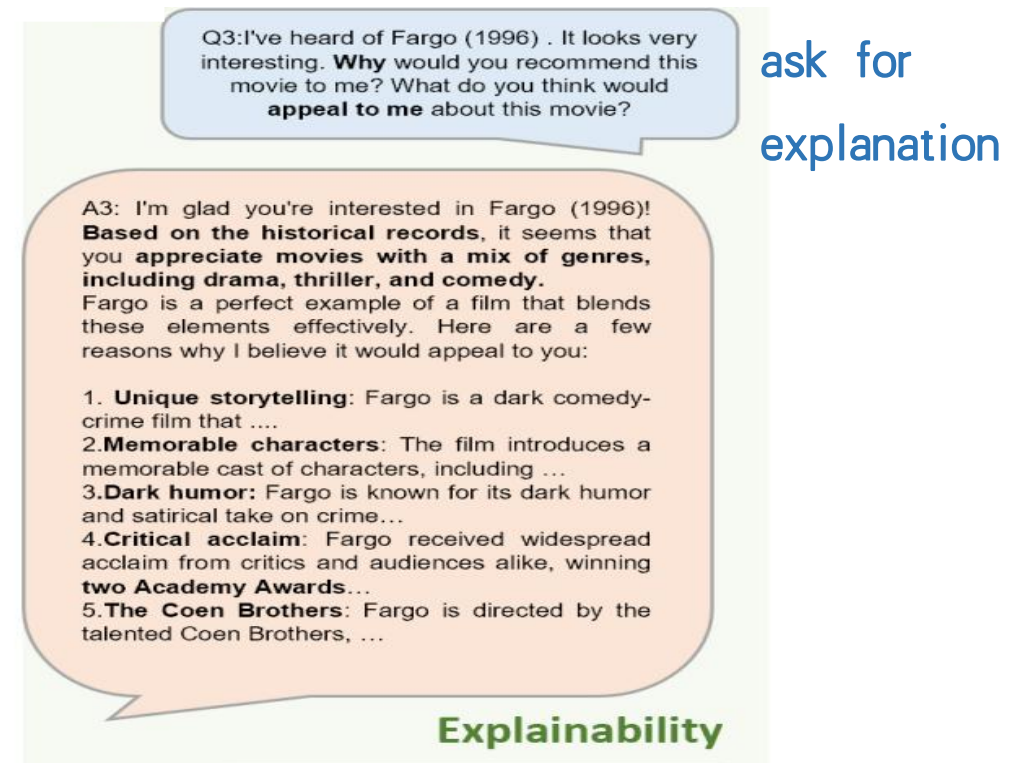
A1: Hello! How can I help you today?

Q2: I want to watch a movie now, do you have any recommendations?

A2: Based on your preference for action, adventure, comedy, and romance movies, I would recommend the following top 5 movies :

1. **Star Wars (1977)** - An epic space opera that ...
2. **Star Trek: First Contact (1996)** - A science-fiction adventure film that
3. **Fargo (1996)** - A dark comedy-crime film following
4. **The Fifth Element (1997)** - A visually stunning science-fiction film about...
5. **Face/Off (1997)** - An action-thriller film where...

Personalized Recommendations



Q3: I've heard of Fargo (1996) . It looks very interesting. **Why** would you recommend this movie to me? What do you think would **appeal to me** about this movie?

ask for explanation

A3: I'm glad you're interested in Fargo (1996)! **Based on the historical records**, it seems that you **appreciate movies with a mix of genres, including drama, thriller, and comedy**. Fargo is a perfect example of a film that blends these elements effectively. Here are a few reasons why I believe it would appeal to you:

1. **Unique storytelling**: Fargo is a dark comedy-crime film that
2. **Memorable characters**: The film introduces a memorable cast of characters, including ...
3. **Dark humor**: Fargo is known for its dark humor and satirical take on crime...
4. **Critical acclaim**: Fargo received widespread acclaim from critics and audiences alike, winning **two Academy Awards**...
5. **The Coen Brothers**: Fargo is directed by the talented Coen Brothers, ...

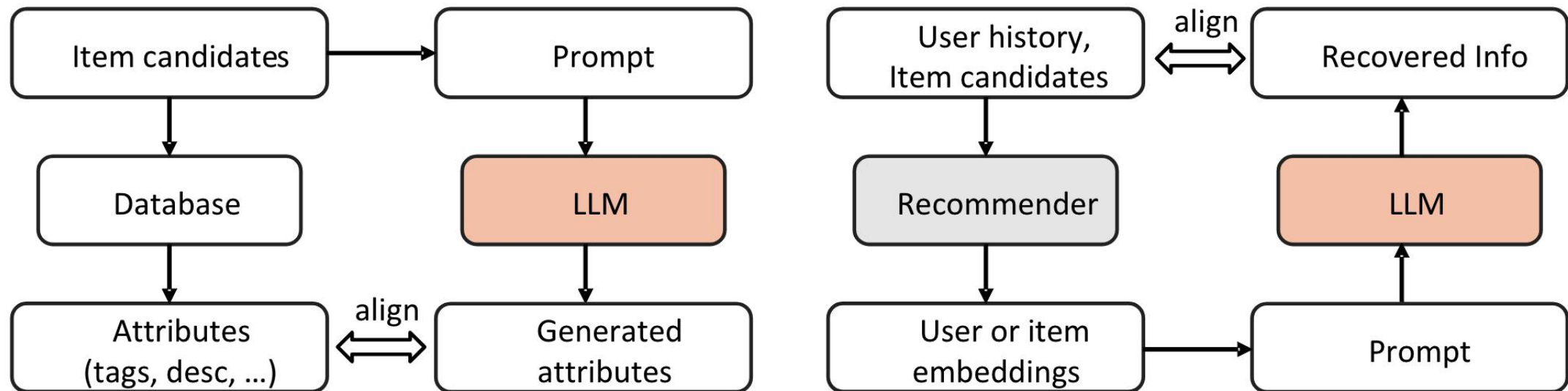
Explainability

[1] Gao Yunfan, et al. "Chat-rec: Towards interactive and explainable llms-augmented recommender".

[2] Junling Liu, et al. "Is ChatGPT a Good Recommender? A Preliminary Study".

Finetune LLM for Rec Explanation

- Design different tasks to finetune LLM for Recommendation Explanation
 - Besides finetuning for recommendation performance, RecExplainer finetunes LLM on different task related to recommendation explanation, such as Item discrimination and history reconstruction.



- Introduction
- Background: LM & LM4Rec
- Development of LLMs
- Progress of LLM4Rec
- **Open Problems**
 - Modeling
 - Cost
 - Evaluation
- Future Direction & Conclusions

Open Problems & Challenges

Three aspects:

Modeling

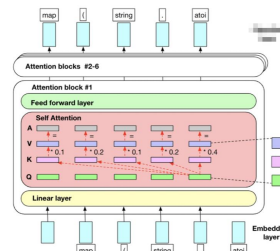
LLM: modeling text/language



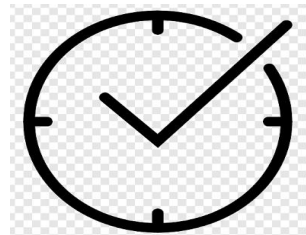
RecSys: modeling behaviors

Cost

LLM: high cost/delay



computation/m
emory—costly

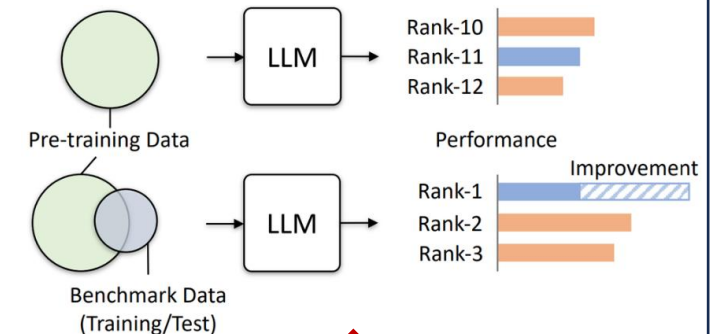


Real—time,
focus on cost

RecSys: cost sensitive

Evaluation

LLM: Trained on many data,
text—focused, language



Evaluation?

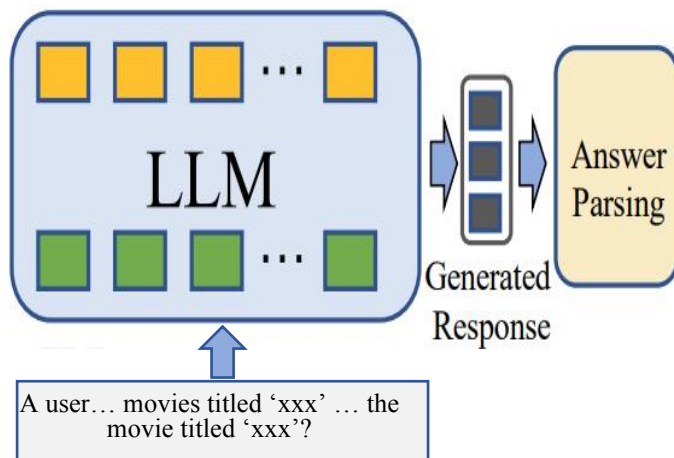
RecSys research: interactions,
offline, anonymous data

Modeling: User/Item Representation

- **Recommendation: user behavior modeling**
- **How should we represent user behaviors (represent users/items) in LLM4Rec?**

LLM4Rec methods

User/Item: Text

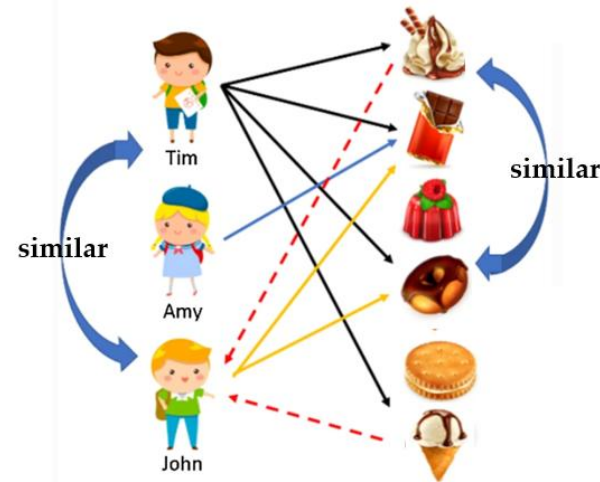


May lack of some information

Textually similar item
may have distinct collab.
info.

Traditional methods

User/Item: features + ID



LLMs are constructed using texts,
making the representation of users/items
in texts the natural choice.

Features (content) alone **are insufficient** to
depict users and items, mainly behavioral
similarities (**collaborative info**). **IDs are**
utilized.

Modeling: User/Item Representation

Integrate collaborative information:

- Why?



LLM Rec vs Traditional CF Model:

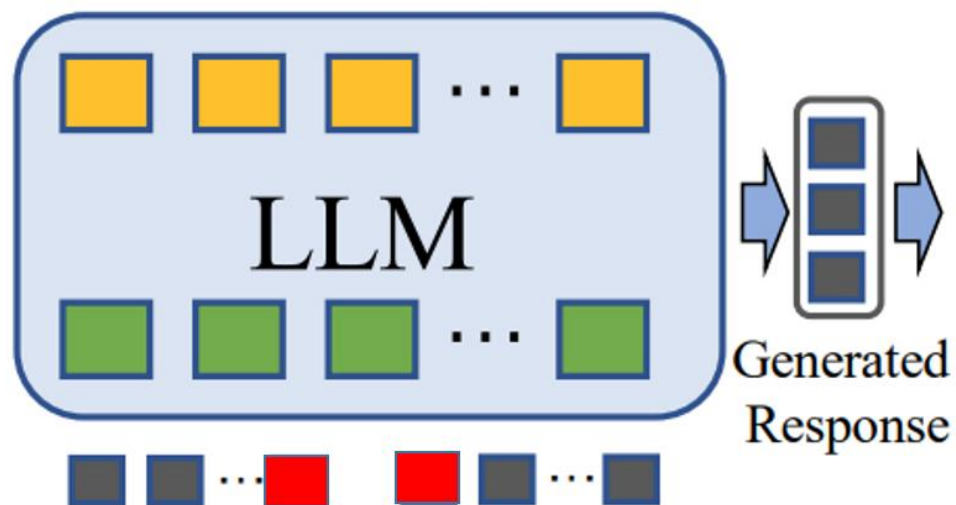
#:Excellent at old-start scenarios

#: Poor at warm-start scenarios

Modeling: User/Item Representation

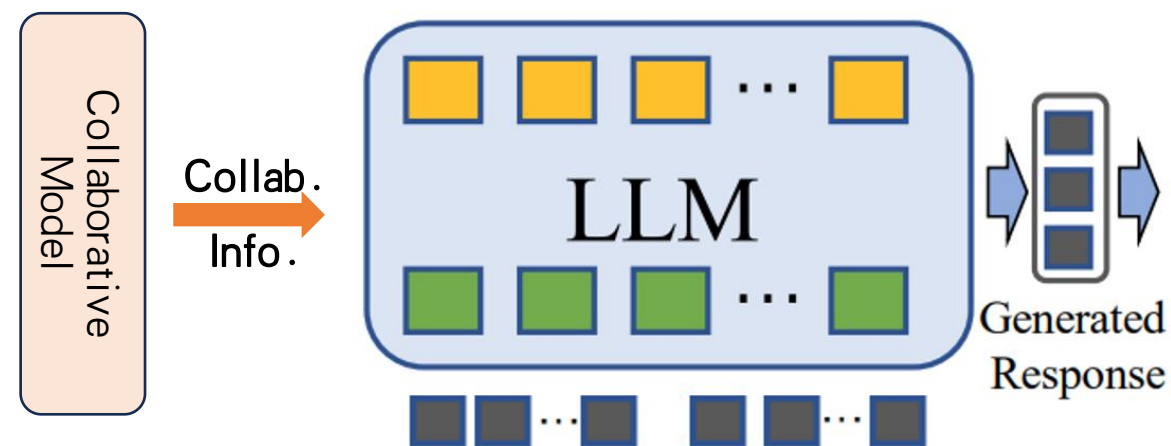
Integrate collaborative information: How?

#1: learn user/item embedding by fitting interactions



Add tokens to represent users and items in LLM
Learn LLM token embeddings by fitting interaction data
↓
Large space, low learning efficacy
Design better tokenization

#2: Feed the collaborative information extracted by external models into LLM

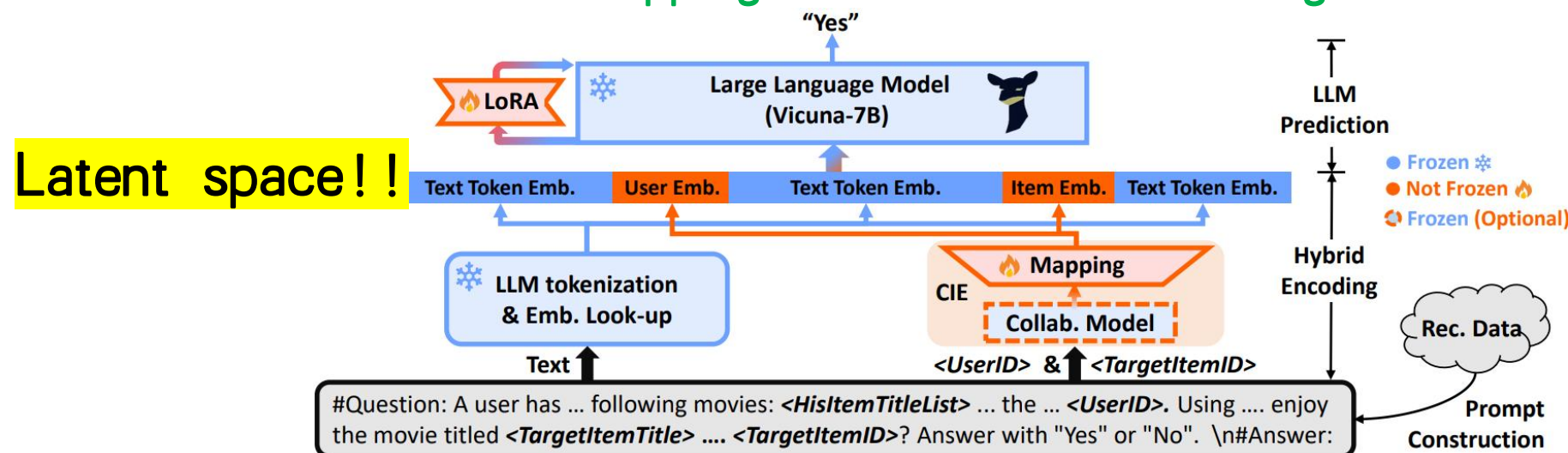


Extract collaborative information with traditional models
Feed the extracted information into LLMs
↓
Focus on how to feed the info.

Modeling: User/Item Representation

Integrate collaborative information: **feed external collaborative information into LLM**

- Work#1: CoLLM — **mapping collaborative embeddings into LLM's Latent space**



- **Prompt construction:** add <UserID> and <TargetID> for placing the Collab. Info.
- **Hybrid Encoding:**
 - text: tokenization & LLM emb Lookup;
 - user/item ID: CIE --- extract info with collab. model (**low rank**), then map it to the token embedding space
- LLM prediction: add a LoRA module for recommendation task learning

Modeling: User/Item Representation

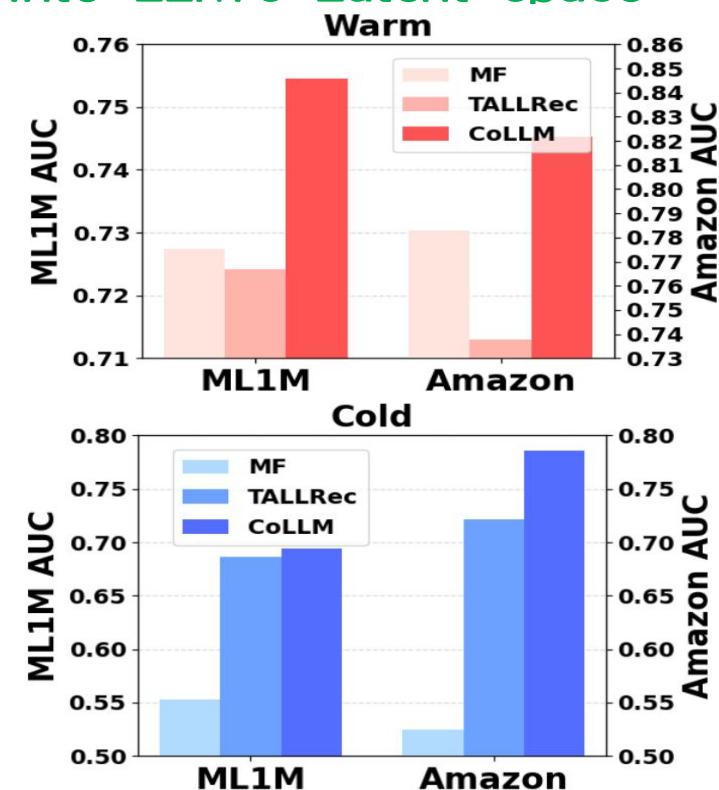


Integrate collaborative information: **feed external collaborative information into LLM**

- Work#1: CoLLM — **mapping collaborative embeddings into LLM's Latent space**

Overall Performance

| Dataset | | ML-1M | | | Amazon-Book | | |
|---------|----------------|--------|--------|-----------|-------------|--------|-----------|
| Methods | | AUC | UAUC | Rel. Imp. | AUC | UAUC | Rel. Imp. |
| Collab. | MF | 0.6482 | 0.6361 | 10.3% | 0.7134 | 0.5565 | 12.8% |
| | LightGCN | 0.5959 | 0.6499 | 13.2% | 0.7103 | 0.5639 | 10.7% |
| | SASRec | 0.7078 | 0.6884 | 1.9% | 0.6887 | 0.5714 | 8.4% |
| LLMRec | ICL | 0.5320 | 0.5268 | 33.8% | 0.4820 | 0.4856 | 48.2% |
| | Soft-Prompt | 0.7071 | 0.6739 | 2.7% | 0.7224 | 0.5881 | 10.4% |
| | TALLRec | 0.7097 | 0.6818 | 1.8% | 0.7375 | 0.5983 | 8.2% |
| Ours | CoLLM-MF | 0.7295 | 0.6875 | - | 0.8109 | 0.6225 | - |
| | CoLLM-LightGCN | 0.7100 | 0.6967 | - | 0.7978 | 0.6149 | - |
| | CoLLM-SASRec | 0.7235 | 0.6990 | - | 0.7746 | 0.5962 | - |



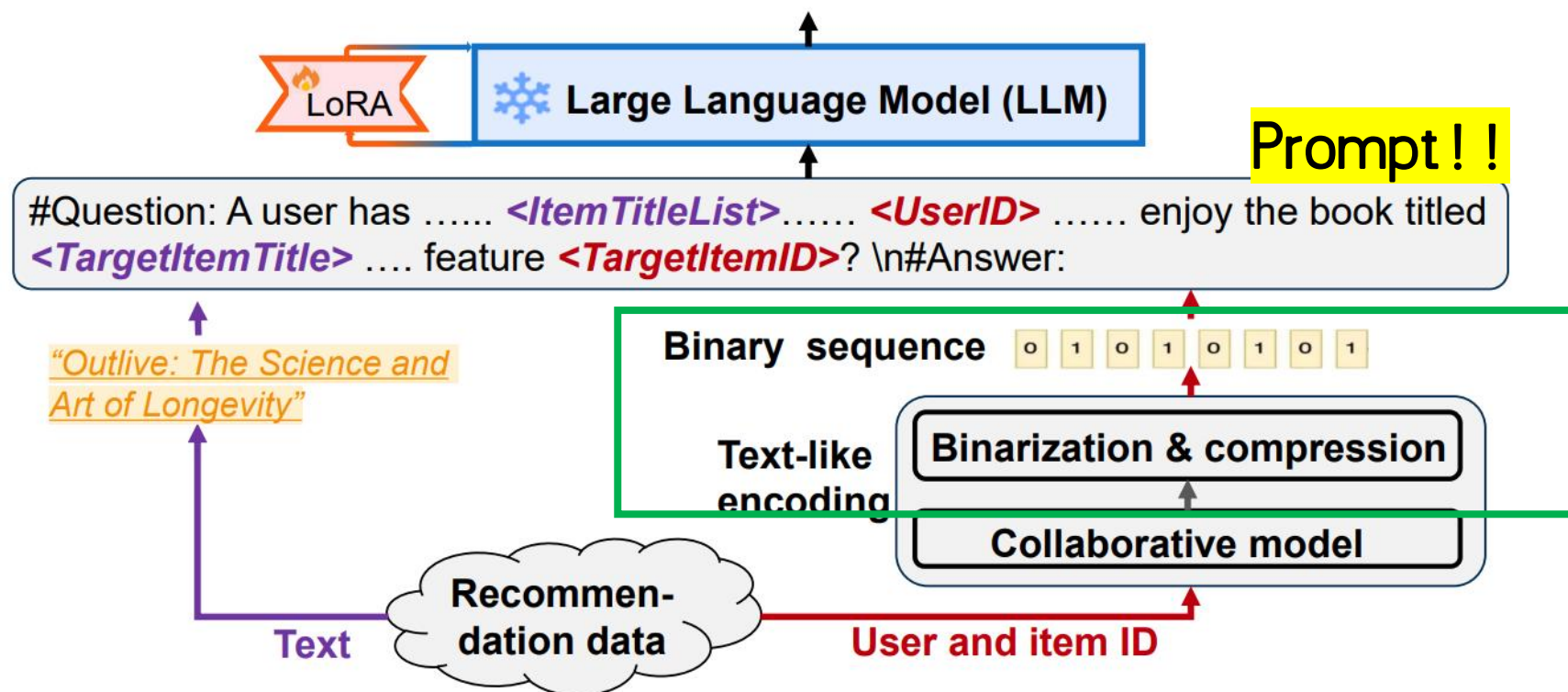
- CoLLM brings performance improvements over traditional models and current LLM Rec in most cases

- CoLLM significantly improves the warm performance of LLM4Rec, while ensuring cold performance

Modeling: User/Item Representation

Integrate collaborative information: feed external collaborative information into LLM

- Work#2: BinLLM — Encoding collaborative embeddings in a text-like format for LLM



transform the collaborative embeddings into **binary sequence**, treating them as **textual features** directly usable by LLMs

- LLMs could naturally perform bitwise operations
- Binarizing collaborative embeddings could keep performance.

Feed collaborative information into prompts

Modeling: User/Item Representation



Integrate collaborative information: **feed external collaborative information into LLM**

- More works

[1] Liao et al. Large Language—Recommendation Assistant. ArXiv 2023.

[2] Yang et al. Large Language Model Can Interpret Latent Space of Sequential Recommender. ArXiv 2023.

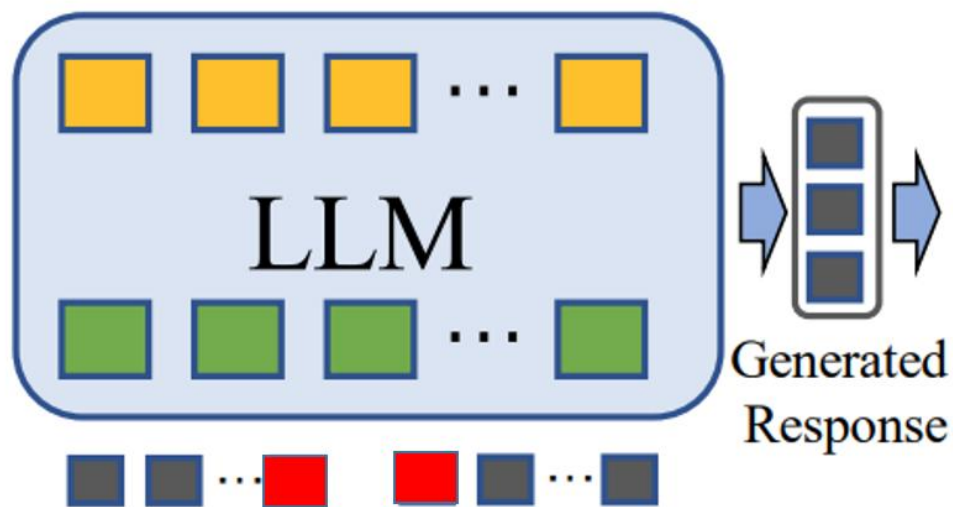
[3] Yu et al. "RA—Rec: An Efficient ID Representation Alignment Framework for LLM—based Recommendation." arXiv 2024.

[4] Li et al. "E4SRec: An elegant effective efficient extensible solution of large language models for sequential recommendation." arXiv 2023.

Modeling: User/Item Representation

Integrate collaborative information: **learn user/item-specific token embedding**

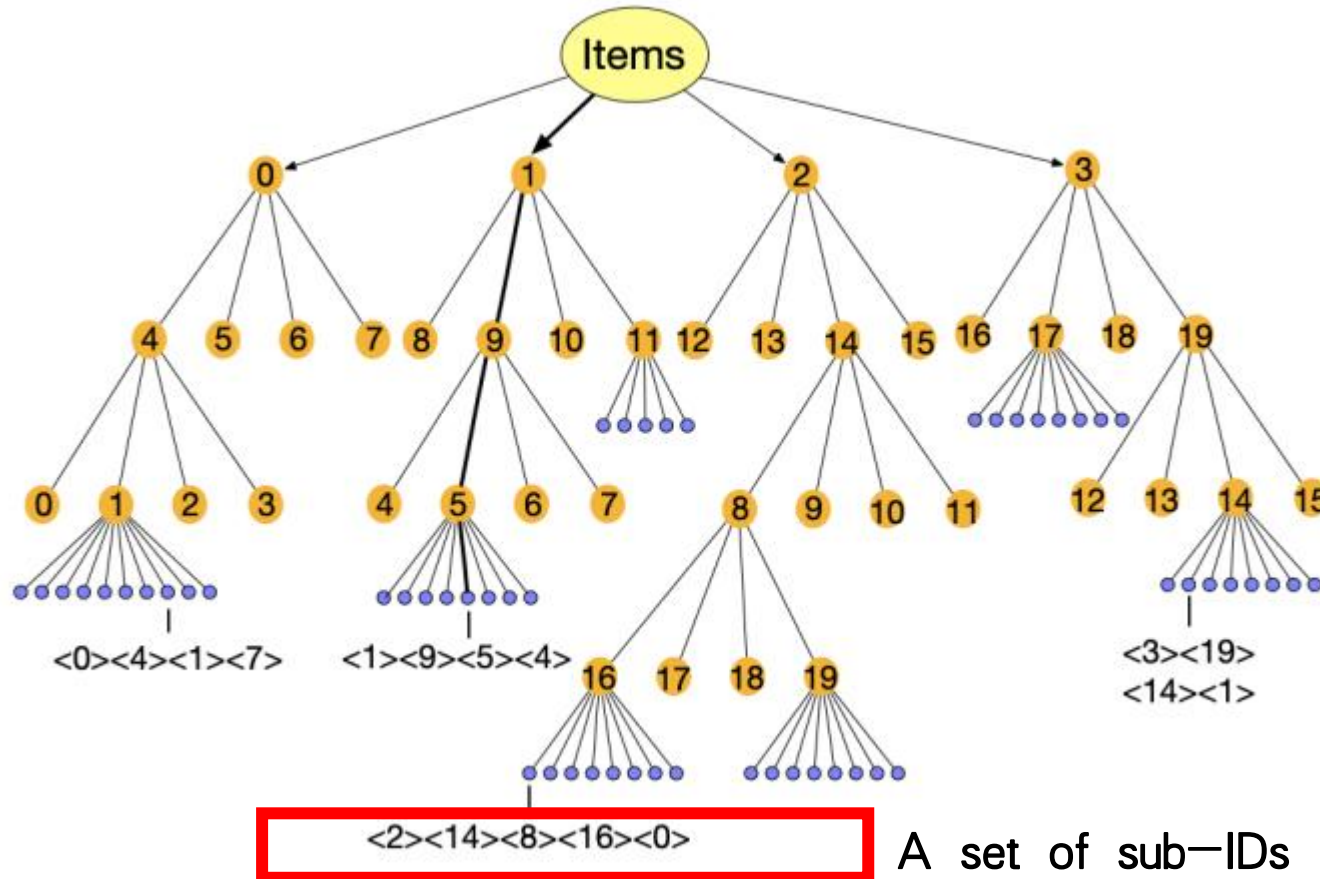
learn user/item embedding by
fitting interactions



- Add new tokens to represent users and items in LLM
- Learn LLM token embeddings by fitting interaction data
- **Default choice: Random IDs as tokens**
- **Issues:**
 - **Large learning space --- low learning efficacy**
 - **Semantic gaps between text tokens and recommendation tokens**
 - **Generalization issues --- cannot deal with new items**

Integrate collaborative information: **learn user/item-specific token embedding**

- Work#1: Collaborative indexing: Clustering collaborative information to create IDs



- Generate collaborative embeddings
- Hierarchically cluster the collaborative embedding
- generate IDs based on category indices.

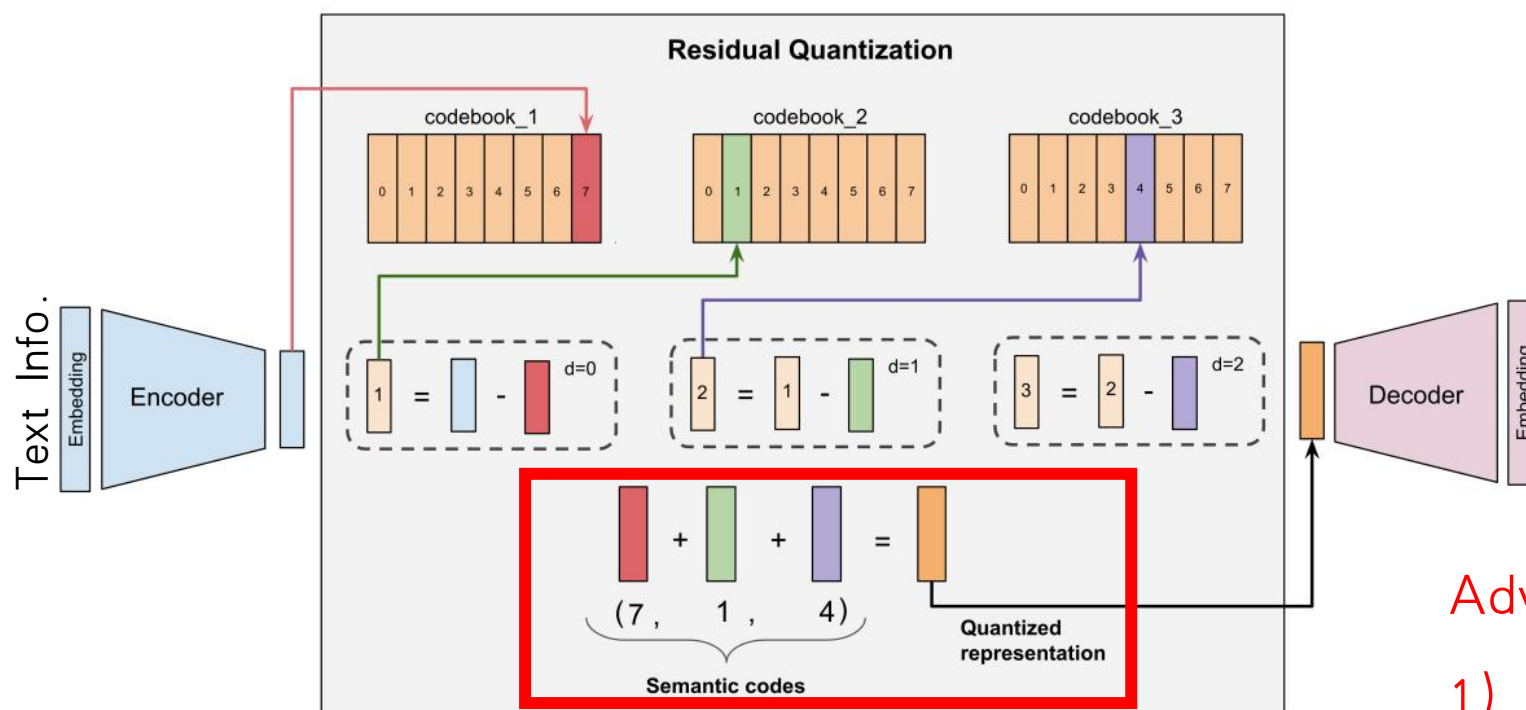
Advantages:

- 1) Add constraints on item IDs
 - 2) Reduce the token spaces
- Increase the learning efficacy.

Modeling: User/Item Representation

Integrate collaborative information: **learn user/item-specific token embedding**

- Work#2: Semantic-aware ID (Tiger/LC-Rec): quantizing text embedding to generate IDs



Quantization: RQ-VAE

- Convert text content information into embeddings
- Quantization: represent the text embedding with several sub-embeddings generating semantic ID
- Several sub-IDs form a semantic ID

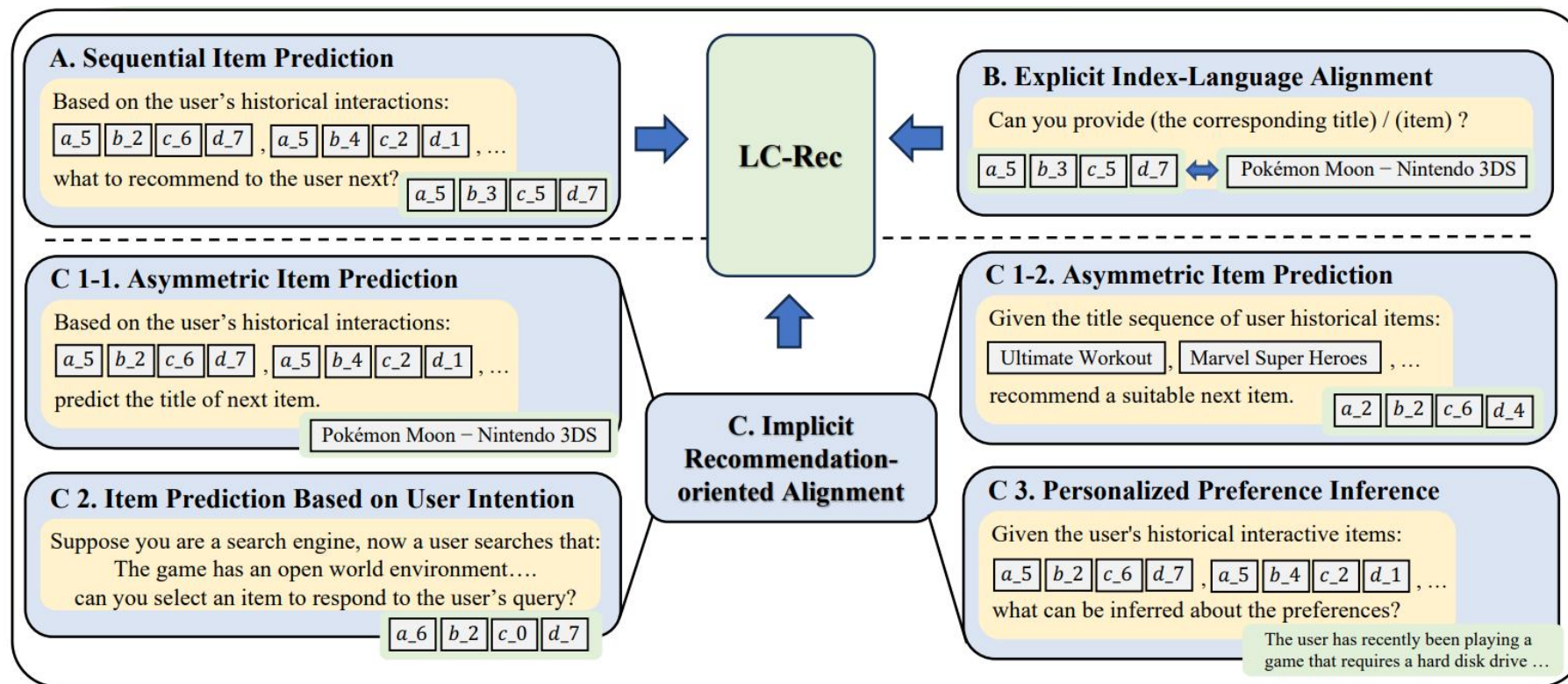
Advantages:

- 1) Reduce the token spaces, $N \rightarrow K \cdot N^{1/K}$
- 2) Could deal with new items

Modeling: User/Item Representation

Integrate collaborative information: **learn user/item-specific token embedding**

- Work#2: Semantic-aware ID (LC-Rec): quantizing text embedding to generate IDs



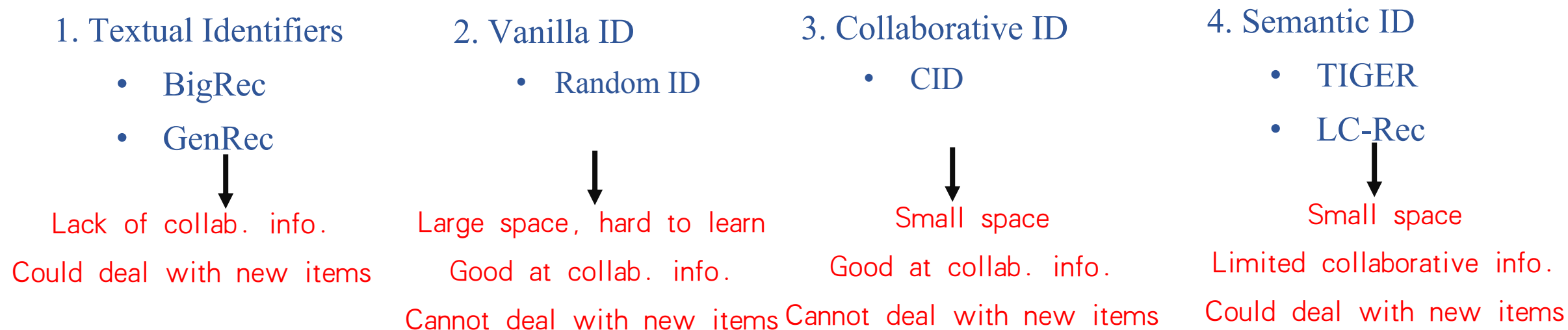
Use text tokens (like titles) and semantic IDs to predict each other, bridging the gap between text and collaborative information

Modeling: User/Item Representation



Integrate collaborative information: learn user/item-specific token embedding

- **Summary of tokenizer (item-side):**



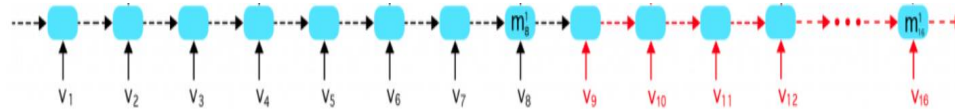
- **Open Problem:**

- Tokenization on user behaviors
- Tokenization on cross-domain items

Modeling: Lifelong Modeling

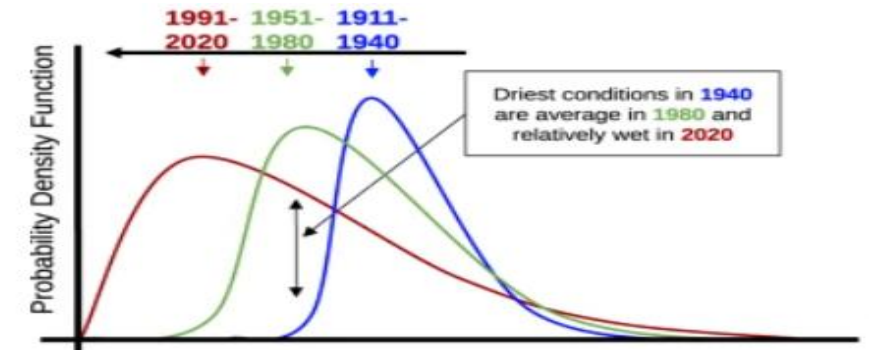
- Users are anticipated to engage with the recommender system continuously
- Raise the need of lifelong behavior modeling for users

Lifelong sequential behavior modeling



- The length of historical interaction sequences grows significantly, easily exceeding 1000
- How to model such long sequence effectively?

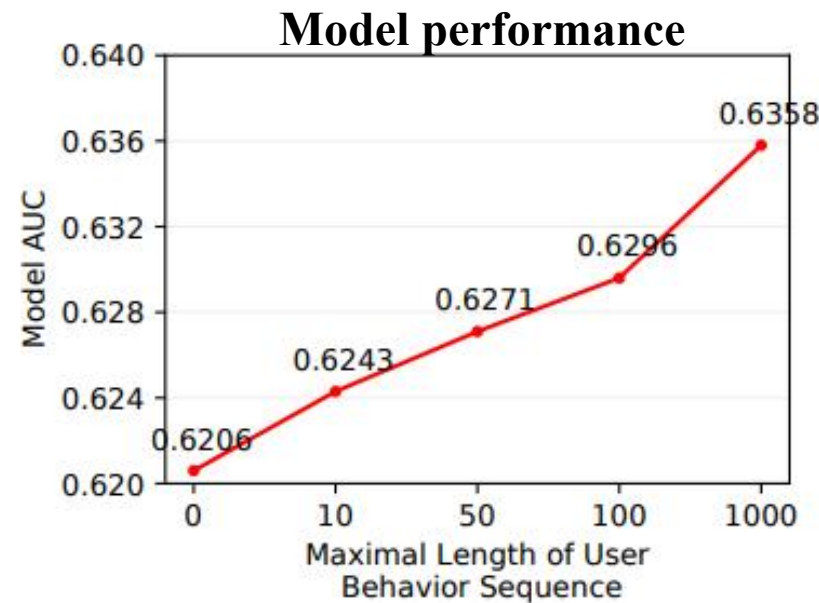
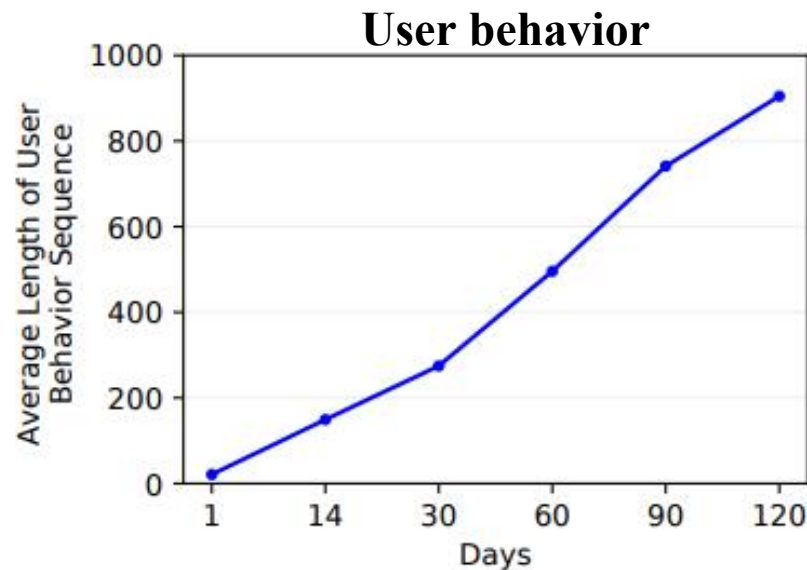
Continual learning



- User interests drift with time going
- How to continuously/incremental learn user interests?

Lifelong sequential behavior modeling:

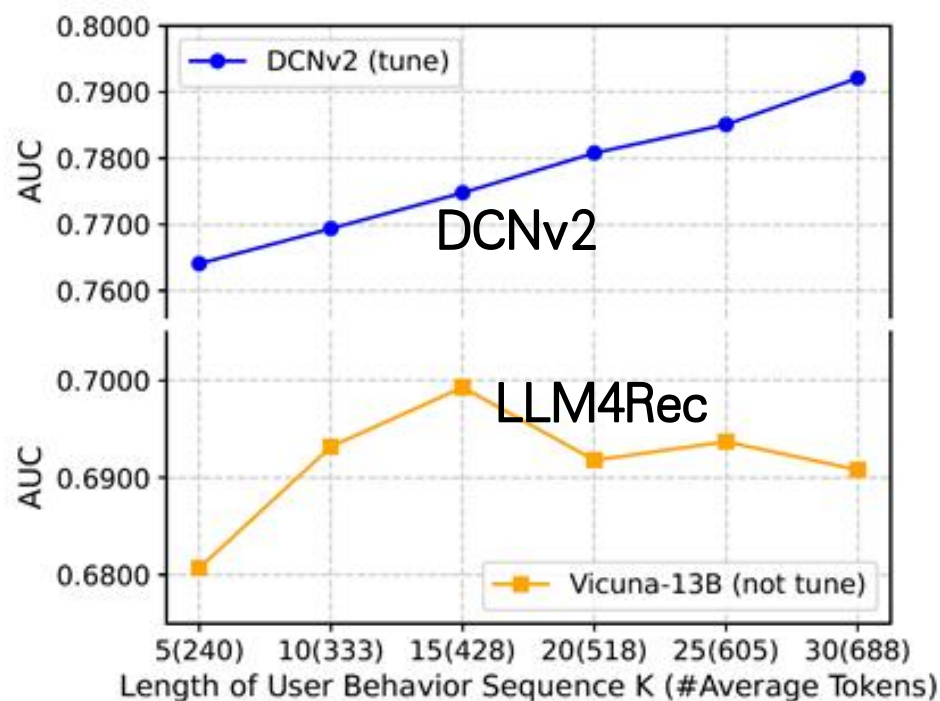
- A longer history signifies **richer personalization information**, and modeling this can lead to heightened prediction accuracy.



An example in the advertising system in Alibaba.

Lifelong sequential behavior modeling:

LLM cannot effectively model long user Behavior sequence

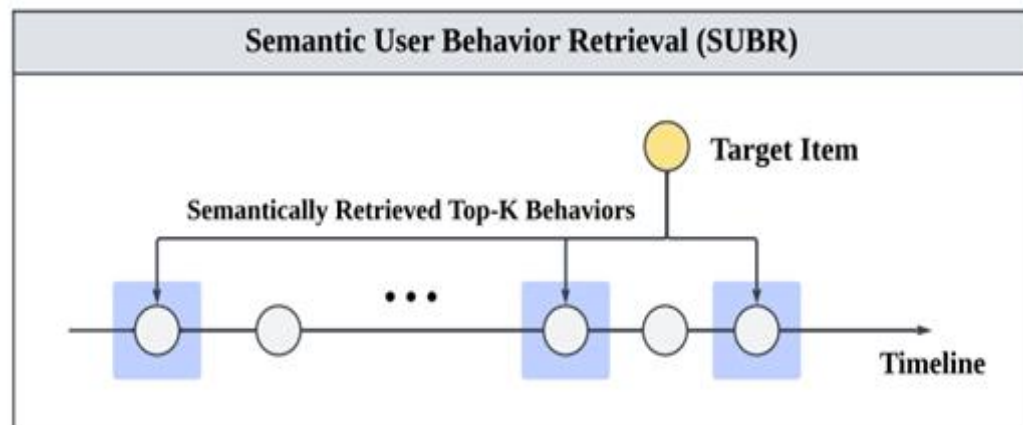


- Extending user behavior sequences **doesn't necessarily enhance recommendation performance**, even if the input length is far below the length limit of LLMs (e.g., Vicuna-13B has an upper limit of 2048 tokens).

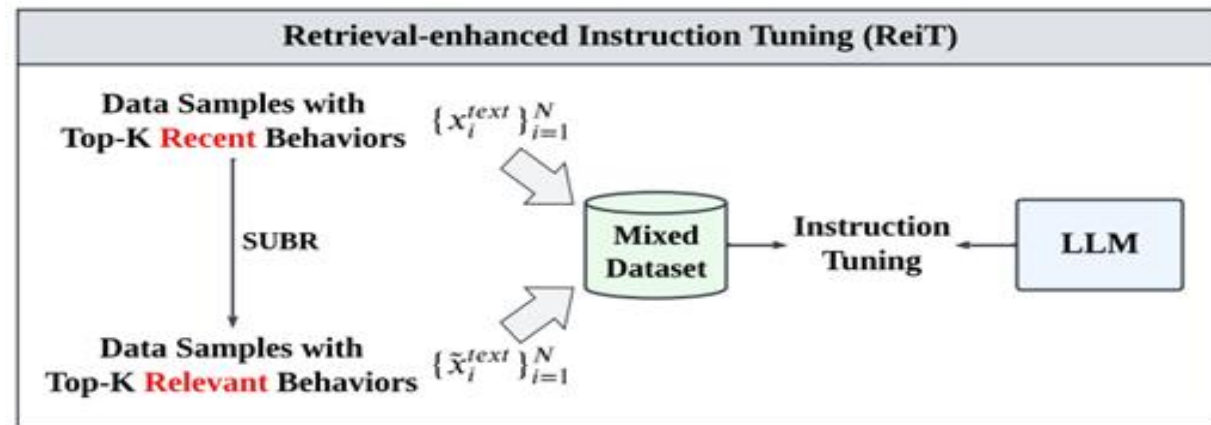
Lifelong sequential behavior modeling:

- **Work#1: Rella --- just retrieve most (semantically) similar items from the history**

Step1: For a **target item**, retrieve the top-K **semantically similar items from the history**, forming a new sample



Step2: Leverage the original sample and new sample to fine tune LLM for recommendation

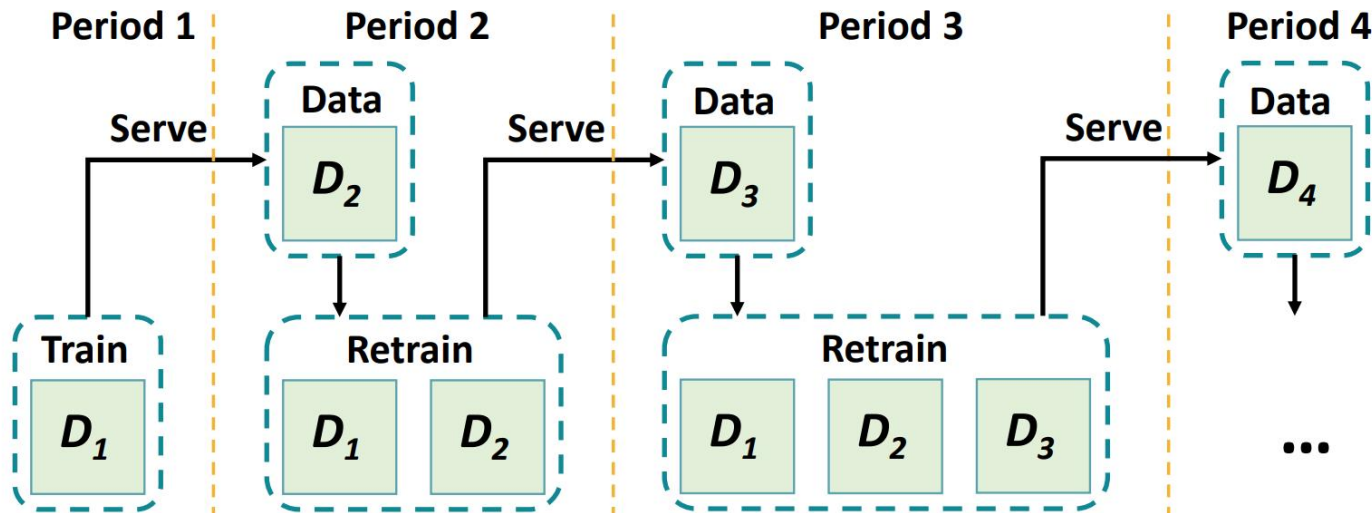


- Limitations: heavily depends on “target attention, not applicable when the input lacks target items.
- Future: may need to explore other solutions like memory.

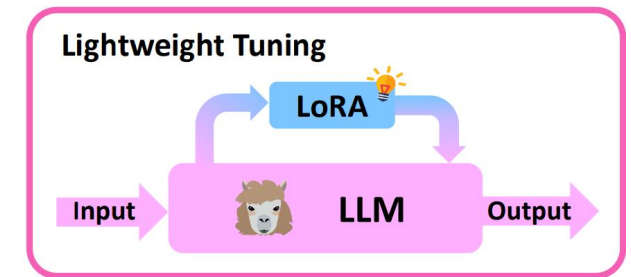
Modeling: Lifelong Modeling

Continual learning:

- How to incrementally learn user interests?
- There is work [1] studying the common used methods: periodic retraining

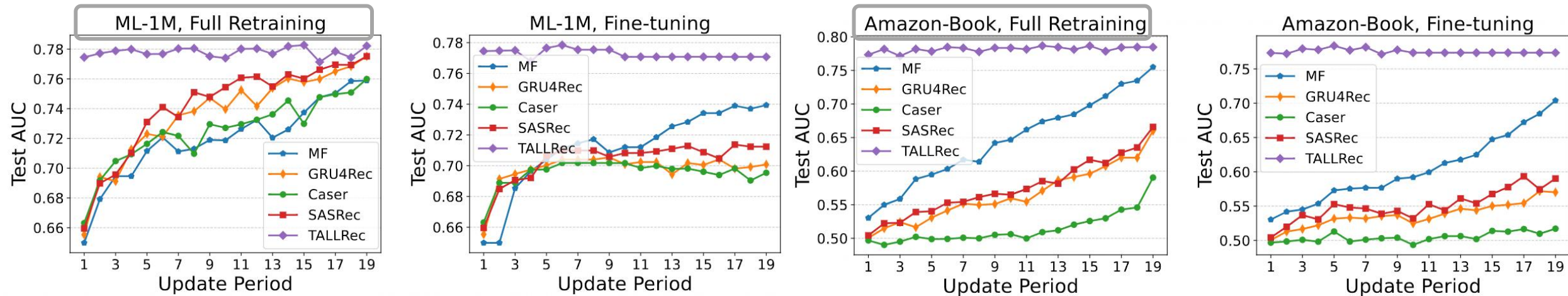


Just retrain LoRA
(TALLRec)



Continual learning:

Work#1: The effectiveness of full-retraining and fine-tuning for TALLRec

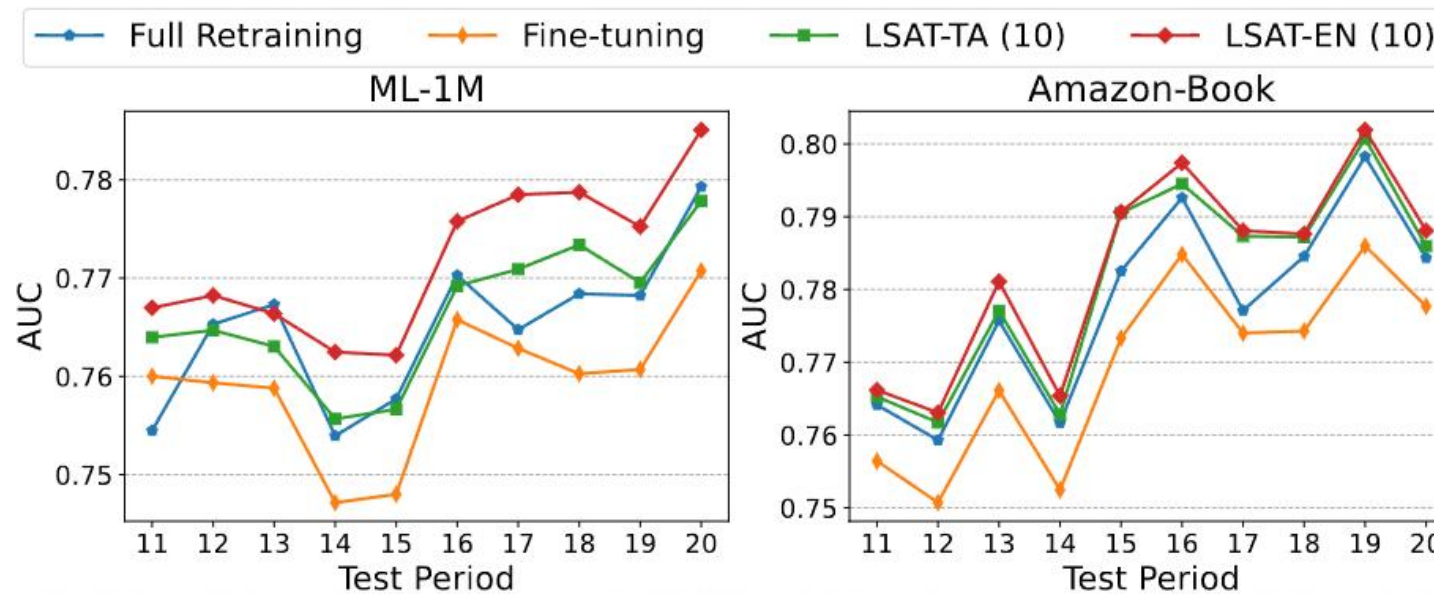


- ❑ Periodically update TALLRec does not bring significant performance improvements.
- ❑ LLM4Rec **may struggle to capture short-term preferences in the latest data** with traditional periodic updates, limiting performance improvement.

Continual learning:

Work#1: Long- and short-term Adaptation-aware Tuning

- Long-term LoRA fits all historical data to capture long-term preferences. (updates less frequently)
- Short-term LoRA retraines frequently with the latest data to capture short-term preferences.

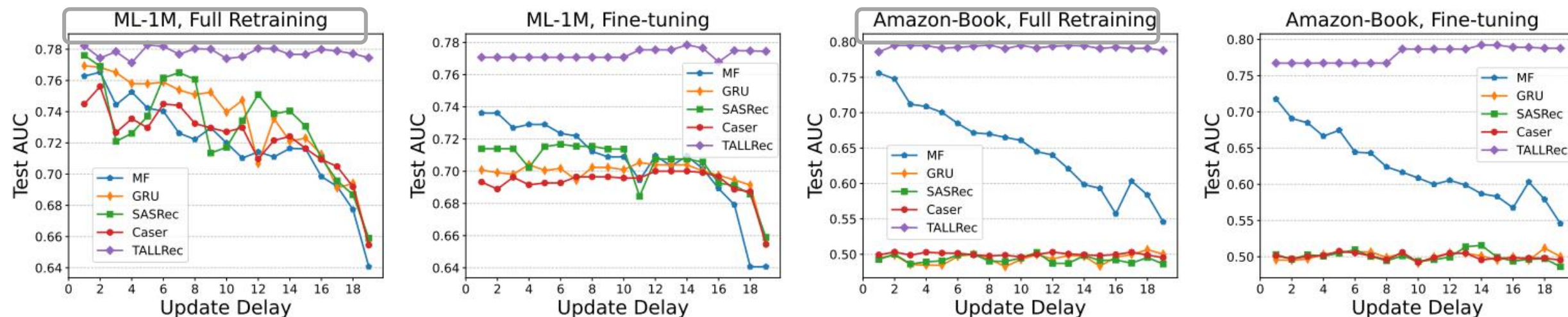


- Just focus on TALLRec and retrain-based methods

Modeling: Lifelong Modeling

Continual learning:

Work#1: The effectiveness of full-retraining and fine-tuning for TALLRec

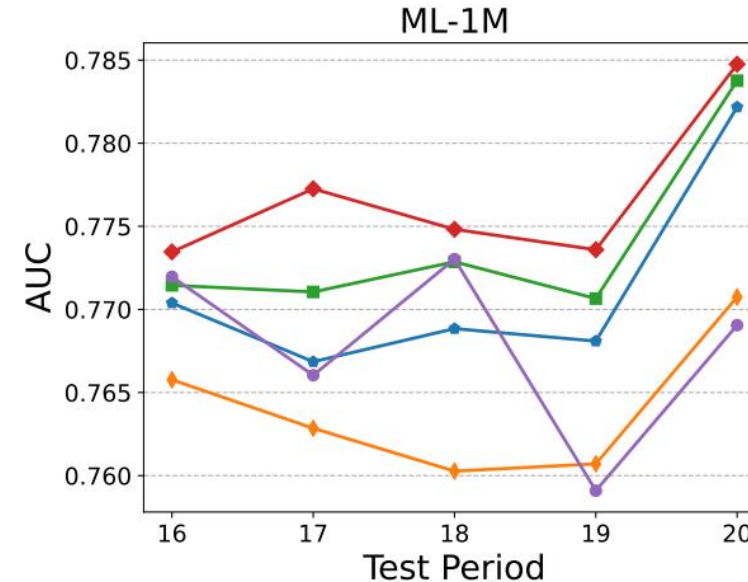
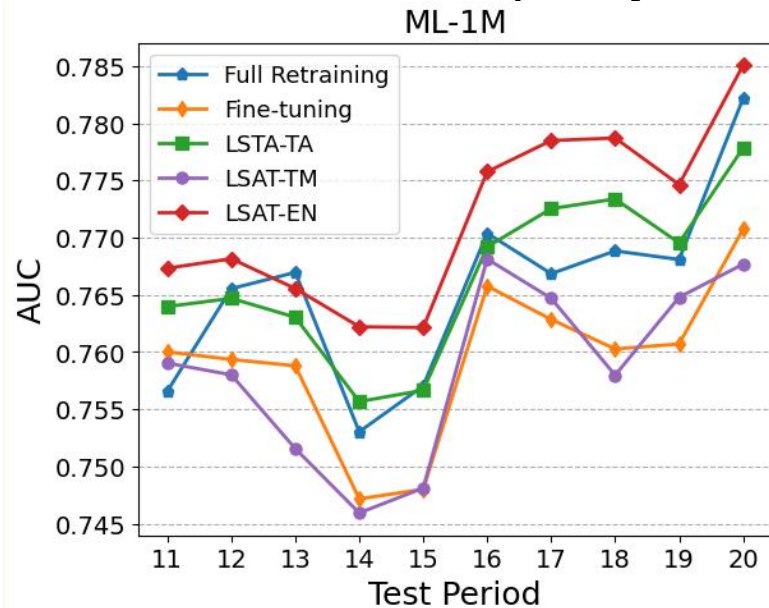


- ❑ Periodically update TALLRec does not bring significant performance improvements.
- ❑ LLM4Rec **may struggle to capture short-term preferences in the latest data** with traditional periodic updates, limiting performance improvement.

Continual learning:

Work#1: Long- and short-term Adaptation-aware Tuning

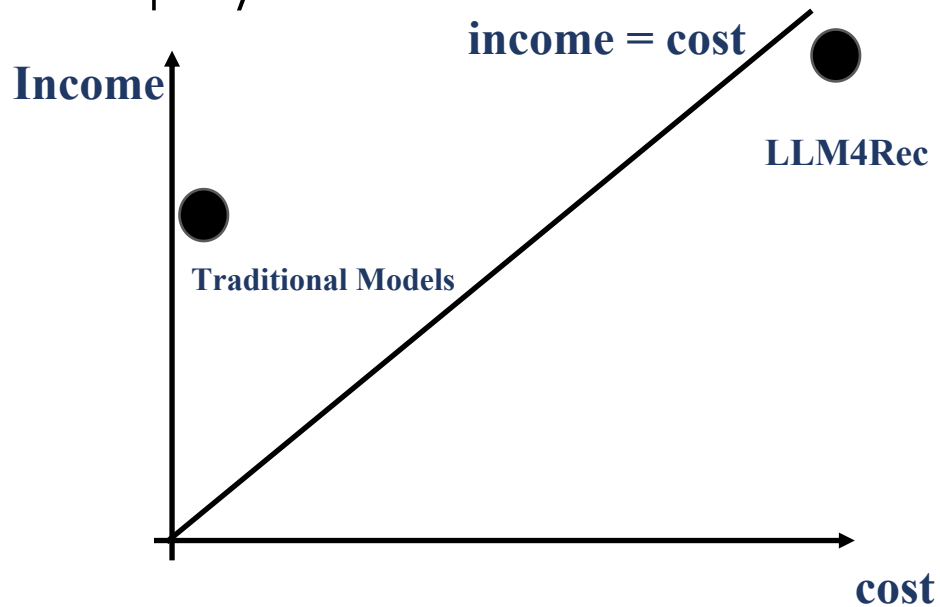
- Long-term LoRA fits all historical data to capture long-term preferences. (updates less frequently)
- Short-term LoRA retraines frequently with the latest data to capture short-term preferences.



- Just focus on TALLRec and retrain-based methods

- Introduction
- Background: LM & LM4Rec
- Development of LLMs
- Progress of LLM4Rec
- **Open Problems**
 - Modeling
 - Cost
 - Evaluation
- Future Direction & Conclusions

- The income—cost trade—off is sensitive for recommendation
- Deployment cost of LLM4Rec is high



LLM Parameters: tens/hundreds of billions

Training and inference:

- High demand on GPUs/Memory
- Slow

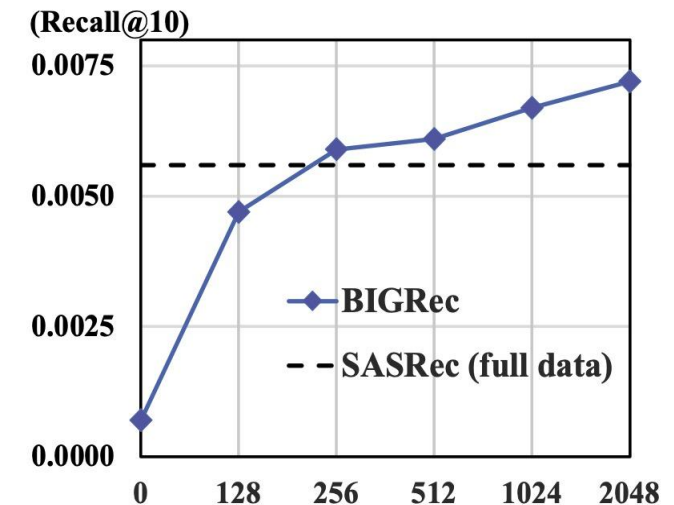
How to reduce the cost?

One exploration: Data-efficient training

- ❑ Fine-tuning LLM is necessary
 - ❑ LLMs are not particularly trained on recommendation data
- ❑ LLM fine-tuning is expensive, *e.g.*, high computational costs, time-consuming
- ❑ Few-shot fine-tuning is a promising solution
- ❑ **Data pruning for efficient LLM-based recommendation**
 - ❑ identify **representative samples** tailored for LLMs

Statistics from Tiktok¹ (per day)

- New videos: ~160M
- New interactions: ~942B

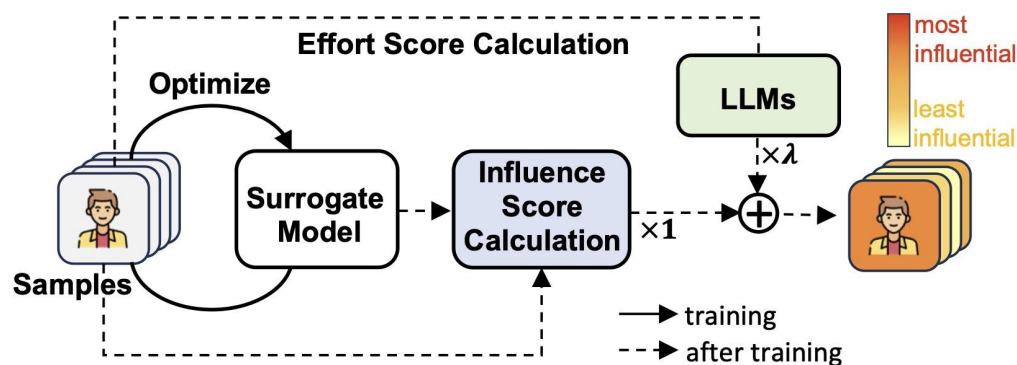


(a) Few-shot performance on MicroLens-50K.

Cost: Training

One exploration: Data-efficient training

- Two objectives for data pruning
 - high accuracy**: select the samples that can lead to higher performance \rightarrow **influence score**
 - high efficiency**: emphasize the low costs of the data pruning process
 - surrogate model to improve efficiency
 - effort score** to bridge between surrogate model and LLMs

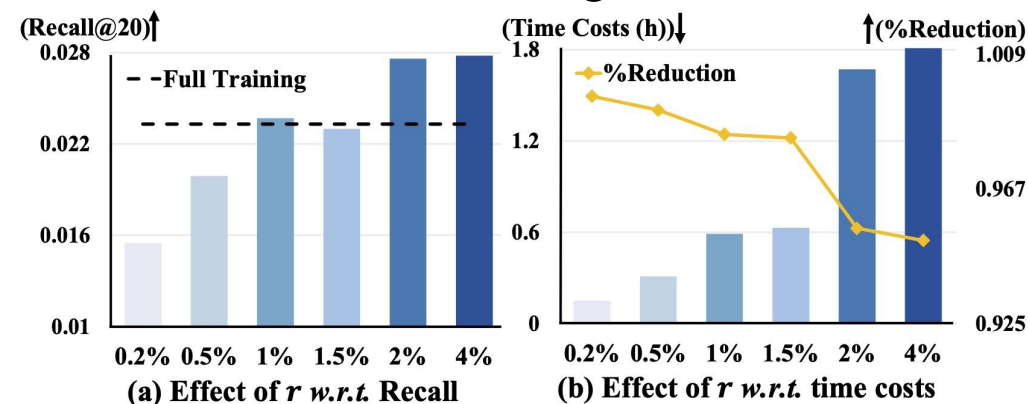


Experimental results

- fine-tune with 1024 samples

| | 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% |

- Increasing samples from 0.2% to 4% of all training data



Cost: Inference

One solution: distillation

Distill LLM's knowledge to smaller models and utilize small models for inference

- Work#1: distill recommendation results

| Dataset | Model | HR@20 | NDCG@20 | Inference time |
|---------|-------------|---------|---------|-----------------------|
| Games | DROS | 0.0473 | 0.0267 | 1.8s |
| | BIGRec | 0.0532 | 0.0341 | $2.3 \times 10^4 s$ |
| | <i>Gain</i> | +12.47% | +27.72% | $-1.3 \times 10^6 \%$ |
| Toys | DROS | 0.0231 | 0.0144 | 1.6s |
| | BIGRec | 0.0420 | 0.0207 | $1.1 \times 10^4 s$ |
| | <i>Gain</i> | +81.82% | +43.75% | $-6.8 \times 10^5 \%$ |

The inference latency of BIGRec far exceeds that of DROS.

| Dataset | Condition | Relative Ratio |
|-----------|---------------|----------------|
| Games | BIGRec > DROS | 53.90% |
| | BIGRec < DROS | 46.10% |
| MovieLens | BIGRec > DROS | 40.90% |
| | BIGRec < DROS | 59.10% |
| Toys | BIGRec > DROS | 66.67% |
| | BIGRec < DROS | 33.33% |

BIGRec does not always outperform DROS.

❑ Distillation challenges:

- ❑ 1) The teacher's knowledge may not always be reliable.
- ❑ 2) The divergence in semantic space poses a challenge to distill the knowledge from embeddings.

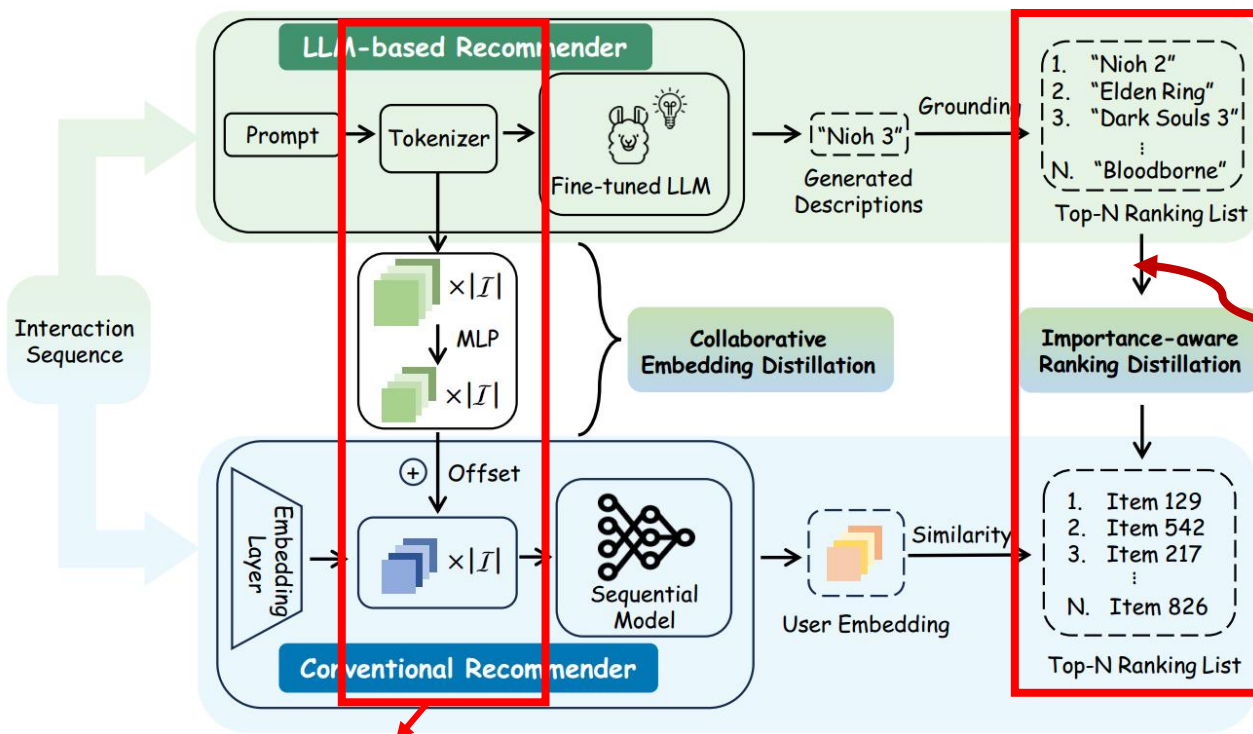


Cost: Inference

One solution: distillation

Distill LLM's knowledge to smaller models and utilize small models for inference

- Work#1: distill recommendation results



- **Importance-aware Ranking Distillation**
filter reliable and student-friendly knowledge by **weighting instances**

- **Confidence of LLMs**
The distance between the generated descriptions with the target item
- **Teacher-Student Consensus**
The items recommended by both teacher and student are more likely to be positive
- **Ranking Position**
Higher ranked items by teachers are more reliable

Collaborative Embedding Distillation

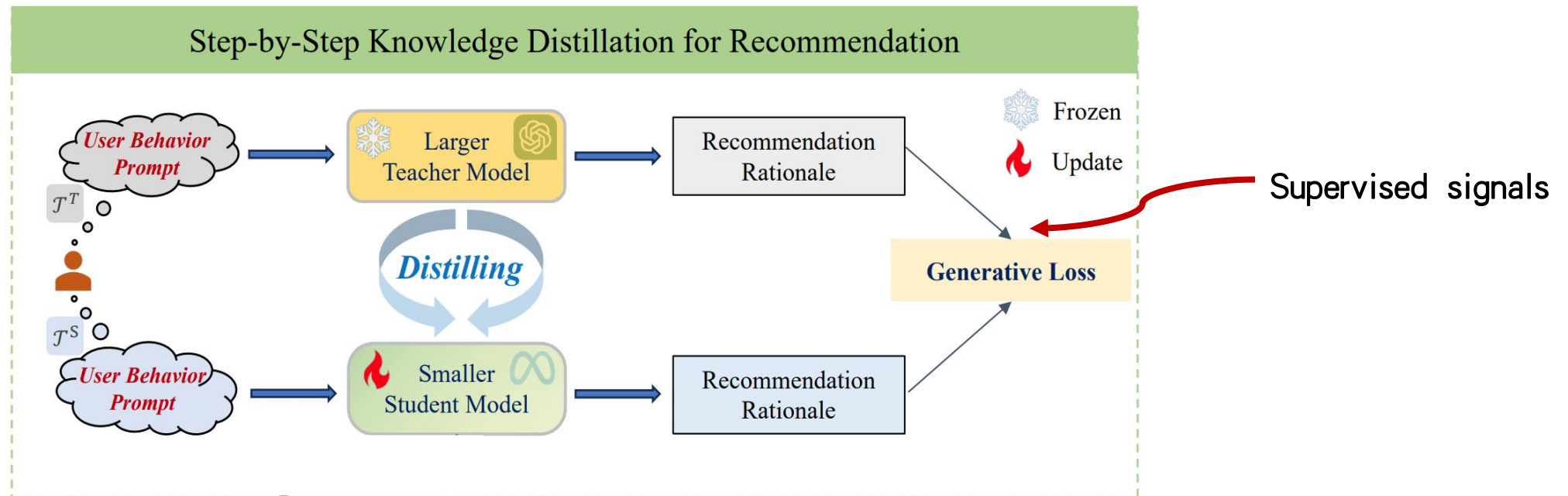
integrate knowledge from teacher embeddings with student's

Cost: Inference

One solution: distillation

Distill LLM's knowledge to smaller models and utilize small models for inference

- Work#2: distill recommendation rationales

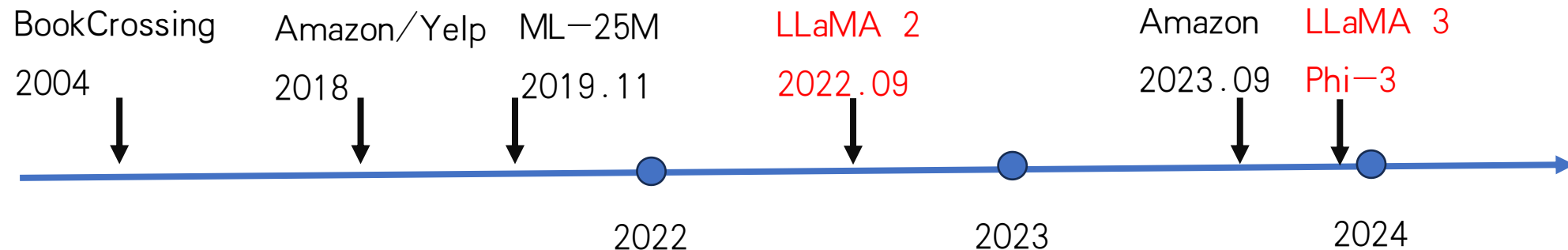


- ❑ Distill recommendation rationale from ChatGPT to Llama-7B
- ❑ Empowering recommendation with rationale embedding
- ❑ Combining the rationale embedding and item description embedding for prediction

- Introduction
- Background: LM & LM4Rec
- Development of LLMs
- Progress of LLM4Rec
- **Open Problems**
 - Modeling
 - Cost
 - Evaluation
- Future Direction & Conclusions

- ❑ **Challenge#1: Lack of data for evaluation**

- ❑ **Most of benchmarks are proposed ahead of pre-training stage of LLMs, e.g., ChatGPT, LLaMA.**



- ❑ The information of recommendation datasets (e.g., reviews,) may be include in LLMs.
- ❑ Existing works usually did not discuss this.
- ❑ Evaluations on the data that is not include in pretraining data of LLMs.

❑ Challenge#1: Lack of data for evaluation

❑ Insufficient features

- ❑ Lack of raw feature
 - ❑ Anonymous (e.g., just feature ID)
 - ❑ Lack of content (e.g., video content)
- ❑ Currently, many works just utilize titles

- Underutilization of LLM capabilities;
- Underassessment of the effectiveness of LLM4Rec

❑ Data homogeneity

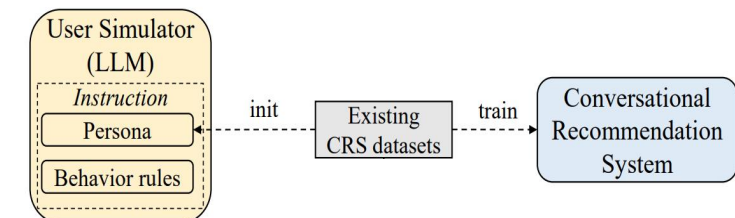
- ❑ content homogeneity:
 - mostly from E-commerce platform /
 - entertaining content or places
- ❑ biased user distributions: mostly from China and U.S.

- Not comprehensive evaluation
- Biased evaluation

❑ Challenge#2: Evaluate interactive recommendation

❑ Conversational recommendation

- ❑ provide personalized recommendation via multi-turn dialogs in natural language
- ❑ focus on conversational quality and recommendation quality
- Issues of traditional evaluation:
 - **Simulated users** are overly simplified representations of human users
 - Conversations are often vague about the user preference, but not focus on exactly match the ground-truth items
 - Evaluation protocol is based on fixed conversations, but the conversation could be diverging.
- New evaluation: simulation with LLM-based agents?
 - Challenges: how to design simulators is still an open problem.



- ❑ **Challenge#2: Evaluate interactive recommendation**

- ❑ **Long-term recommendation**

- ❑ Multi-turn user-system interactions
 - ❑ Focus on long-term user engagement, e.g., user retention

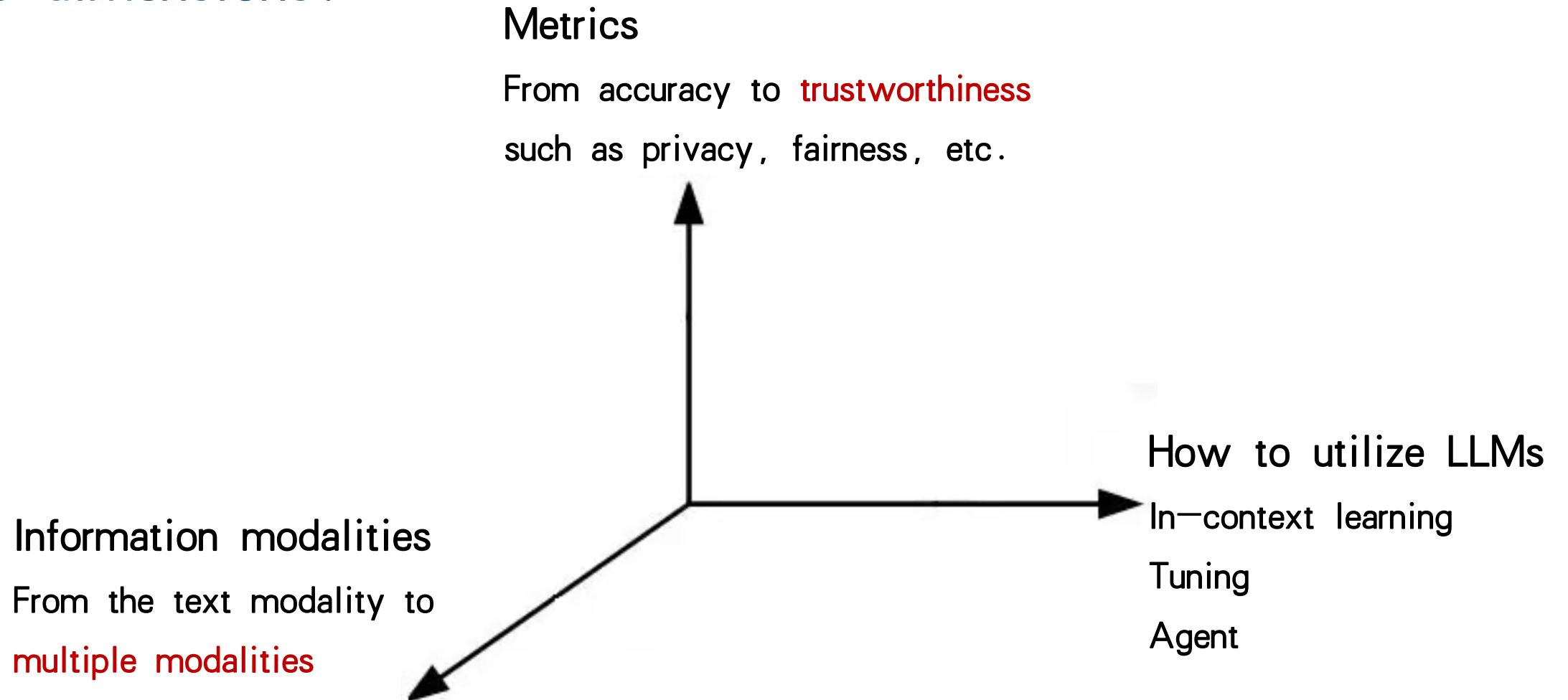
- ❑ **How to evaluate long-term engagement is a big challenge.**

- ❑ We have not feedback about the unseen interaction trajectory
 - ❑ Evaluation with agent-based simulator is a potential solution

- Introduction
- Background: LM & LM4Rec
- Development of LLMs
- Progress of LLM4Rec
- Open Problems
- **Future Direction & Conclusions**

Progress of LLM4Rec

Three dimensions:



Open Problems

Three aspects:

Modeling

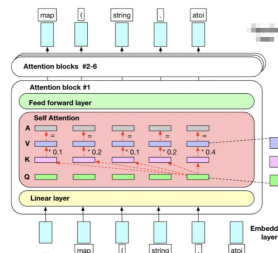
LLM: modeling text/language



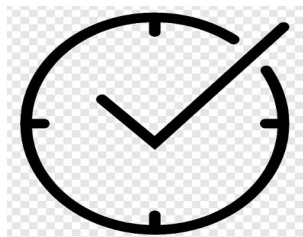
RecSys: modeling behaviors

Cost

LLM: high cost/delay



computation/m
emory—costly

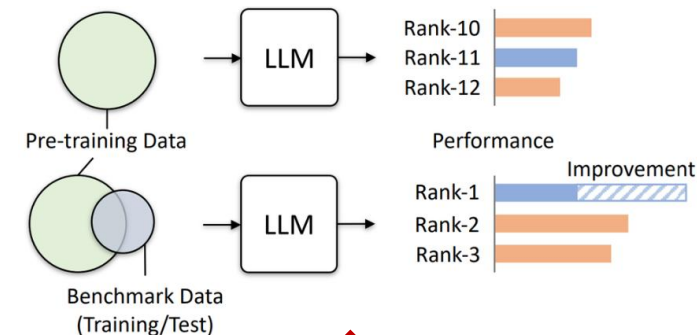


Real—time,
focus on cost

RecSys: cost sensitive

Evaluation

LLM: Trained on many data,
text—focused, language



Evaluation?

RecSys research: interactions,
offline, anonymous data

Generative Recommendation Paradigm

□ Generative AI for recommendation

- Personalized **content generation**, including item repurposing and creation.
 - **Application:** News, fashion products, micro-videos, virtual products in games, etc.

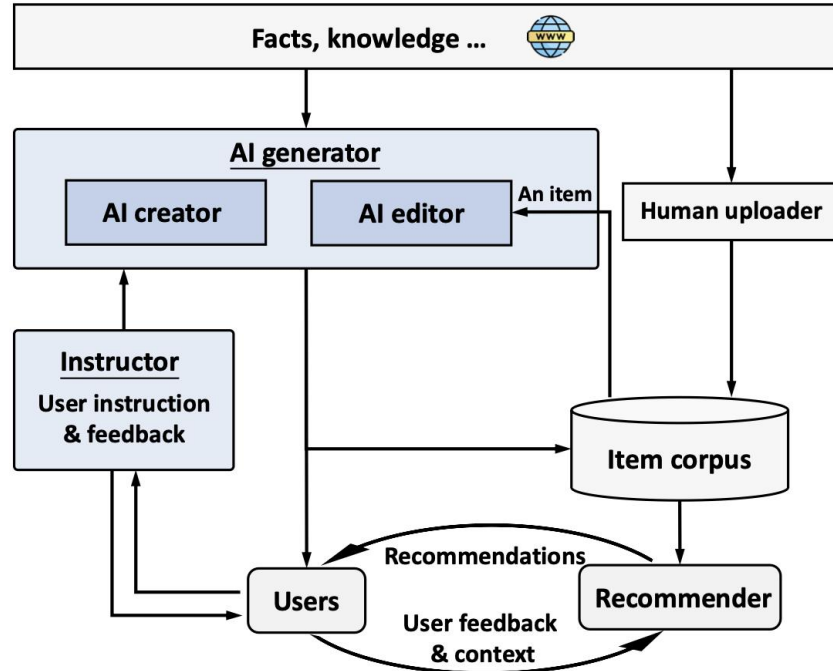


Figure 4: A demonstration of GeneRec. The instructor collects user instructions and feedback to guide content generation. The AI editor aims to repurpose existing items in the item corpus while the AI creator directly creates new items.

Instructor:

- Pre-process user instructions and feedback to guide the content generation of the AI generator.
- ### AI Editor:
- Refine or repurpose existing items according to personalized user instructions and feedback.
 - External facts and knowledge might be used for content generation.

AI Creator:

- Generate new items based on personalized user instructions and feedback.

AI Checker:

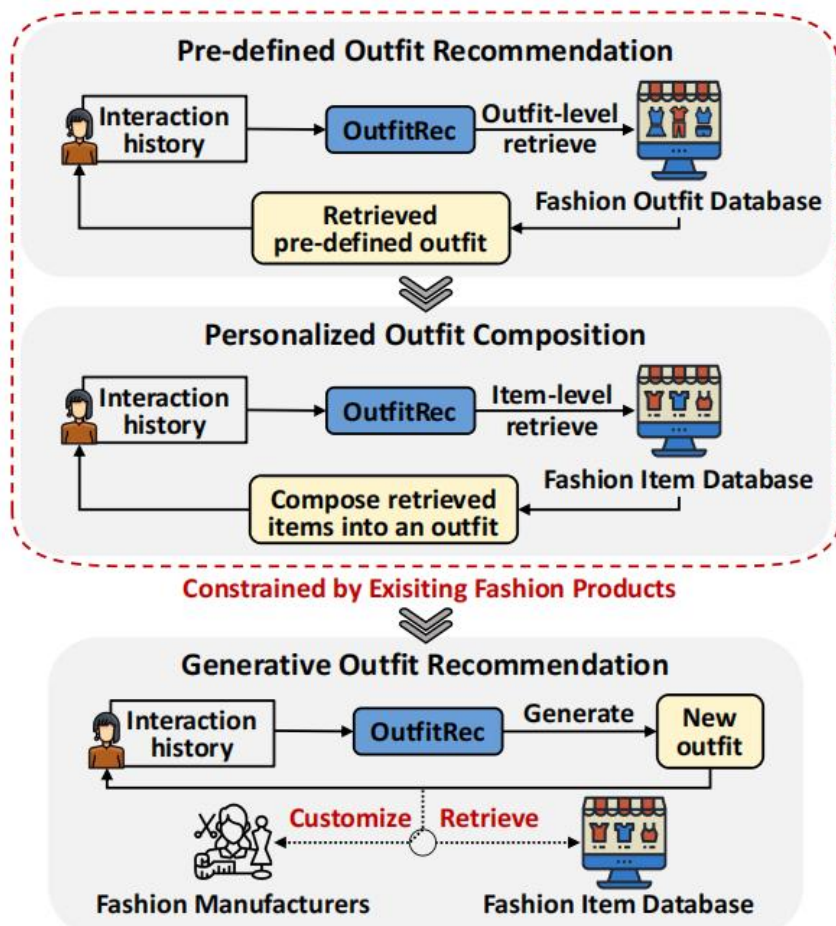
- Generation quality checks.
- Trustworthiness checks.

Applicable to many domains, including images, micro-videos, movies, news, books, and even products (for manufacture).

Generative Recommendation Paradigm

Generative Recommendation in Fashion Domain

The Evolution of Fashion Outfit Recommendation

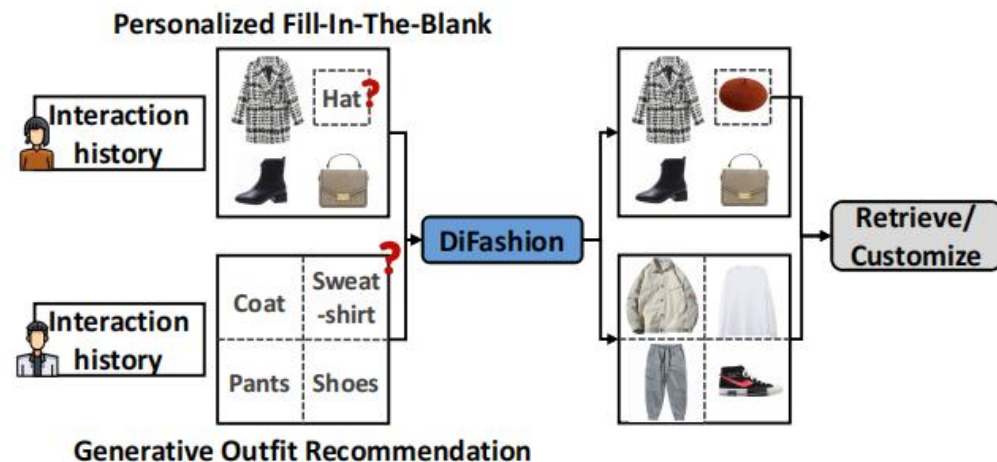


New Task

Generative Outfit Recommendation

Objective: generating a set of new personalized fashion products to compose a visually compatible outfit catering to users' fashion tastes.

Practical Implementation: retrieve or customize

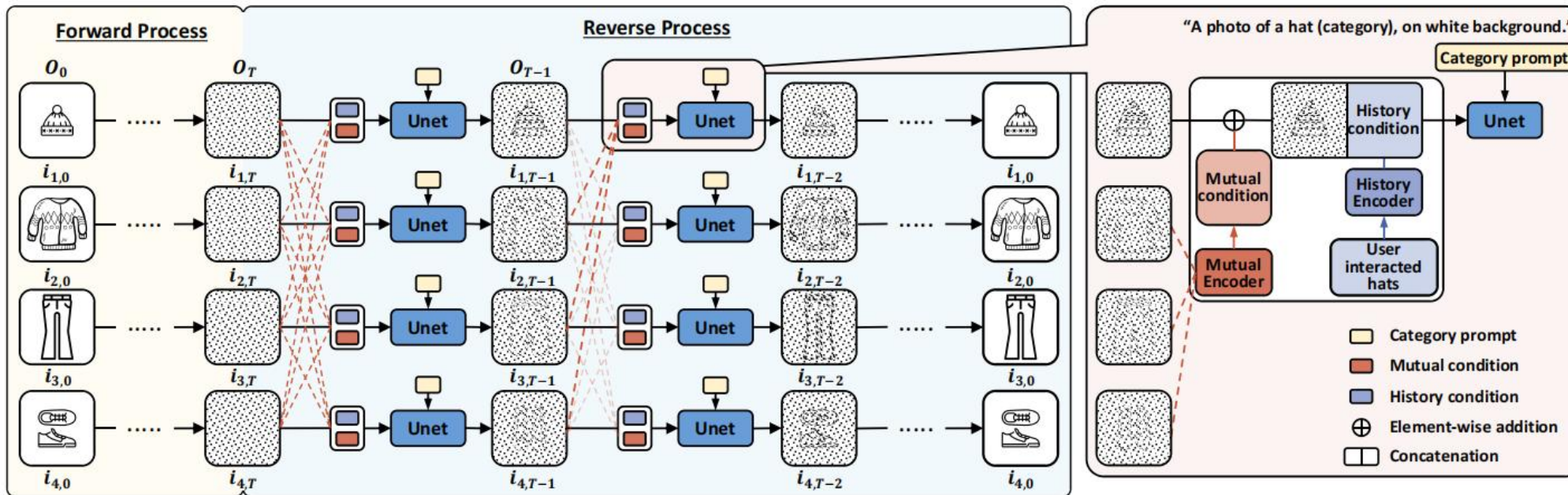


Generative Recommendation Paradigm

Generative Recommendation in Fashion Domain

Overview of DiFashion

- **Forward Process:** gradually corrupting outfit images with Gaussian noise
- **Reverse Process:** incorporating three conditions to guide the parallel generation process



Three criteria of GOR

- **High Fidelity:** category prompts & classifier-free guidance
- **Compatibility:** mutual encoder
- **Personalization:** history encoder

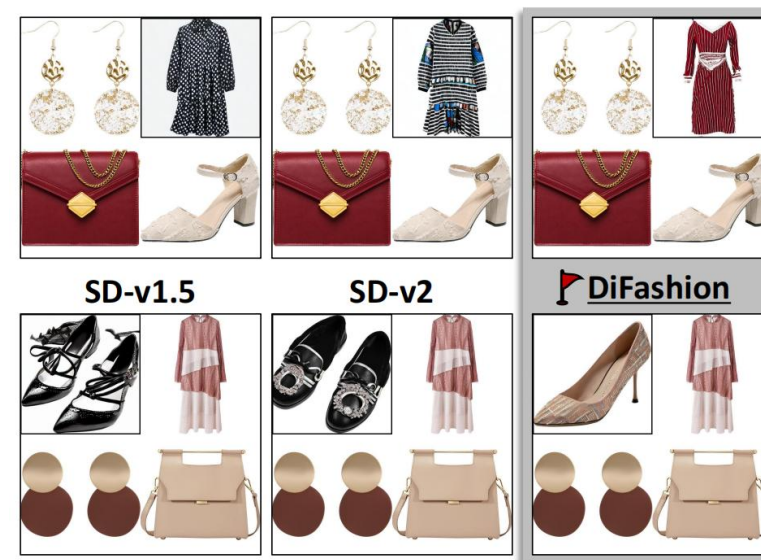
Generative Recommendation Paradigm

❑ Experiments

- ❑ **Datasets:** iFashion, Polyvore-U
- ❑ **Baselines:** generative models, retrieval-based models
- ❑ **Tasks:** personalized Fill-In-The-Blank (PFITB), GOR
- ❑ **Evaluation**
 - **Quantitative Evaluation**
 - **Human-involved Qualitative Evaluation**

Table 5: The human-involved qualitative evaluation results, where “±” denotes 95% confidence interval. DiFashion is consistently preferred ($\geq 50\%$) over the baselines across all evaluation metrics for both PFITB and GOR tasks.

| | DiFashion | Fidelity | Compatibility | Personalization |
|-------|-----------|--------------------|--------------------|--------------------|
| PFITB | SD-v1.5 | $64.08 \pm 3.08\%$ | $60.44 \pm 2.42\%$ | $68.32 \pm 3.47\%$ |
| | SD-v2 | $70.04 \pm 4.16\%$ | $57.48 \pm 1.90\%$ | $66.40 \pm 3.39\%$ |
| GOR | SD-v1.5 | $61.56 \pm 1.93\%$ | $61.20 \pm 2.00\%$ | $60.80 \pm 2.57\%$ |
| | SD-v2 | $66.52 \pm 2.15\%$ | $60.56 \pm 1.88\%$ | $63.72 \pm 1.95\%$ |



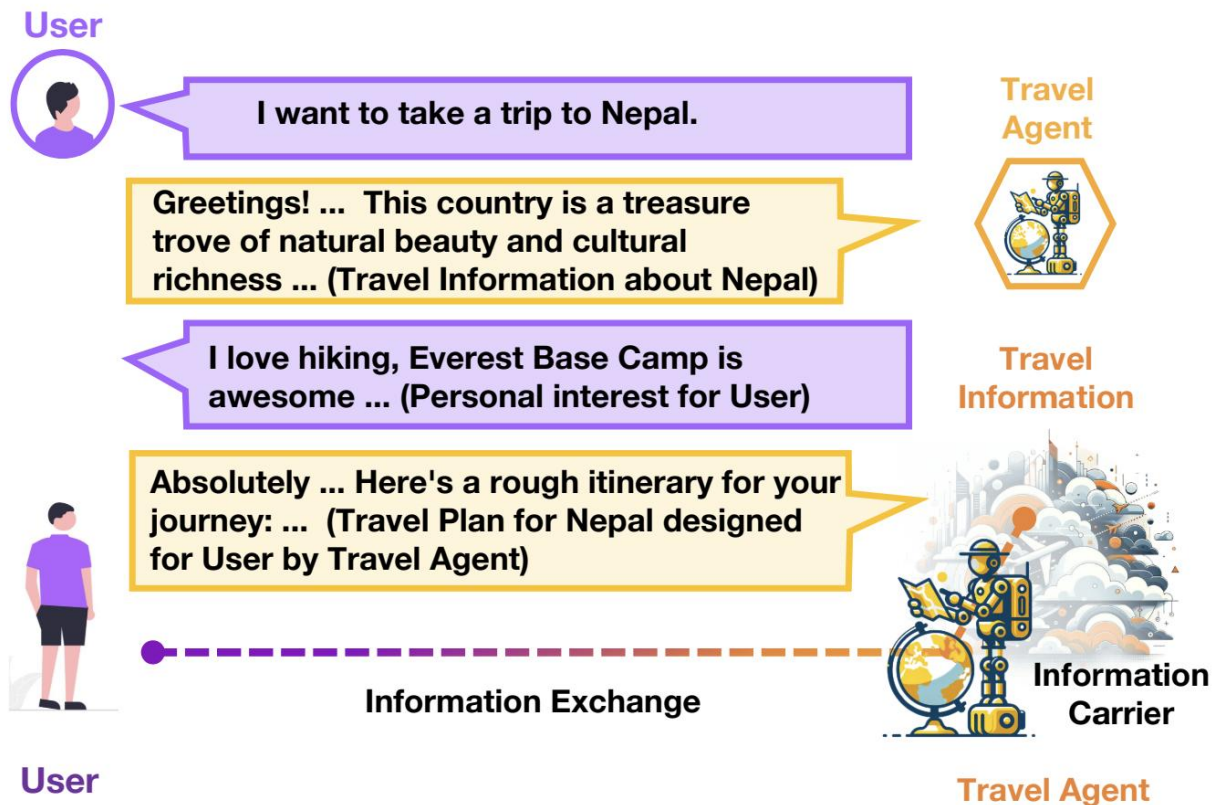
PFITB



GOR

Recommender for Agent Platform

- Existing agent platforms such as GPTs (OpenAI), Poe (Quora), and DouBao (ByteDance) possess a vast number of LLM-based agents.
- How to recommend LLM-based Agent to the user?

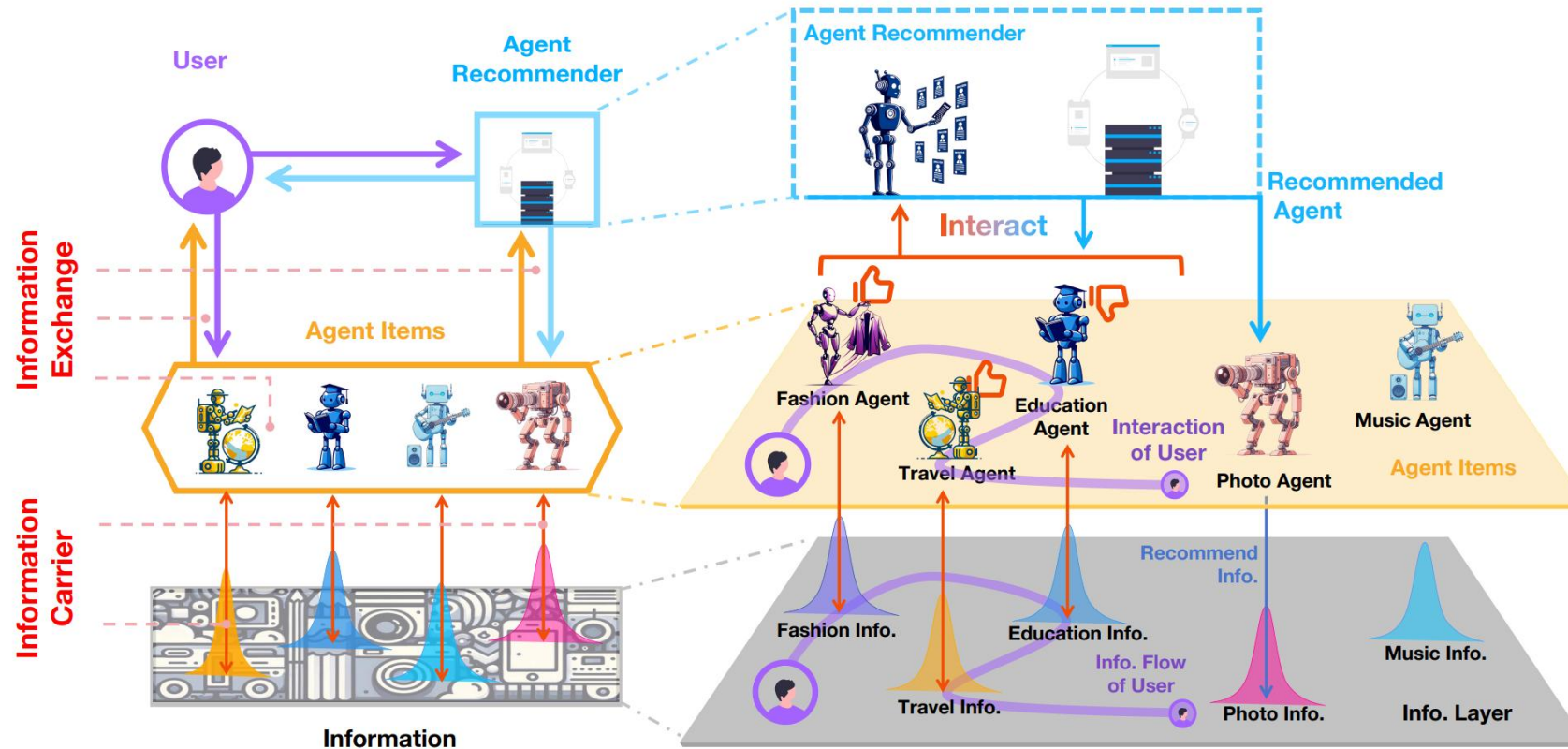


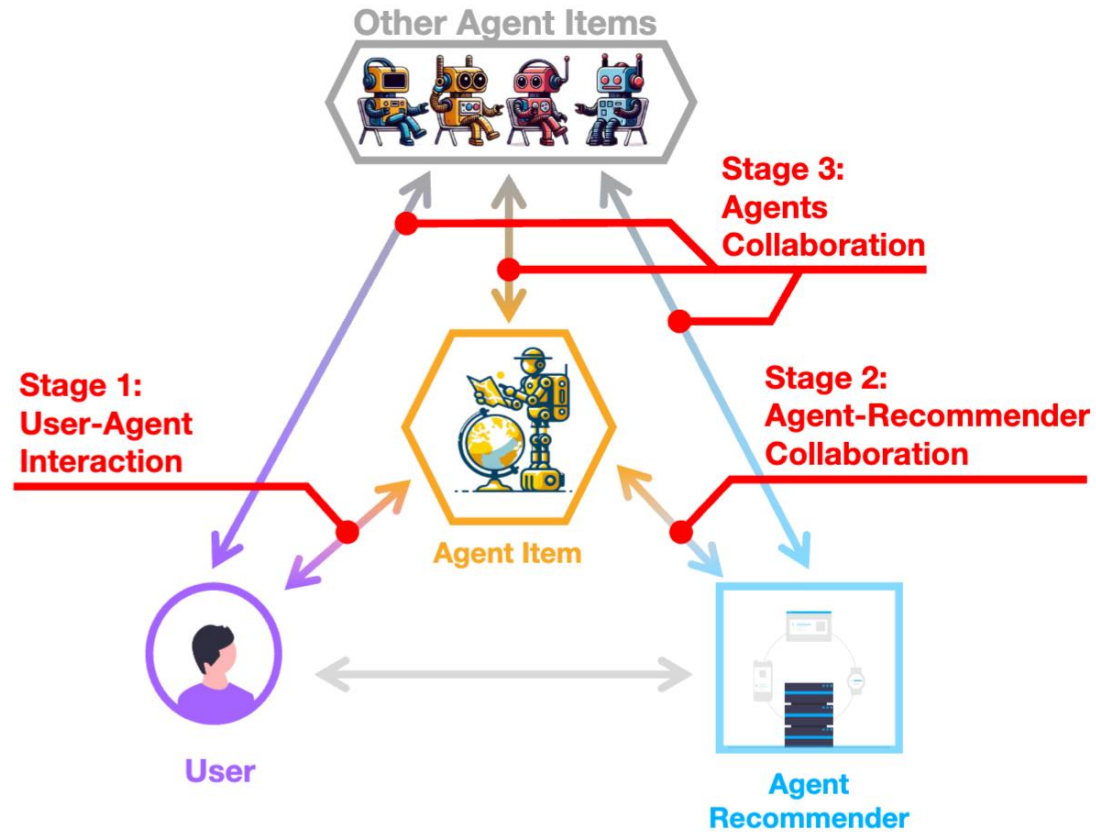
Different from Items in Traditional Recommender System, LLM-based Agent holds the potential to extend the format of information carriers and the way of information exchange.

- > Formulate **new Information System**
- > **New Rec paradigm Rec4Agentverse**

Rec4Agentverse

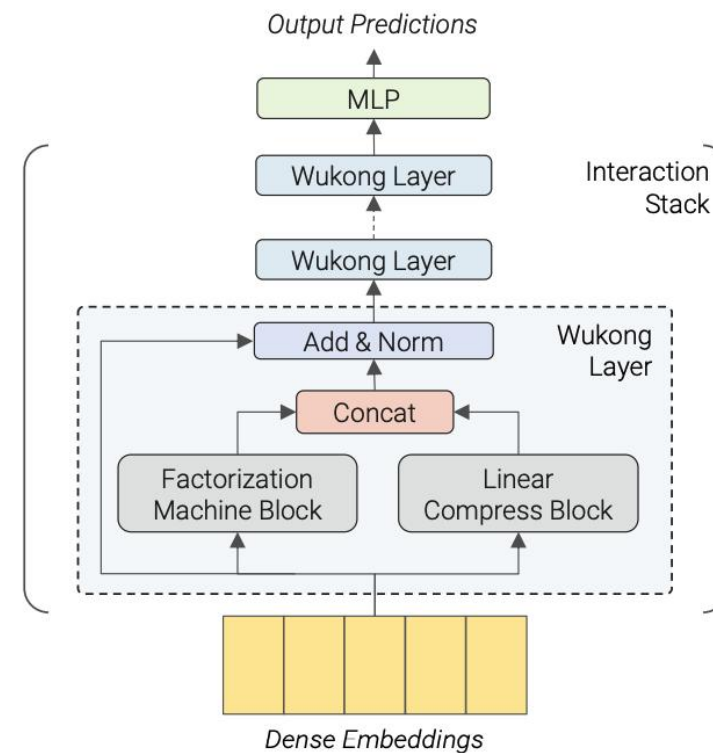
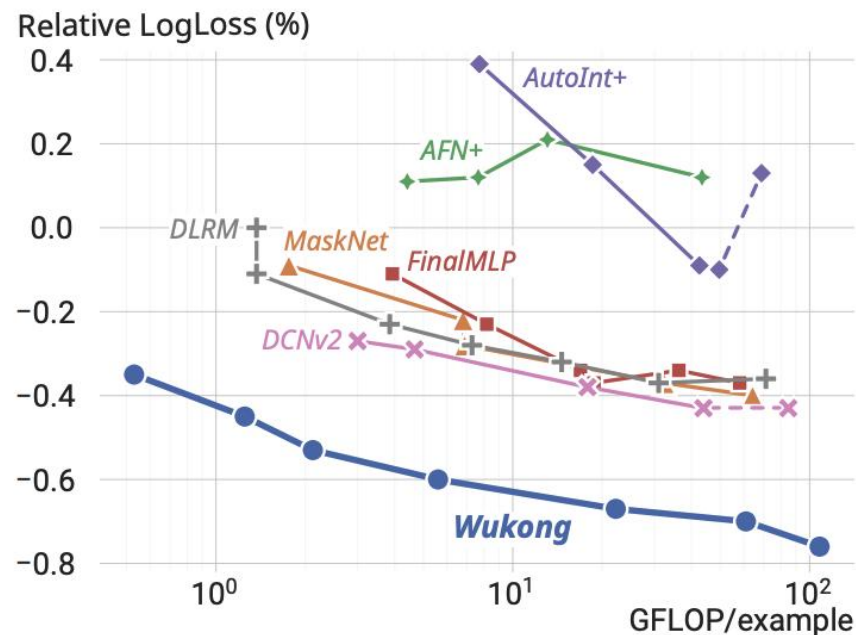
In Rec4Agentverse, the relationship between user, Agent Item and Agent Recommender may be much closer. Agent Recommender can collaborate with Agent Items to affect the information flow of users and offer personalized information services.





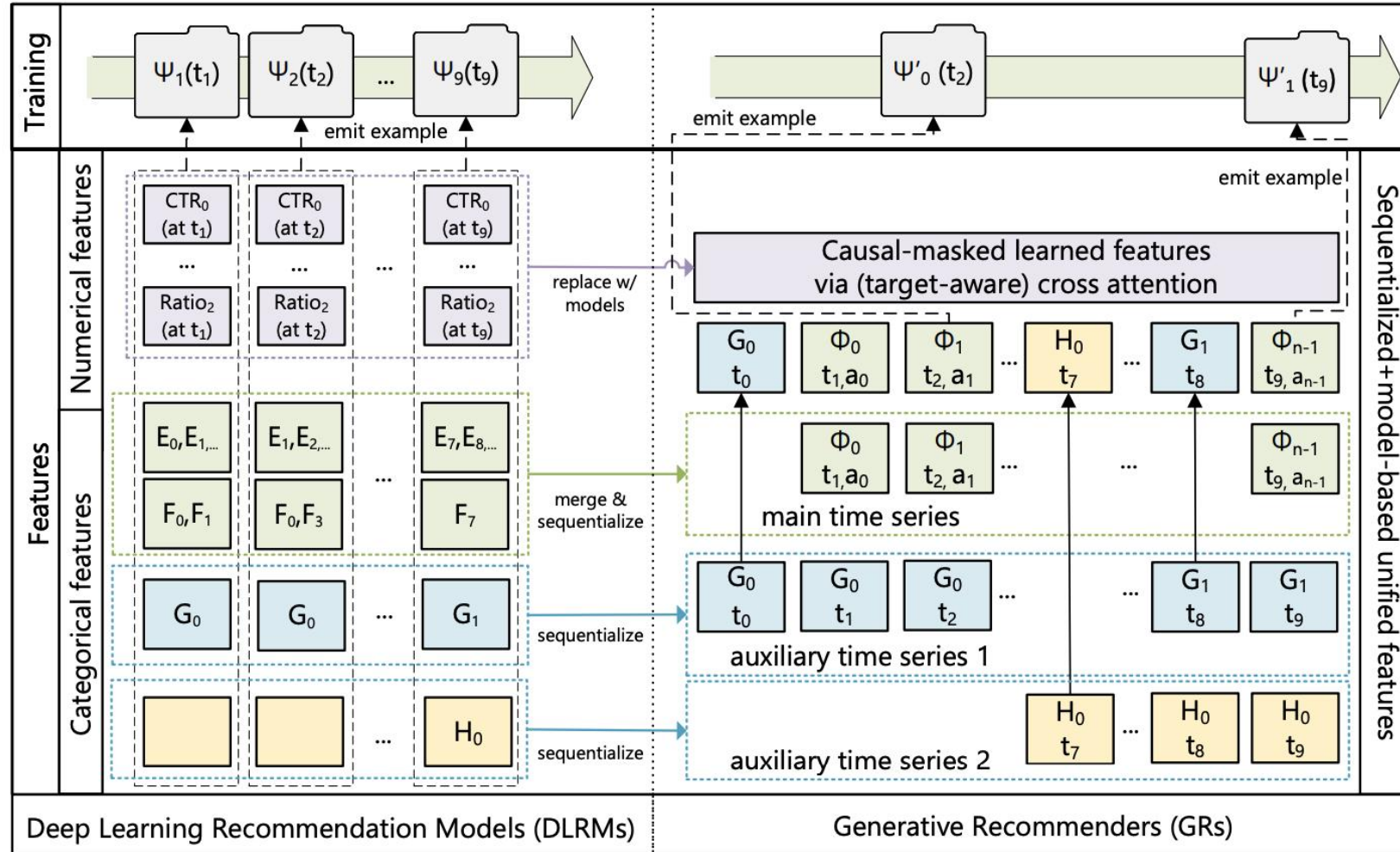
Three stages of Rec4Agentverse . The bidirectional arrows depicted in the Figure symbolize the flow of information.

- **User-Agent interaction stage:** Information flows between the user and Agent Item.
- **Agent-Recommender Collaboration stage:** Information flows between Agent Item and Agent Recommender.
- **Agents Collaboration stage:** Information flows between Agent Items.

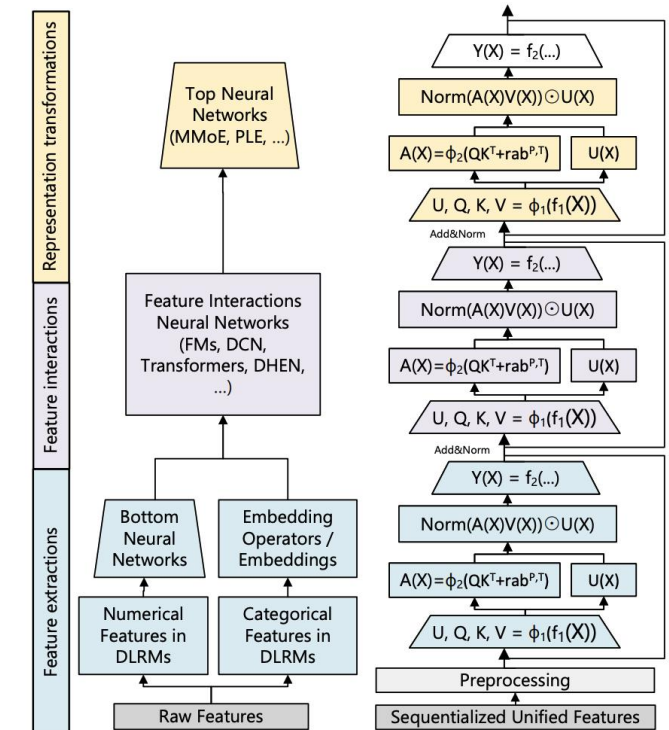


- ❑ The scaling properties of the CTR model have been verified, showing excellent performance on both internal and open-source data.
- ❑ Demonstrates the possibility of increasing the size of CTR models through clever structural design and appropriate scaling settings
- ❑ Exhibits better scaling performance than previous models.

Action Speaker Louder than Words



| Task | | Specification (Inputs / Outputs) |
|-----------|---------|---|
| Ranking | $x_i s$ | $\Phi_0, a_0, \Phi_1, a_1, \dots, \Phi_{n_c-1}, a_{n_c-1}$ |
| | $y_i s$ | $a_0, \emptyset, a_1, \emptyset, \dots, a_{n_c-1}, \emptyset$ |
| Retrieval | $x_i s$ | $(\Phi_0, a_0), (\Phi_1, a_1), \dots, (\Phi_{n_c-1}, a_{n_c-1})$ |
| | $y_i s$ | $\Phi'_1, \Phi'_2, \dots, \Phi'_{n_c-1}, \emptyset$ $(\Phi'_i = \Phi_i \text{ if } a_i \text{ is positive, otherwise } \emptyset)$ |



Generative Recommender

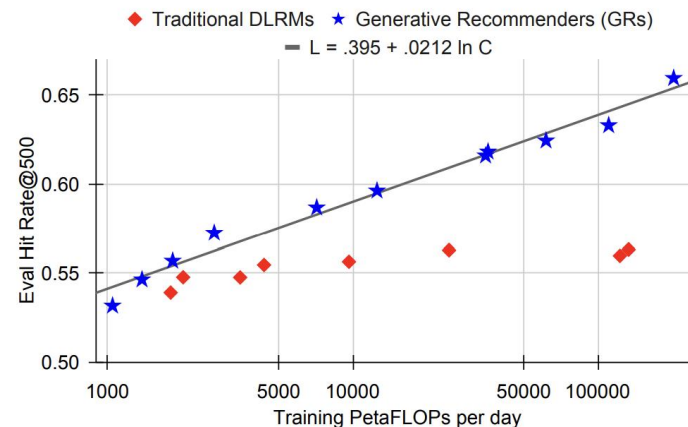
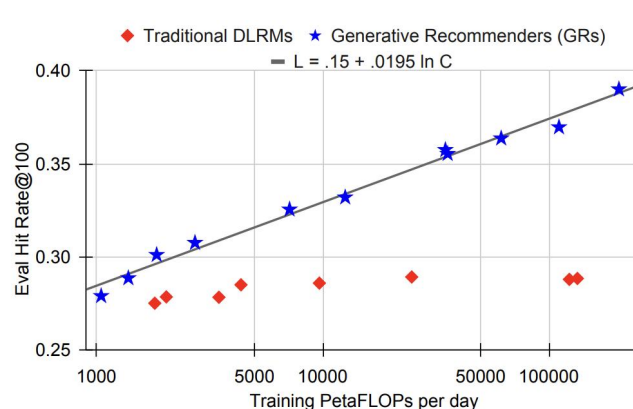
New model architecture and feature processing methods.

Action Speaker Louder than Words



Table 4. Evaluations of methods on public datasets in multi-pass, full-shuffle settings.

| | Method | HR@10 | HR@50 | HR@200 | NDCG@10 | NDCG@200 |
|--------|---------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| ML-1M | SASRec (2023) | .2853 | .5474 | .7528 | .1603 | .2498 |
| | HSTU | .3097 (+8.6%) | .5754 (+5.1%) | .7716 (+2.5%) | .1720 (+7.3%) | .2606 (+4.3%) |
| | HSTU-large | .3294 (+15.5%) | .5935 (+8.4%) | .7839 (+4.1%) | .1893 (+18.1%) | .2771 (+10.9%) |
| ML-20M | SASRec (2023) | .2906 | .5499 | .7655 | .1621 | .2521 |
| | HSTU | .3252 (+11.9%) | .5885 (+7.0%) | .7943 (+3.8%) | .1878 (+15.9%) | .2774 (+10.0%) |
| | HSTU-large | .3567 (+22.8%) | .6149 (+11.8%) | .8076 (+5.5%) | .2106 (+30.0%) | .2971 (+17.9%) |
| Books | SASRec (2023) | .0292 | .0729 | .1400 | .0156 | .0350 |
| | HSTU | .0404 (+38.4%) | .0943 (+29.5%) | .1710 (+22.1%) | .0219 (+40.6%) | .0450 (+28.6%) |
| | HSTU-large | .0469 (+60.6%) | .1066 (+46.2%) | .1876 (+33.9%) | .0257 (+65.8%) | .0508 (+45.1%) |



❑ Better performance than traditional models on in-house data and open source data (Above two table)

❑ Far more scaling ability than traditional DLRMs

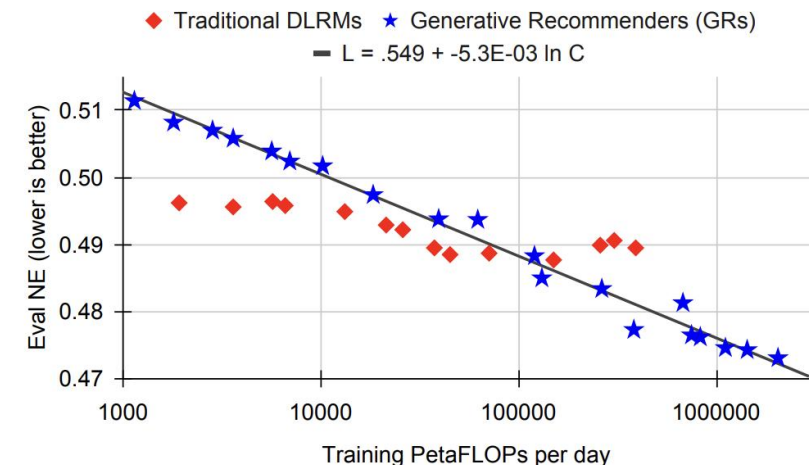
Jiaqi Zhai et al., 2024 Actions Speak Louder than Words: Trillion-Parameter Sequential Transducers for Generative Recommendations ICML 2024

Table 6. Offline/Online Comparison of Retrieval Models.

| Methods | Offline HR@K | | Online metrics | |
|------------------------|--------------|--------------|----------------|--------------|
| | K=100 | K=500 | E-Task | C-Task |
| DLRM | 29.0% | 55.5% | +0% | +0% |
| DLRM (abl. features) | 28.3% | 54.3% | — | — |
| GR (content-based) | 11.6% | 18.8% | — | — |
| GR (interactions only) | 35.6% | 61.7% | — | — |
| GR (new source) | 36.9% | 62.4% | +6.2% | +5.0% |
| GR (replace source) | | | +5.1% | +1.9% |

Table 7. Offline/Online Comparison of Ranking Models.

| Methods | Offline NEs | | Online metrics | |
|------------------------|--------------|--------------|----------------|--------------|
| | E-Task | C-Task | E-Task | C-Task |
| DLRM | .4982 | .7842 | +0% | +0% |
| DLRM (DIN+DCN) | .5053 | .7899 | — | — |
| DLRM (abl. features) | .5053 | .7925 | — | — |
| GR (interactions only) | .4851 | .7903 | — | — |
| GR | .4845 | .7645 | +12.4% | +4.4% |



Large Behaviour Model



❑ What we have know?

- ❑ **Scaling Law remains effective** on industrial-scale data when combined with an appropriate architecture in the context of recommendation scenario.
- ❑ When the model is large enough and captures high-order information, it exhibits a certain **generalization ability across scenes and domains**.
- ❑ Generative recommender is more stronger than traditional methods

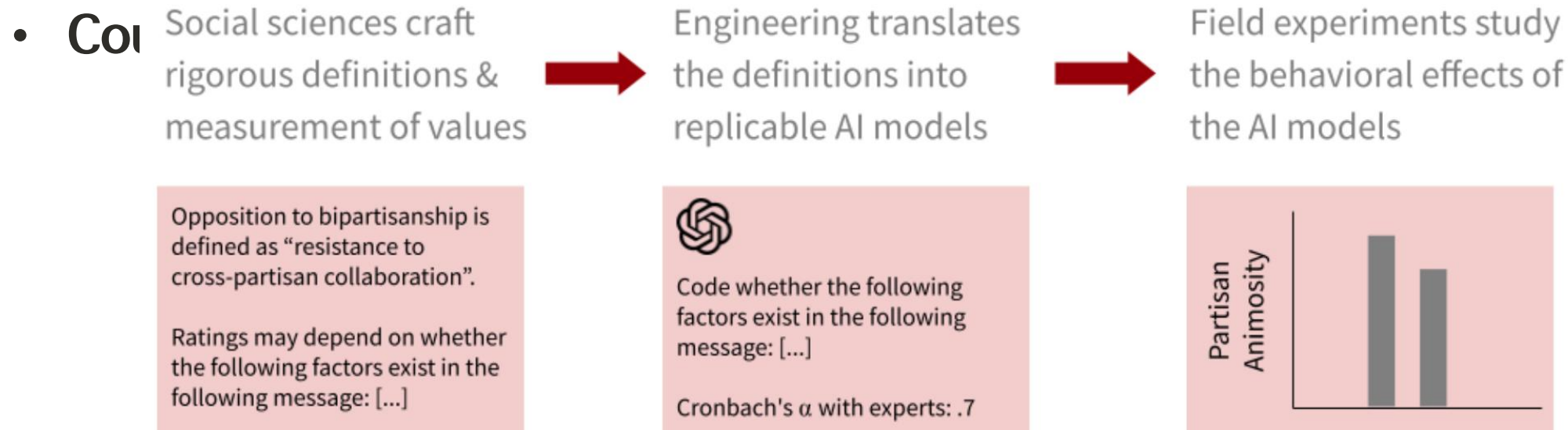
❑ What we haven't know?

- ❑ How to **integrate world knowledge**, and whether it can be **combined with LLMs**.
- ❑ In addition to features and simple action, how do we **model more complex short-term and long-term user behaviors**? And how is the **scalability** of these behaviors manifested?
- ❑ How to model the shared information between items/users and items/users?

Embed Social Values into LLMRec



- **Social media AI (RecSys) already embed values** --- maximize each user's individual experience---as predicted through likes
- **It can harm societal values** --- wellbeing, social capital, mitigating harm to minoritized groups, democracy, and maintaining pro-social norms.



Thanks for Your Listening !

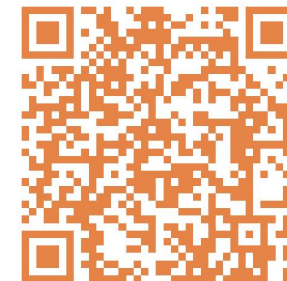
Tutorial on Large Language Models for Recommendation:
Progresses and Future Direction

Find our slides at

<https://generative-rec.github.io/tutorial/>



Tutorial



Survey: A Survey of Generative Search and Recommendation
in the Era of Large Language Models

<https://arxiv.org/pdf/2404.16924>

智荐阁



Survey



Follow our WeChat account “智荐阁”!

- ❑ The immense ability of LLMs may exceed the capabilities of traditional recommendation benchmark!
- ❑ The LLM may recommend items that are not in the dataset but are in line with user's real preference, how will it be evaluated?
- ❑ The LLM may recommend non-existent but meaningful items that meet the user's preferences. How will this situation be evaluated?

Evaluation & Benchmark

- ❑ The immense ability of LLMs may exceed the capabilities of traditional recommendation benchmark!

