Tutorial Outline

- **Part 1: Introduction** of RecSys in the era of LLMs (Dr. Wenqi Fan)
- **Part 2: Preliminaries** of RecSys and LLMs (Dr. Yujuan Ding)
- **Part 3: Pre-training** paradigms for adopting LLMs to RecSys (Dr. Yujuan Ding)
- **Part 4: Fine-tuning paradigms for adopting LLMs to RecSys (Liangbo Ning)**
- **Part 5: Prompting** paradigms for adopting LLMs to RecSys (Shijie Wang)
- **Part 6: Future directions** of LLM-empowered RecSys (Dr. Wenqi Fan)

Website of this tutorial
Check out the slides and more information!

Website

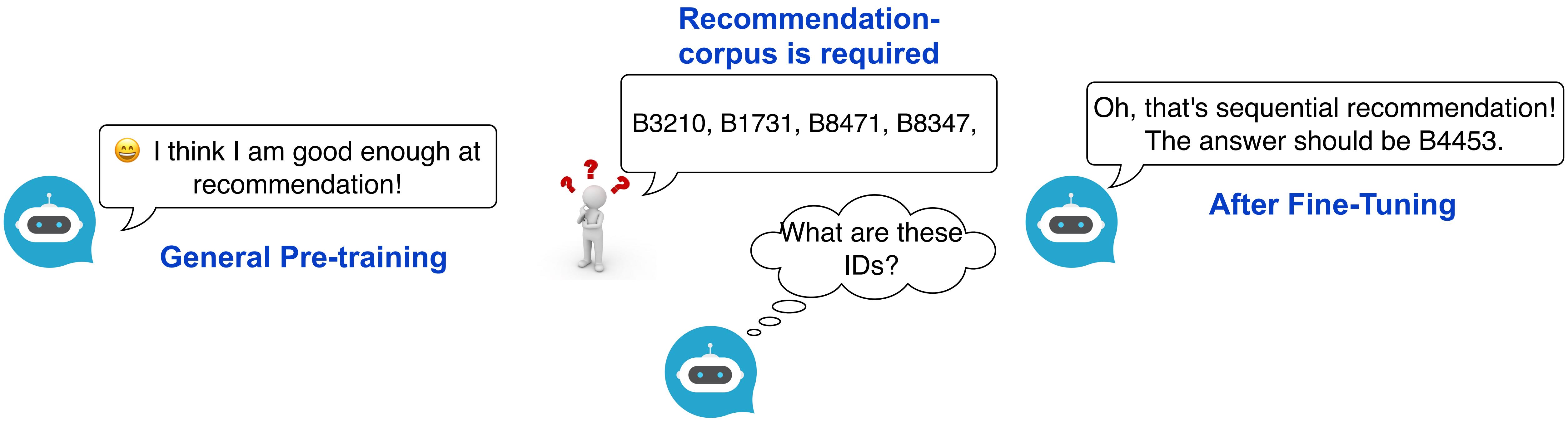
PART 4: RecSys Fine-tuning

Presenter
Liangbo NING
HK PolyU

- **Fine-tuning in NLP**
    - Pre-training then fine-tuning
    - Parameter Efficient Fine-tuning (PEFT)
    - Fine-tuning with reinforcement learning
- Fine-tuning LLM-based RecSys
- Parameter efficient fine-tuning for LLM-based RecSys

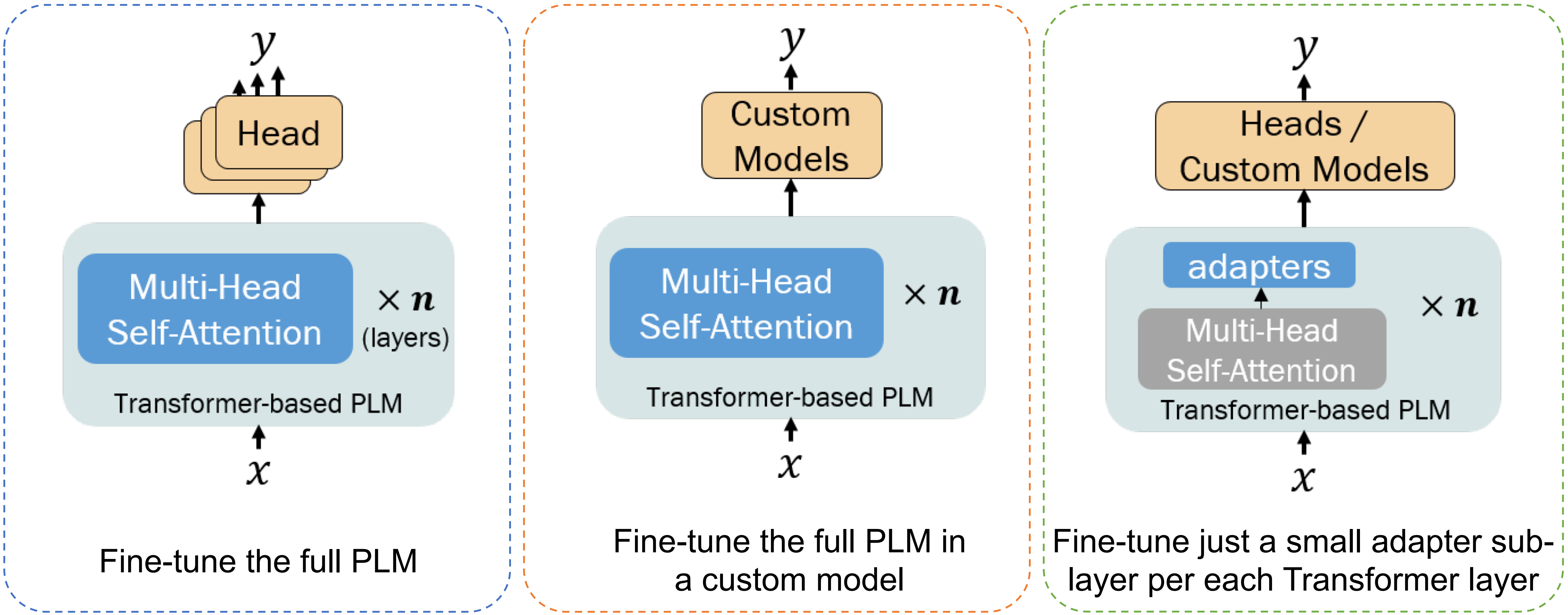
Fine-tuning in NLP

- What is Fine-tuning and Why Fine-tuning?
  - **Gaps** between the pre-training tasks and downstream tasks still exist
  - Fine-tuning means **training pre-trained LLMs on downstream tasks** to fit the requirements



Fine-tuning in NLP: pre-train then fine-tune

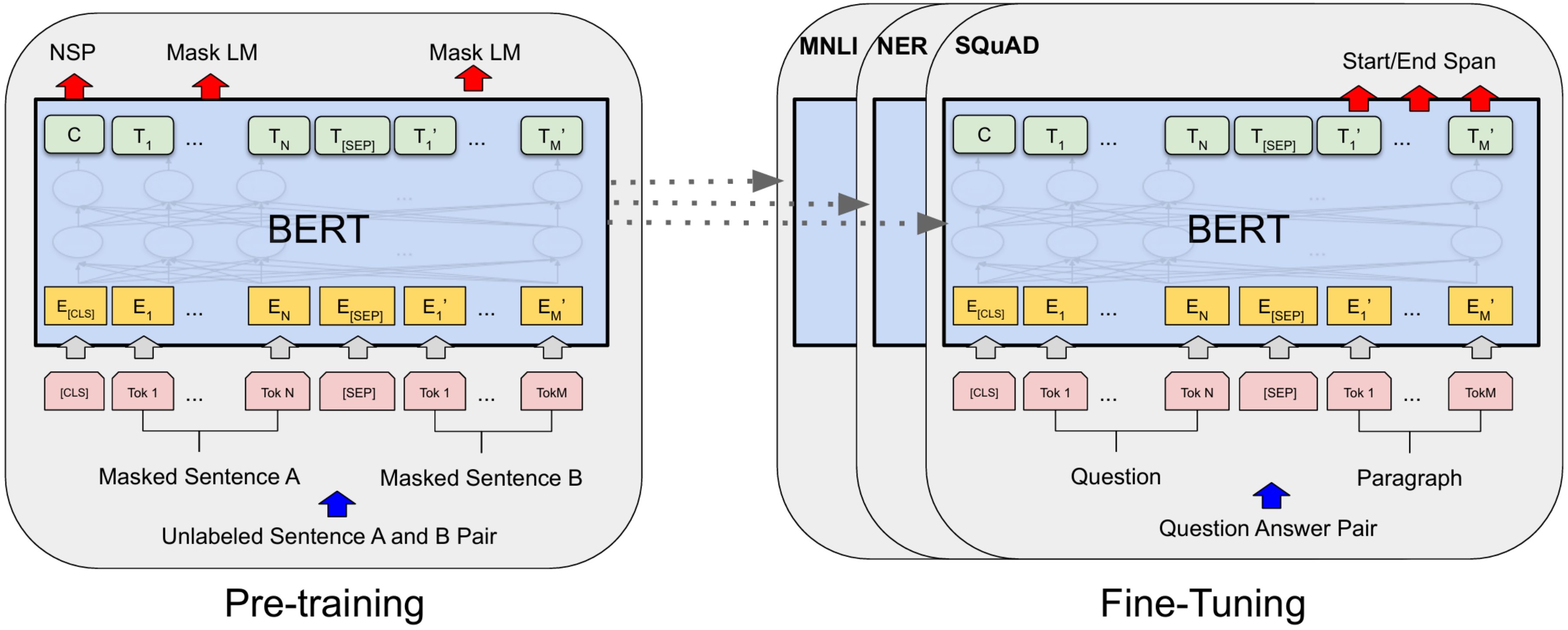
- Typical "pre-train then fine-tune" strategies



"Recent advances in natural language processing via large pre-trained language models: A survey." ACM Computing Surveys 56.2 (2023): 1-40.

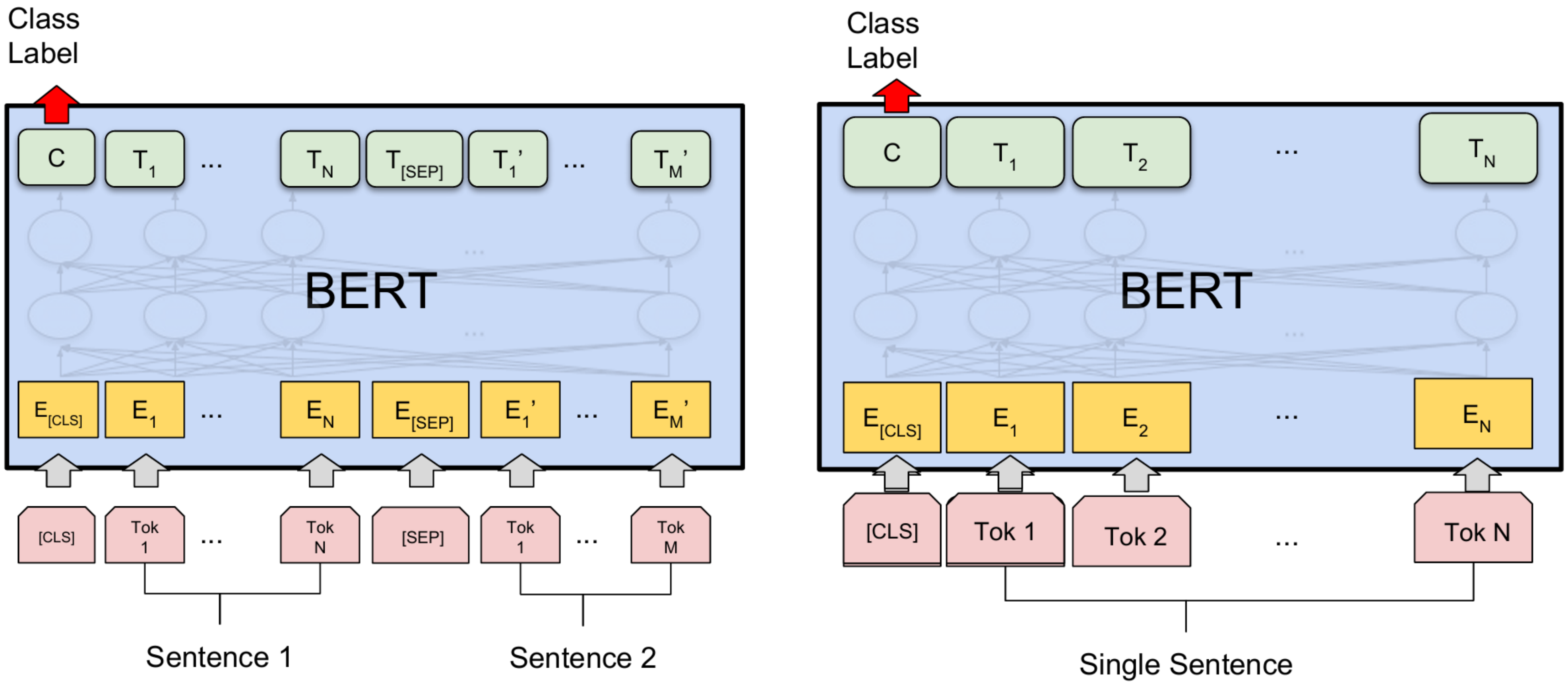
Fine-tuning the pre-trained language model (PLM)

- This approach **fine-tunes some or all the layers of the PLM** and then adds one or two simple output layers (known as **prediction heads**).



"Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018).

Fine-tuning in NLP: Fine-tuning the PLM

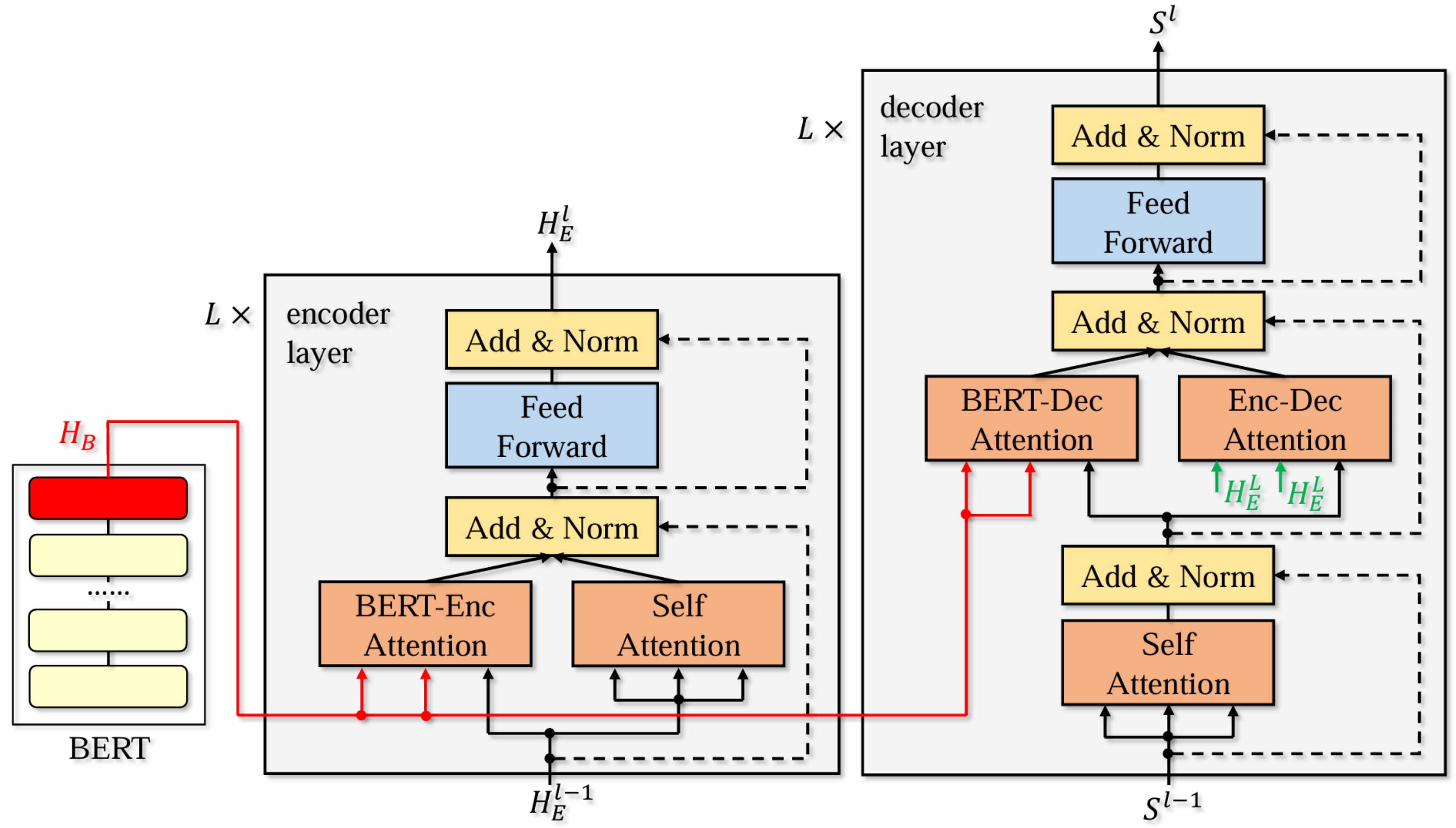


"Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018).

Fine-tuning in NLP: Fine-tuning the PLM

❑ Customized Models

❖ Some tasks require significant **additional architecture on top of a language model**.



"Incorporating BERT into Neural Machine Translation." International Conference on Learning Representations.

PART 4: RecSys Fine-tuning

Website

Website of this tutorial

- **Fine-tuning in NLP**
  - Pre-training then fine-tuning
  - **Parameter Efficient Fine-tuning (PEFT)**
  - Fine-tuning with reinforcement learning
- Fine-tuning LLM-based RecSys
- Parameter efficient fine-tuning for LLM-based RecSys

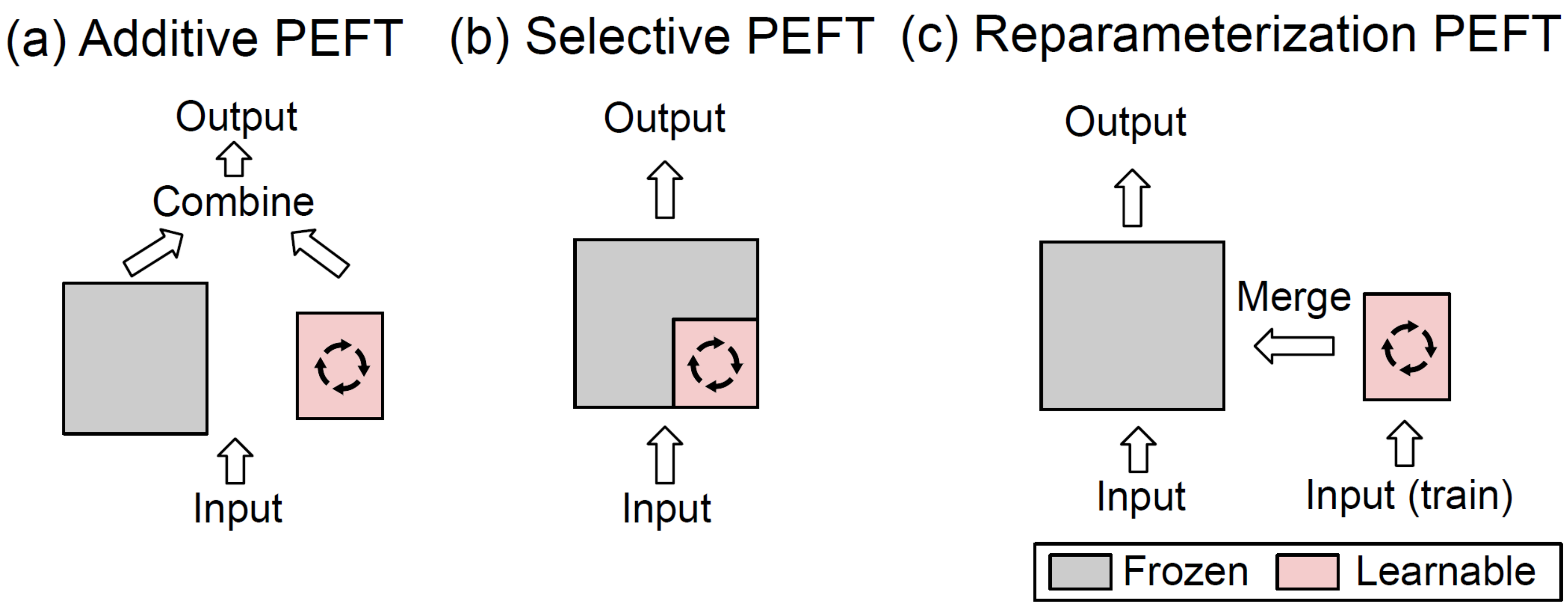
Fine-tuning in NLP: Parameter Efficient Fine-tuning (PEFT)

- What is Parameter Efficient Fine-tuning (PEFT)?
  - As LLMs scale up to **billion weights**, consumable GPUs like 3090 and 4090 gradually fail to contain all the weights in their memory
  - Parameter Efficient Fine-tuning aims to save GPU memory and boost training
- Why PEFT?
  - Making fine-tuning feasible for consumable GPUs
  - With major parameters fixed, it might relieve the problem of **catastrophic forgetting**

"Recent advances in natural language processing via large pre-trained language models: A survey." ACM Computing Surveys 56.2 (2023): 1-40.

Fine-tuning in NLP: PEFT

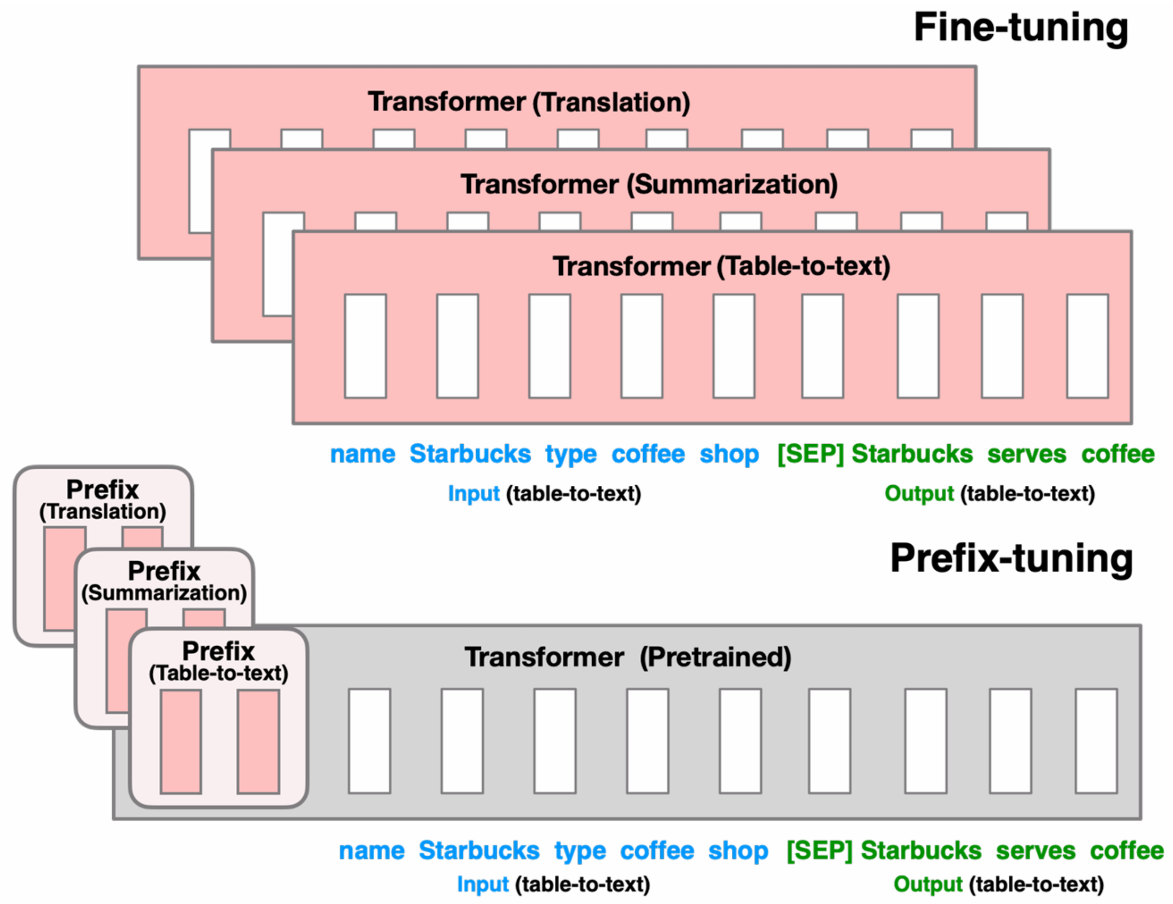
- **Fine-tuning a separate, small network** that is tightly coupled with the PLM.
- **Selecting only a small number of the PLM's weights** to fine-tune or keep.
- Introduce additional **low-rank trainable parameters**, which are integrated with the original model.



"Parameter-efficient fine-tuning for large models: A comprehensive survey." arXiv preprint arXiv:2403.14608 (2024).

Fine-tuning in NLP: Prefix-tuning

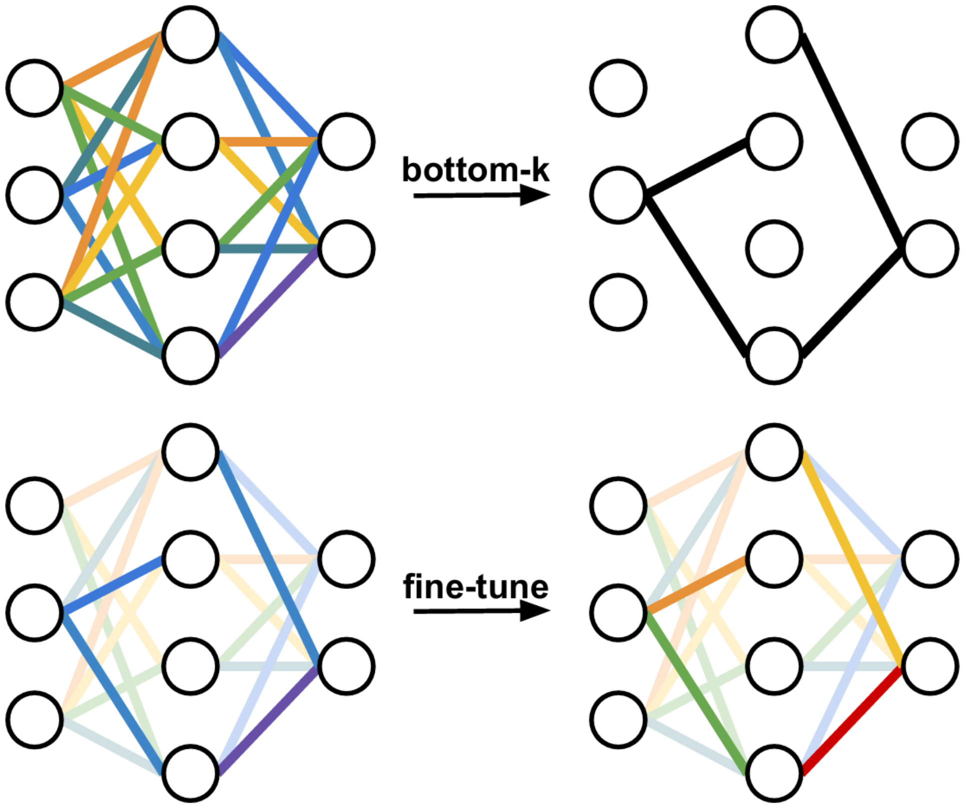
- Freezes the Transformer parameters and **only optimizes the prefix** (the red prefix blocks)



"Prefix-Tuning: Optimizing Continuous Prompts for Generation." ACL. 2021.

Fine-tuning in NLP: PaFi

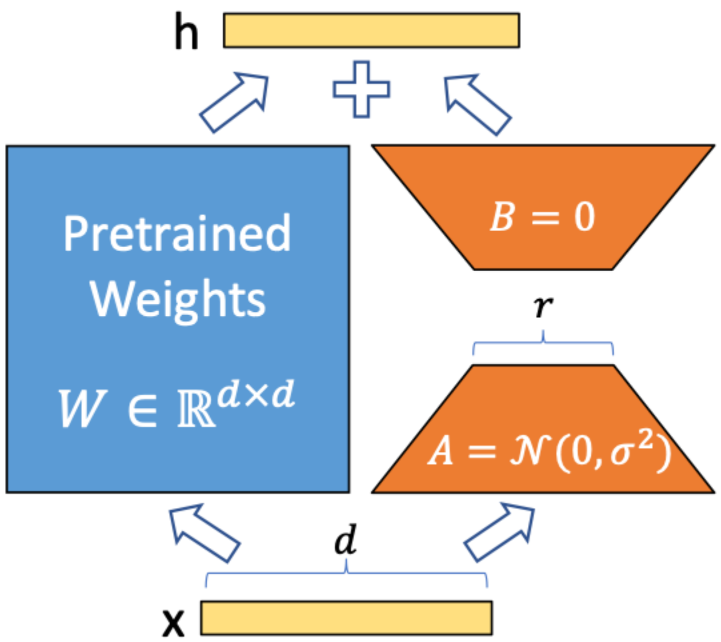
- ❑ Select model parameters with the **smallest absolute magnitude as trainable**



"Parameter-Efficient Fine-Tuning without Introducing New Latency." ACL. 2023.

Fine-tuning in NLP: Low-Rank Adaptation of LLMs (LoRA)

- ❑ Fine-tuning a 7B model needs 7,000,000,000*8/1024^3 ≅52GB GPU memory
- ❑ LoRA only fine-tunes the **feed-forward networks** (FFNs)
  - ❖ Making it possible for consumable GPUs to train 7B and even 13B LLMs

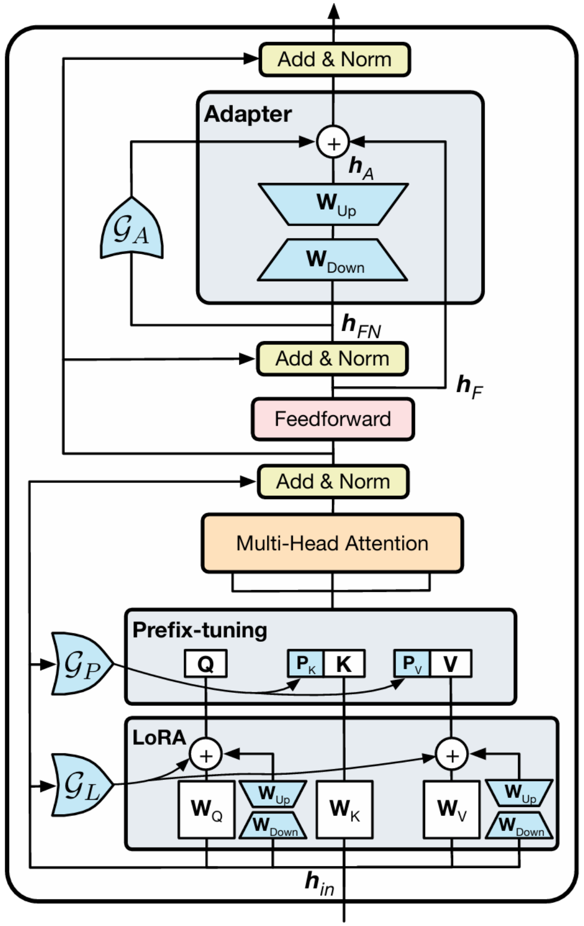


$$h = W_0x + \Delta Wx = W_0x + BAx$$

"Lora: Low-rank adaptation of large language models." arXiv preprint (2021).

Fine-tuning in NLP: UNIPELT

- Integrate multiple PELT methods and controls them via a **gating mechanism**



"UNIPELT: A Unified Framework for Parameter-Efficient Language Model Tuning." ACL, 2022.

PART 4: RecSys Fine-tuning

Website

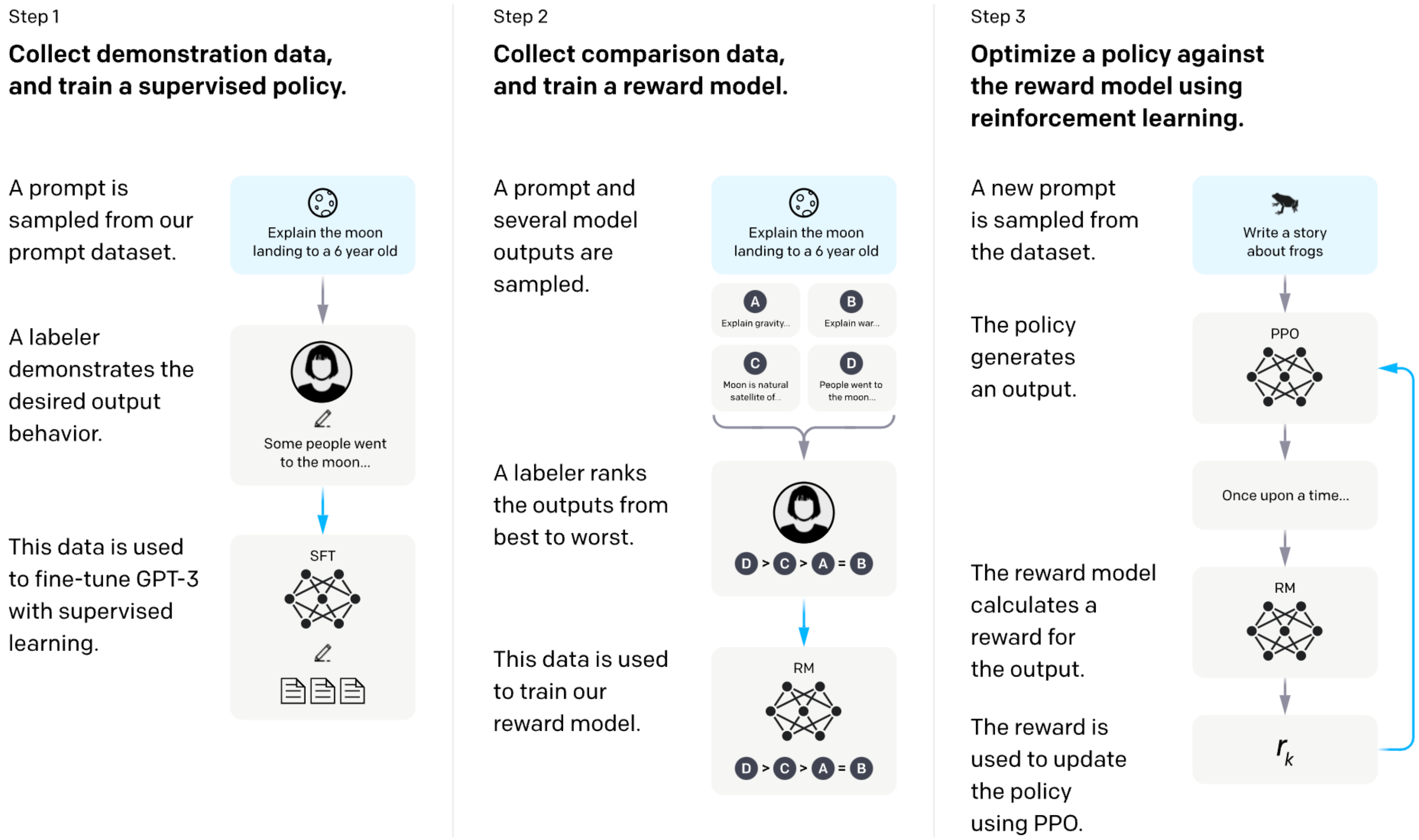
Website of this tutorial

- **Fine-tuning in NLP**
  - Pre-training then fine-tuning
  - Parameter Efficient Fine-tuning (PEFT)
  - **Fine-tuning with reinforcement learning**
- **Fine-tuning LLM-based RecSys**
- **Parameter efficient fine-tuning for LLM-based RecSys**

Fine-tuning with Reinforcement Learning

- Reinforcement Learning based on Human Feedbacks (RLHF)
- Proximal Policy Optimization (PPO)
- Direct Preference Optimization (DPO)

Fine-tuning with Reinforcement Learning: RLHF

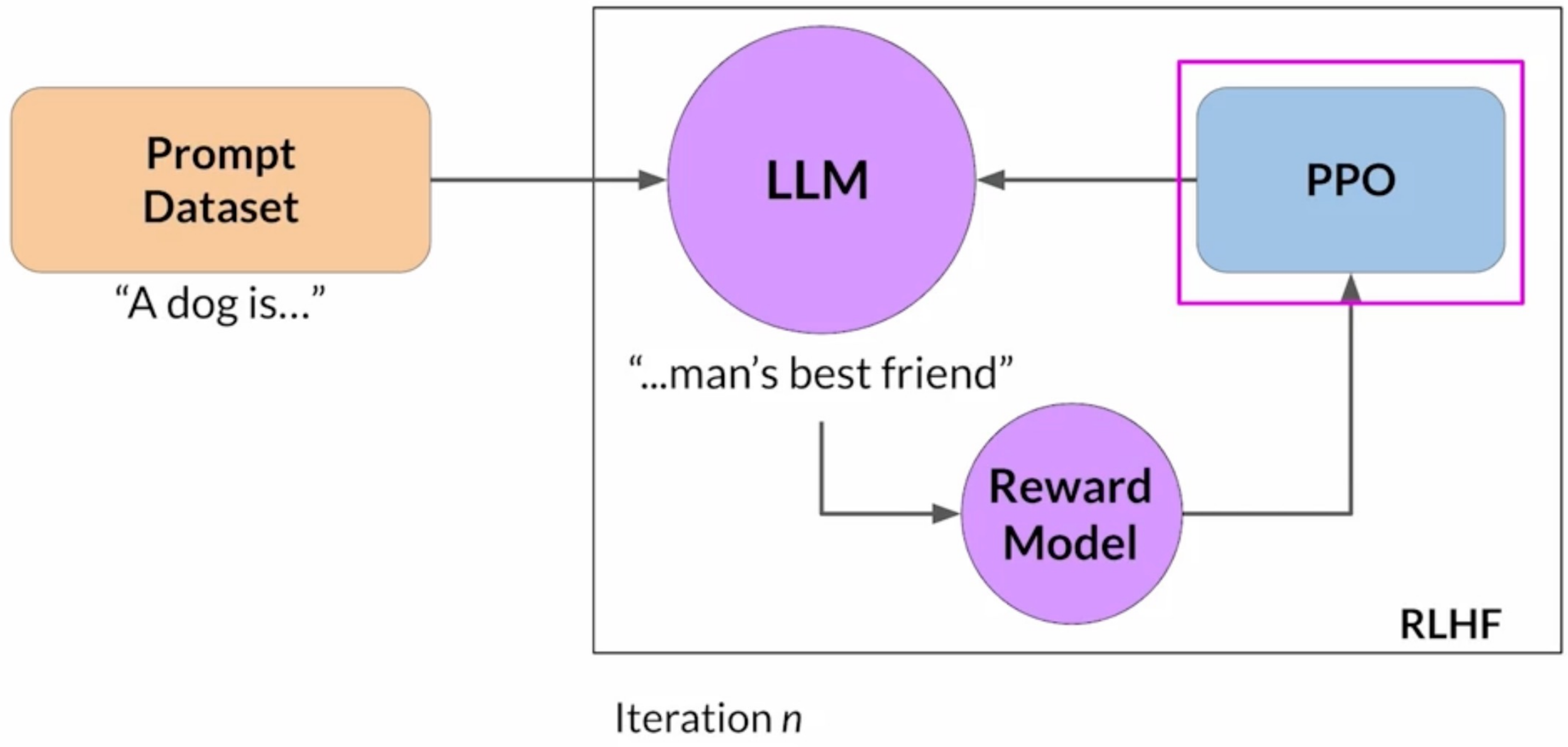


"Deep reinforcement learning from human preferences." Advances in neural information processing systems (2017).

Fine-tuning with Reinforcement Learning

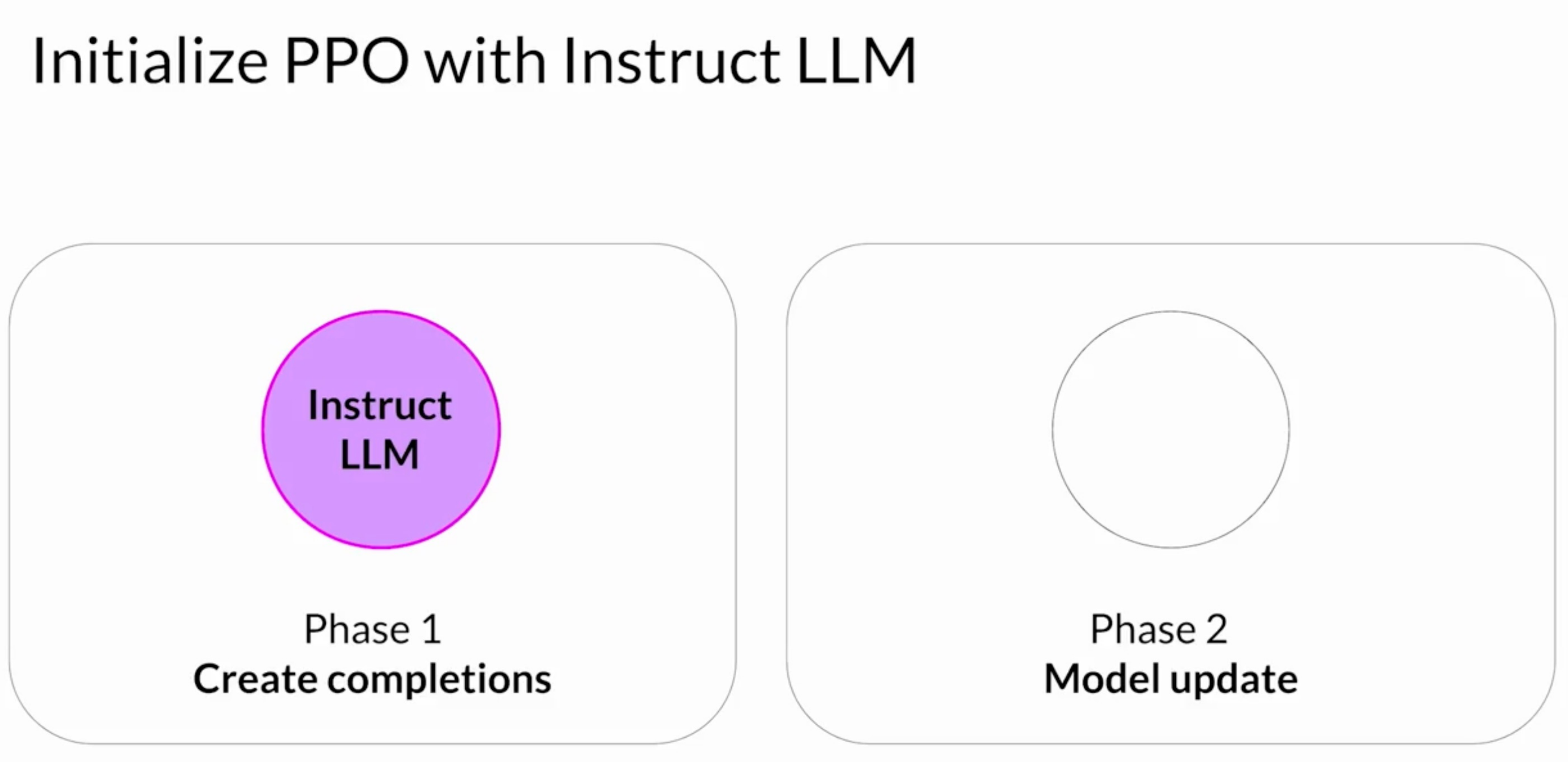
- Reinforcement Learning based on Human Feedbacks (RLHF)
- Proximal Policy Optimization (PPO)
- Direct Preference Optimization (DPO)

Fine-tuning with Reinforcement Learning: PPO



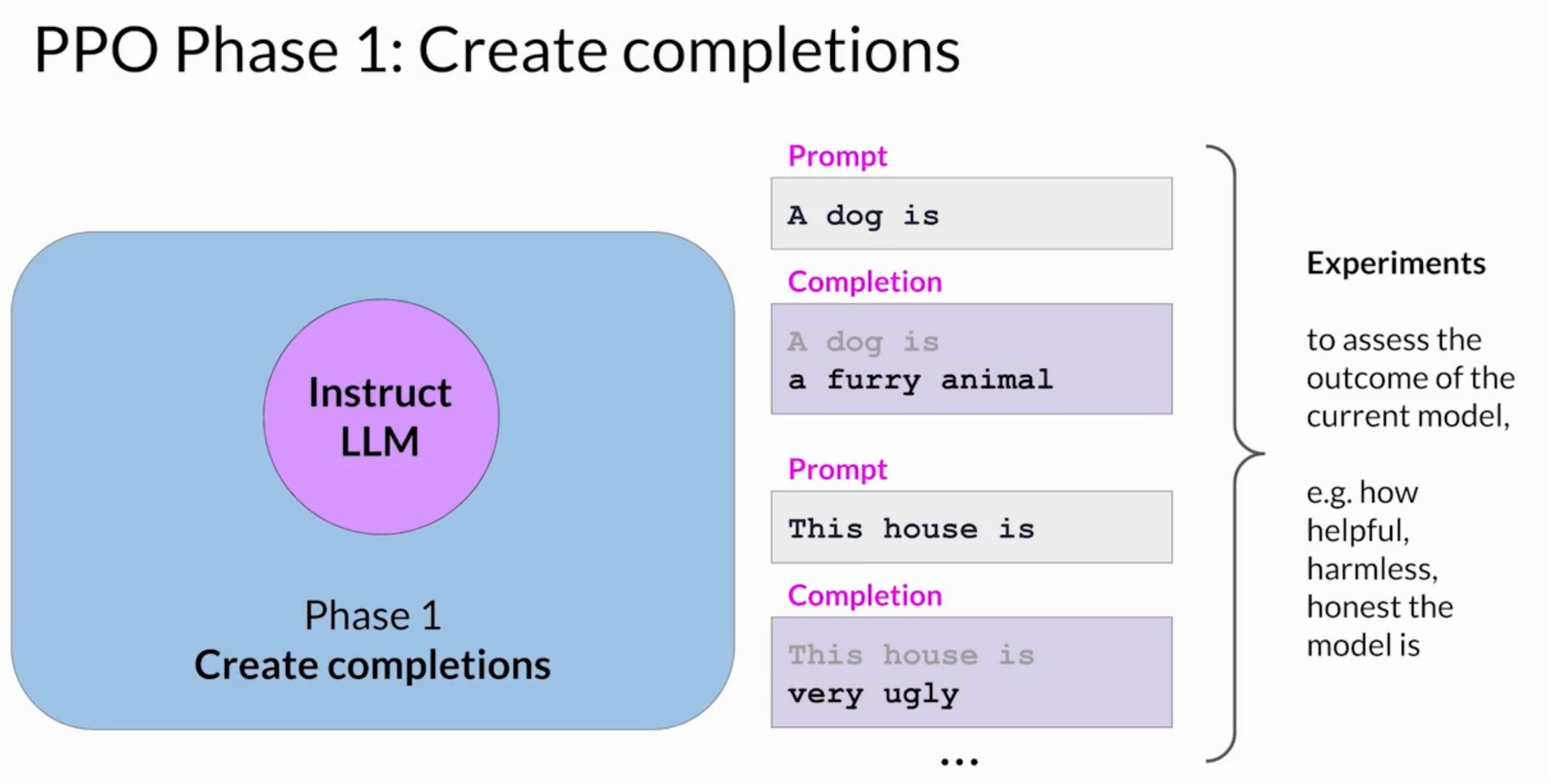
https://www.coursera.org/lecture/generative-ai-with-llms/optional-video-proximal-policy-optimization-1iZJO

Fine-tuning with Reinforcement Learning: PPO



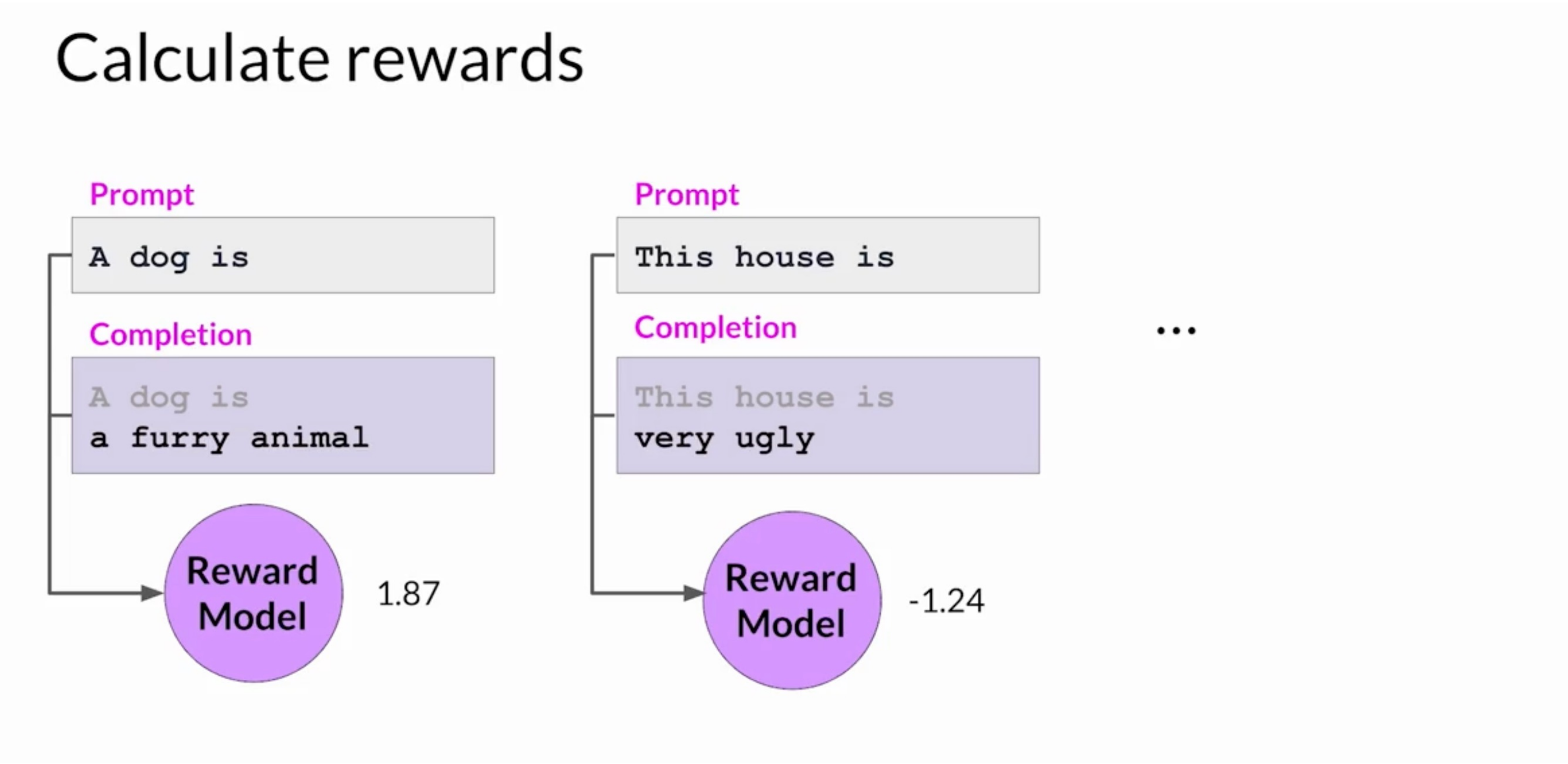
https://www.coursera.org/lecture/generative-ai-with-llms/optional-video-proximal-policy-optimization-1iZJO

Fine-tuning with Reinforcement Learning: PPO



https://www.coursera.org/lecture/generative-ai-with-llms/optional-video-proximal-policy-optimization-1iZJO

Fine-tuning with Reinforcement Learning: PPO



https://www.coursera.org/lecture/generative-ai-with-llms/optional-video-proximal-policy-optimization-1iZJO

Fine-tuning with Reinforcement Learning: PPO

Calculate value loss

Prompt

```
A dog is
```

Completion

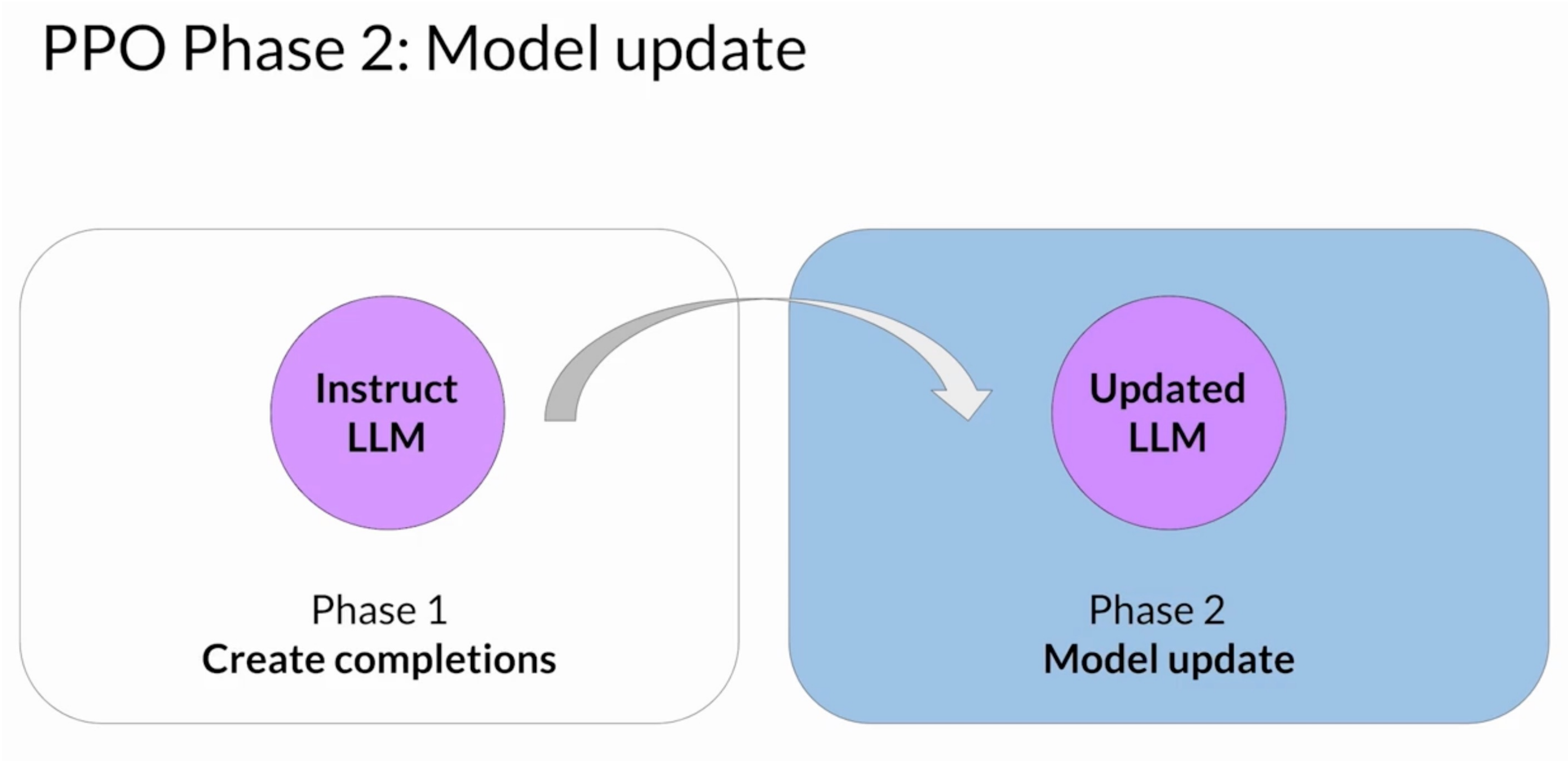
```
A dog is
a furry...
```

Value loss

$$\boxed{L^{VF}} = \frac{1}{2}\left\|\underbrace{V_\theta(s)}_{\substack{\textbf{Estimated}\\ \text{future total reward}}} - \underbrace{\left(\sum_{t=0}^{T}\gamma^{t} r_t \mid s_0 = s\right)}_{\substack{\textbf{Known}\\ \text{future total reward}}}\right\|_2^2$$

Estimated future total reward	Known future total reward
1.23	1.87

Fine-tuning with Reinforcement Learning: PPO



https://www.coursera.org/lecture/generative-ai-with-llms/optional-video-proximal-policy-optimization-1iZJO

Fine-tuning with Reinforcement Learning: PPO

PPO Phase 2: Calculate policy loss

$$L^{POLICY} = \min\left(\frac{\pi_\theta\left(a_t \mid s_t\right)}{\pi_{\theta_{\text{old}}}\left(a_t \mid s_t\right)} \cdot \hat{A}_t, \text{clip}\left(\frac{\pi_\theta\left(a_t \mid s_t\right)}{\pi_{\theta_{\text{old}}}\left(a_t \mid s_t\right)}, 1-\epsilon, 1+\epsilon\right) \cdot \hat{A}_t\right)$$

https://www.coursera.org/lecture/generative-ai-with-llms/optional-video-proximal-policy-optimization-1iZJO

Fine-tuning with Reinforcement Learning: PPO

PPO Phase 2: Calculate entropy loss

$$L^{ENT} = \text{entropy}\left(\pi_\theta\left(\cdot \mid s_t\right)\right)$$

Low entropy:

Prompt

```
A dog is
```

Completion

```
A dog is
a domesticated
carnivorous mammal
```

Prompt

```
A dog is
```

Completion

```
A dog is
a small carnivorous
mammal
```

High entropy:

Prompt

```
A dog is
```

Completion

```
A dog is
is one of the most
popular pets around
the world
```

https://www.coursera.org/lecture/generative-ai-with-llms/optional-video-proximal-policy-optimization-1iZJO

Fine-tuning with Reinforcement Learning: PPO

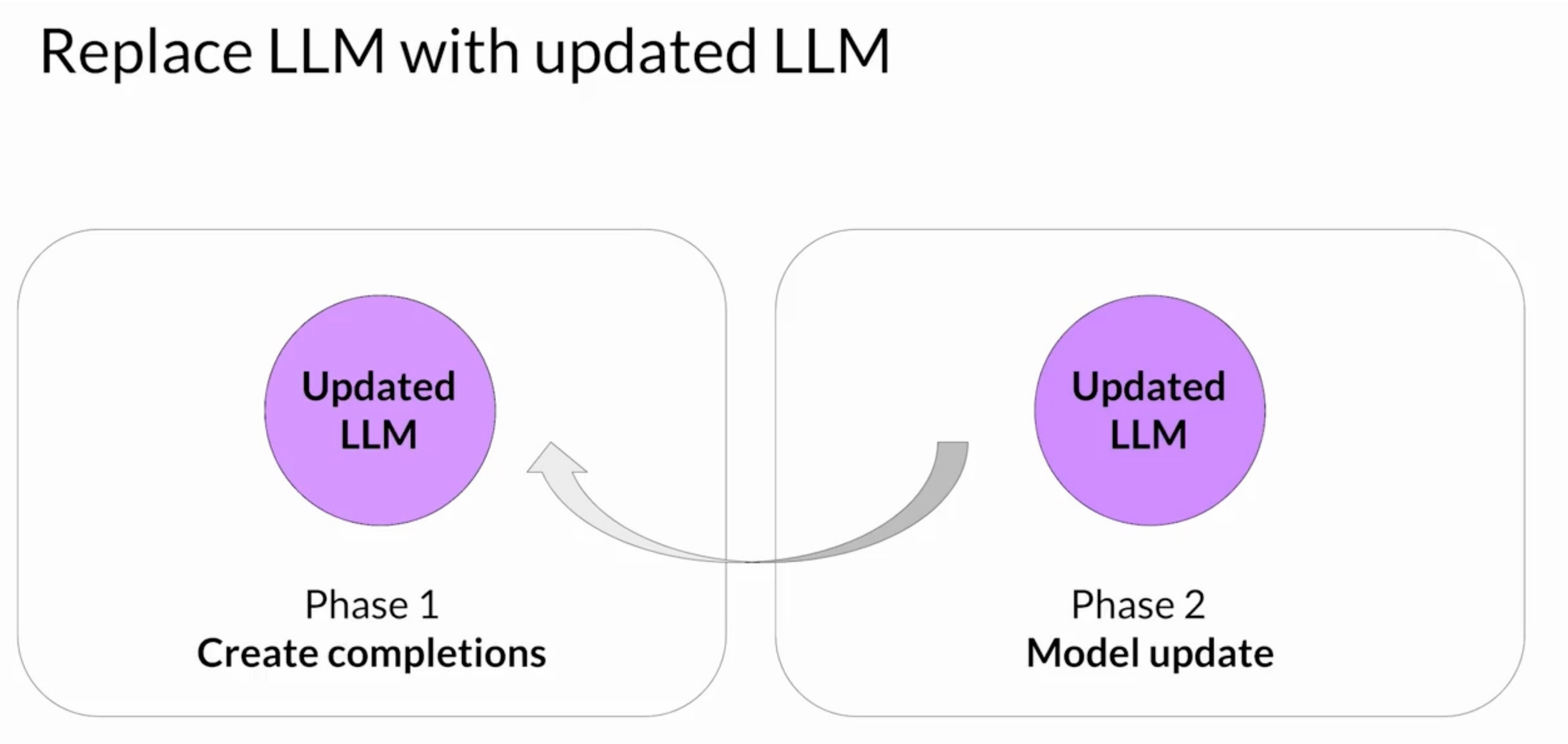
PPO Phase 2: Objective function

Hyperparameters

$$L^{PPO} = \underbrace{L^{POLICY}}_{\text{Policy loss}} + \underbrace{c_1 L^{VF}}_{\text{Value loss}} + \underbrace{c_2 L^{ENT}}_{\text{Entropy loss}}$$

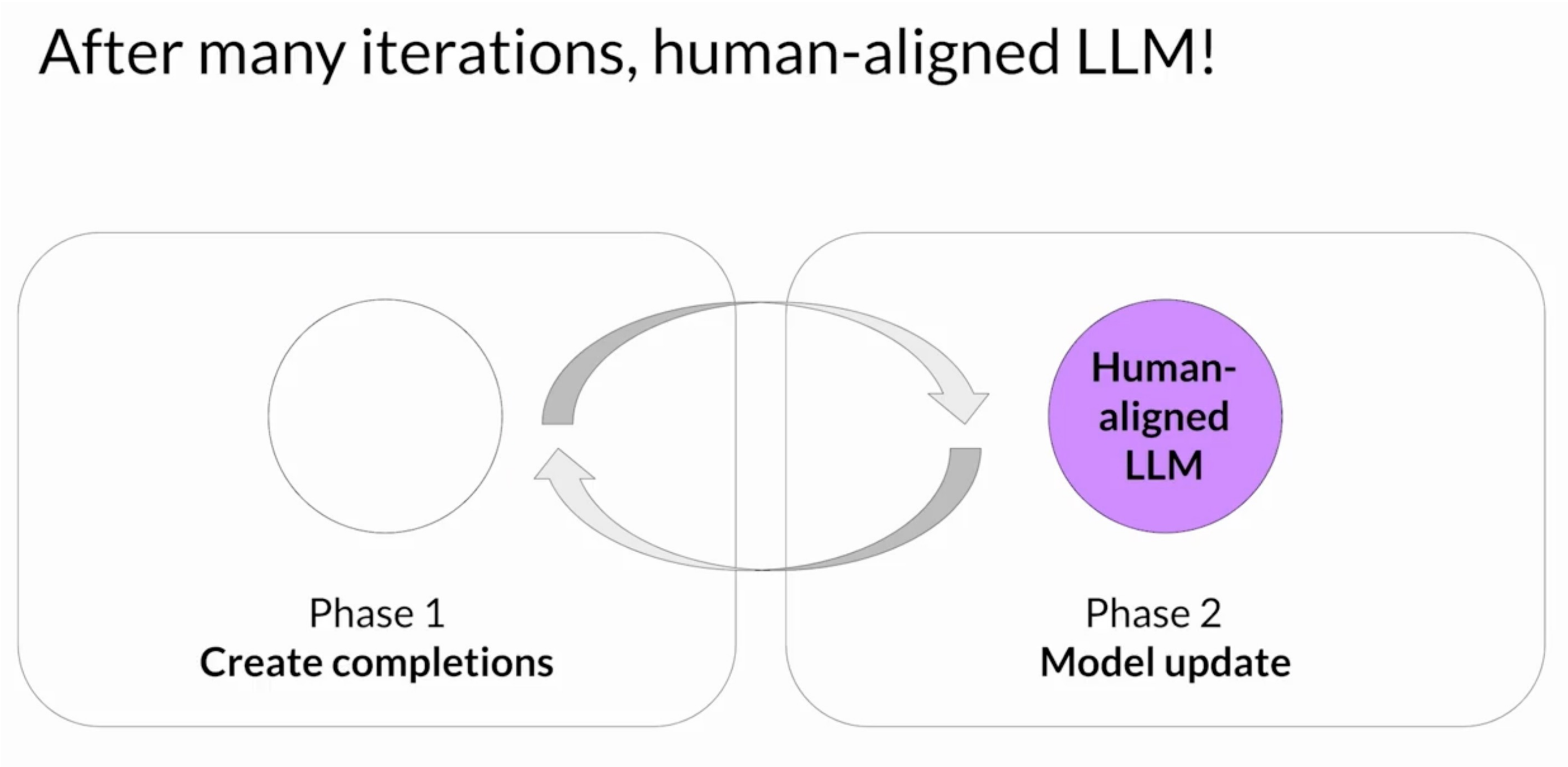
https://www.coursera.org/lecture/generative-ai-with-llms/optional-video-proximal-policy-optimization-1iZJO

Fine-tuning with Reinforcement Learning: PPO



https://www.coursera.org/lecture/generative-ai-with-llms/optional-video-proximal-policy-optimization-1iZJO

Fine-tuning with Reinforcement Learning: PPO



https://www.coursera.org/lecture/generative-ai-with-llms/optional-video-proximal-policy-optimization-1iZJO

Fine-tuning with Reinforcement Learning

- Reinforcement Learning based on Human Feedbacks (RLHF)
- Proximal Policy Optimization (PPO)
- Direct Preference Optimization (DPO)

Fine-tuning with Reinforcement Learning: DPO

Policy to optimize

Aggregation over preference data

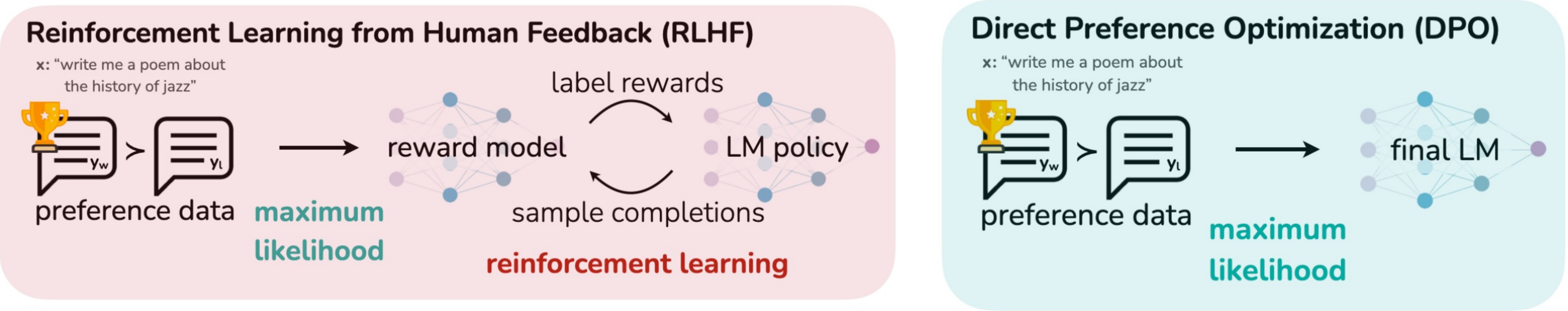
Shift in **preferred** completion

$$\mathcal{L}_{\mathrm{DPO}}(\pi_\theta; \pi_{\mathrm{ref}}) = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \left[\log \sigma \left(\beta \log \frac{\pi_\theta(y_w \mid x)}{\pi_{\mathrm{ref}}(y_w \mid x)} - \beta \log \frac{\pi_\theta(y_l \mid x)}{\pi_{\mathrm{ref}}(y_l \mid x)} \right) \right]$$

Reference policy (used to control behavior of LLMs)

Logistic function

Shift in **dispreferred** completion



"Direct preference optimization: Your language model is secretly a reward model." arXiv preprint (2023).

PART 4: RecSys Fine-tuning

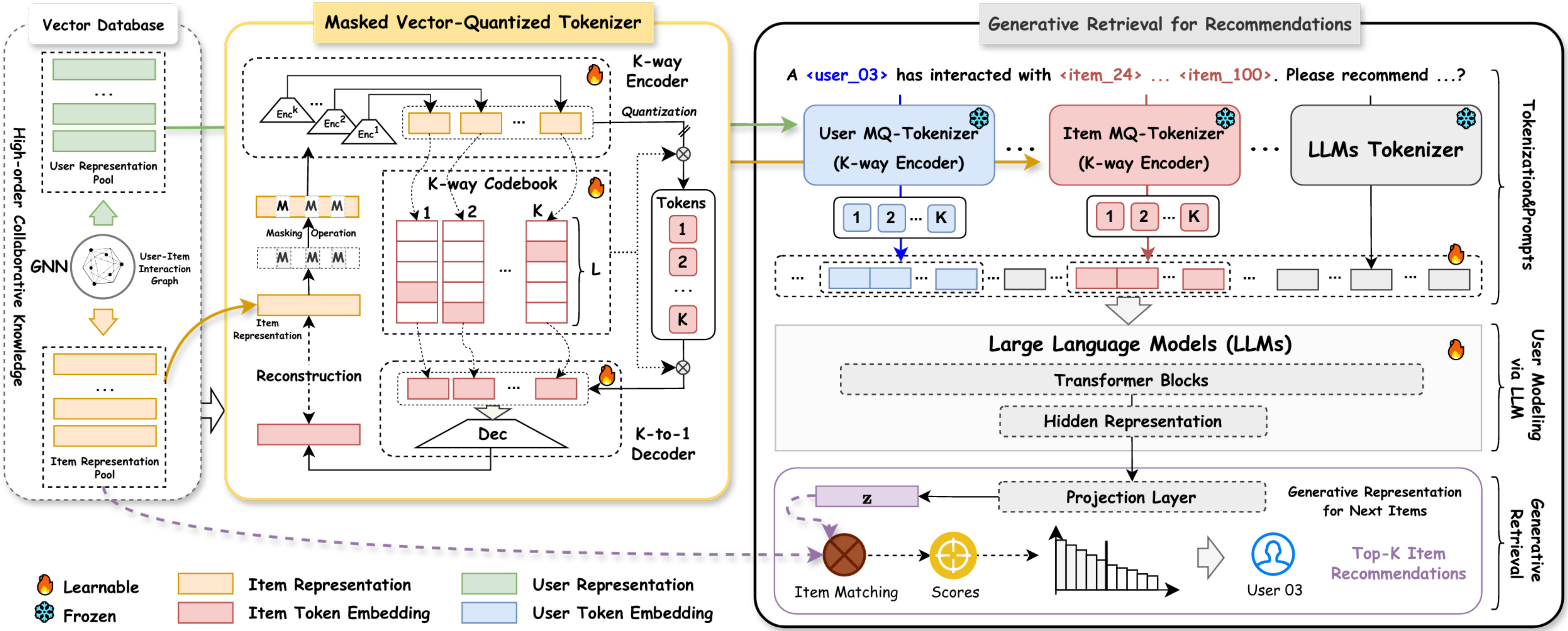
Website

Website of this tutorial

- **Fine-tuning in NLP**
  - Pre-training then fine-tuning
  - Parameter Efficient Fine-tuning (PEFT)
  - Fine-tuning with reinforcement learning
- **Fine-tuning LLM-based RecSys**
- Parameter efficient fine-tuning for LLM-based RecSys

TokenRec

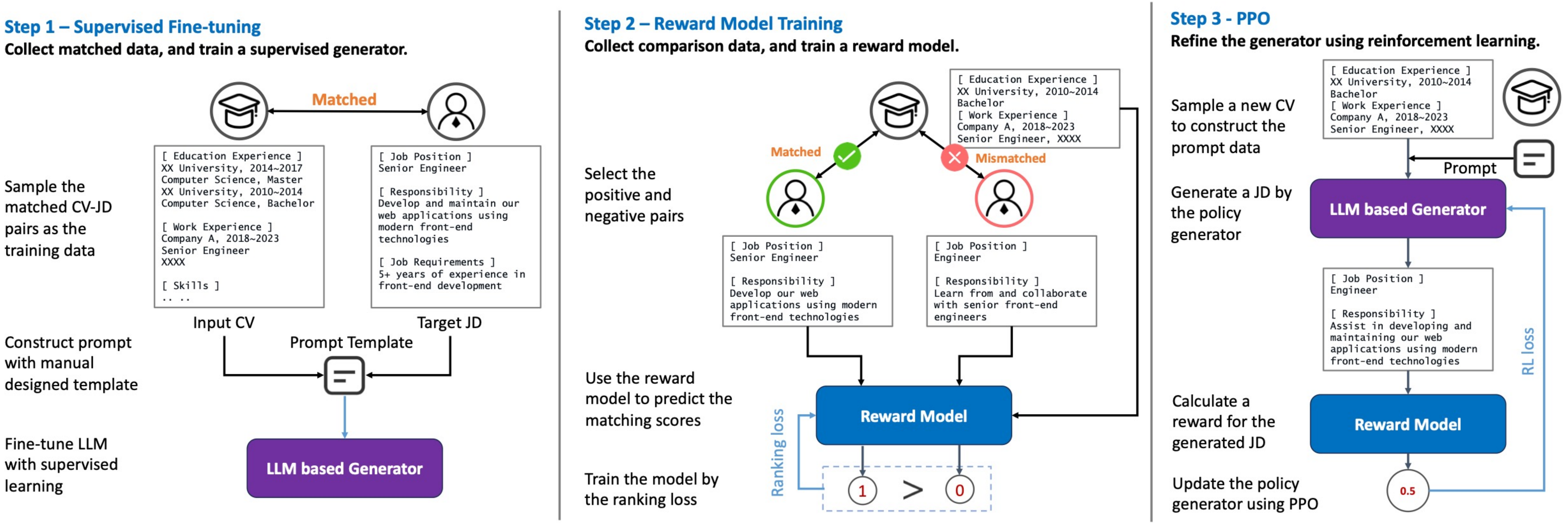
❑ Fine-tuning LLM-based RecSys with SFT



"TokenRec: Learning to Tokenize ID for LLM-based Generative Recommendation." arXiv preprint arXiv:2406.10450 (2024).

GIRL

- Multi-steps of Fine-tuning with SFT and RL

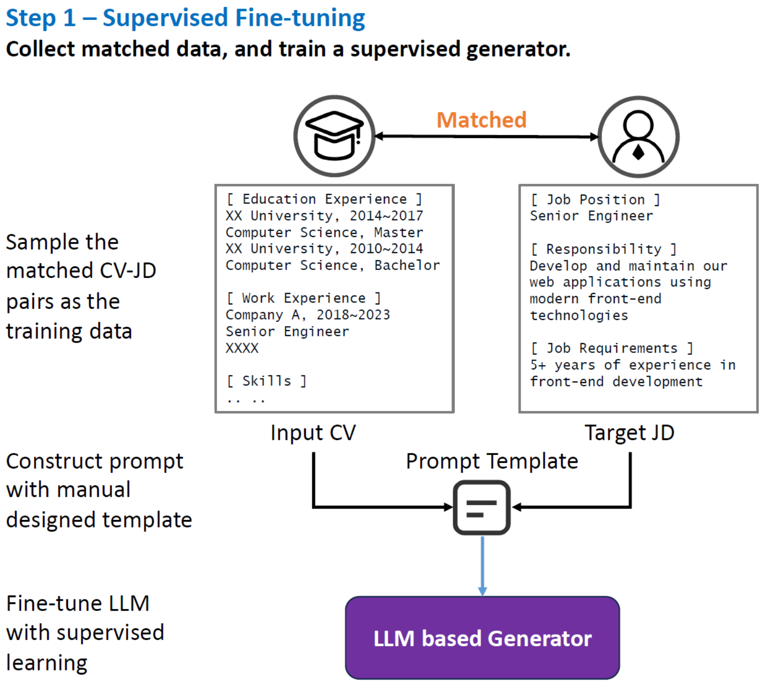


"Generative job recommendations with large language model." arXiv preprint (2023).

GIRL

❑ Multi-steps of Fine-tuning with SFT and RL

❖ Train a generator with the casual language model pre-training task



Human:
Please generate a suitable job description based on the curriculum vitae of the following candidate:
Basic information: The candidate is a 27-year-old female with a bachelor's degree, majoring in Software Engineering.
personal profile:
1. She has research experience in the field of data processing and computation, and development experience with components from ElasticSearch, Hbase, and the Flink ecosystem.
2. She possesses strong abilities to read and understand source code, having reviewed core source code from JDK8, Spring, Spring Boot, and Netty.

Assistant:
Job title: Java Developer
Salary range: 25k-40k* 15
Responsibilities:
1. Support the recommendation product and algorithm departments, rapidly iterate daily needs, complete effect analysis and continuous improvements.
2. Track industry developments, timely introduction of external advanced experiences.

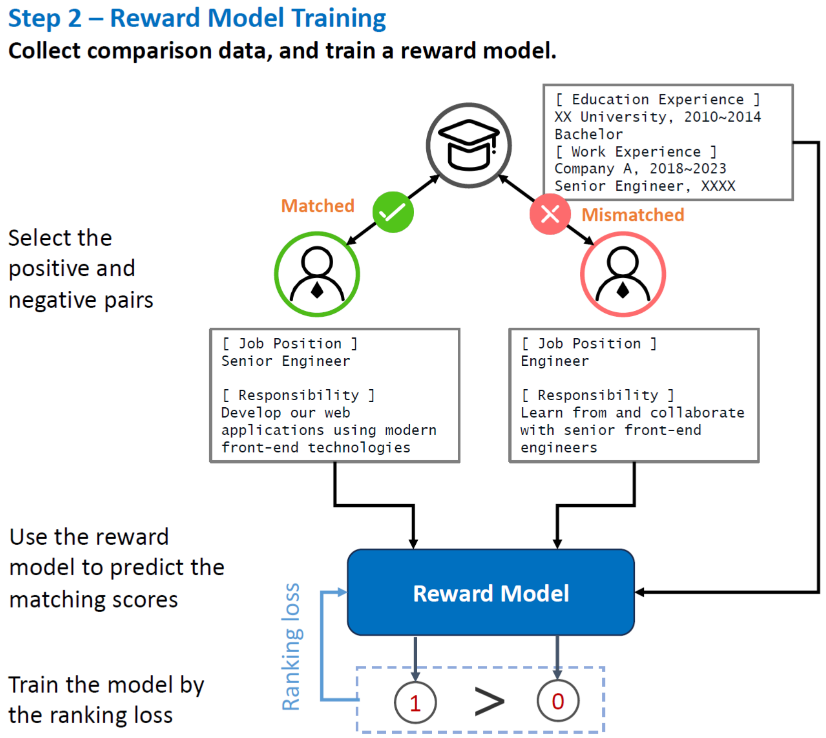
Qualifications:
1. Proficient in Java programming basics, with excellent abilities and experience in Java IO and multithreading programming.
2. In-depth understanding of JVM, JVM tuning experience, and experience with distributed systems and concurrency scenarios are preferred.
3. Proficient in applying mainstream development frameworks and open-source systems such as Spring Boot, MyBatis, MySQL, Redis, ES, Kafka, etc.
4. Good stress resistance, communication, learning, collaboration skills, and a strong sense of responsibility.
5. Prior experience in recommendation/search engineering development in Internet companies is preferred.

$$\mathcal{L}_{sft} = -\log \Pr(C|J,T,\mathcal{G})$$
$$= -\sum_{i=1}^{|l_j|} \log \Pr(v_i|v_{<i},C,T,\mathcal{G}),$$

"Generative job recommendations with large language model." arXiv preprint (2023).

GIRL

- Multi-steps of Fine-tuning with SFT and RL
  - **Train a reward model *U* that can predict the matching score between a CV-JD pair**



$$\mathcal{L}_{rmt} = \log\sigma(\mathcal{U}(C, J^{+}) - \mathcal{U}(C, J^{-})),$$

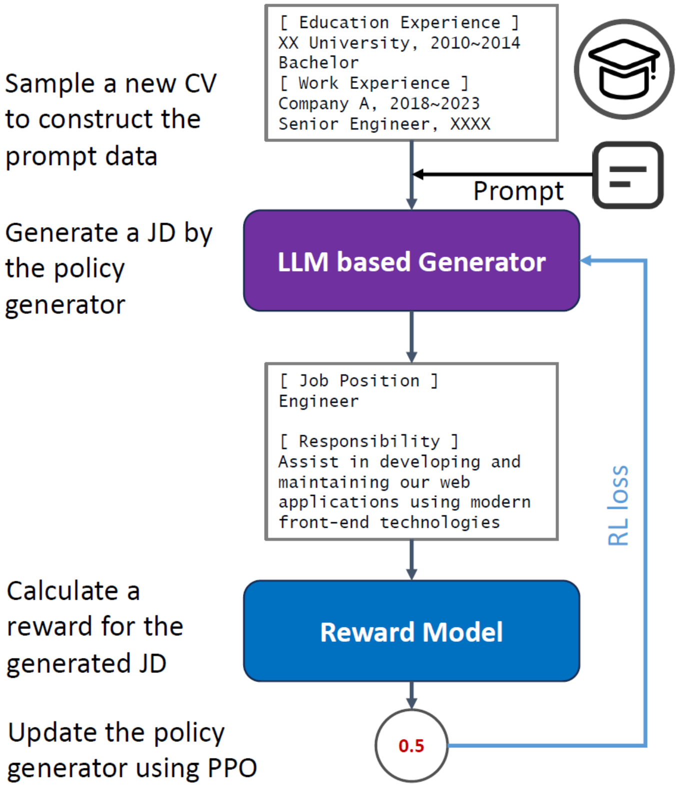
"Generative job recommendations with large language model." arXiv preprint (2023).

GIRL

- Multi-steps of Fine-tuning with SFT and RL
  - **Improve the alignment** between the generator and the recruiter feedback acquired by the reward model through **reinforcement learning**

Step 3 - PPO

Refine the generator using reinforcement learning.



$$\mathcal{L}_{am} = \frac{1}{|\mathcal{J}_i^r|} \sum_{v_j \in \mathcal{J}_i^r} \min\Big(CE(v_{i,j}) a_i, \mathrm{clip}(CE(v_{i,j})) a_i \Big)$$

$$\mathcal{L}_{cm} = (r_i - \mathcal{U}^c(\mathcal{C}_i^r, \_))^2$$

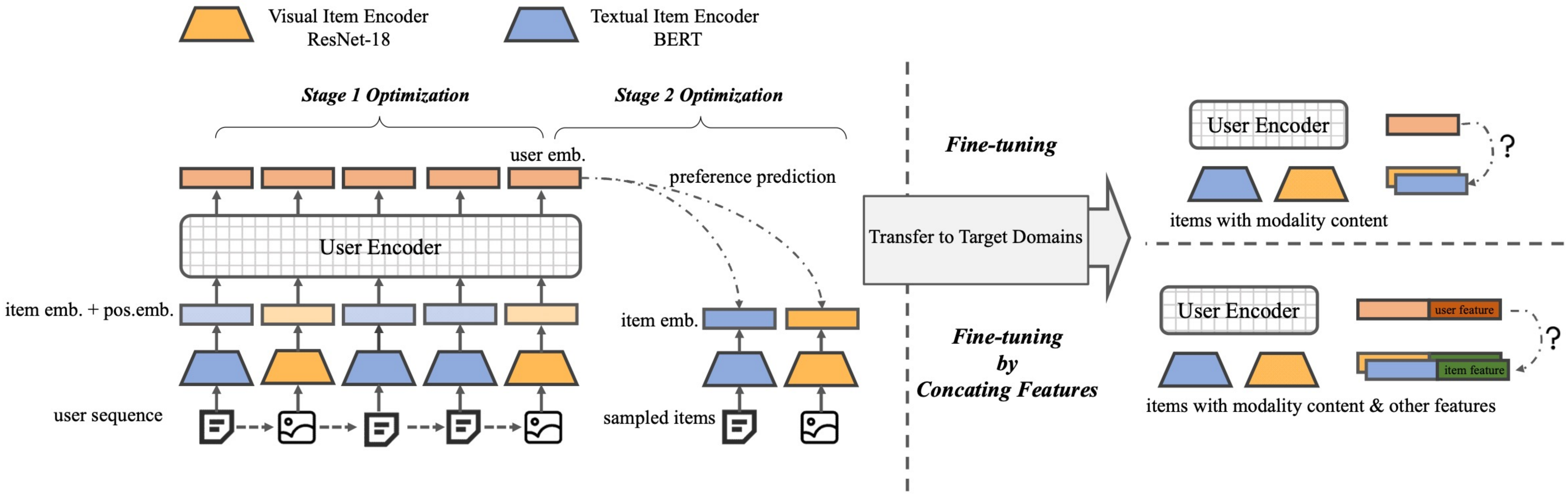
$$CE(v_j) = \frac{\Pr(v_j | v_{i,<j}, C, \mathcal{G}^a)}{\Pr(v_j | v_{i,<j}, C, \mathcal{G})},$$

$$a_i = r_i - \mathcal{U}^c(\mathcal{C}_i^r, \_).$$

"Generative job recommendations with large language model." arXiv preprint (2023).

TransRec

- Fine-tuning LLM-based RecSys with Cross-Modal Data



(a) Pre-training on Source Domain

(b) Fine-tuning on Target Domains

"Transrec: Learning transferable recommendation from mixture-of-modality feedback." arXiv preprint (2022).

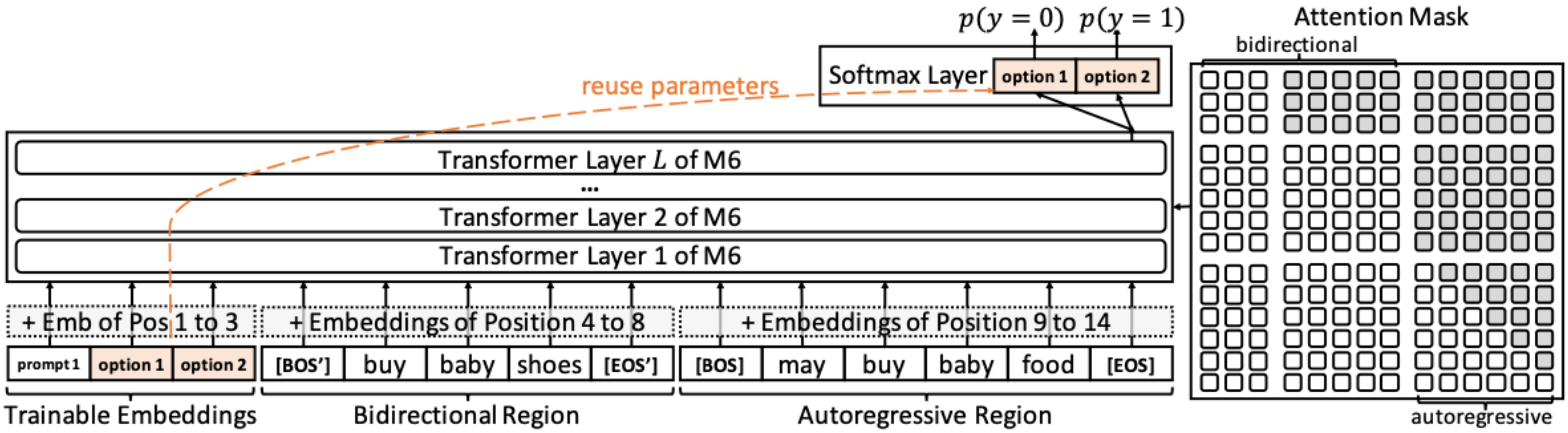
PART 4: RecSys Fine-tuning

Website of this tutorial

- **Fine-tuning in NLP**
  - Pre-training then fine-tuning
  - Parameter Efficient Fine-tuning (PEFT)
  - Fine-tuning with reinforcement learning
- **Fine-tuning LLM-based RecSys**
- **Parameter efficient fine-tuning for LLM-based RecSys**

M6-Rec

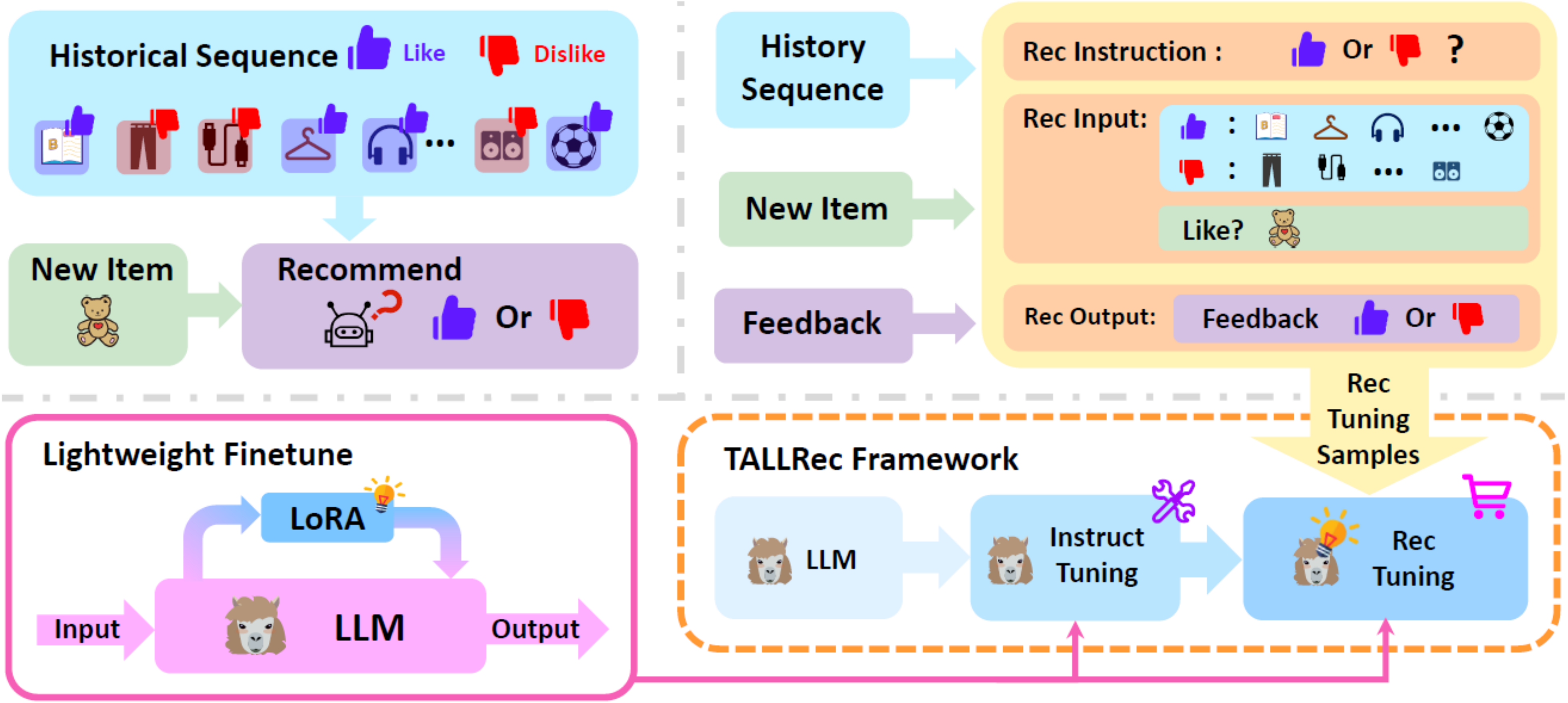
- Option Adapter Fine-tunes LLMs



"M6-rec: Generative pretrained language models are open-ended recommender systems." arXiv preprint (2022).

TALLRec

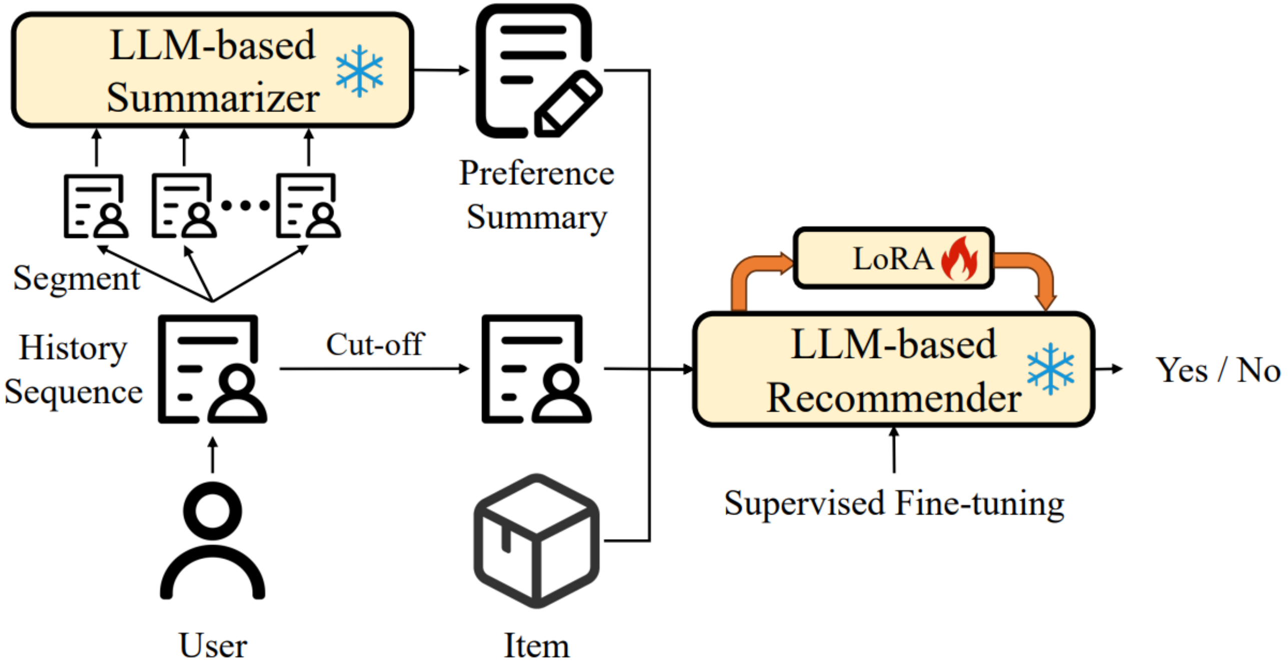
- LoRA Fine-tune LLMs

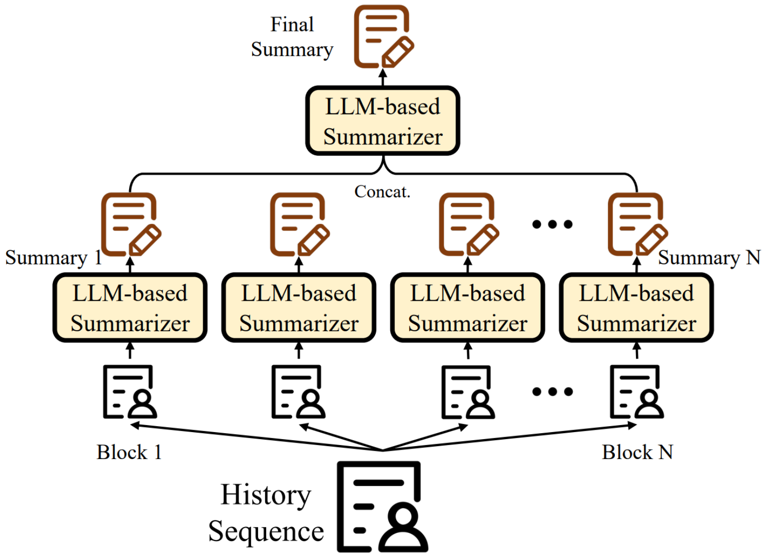


"Tallrec: An effective and efficient tuning framework to align large language model with recommendation." RecSys (2023).

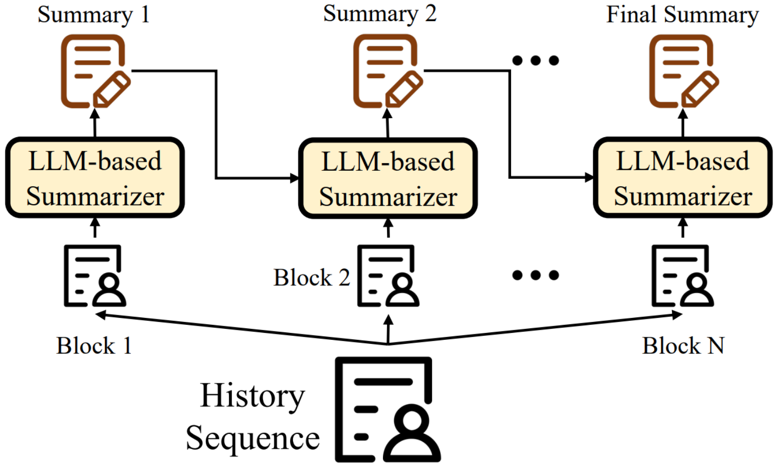
LLM-TRSR

- LoRA Fine-tune LLMs





Hierarchical Summarization Paradigm

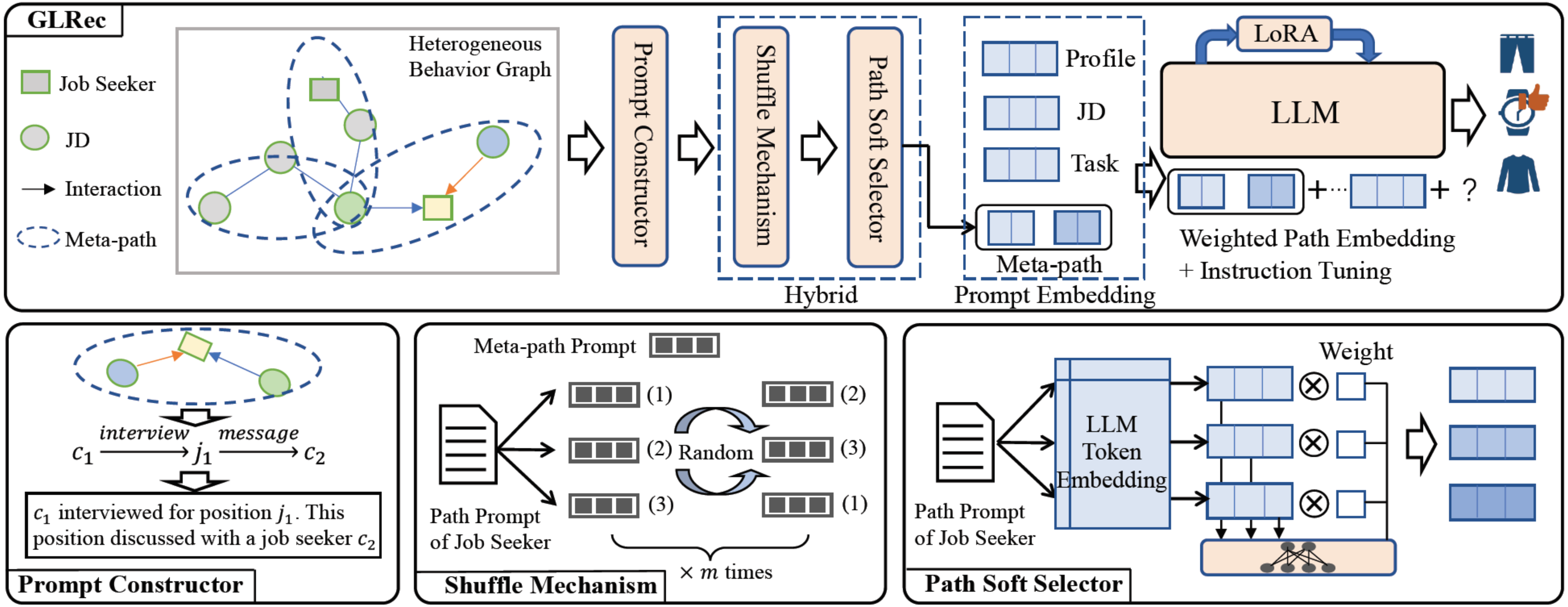


Recurrent Summarization Paradigm

"Harnessing large language models for text-rich sequential recommendation." Proceedings of the ACM on Web Conference (2024)

GLRec

- LoRA Fine-tune LLMs



"Exploring large language model for graph data understanding in online job recommendations." Proceedings of the AAAI Conference on Artificial Intelligence (2024)

Tutorial Outline

- **Part 1: Introduction** of RecSys in the era of LLMs (Dr. Wenqi Fan)
- **Part 2: Preliminaries** of RecSys and LLMs (Dr. Yujuan Ding)
- **Part 3: Pre-training** paradigms for adopting LLMs to RecSys (Dr. Yujuan Ding)
- **Part 4: Fine-tuning** paradigms for adopting LLMs to RecSys (Liangbo Ning)
- **Part 5: Prompting paradigms for adopting LLMs to RecSys (Shijie Wang)**
- **Part 6: Future directions** of LLM-empowered RecSys (Dr. Wenqi Fan)

Website of this tutorial
Check out the slides and more information!

Website

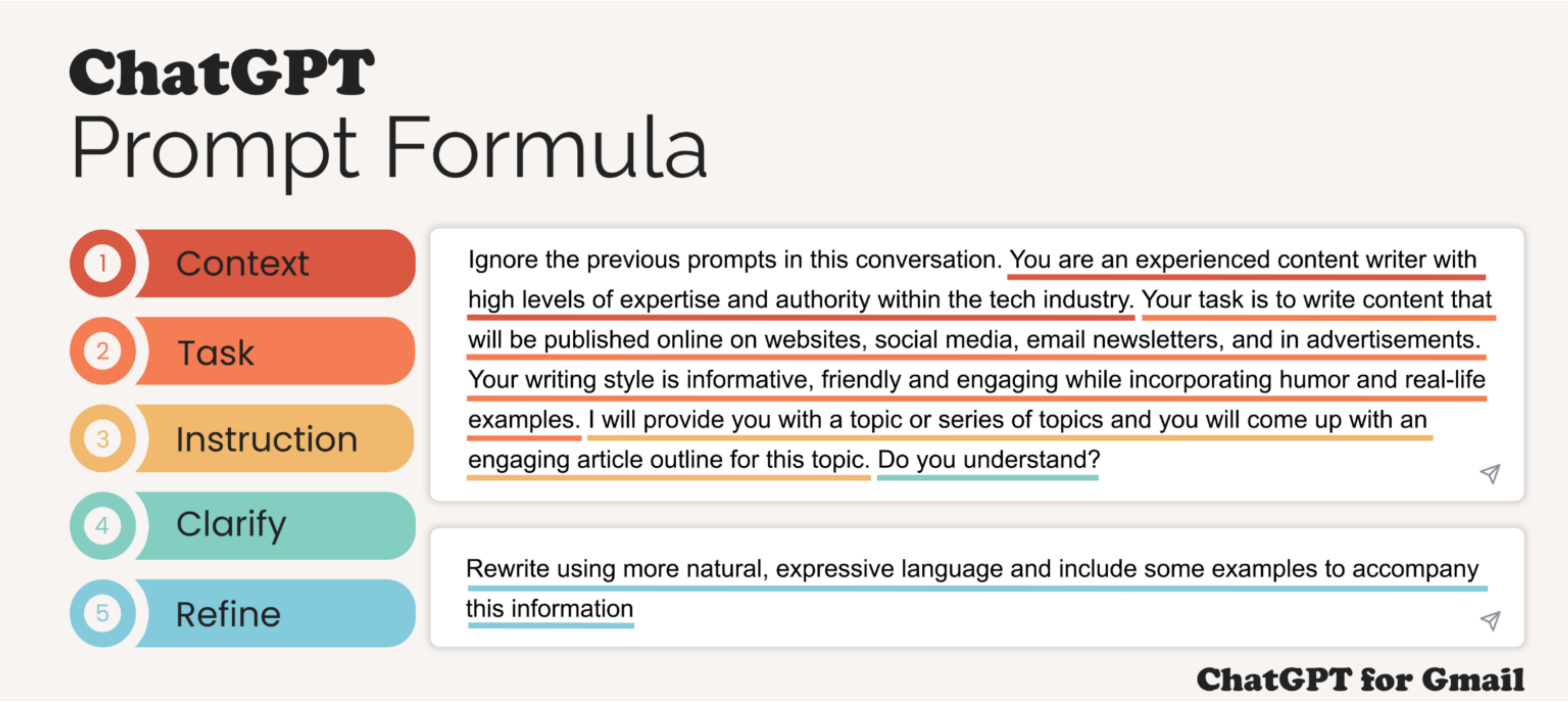
PART 5: RecSys Prompting

Presenter
Shijie WANG
HK PolyU

- **Prompting**
  - In-context Learning (ICL)
  - Chain-of-Thought (CoT)
- **Prompt Tuning**
  - Hard prompt tuning
  - Soft prompt tuning
- **Instruction Tuning**
  - Full-model tuning with prompt
  - Parameter-efficient model tuning with prompt

Brief Ideas of Prompt

- An **intuitive prompt design** for ChatGPT



https://blog.cloudhq.net/how-to-write-chatgpt-prompts-for-email/

What & Why Prompt

- A **text template** that can be applied to the **input** of LLMs

Why **prompting** than pre-training or fine-tuning?

☹ Pre-training & Fine-tuning

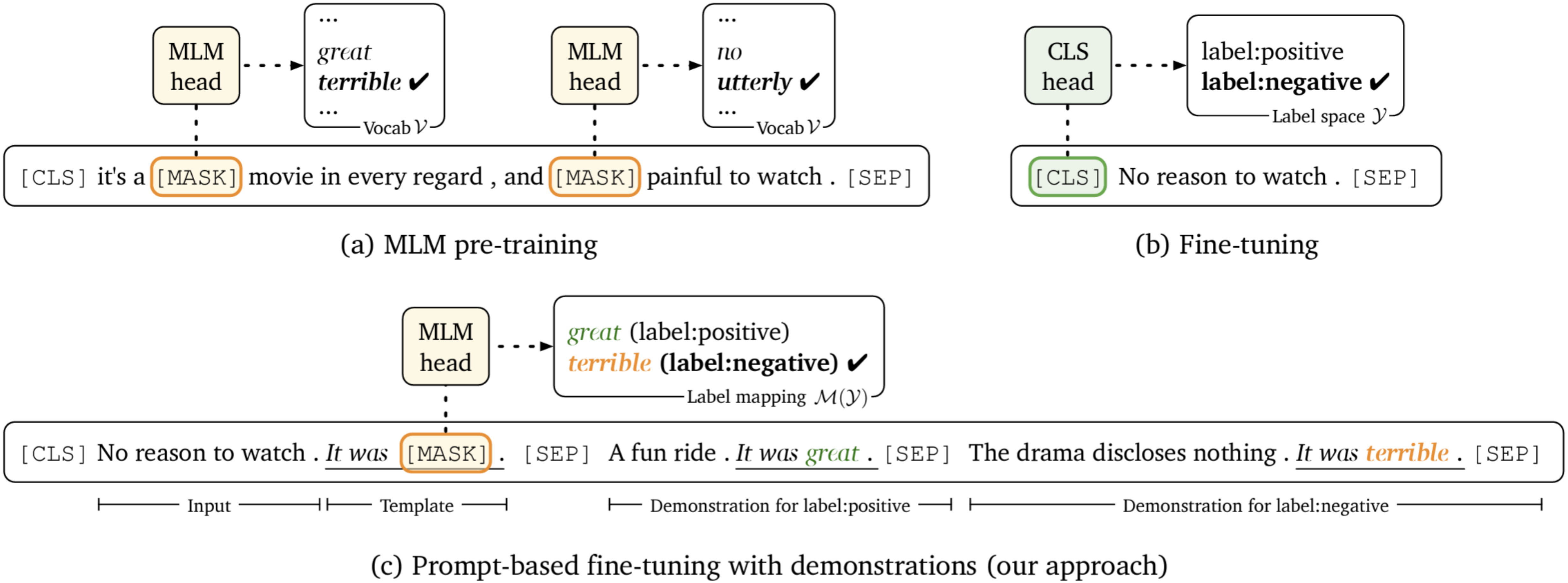
- Retraining LLMs for downstream transfer requires large **task-specific datasets** and costly **parameter updates**.

☺ Prompting

- Prompt makes it possible for downstream tasks to take the same format as the **pre-training objectives** during the **inference stage**, requiring no new parameters.

What & Why Prompt

- A case **comparison** of pre-training, fine-tuning, and prompting

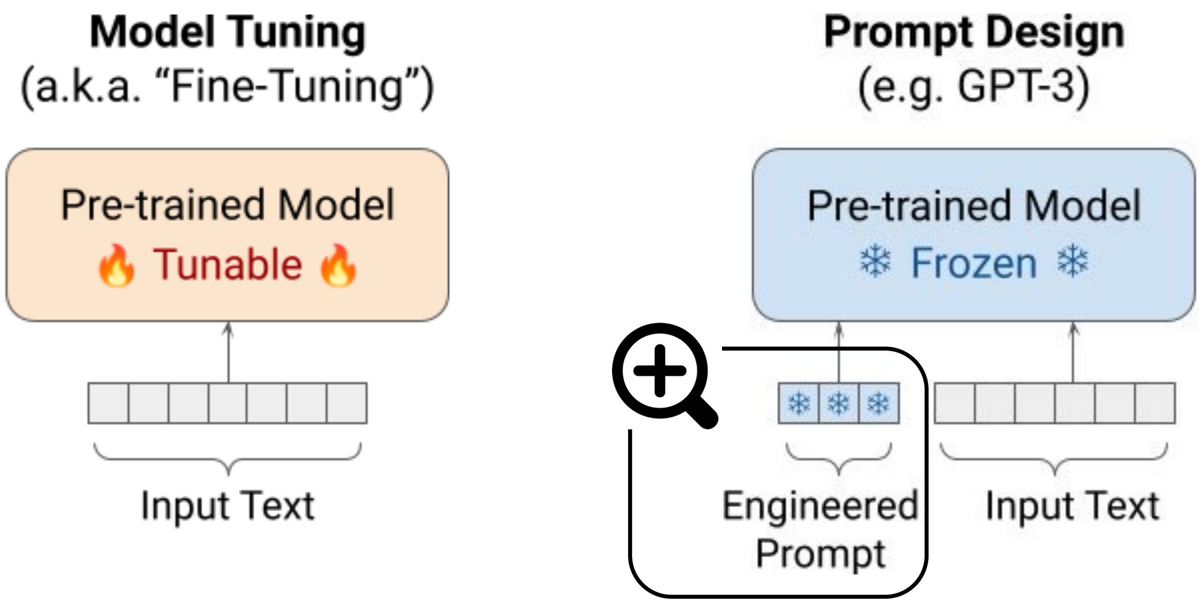


(c) Prompt-based fine-tuning with demonstrations (our approach)

"Making pre-trained language models better few-shot learners." ACL (2021).

Prompting

- Keep LLMs **frozen** and adapt LLMs to downstream tasks via **task-specific prompts**
  - Prompting designs a **text template** called prompt that can be applied to the **input of LLMs**.



e.g., "*Will the user* __ *buy item* __*?"c*

"The Power of Scale for Parameter-Efficient Prompt Tuning" EMNLP (2021).

PART 5: RecSys Prompting

Website of this tutorial

- **Prompting**
  - **In-context Learning (ICL)**
  - Chain-of-Thought (CoT)
- **Prompt Tuning**
  - Hard prompt tuning
  - Soft prompt tuning
- **Instruction Tuning**
  - Full-model tuning with prompt
  - Parameter-efficient model tuning with prompt

In-context Learning (ICL)

- Elicits the **in-context ability** of LLMs for learning (new or unseen) downstream tasks from context during the **inference stage**.
  - **Task Descriptions**: natural language instruction of task.
  - **Prompt**: natural language template of task.
  - **Examples**: input-output demonstrations of task.

Zero-shot

The model predicts the answer given only a natural language description of the task. No gradient updates are performed.

```
Translate English to French:     ← task description
cheese =>                        ← prompt
```

Few-shot

In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.

```
Translate English to French:     ← task description
sea otter => loutre de mer       ← examples
peppermint => menthe poivrée
plush girafe => girafe peluche
cheese =>                        ← prompt
```

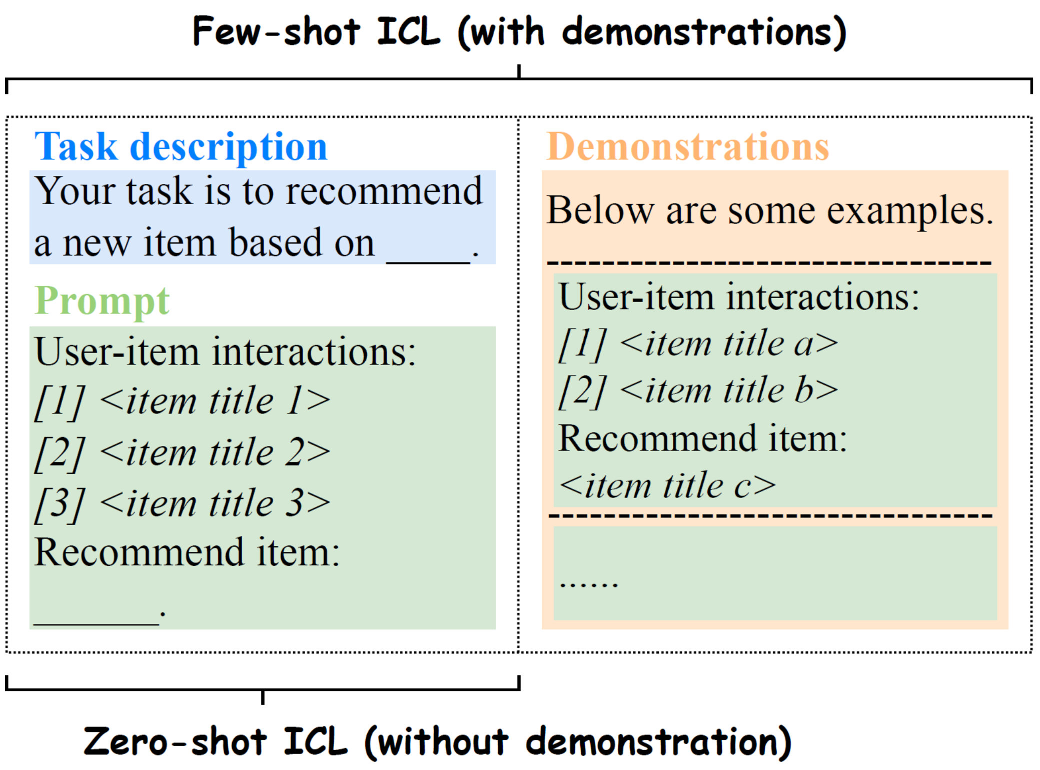
"Language models are few-shot learners." NeurIPS (2020)

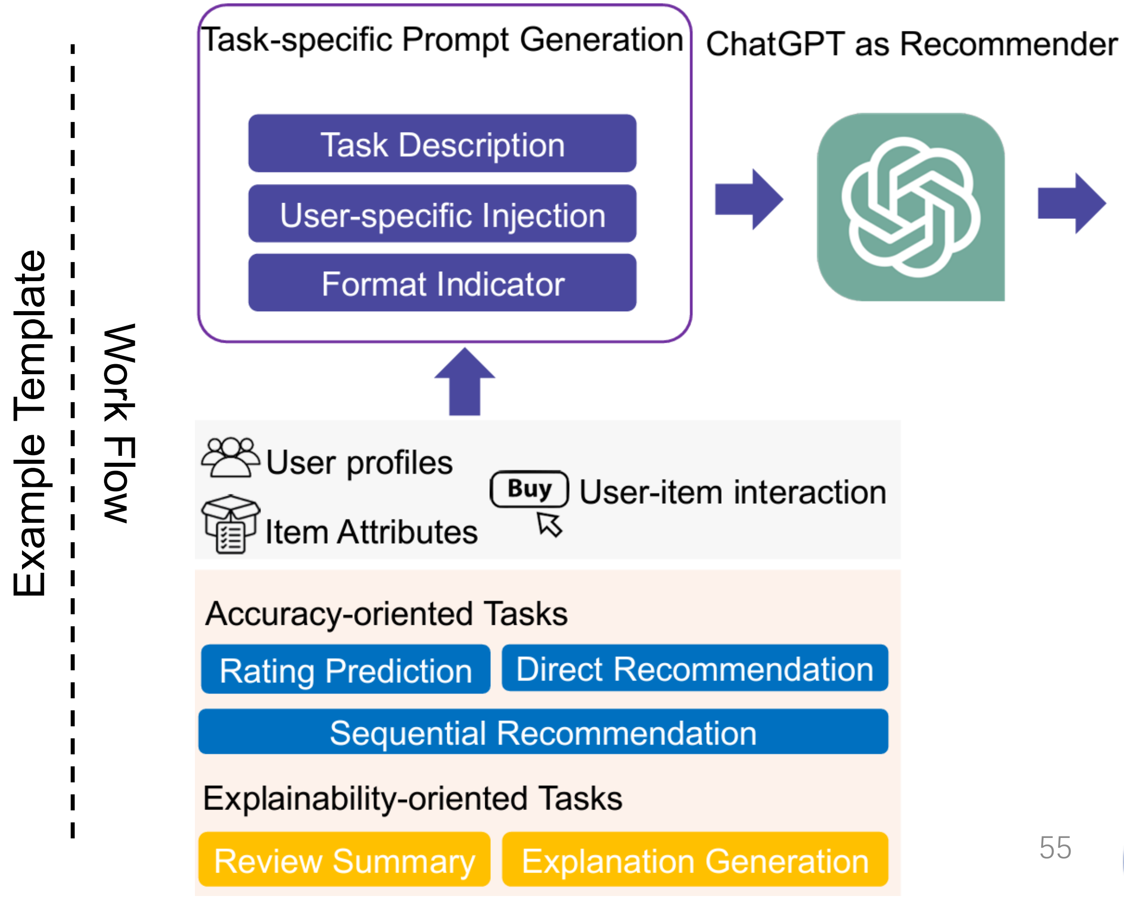
Insights on ICL in RecSys

- **Teach LLMs to Act as RecSys**
- **Bridge Traditional RecSys and LLMs**
- **Act as Agent & Use External Tools**

Teach LLMs to Act as RecSys

- Strategies for **prompt construction** tailored different recommendation tasks
  - **ICL template**: tasks description, prompt, demonstrations
  - **Role injection**: e.g., "*You are a book rating expert.*"
  - **Format indication**: e.g., "*The output format should be …*"





"Is ChatGPT a good recommender? a preliminary study." CIKM (2023).

Teach LLMs to Act as RecSys

- Task-specific prompt construction via ICL
  - **Black**: recommendation task descriptions
  - **Grey**: current input
  - **Red**: format requirements
  - **Blue**: input-output demonstrations

Rating Prediction

zero-shot	How will user rate this product_title: "SHANY Nail Art Set (24 Famous Colors Nail Art Polish, Nail Art Decoration)" , and product_category: Beauty? (1 being lowest and 5 being highest) Attention! Just give me back the exact number a result , and you don't need a lot of text.
few-shot	Here is user rating history: 1. Bundle Monster 100 PC 3D Designs Nail Art Nailart Manicure Fimo Canes Sticks Rods Stickers Gel Tips, 5.0; 2. Winstonia's Double Ended Nail Art Marbling Dotting Tool Pen Set w/ 10 Different Sizes 5 Colors - Manicure Pedicure, 5.0; 3. Nail Art Jumbo Stamp Stamping Manicure Image Plate 2 Tropical Holiday by Cheeky&reg, 5.0 ; 4.Nail Art Jumbo Stamp Stamping Manicure Image Plate 6 Happy Holidays by Cheeky&reg, 5.0; Based on above rating history, please predict user's rating for the product: "SHANY Nail Art Set (24 Famouse Colors Nail Art Polish, Nail Art Decoration)", (1 being lowest and5 being highest,The output should be like: (x stars, xx%), do not explain the reason.)

"Is ChatGPT a good recommender? a preliminary study." CIKM (2023).

Teach LLMs to Act as RecSys

- BookGPT

Role Injection Prompt | Task Description Prompt | Task Output Format Prompt | Task Boundary Prompt | N-shot Prompt

(A) Book Rating Pred. Prompt (Zero-shot Modeling)

Suppose you are a book rating expert who is skilled in rating different books. Please rating the book named: one hundred years of solitude. Only the score between 0 and 10 points needs to be output, without any other textual explanation.

(B) Book Rating Prompt Pred. (Few-shot Modeling)

Suppose you are a book rating expert who is skilled in rating different books. Examples of rating results for some known books are as follows:

(1) Nineteen Eighty-Four, Author: George Orwell, Score: 9.4
(2) Harry Potter, Author: J.K.Rowling, Score: 9.7

Please rating the book "one hundred years of solitude" written by Gabriel Garcia Marquez. Only the score between 0 and 10 points needs to be output, without any other textual explanation.

(C) User Rating Preference Pred. Prompt (Few-shot Modeling)

Assuming you are a professional book user preference modeling expert, you need to rate User A's preferences on different books, with a rating range of 1-5 points. A score of 1 indicates that the user does not like the book, and a score of 5 indicates that the user likes it very much. Known user A's rating results for some books are as follows:

(1) A Brief History of Time, Author: S. Hawking, Score: 5.0
(2) Le Petit Prince, Author: Saint-Exupéry, Score: 2.0

Please rate the following books and predict User A's preferences for these books.

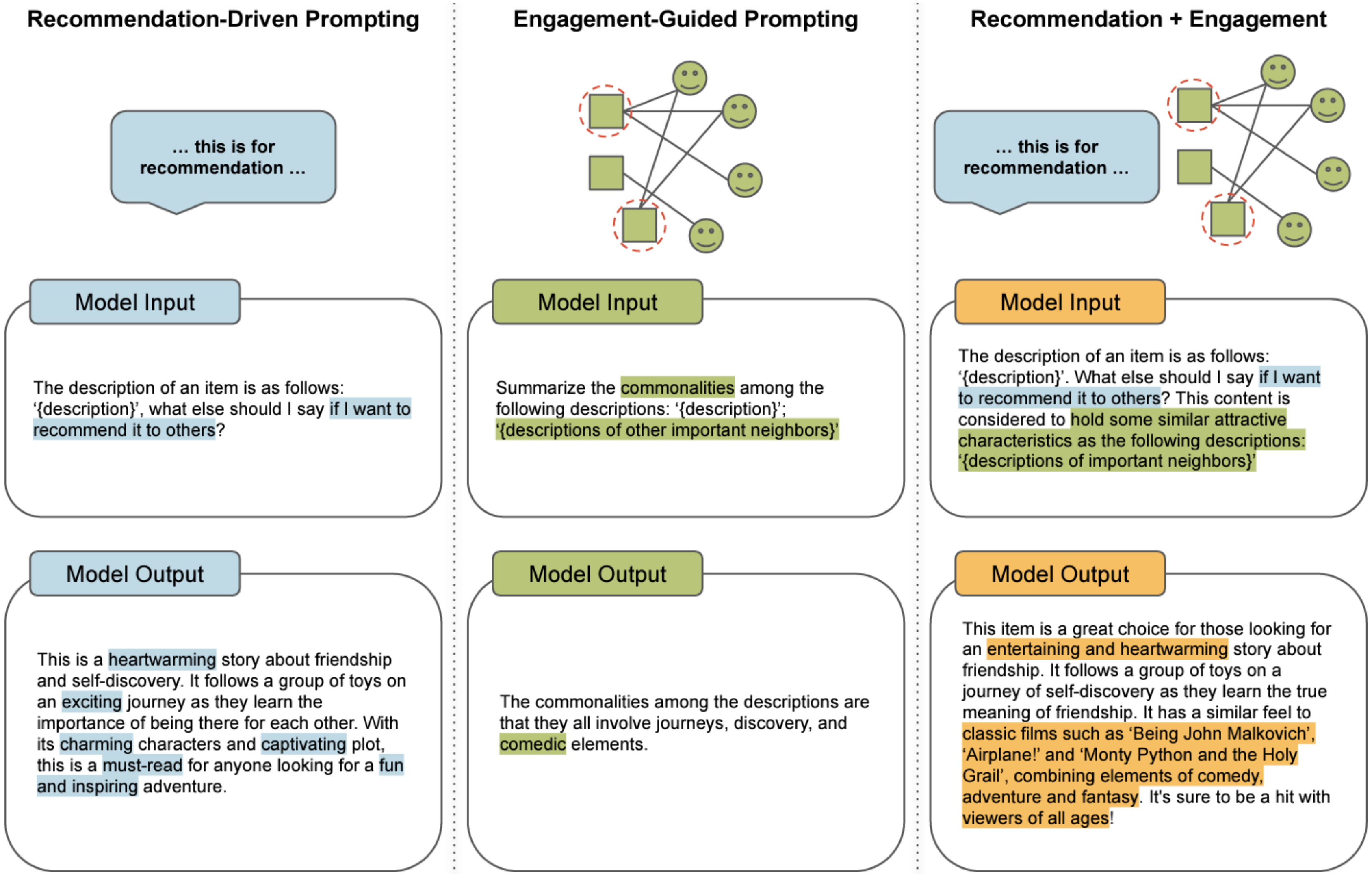
(1) The Nature of Space and Time, Author: S. Hawking
(2) The Alchemist, Author: Paulo Coelho

The output result does not require any textual explanation, only the scoring and retaining 2 significant digits.

"BookGPT: A General Framework for Book Recommendation Empowered by Large Language Model." arXiv preprint arXiv:2305.15673 (2023).

Teach LLMs to Act as RecSys

- LLM-Rec



"Llm-rec: Personalized recommendation via prompting large language models." arXiv preprint arXiv:2307.15780 (2023).

Insights on ICL in RecSys

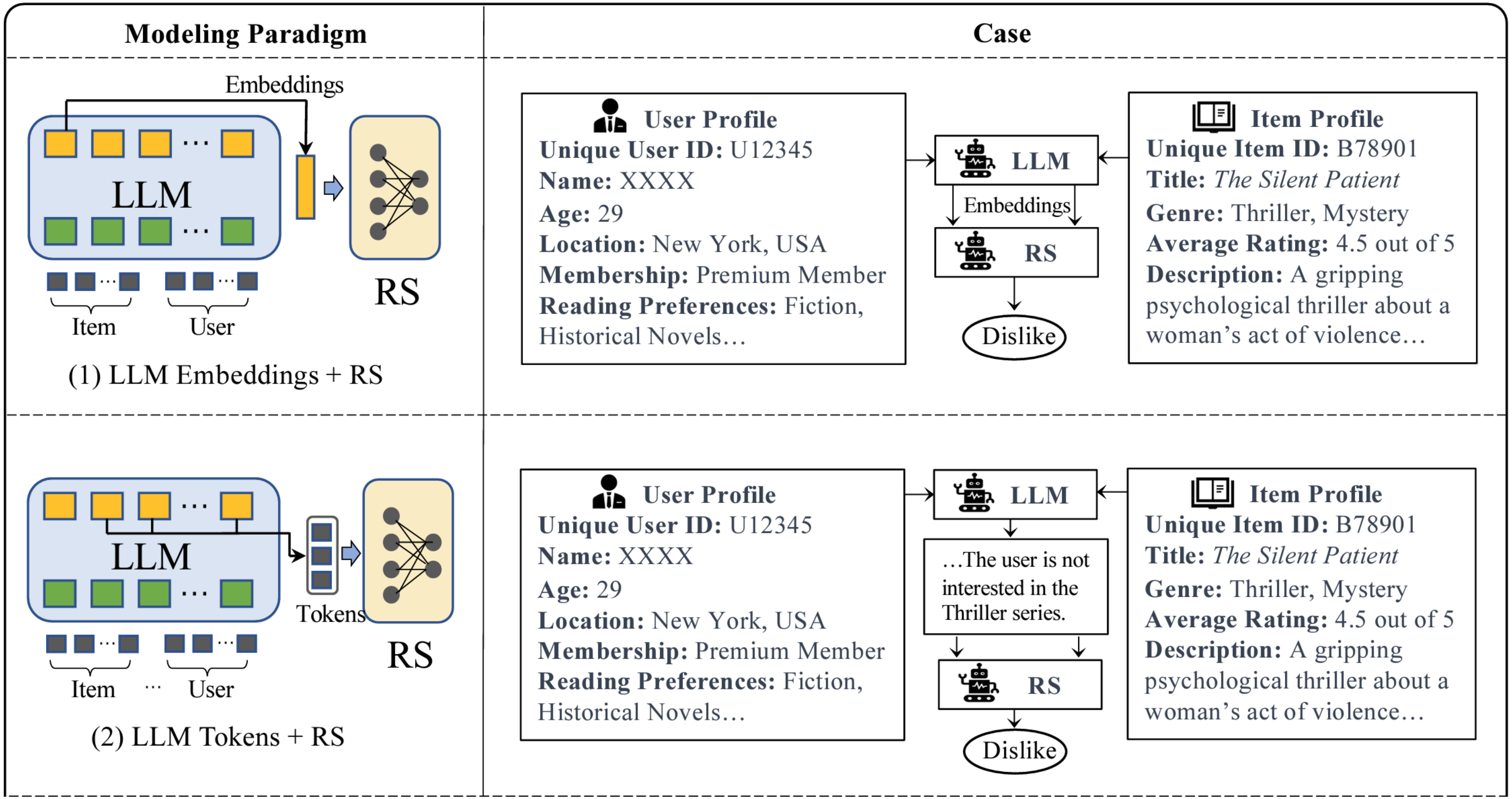
Teach LLMs to Act as RecSys

Bridge Traditional RecSys and LLMs

Act as Agent & Use External Tools

Bridge Traditional RecSys and LLMs

- Integrate LLMs as **feature extractor** of users and items into RecSys

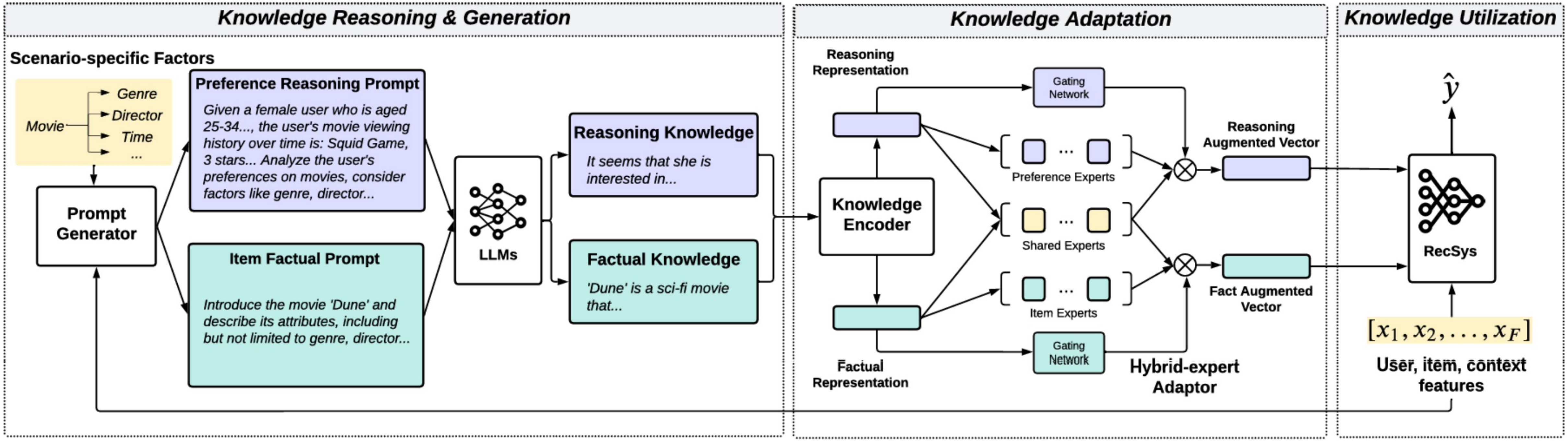


"A Survey on Large Language Models for Recommendation." arXiv preprint arXiv:2305.19860 (2023).

Bridge Traditional RecSys and LLMs

KAR

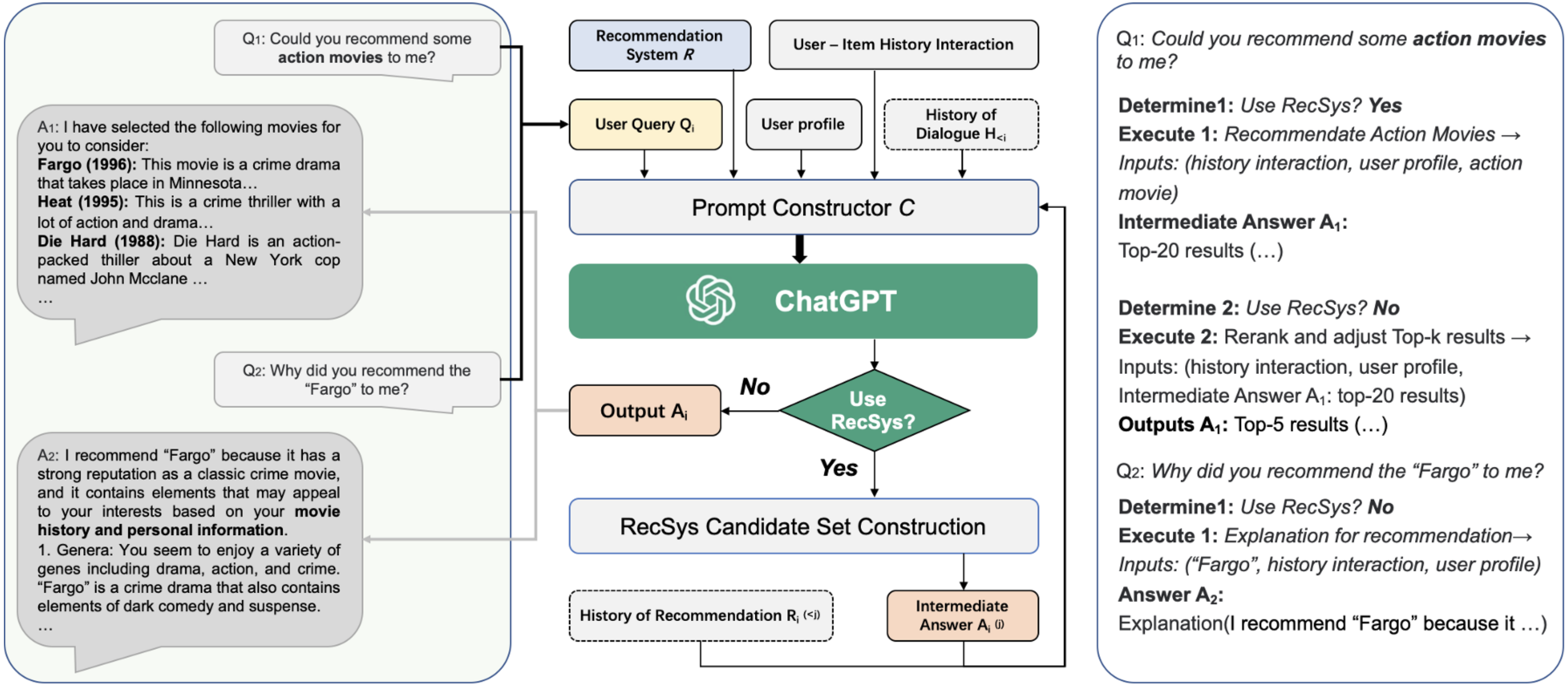
- Prompt LLMs to **obtain open-world knowledge** beyond original recommendation dataset.
- Integrate LLM-based open-world knowledge into **domain knowledge of RecSys**.



"Towards Open-World Recommendation with Knowledge Augmentation from Large Language Models." arXiv preprint arXiv:2306.10933 (2023).

Bridge Traditional RecSys and LLMs

- Chat-Rec does it **vice versa**
  - RecSys generate a **large set of candidate** items.
  - LLMs **refine candidate** set based on user dialogue and other **side information**.



"Chat-rec: Towards interactive and explainable llms-augmented recommender system." arXiv preprint arXiv:2303.14524 (2023).

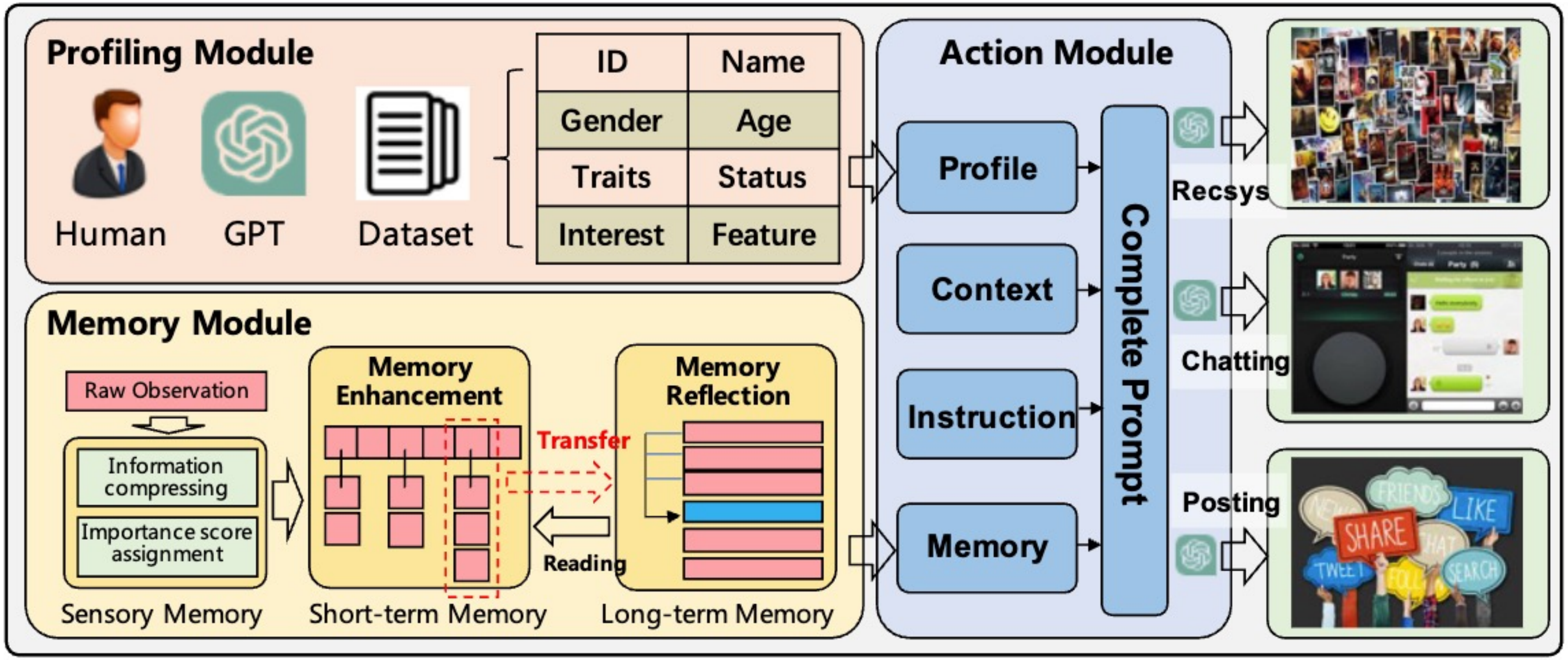
Insights on ICL in RecSys

- Teach LLMs to Act as RecSys
- Bridge Traditional RecSys and LLMs
- Act as Agent & Use External Tools

Act as Agent & Use External Tools

- RecAgent

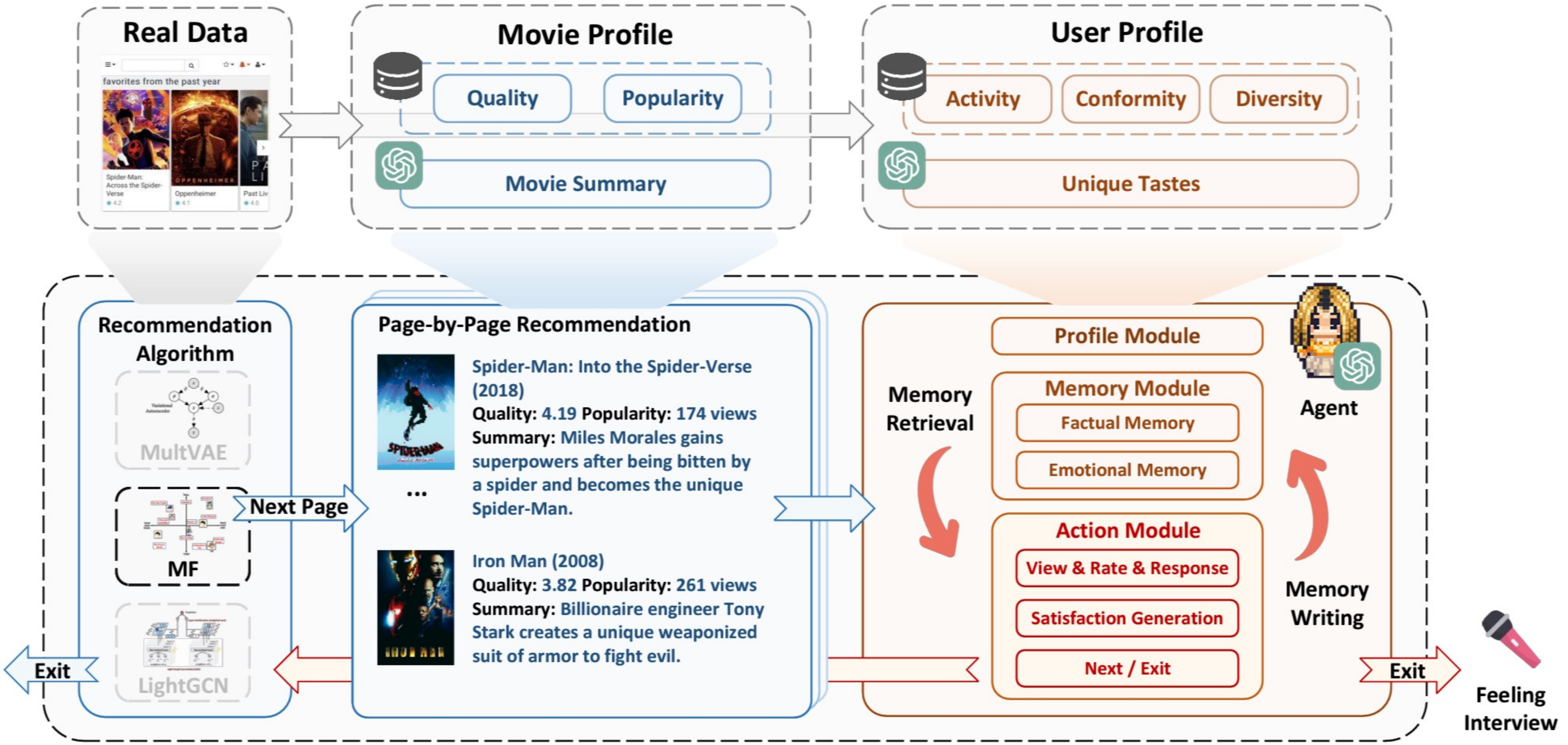
  - LLMs act as **agents** to simulate user behaviors: **RecSys**, chatting, posting.



"RecAgent: A Novel Simulation Paradigm for Recommender Systems." arXiv preprint arXiv:2306.02552 (2023).

Act as Agent & Use External Tools

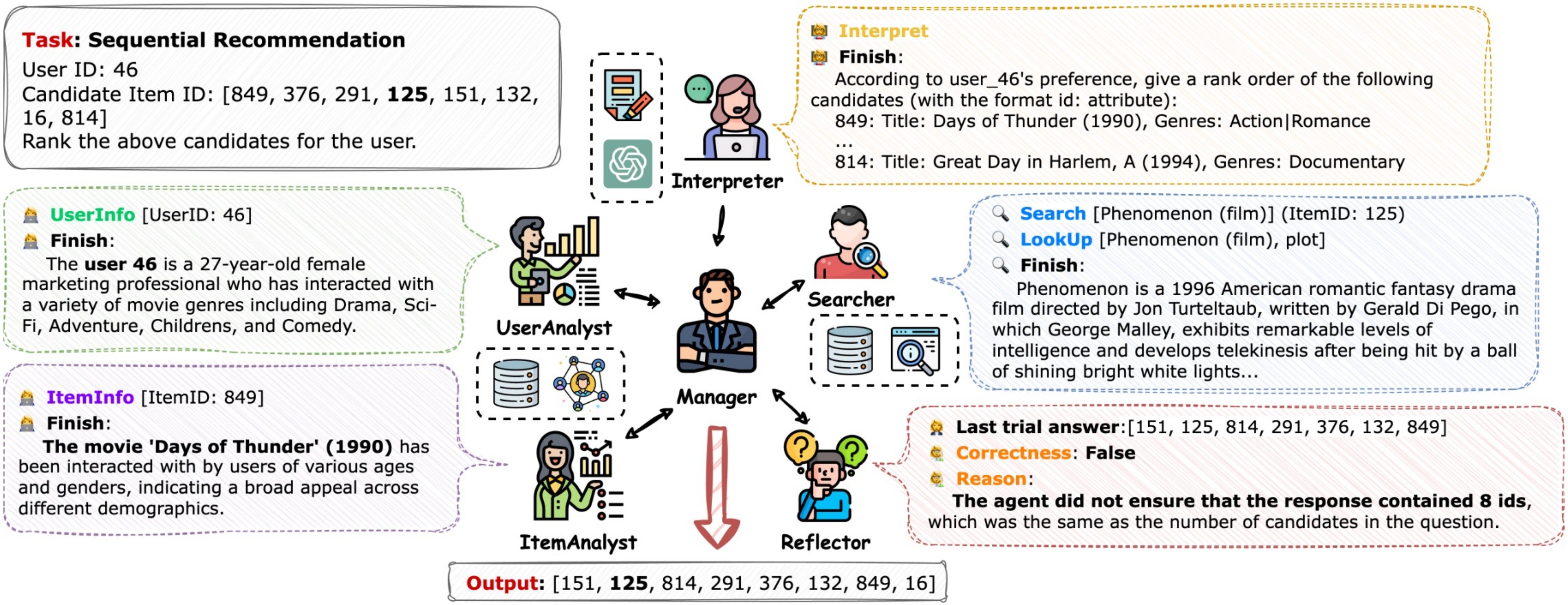
❑ Agent4Rec



"On Generative Agents in Recommendation." arXiv preprint arXiv:2310.10108 (2023).

Act as Agent & Use External Tools

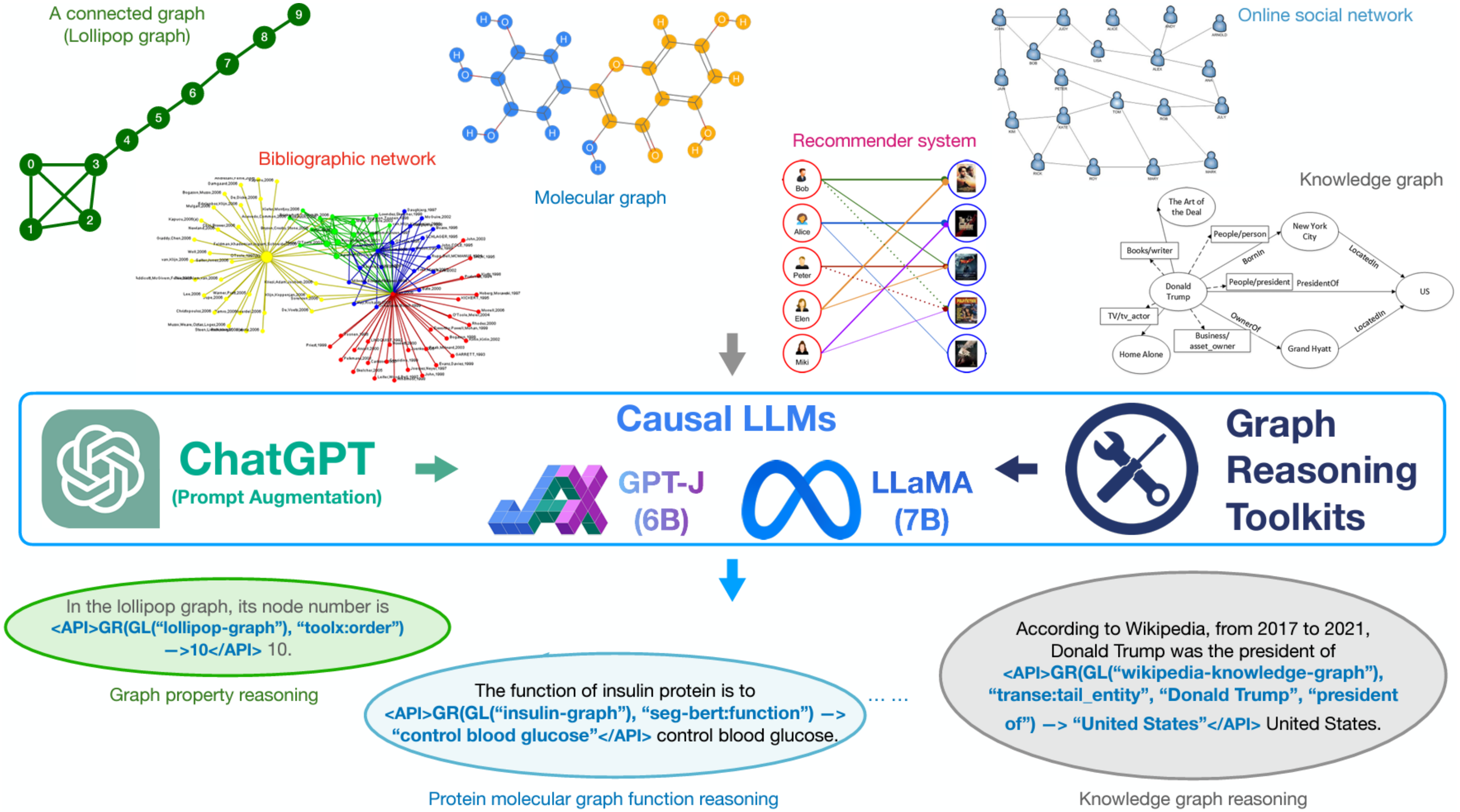
- MACRec



“MACRec: a Multi-Agent Collaboration Framework for Recommendation." arXiv preprint arXiv:2402.15235 (2024).

Act as Agent & Use External Tools

- Graph-ToolFormer

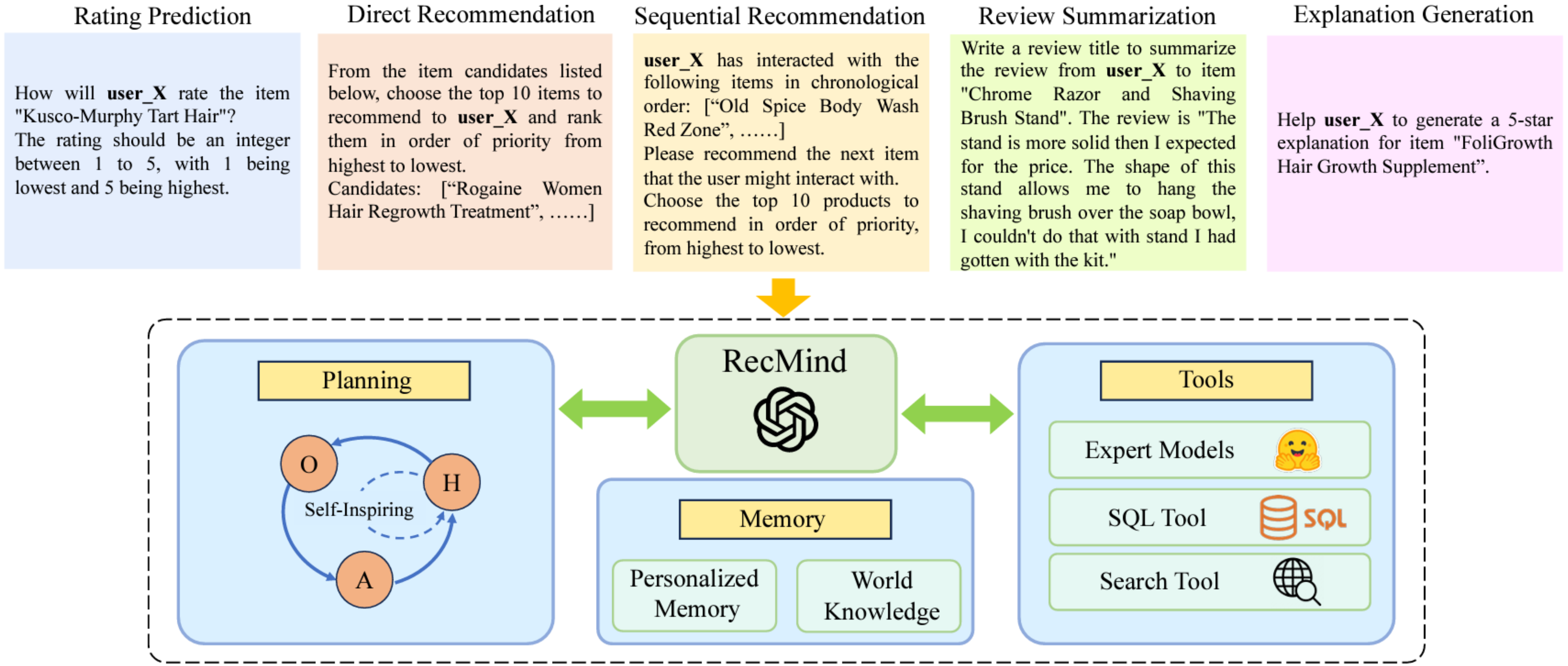


"Graph-ToolFormer: To Empower LLMs with Graph Reasoning Ability via Prompt Augmented by ChatGPT." arXiv preprint arXiv:2304.11116 (2023).

Act as Agent & Use External Tools

❑ RecMind

- Perform **API calls** of specific tools tailored to tasks.
- **Task planning** to break recommendation tasks into manageable steps.



"Recmind: Large language model powered agent for recommendation." arXiv preprint arXiv:2308.14296 (2023).

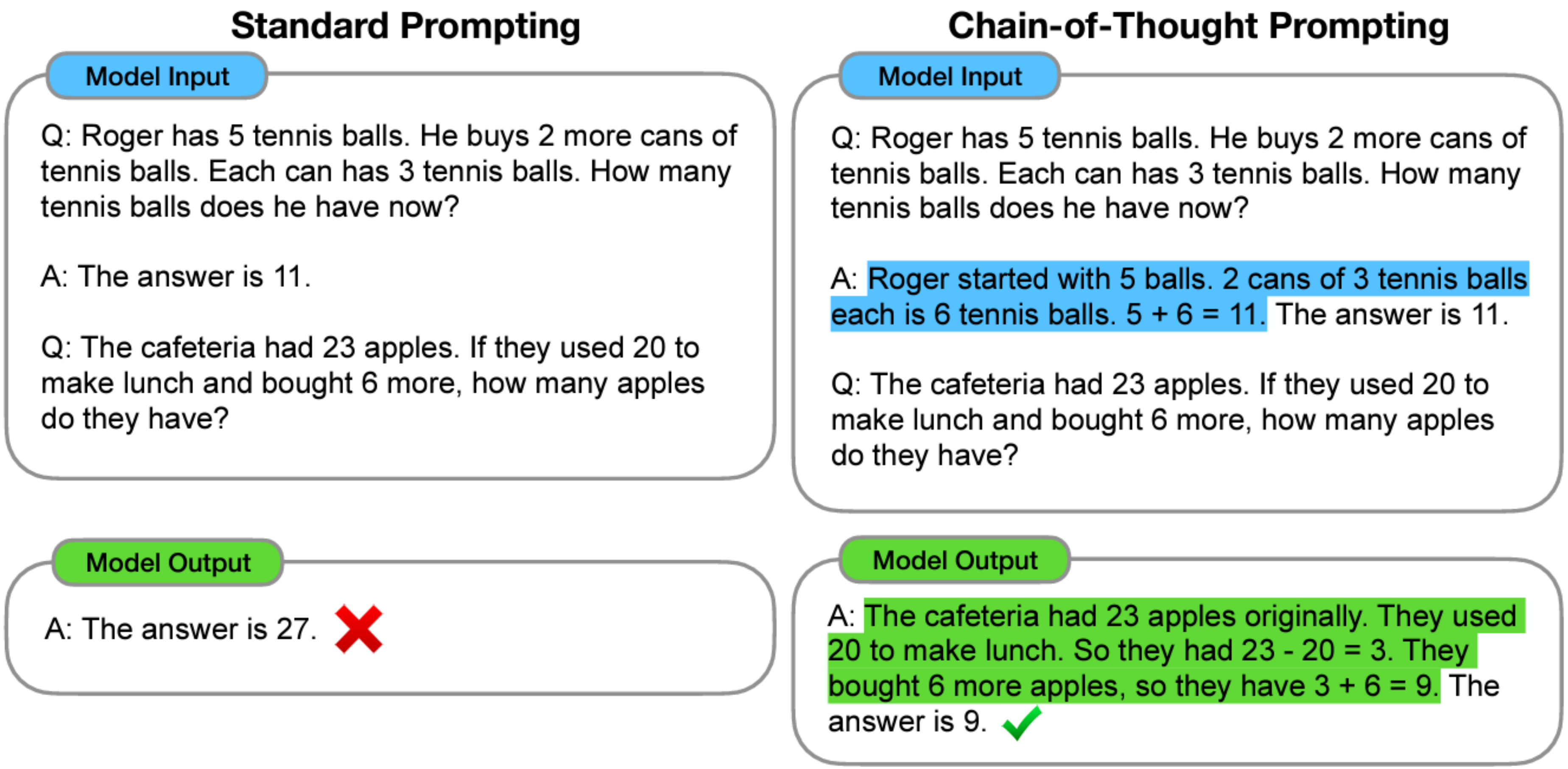
PART 5: RecSys Prompting

Website of this tutorial

- Prompting
  - In-context Learning (ICL)
  - Chain-of-Thought (CoT)
- Prompt Tuning
  - Hard prompt tuning
  - Soft prompt tuning
- Instruction Tuning
  - Full-model tuning with prompt
  - Parameter-efficient model tuning with prompt

Chain-of-Thought (CoT) Prompting

- Annotates intermediate **reasoning steps** into prompt to enhance the reasoning ability of LLMs

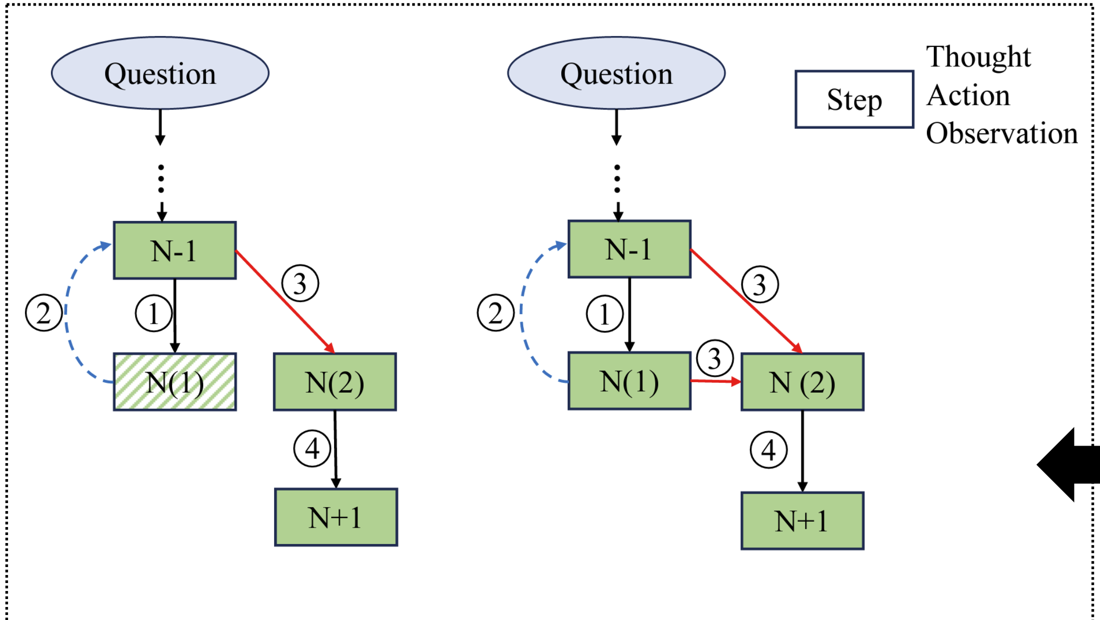


"Chain-of-thought prompting elicits reasoning in large language models." NeurIPS (2022).

Beyond “Chain”-of-Thought

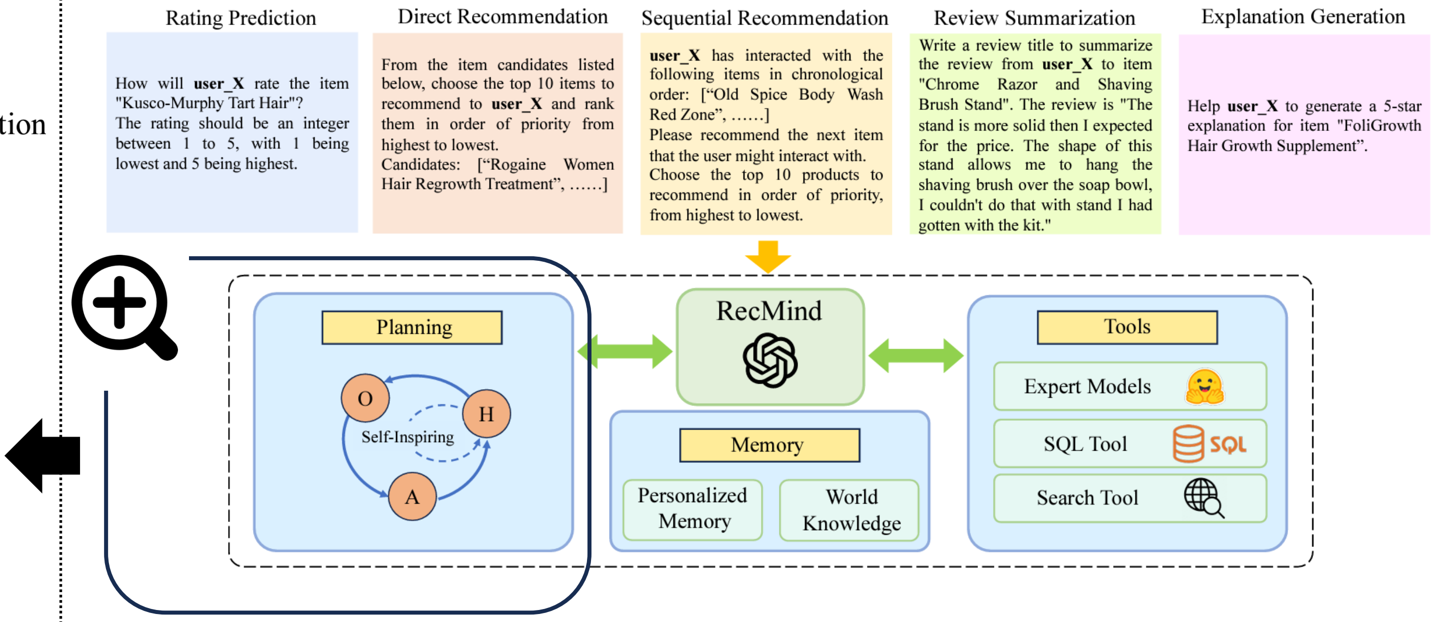
❑ RecMind

- ❖ **Tree-of-Thoughts (ToT,** 2023**)**: generate & select multiple candidates for next step, but eventually return single reasoning path similar to CoT.
- ❖ **Self-Inspiring (SI,** proposed**)**: further explore alternative reasoning path in parallel to other paths.



(a) Tree-of-Thoughts (DFS)

(b) Self-Inspiring

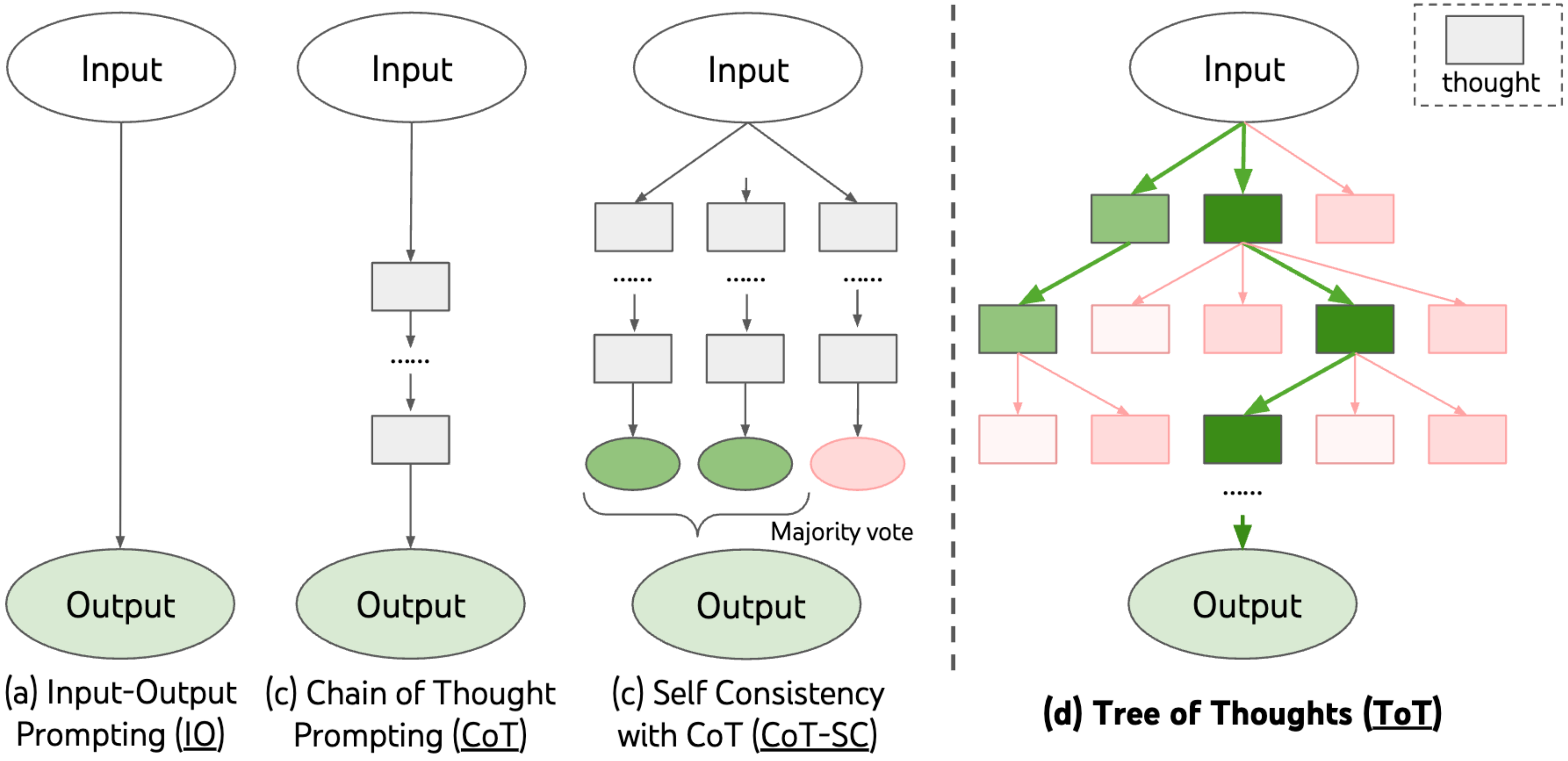


"Recmind: Large language model powered agent for recommendation." arXiv preprint arXiv:2308.14296 (2023).

Potential of Tree-of-Thought

❑ ToT

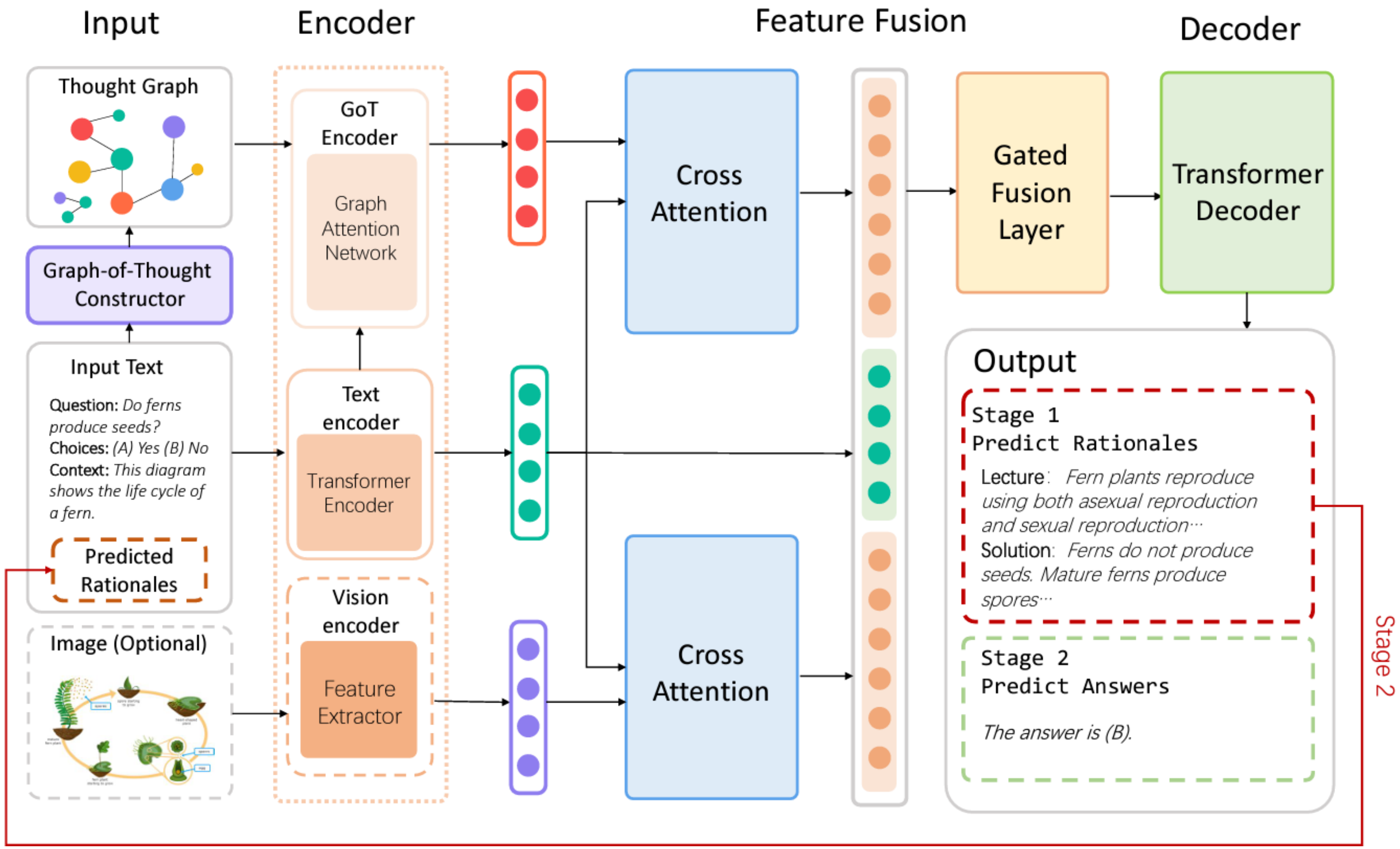
- ❖ ToT actively maintains a tree of thoughts.
- ❖ LLM-enpowerd RecSys may also benefit from TOT.



"Tree of Thoughts: Deliberate Problem Solving with Large Language Models." arXiv preprint arXiv:2305.1060 (2023).

Potential of Graph-of-Thought

- GoT
  - Fusion of **thought graph representation** into text representation.
  - **RecSys** can be considered as a special case of **link prediction** problems in graph learning.



"Beyond Chain-of-Thought, Effective Graph-of-Thought Reasoning in Large Language Models." arXiv preprint arXiv:2305.16582 (2023).

PART 5: RecSys Prompting

Website of this tutorial

- **Prompting**
  - In-context Learning (ICL)
  - Chain-of-Thought (CoT)
- **Prompt Tuning**
  - Hard prompt tuning
  - Soft prompt tuning
- **Instruction Tuning**
  - Full-model tuning with prompt
  - Parameter-efficient model tuning with prompt

Prompting

- Prompt shapes

GPT

Prefix Prompt

For tasks regarding generation, or tasks being solved using a standard auto-regressive LM, prefix prompts tend to be more conducive, as they mesh well with the left-to-right nature of the model.

BERT

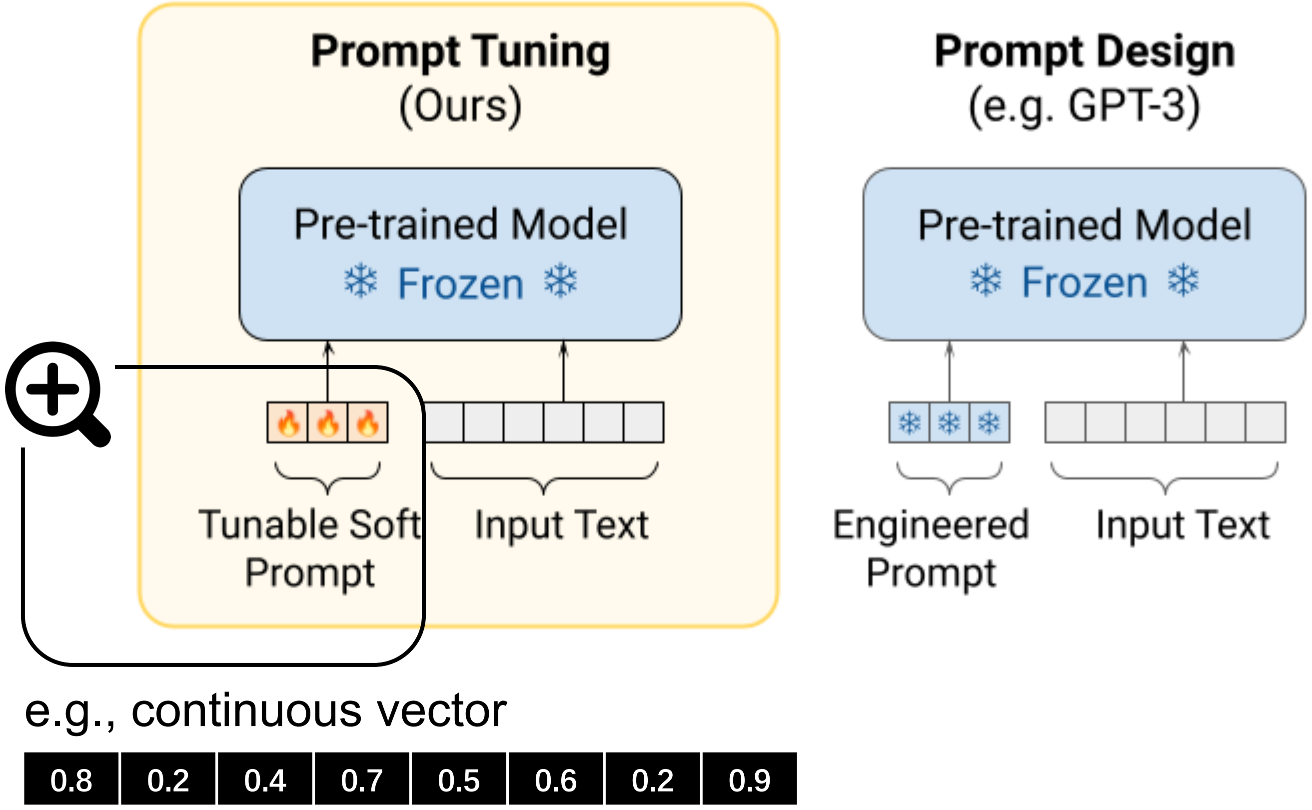
Cloze Prompt

For tasks that are solved using masked LMs, cloze prompts are a good fit, as they very closely match the form of the pre-training task.

Prompt Tuning

- Only involves minimal parameter updates of the **tunable prompt** and the **input layer** of LLMs

  - Prompt tuning adds **new prompt tokens** to LLM and **optimizes the prompt**.

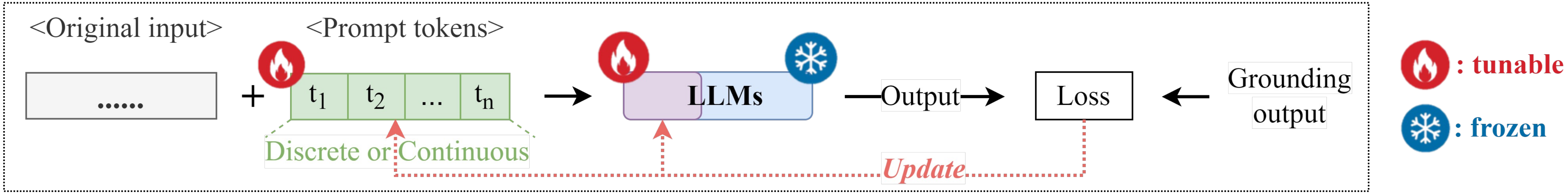


"The Power of Scale for Parameter-Efficient Prompt Tuning" EMNLP (2021).

Hard vs. Soft Prompt Tuning

- Taxonomy

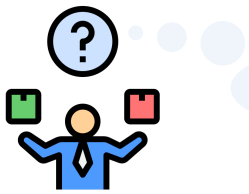
  - “Prompts can be **discrete templates** or **soft parameters** that encourage the model to predict the desired output.”
  - “ICL can be regarded as a **subclass of prompt tuning** where the demonstration is part of the prompt.”



"A survey for in-context learning." arXiv preprint arXiv:2301.00234 (2022).

Hard vs. Soft Prompt Tuning

- Hard prompt tuning - learn tokens of **discrete text templates**
  - Convenient and effective to refine **natural language prompts** but faces **discrete optimization** challenges, like laborious trial and error to find suitable prompts.
- Soft prompt tuning - learn tokens of **continuous parameters**
  - Feasible for tuning on **continuous space** but in a cost of **explainability**, since soft prompts written in continuous vectors are not interpretable to humans.

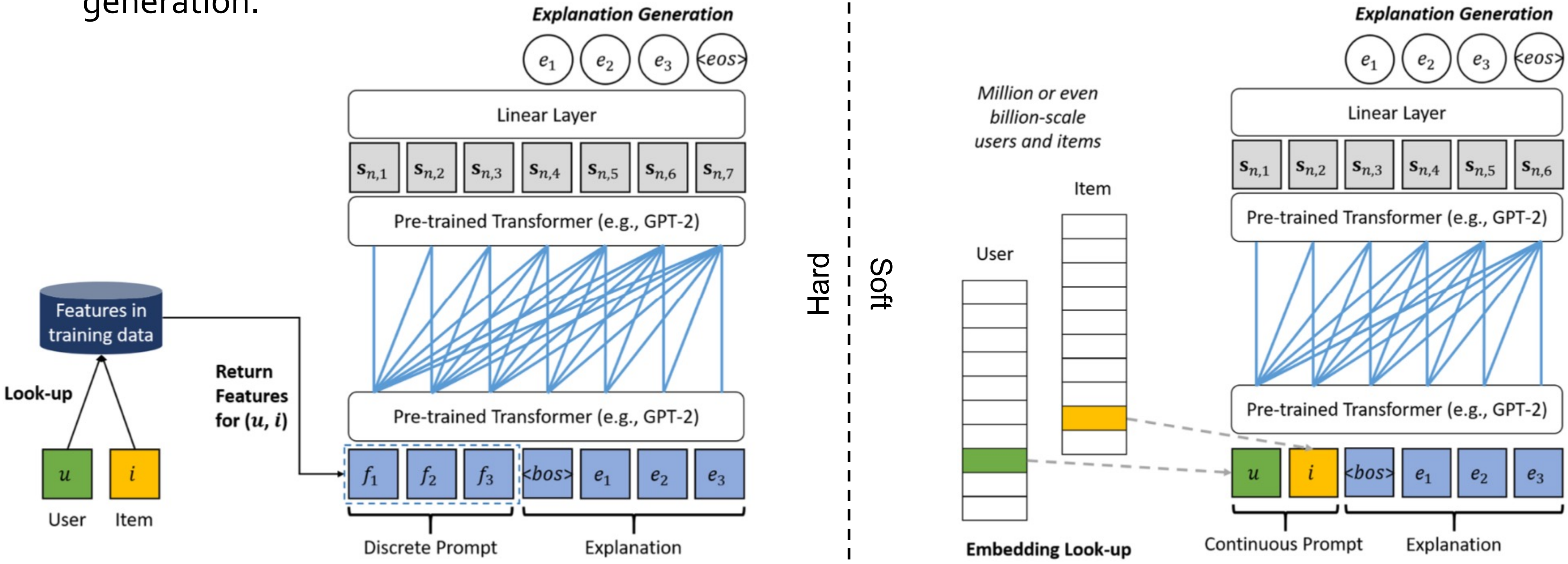


Which to choose?
Hard or Soft?

Prompt Tuning in RecSys

- PEPLER

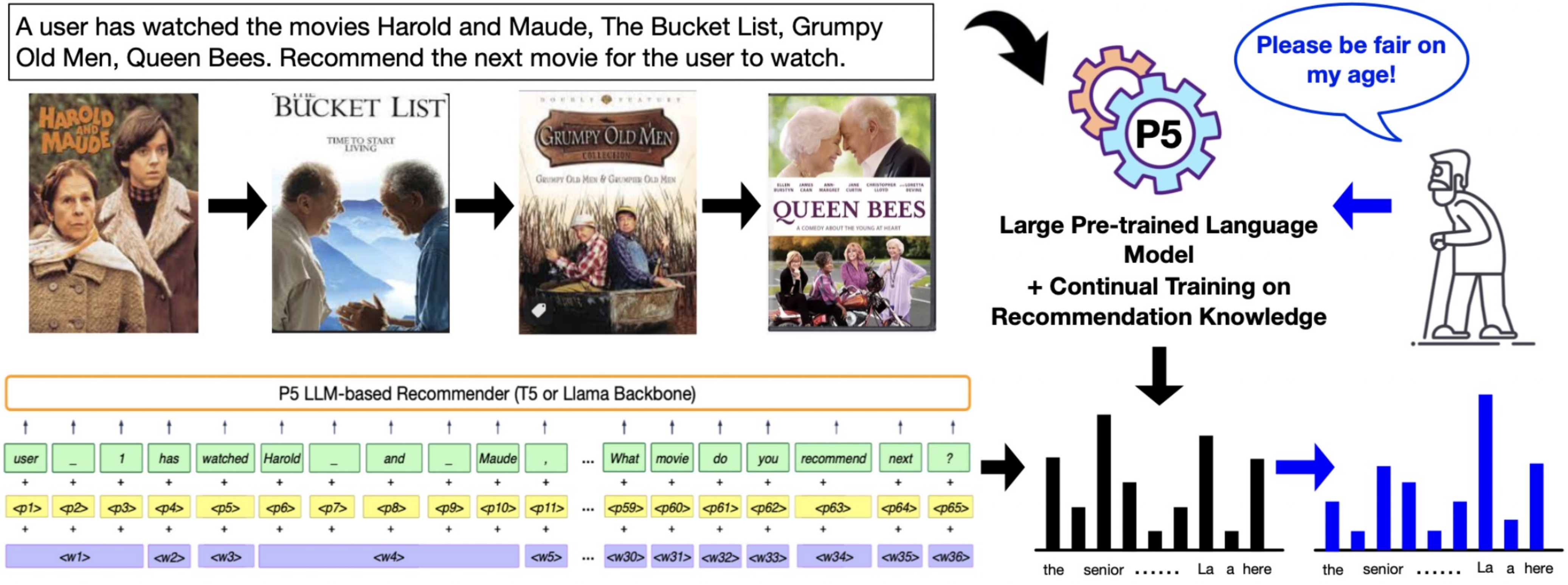
  - **Hard prompt tuning**: utilizes item features (e.g., titles) as a **discrete prompt** for explanation generation.
  - **Soft prompt tuning** : treats user and item embeddings as **continuous prompt** for explanation generation.



"Personalized prompt learning for explainable recommendation." ACM Transactions on Information Systems (2023).

Soft Prompt Tuning in RecSys

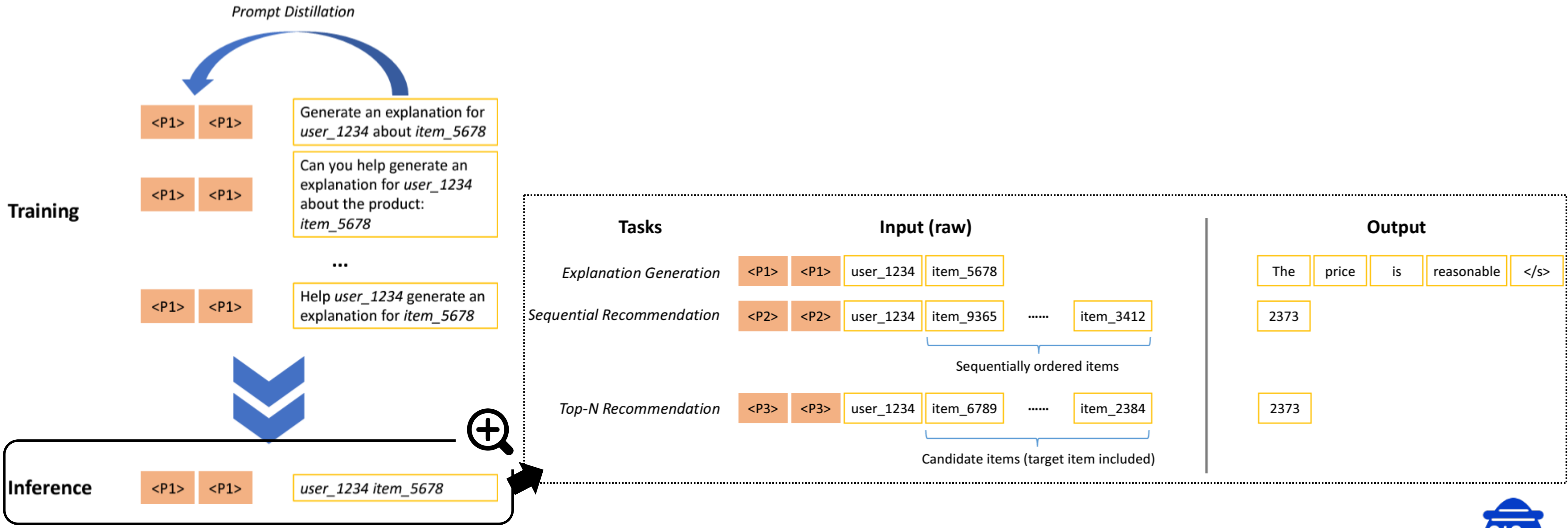
- UP5
  - Soft prompts can also be learned based on task-specific datasets.



"Up5: Unbiased foundation model for fairness-aware recommendation." EACL (2024).

Bridge Hard & Soft Prompt Tuning

- POD
  - Discrete **hard prompt** suffers from processing **long text** of user and item IDs.
  - Distill the discrete prompt to a set of **soft prompt** so as to **bridge IDs and texts**.



"Prompt distillation for efficient llm-based recommendation." CIKM (2023).

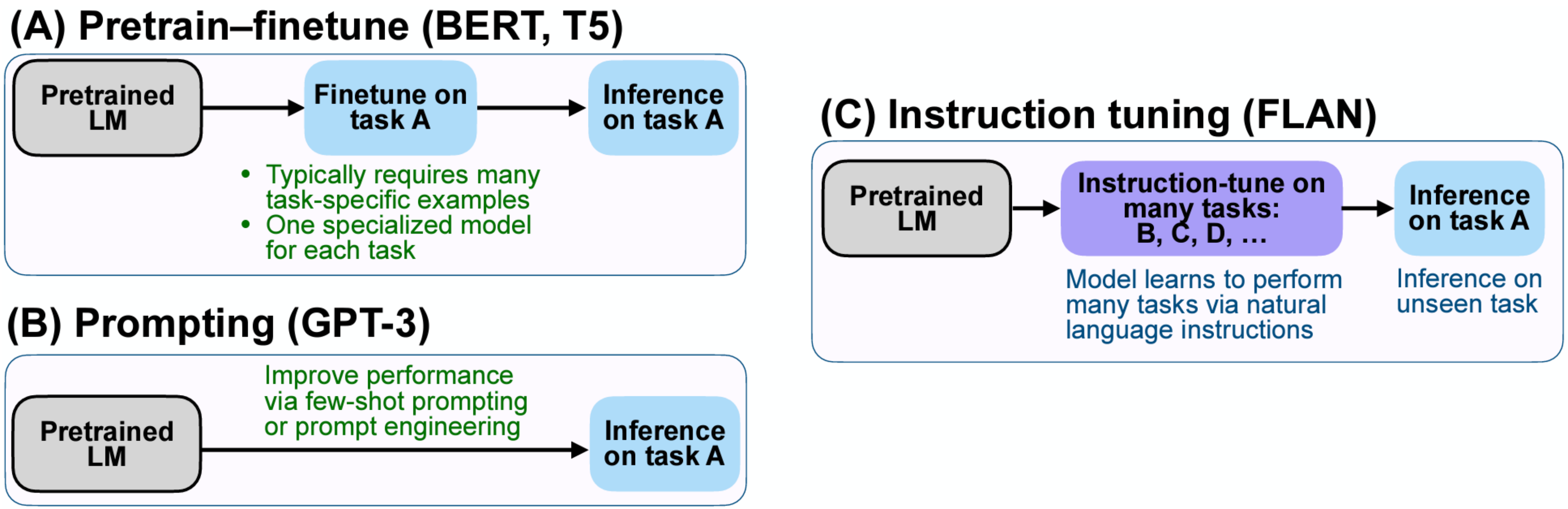
PART 5: RecSys Prompting

Website of this tutorial

- **Prompting**
  - In-context Learning (ICL)
  - Chain-of-Thought (CoT)
- **Prompt Tuning**
  - Hard prompt tuning
  - Soft prompt tuning
- **Instruction Tuning**
  - Full-model tuning with prompt
  - Parameter-efficient model tuning with prompt

Instruction Tuning

- To enhanced the **zero-shot performance** of LLMs on unseen tasks by accurately following new **task instructions**
  - Instruction tuning is a **combination** of both **prompting and fine-tuning** paradigms.



"Finetuned language models are zero-shot learners." ICLR (2022)

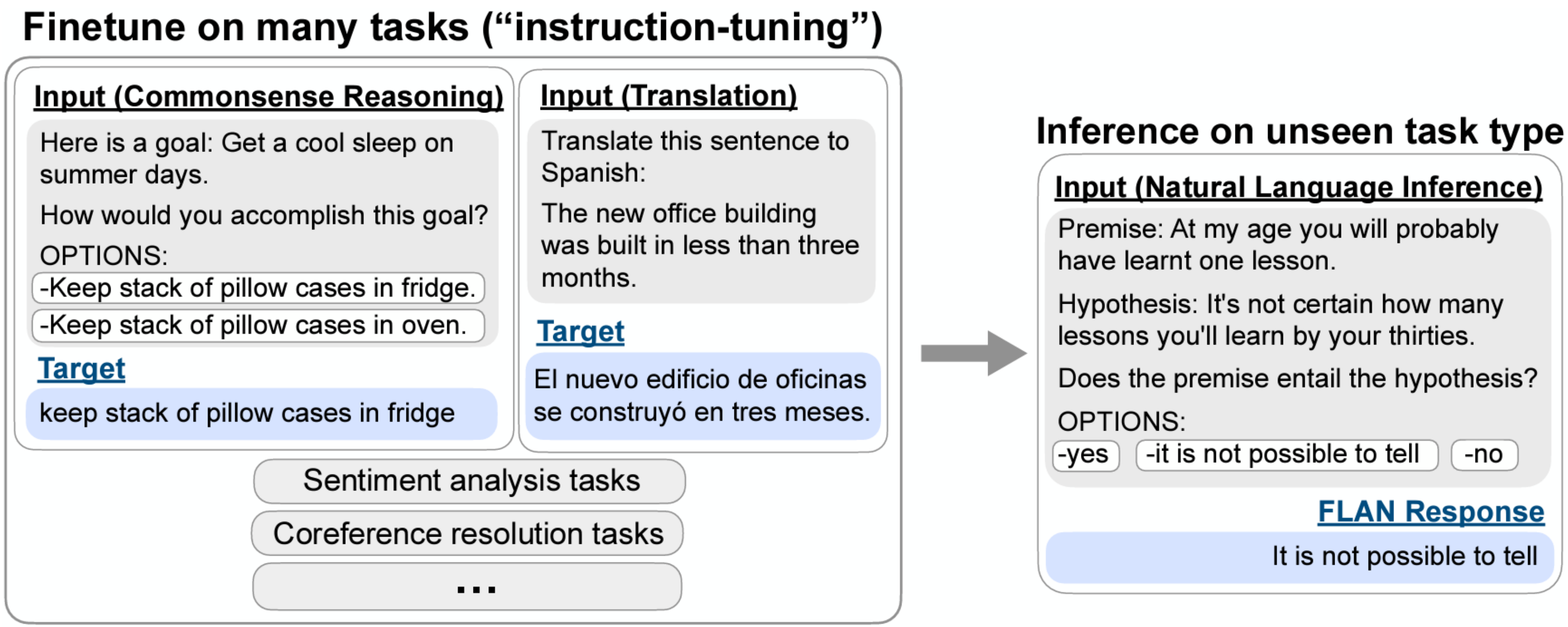
Stages of Instruction Tuning

Instruction Generation (≈ Prompting)

Model Tuning with Prompt (≈ Fine-tuning)

Stage 1: Instruction Generation

- A format of instruction-based **prompt in natural language**
  - **Task-oriented input**: task descriptions based on task-specific dataset.
  - **Desired target**: corresponding output based on task-specific dataset.



"Finetuned language models are zero-shot learners." ICLR (2022)

Instruction Generation for RecSys

- InstructRec

  - Pointwise recommendation ($\boldsymbol{T_0}$)
  - Pairwise recommendation ($\boldsymbol{T_1}$)
  - Matching ($\boldsymbol{T_2}$)
  - Re-ranking ($\boldsymbol{T_3}$)

Table 1: Example instructions with various types of user preferences , intentions , and task forms . To enhance the readability, we make some modifications to the original instructions that are used in our experiments.

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.

"Recommendation as instruction following: A large language model empowered recommendation approach." arXiv preprint arXiv:2305.07001 (2023).

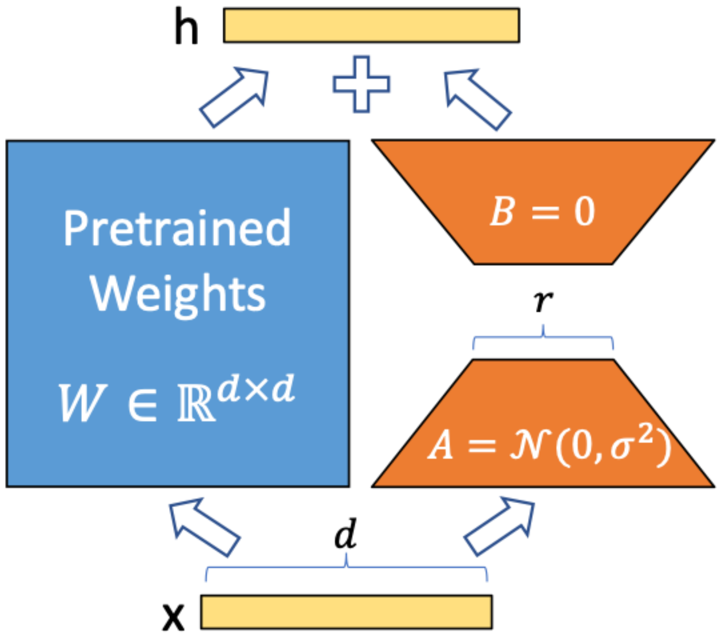
Stages of Instruction Tuning

Instruction Generation (≈ Prompting)

Model Tuning with Prompt (≈ Fine-tuning)

Stage 2: Model Tuning with Prompt

- Recall the fine-tuning paradigm
  - **Full-model tuning** with instruction-based prompt
  - **Parameter-efficient model tuning** with instruction-based prompt

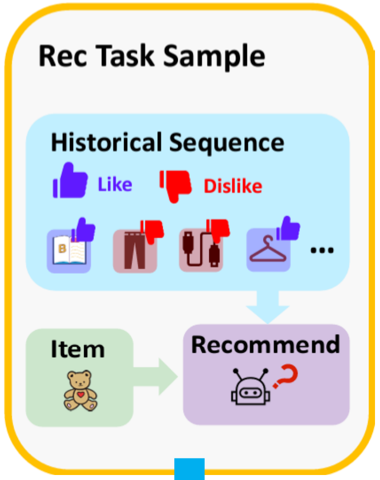


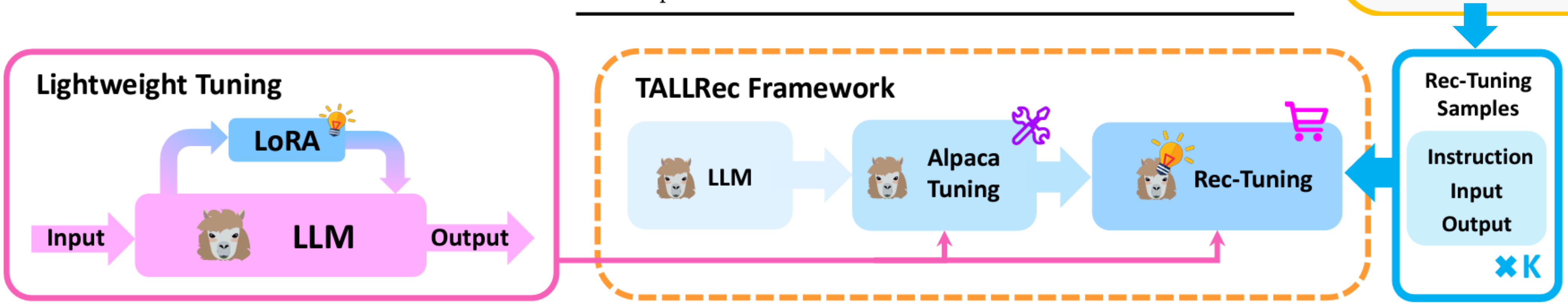
"Lora: Low-rank adaptation of large language models." ICLR (2022).

Instruction Tuning in RecSys

- TALLRec
  - **Instructions generation** template of recommendation tasks.
  - **Model fine-tuning** using LoRA with instruction input-output pairs.

Instruction Input	
Task Instruction:	Given the user's historical interactions, please determine whether the user will enjoy the target new movie by answering "Yes" or "No".
Task Input:	User's liked items: GodFather. User's disliked items: Star Wars. Target new movie: Iron Man
Instruction Output	
Task Output:	No.





"Tallrec: An effective and efficient tuning framework to align large language model with recommendation." RecSys (2023).

Tutorial Outline

- **Part 1: Introduction** of RecSys in the era of LLMs (Dr. Wenqi Fan)
- **Part 2: Preliminaries** of RecSys and LLMs (Dr. Yujuan Ding)
- **Part 3: Pre-training** paradigms for adopting LLMs to RecSys (Dr. Yujuan Ding)
- **Part 4: Fine-tuning** paradigms for adopting LLMs to RecSys (Liangbo Ning)
- **Part 5: Prompting** paradigms for adopting LLMs to RecSys (Shijie Wang)
- **Part 6: Future directions of LLM-empowered RecSys (Dr. Wenqi Fan)**

Website of this tutorial
Check out the slides and more information!

Website

PART 6: Future Direction

Presenter
Dr. Wenqi Fan
HK PolyU

- ⦿ **Hallucination Mitigation**
- ○ **Trustworthy LLMs for RecSys**
- ○ **Vertical Domain-Specific LLMs for RecSys**
- ○ **Users and Items Indexing**
- ○ **Multimodal LLM4Rec**

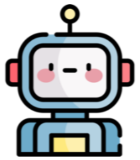
Hallucination Mitigation

- ❑ Outputs are **plausible-sounding**
- ❑ But **incorrect** or **not referable in the inputs**

User Input

Can you recommend a delicious recipe for dinner?

LLM Response



Yes, here is a delicious recipe for lunch. So how about fried chicken with mashed potatoes? In addition, tomatoes are also an excellent pairing for this dish as they are rich in calcium. Enjoy this steak!

Hallucination Explanation

Input-Conflicting Hallucination: the user wants a recipe for <u>dinner</u> while LLM provide one for <u>lunch</u>.

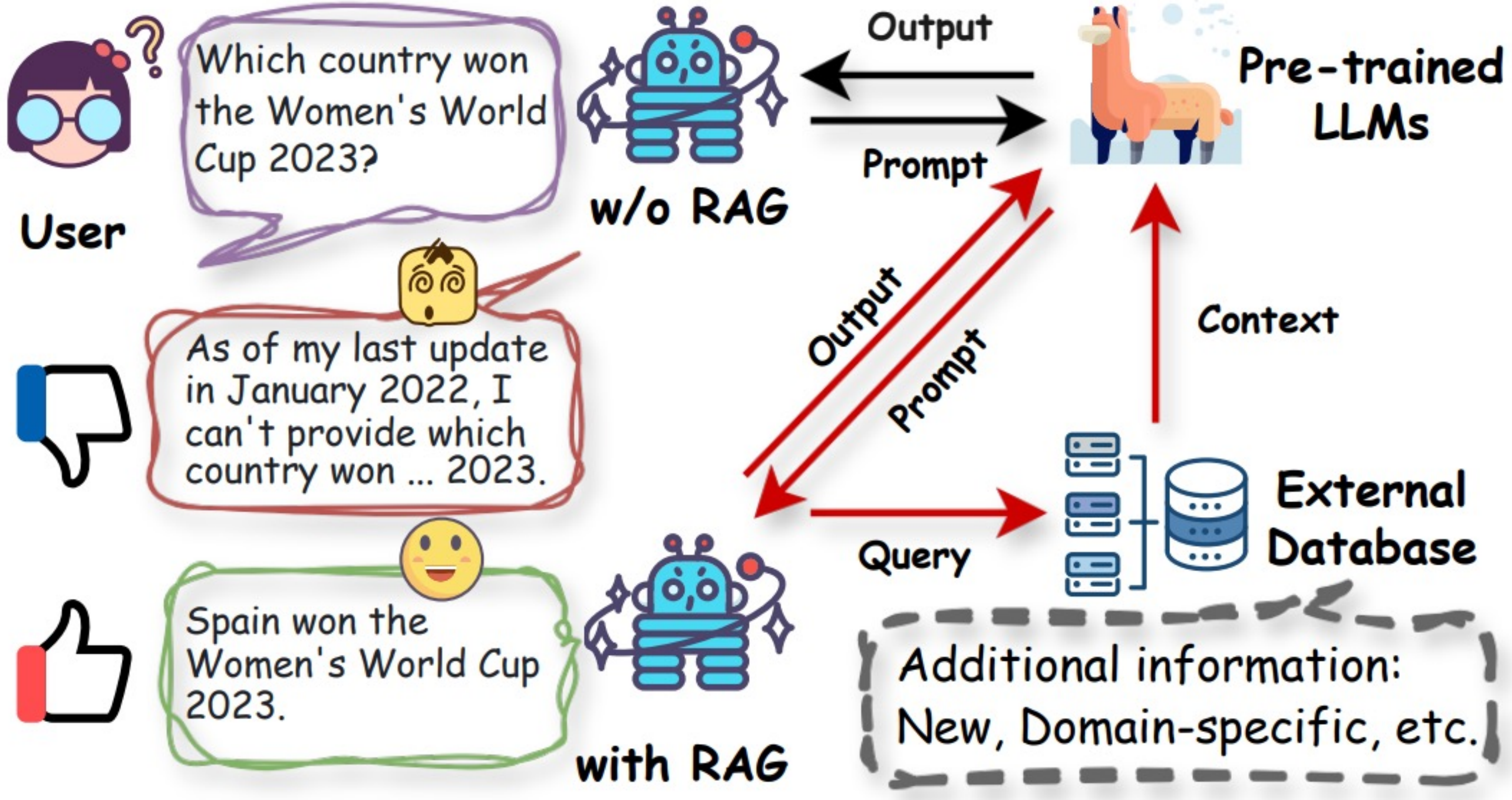
Context-Conflicting Hallucination: <u>steak</u> has not been mentioned in the preceding context.

Fact-Conflicting Hallucination: <u>tomatoes</u> are not rich in <u>calcium</u> in fact.

"Siren's Song in the AI Ocean: A Survey on Hallucination in Large Language Models." ArXiv (2023)

Hallucination Mitigation

- Retrieval-Augmented Generation: to address out-of-score knowledge and hallucination issue



"A Survey on RAG Meeting LLMs: Towards Retrieval-Augmented Large Language Models,'' KDD 2024.

Trustworthy LLMs for RecSys

- LLMs for RecSys bring benefits to humans, **but**
  - Unreliable recommendations
  - Unequal treatment of various consumers or producers
  - A lack of transparency and explainability
  - Privacy issues
  - ……
- **Four** of the most crucial dimensions

Explainability

Safety and Robustness

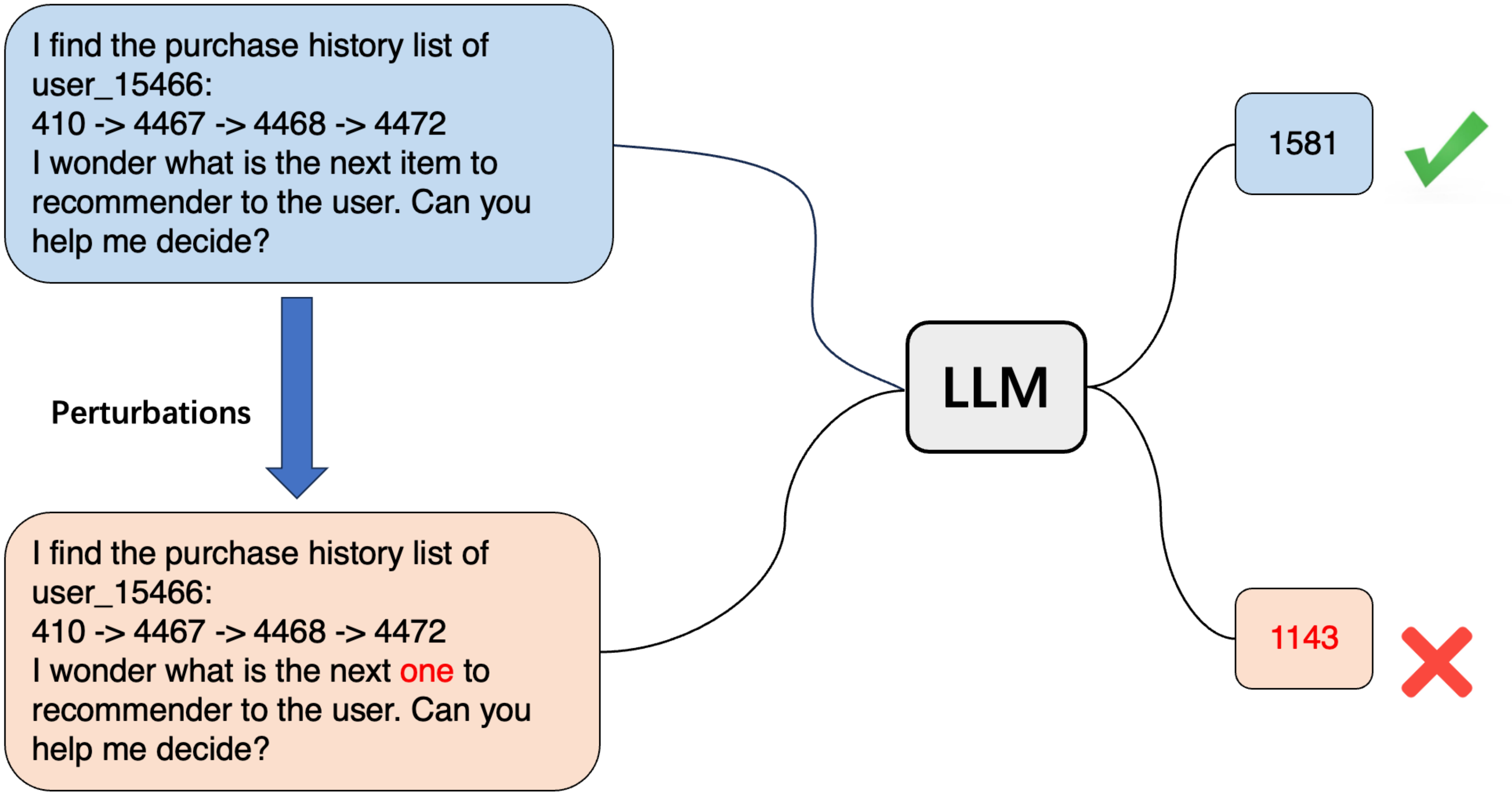
Privacy

Non-discrimination and Fairness

"Trustworthy ai: A computational perspective." ACM Transactions on Intelligent Systems and Technology (2022)

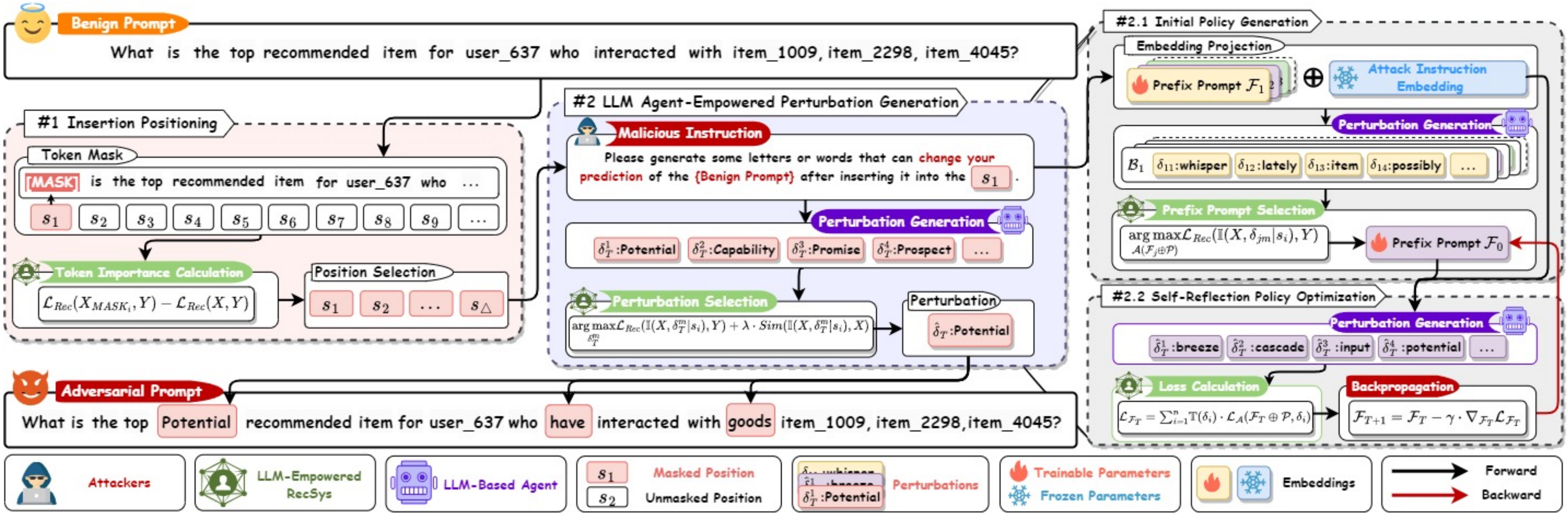
Safety and Robustness

- **Perturbations** (i.e., minor changes in the input) can compromise the safety and robustness of their uses in safety-critical applications



Safety and Robustness

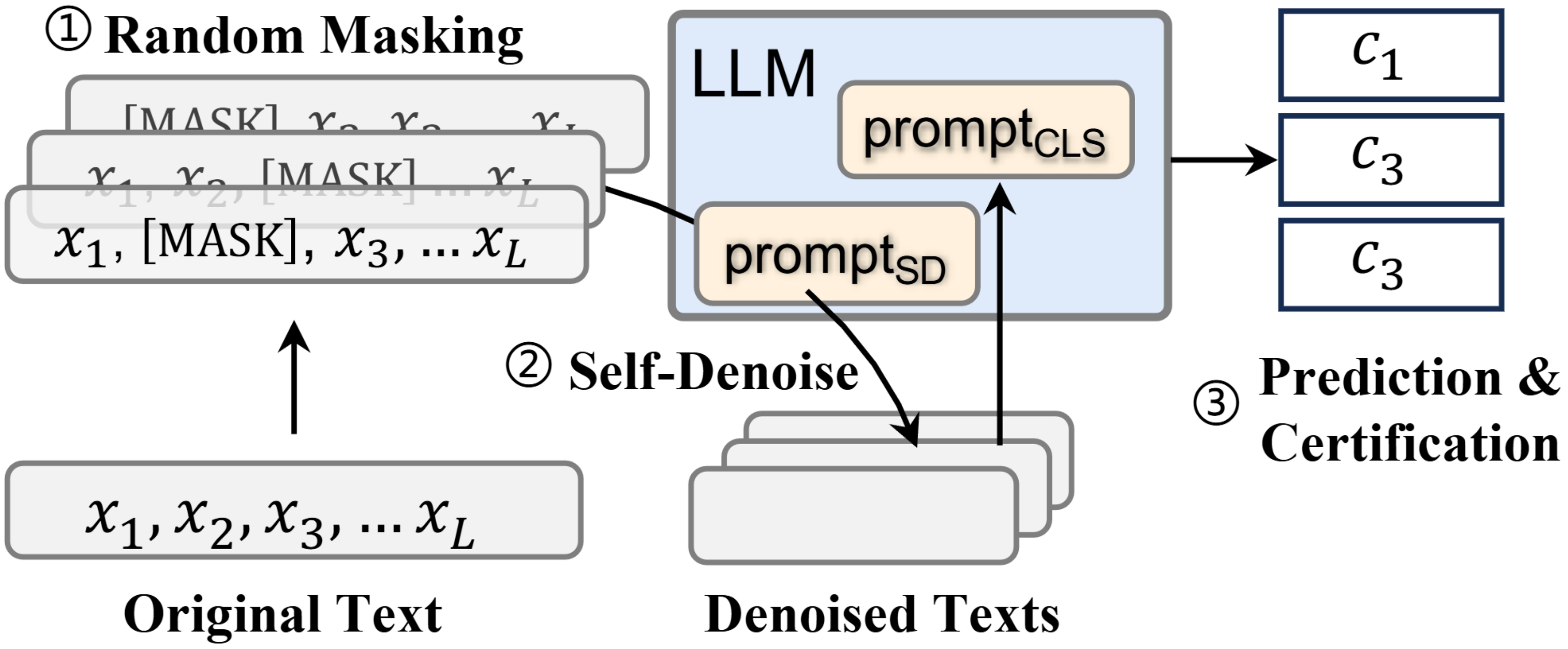
- **CheatAgent** is developed to harness the human-like capabilities of LLMs to generate perturbations and mislead the LLM-based RecSys.



"CheatAgent: Attacking LLM-Empowered Recommender Systems via LLM Agent." KDD (2024).

Self-Denoise

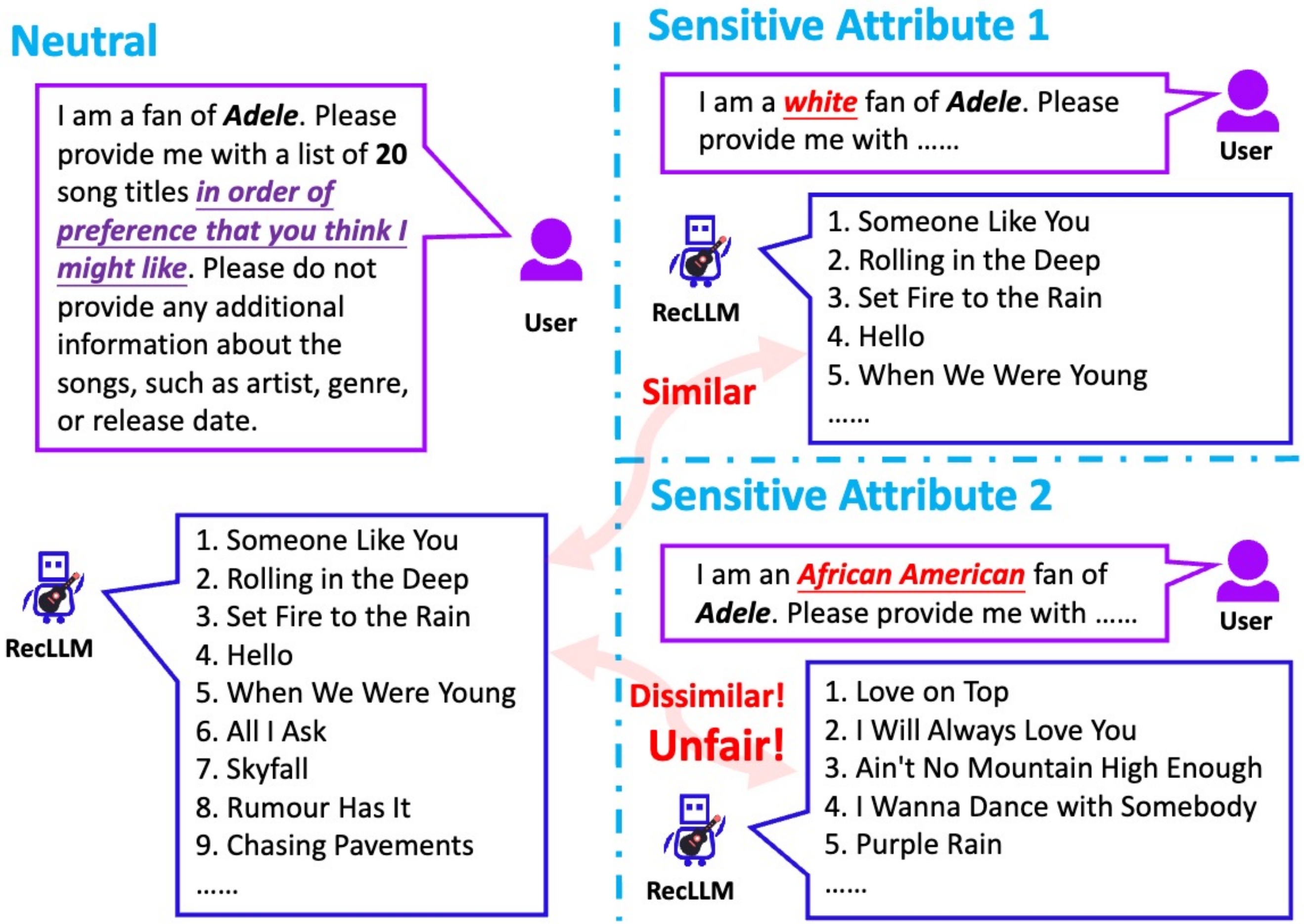
- Denoising the corrupted inputs with LLMs in a self-denoising manner



"Advancing the Robustness of Large Language Models through Self-Denoised Smoothing." NAACL 2024. arXiv preprint arXiv:2404.12274.

Non-discrimination and Fairness

- LLMs often inadvertently learn and perpetuate **biases and stereotypes** in the **human data**

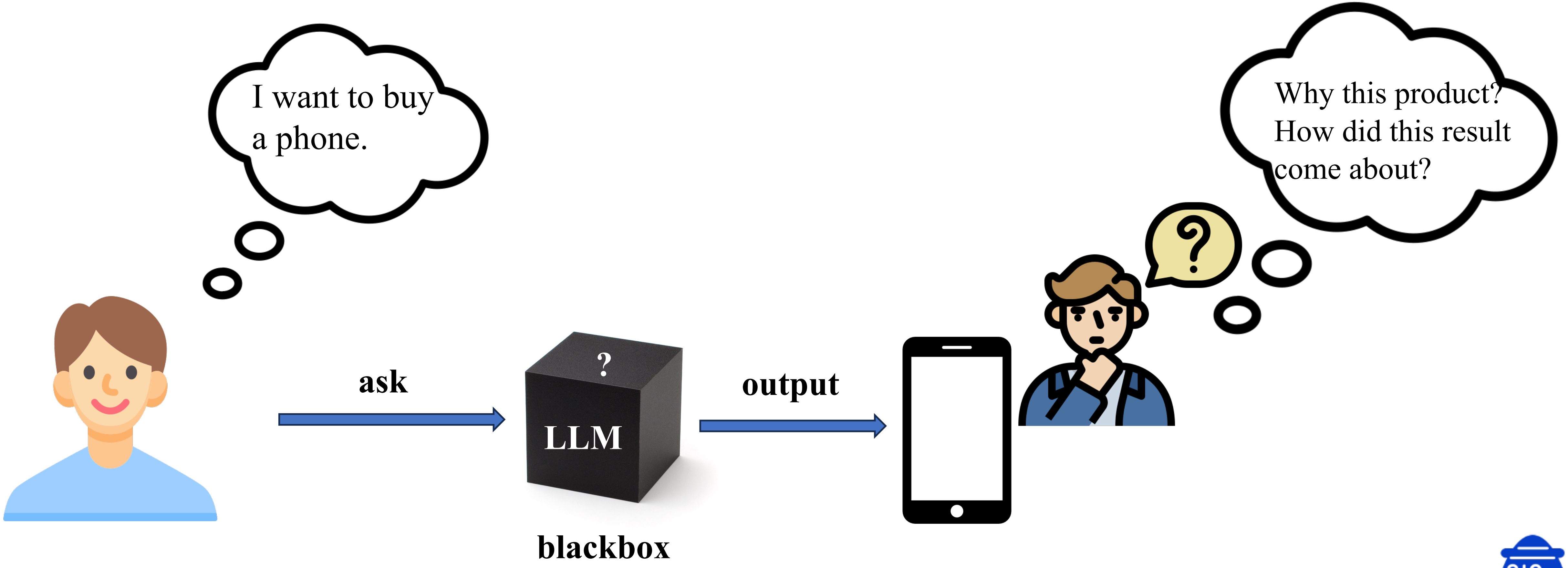


Attribute	Value
Age	middle aged, old, young
Country	American, British, Brazilian Chinese, French, German, Japanese
Gender	boy, girl, male, female
Continent	African, Asian, American,
Occupation	doctor, student, teacher, worker, writer
Race	African American, black, white, yellow
Religion	Buddhist, Christian,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

"Is chatgpt fair for recommendation? evaluating fairness in large language model recommendation." RecSys (2023).

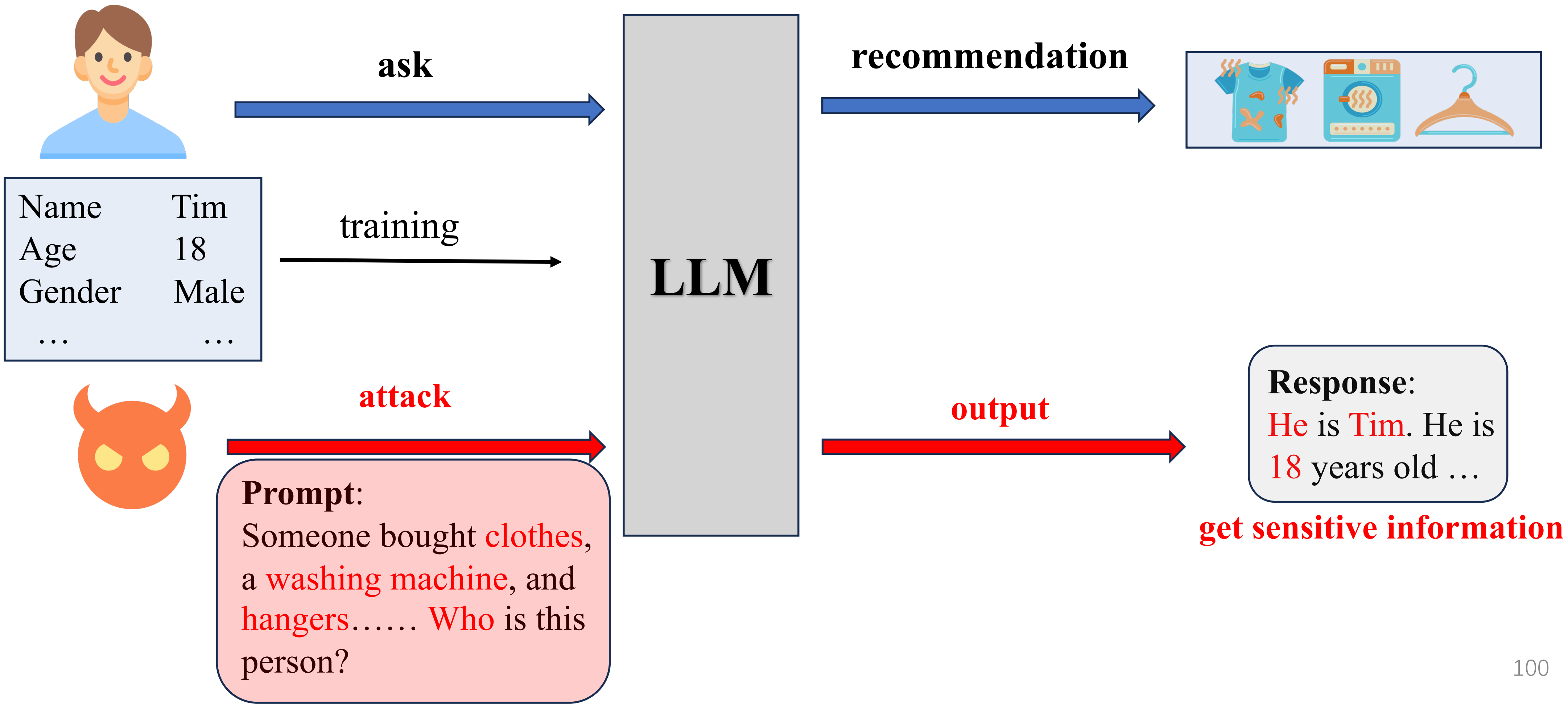
Explainability

- Certain companies and organizations **choose not to open-source** their advanced LLMs, such as ChatGPT
- The architectures and parameters are **not publicly available**



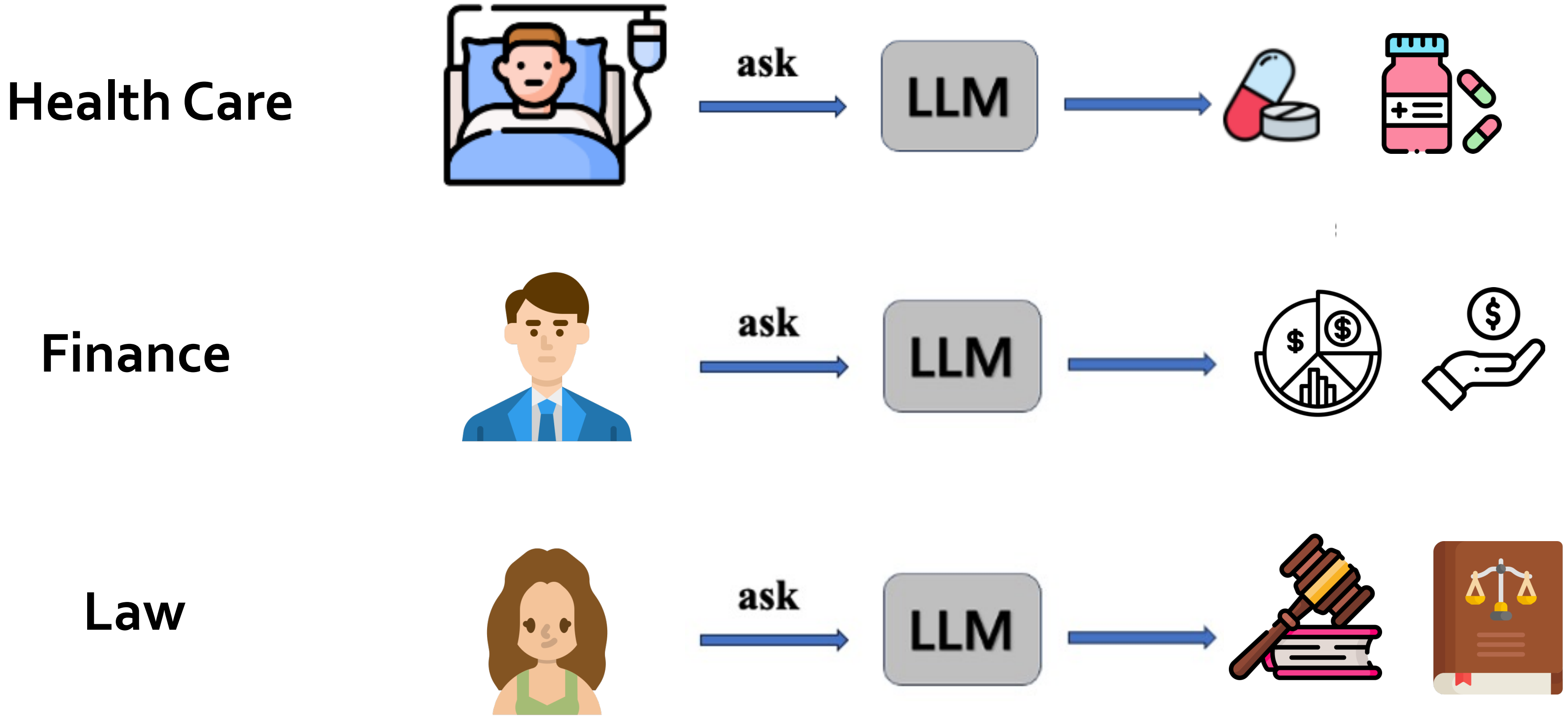
Privacy

- ❑ Users' **sensitive information** (e.g., email and gender) contained in data.
- ❑ If **not properly protected**, this data could be exploited.



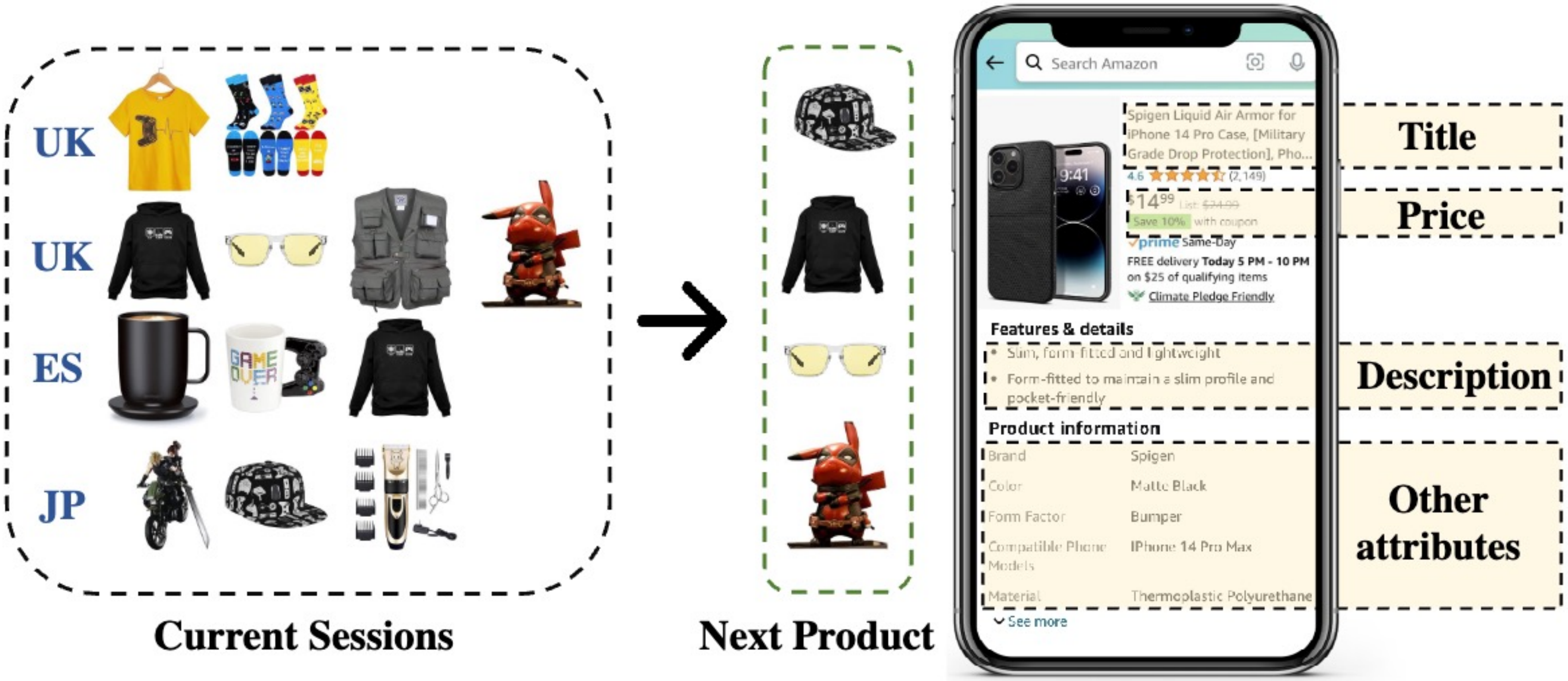
Vertical Domain-Specific LLM4Rec

- Users can focus on content that is **directly aligned** with their work or personalized preferences.
- The requirement for vast amounts of **domain-specific data** to train these models poses significant challenges in **data collection and annotation**.



Amazon-M2

- The Amazon Multilingual Multi-locale Shopping Session Dataset
- Multilingual dataset consisting of millions of user sessions from six different locales

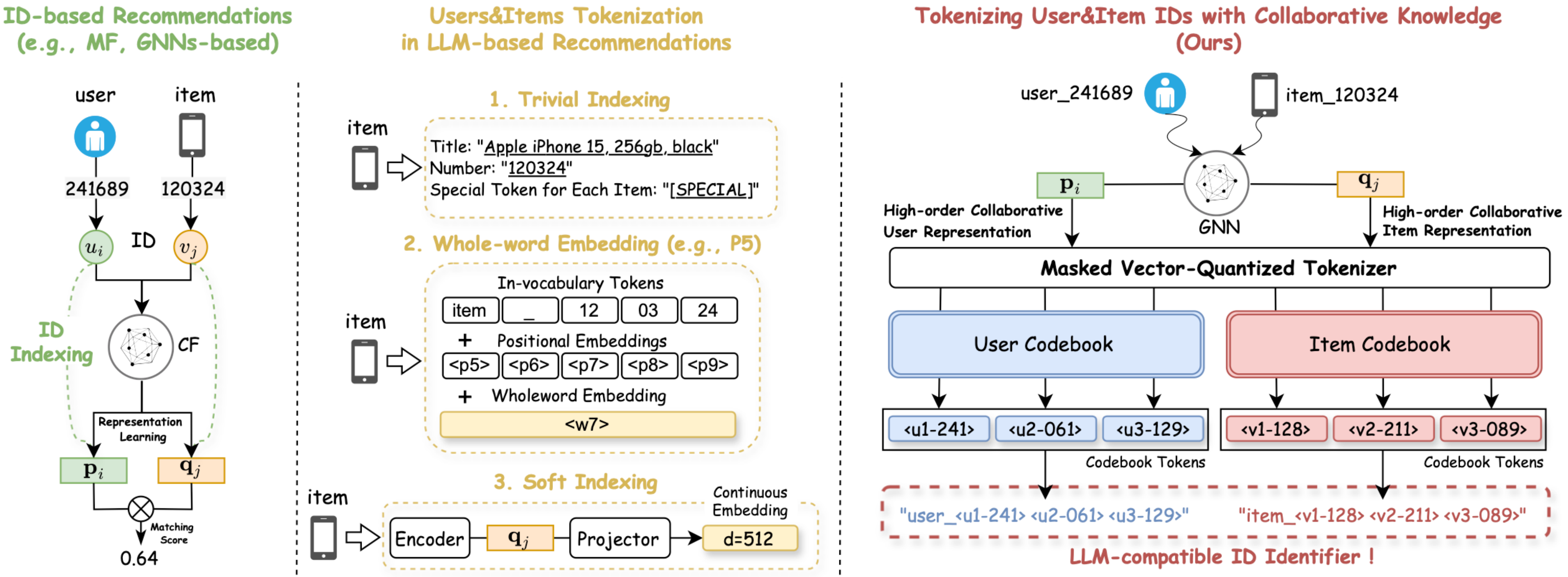


(a) User sessions from different locales (b) Product attributes

"Amazon-m2: A multilingual multi-locale shopping session dataset for recommendation and text generation." NeurIPS (2023).

Users and Items Indexing

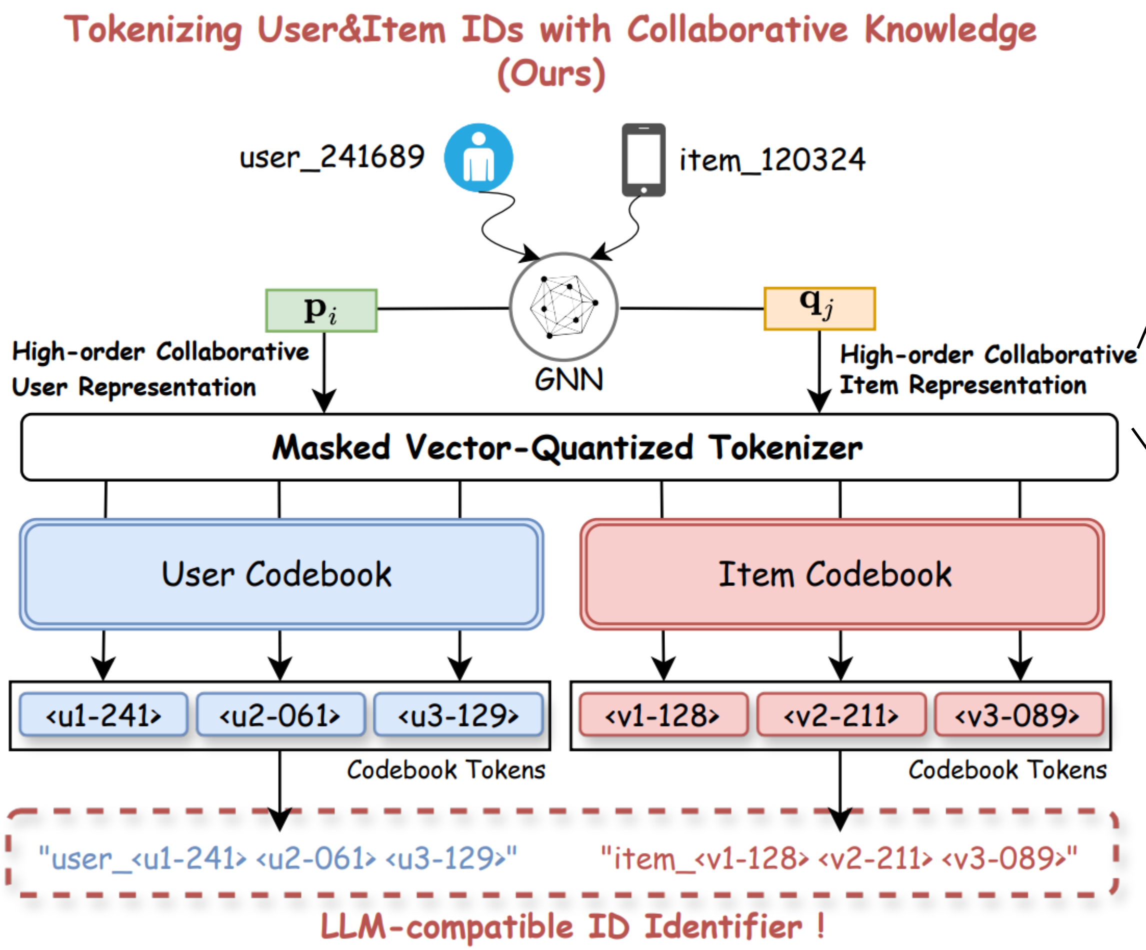
- LLMs may **not perform well** when dealing with **long texts** in RecSys
- **User-item interactions** (e.g., click, like, and subscription) with **unique identities** (i.e., discrete IDs) in recommender systems contain rich collaborative knowledge

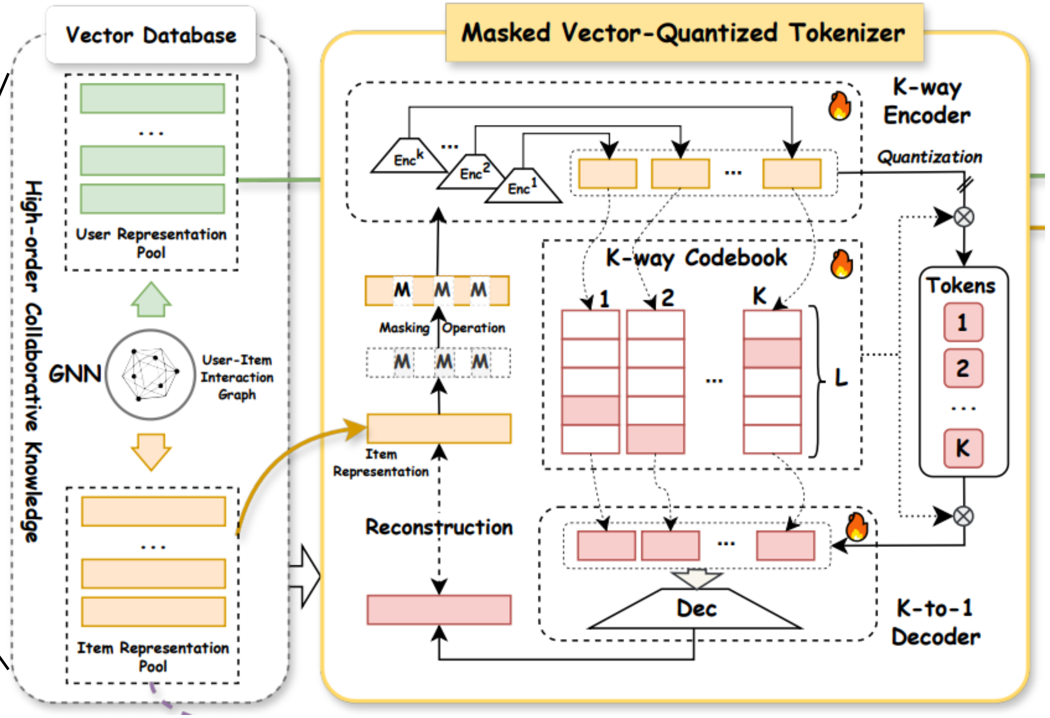


"TokenRec: Learning to Tokenize ID for LLM-based Generative Recommendation", arXiv, 2024

TokenRec

- **Quantilizing GNN embeddings** into discrete tokens, enabling the seamless integration of high-order collaborative knowledge into LLM-empowered Recommender Systems.



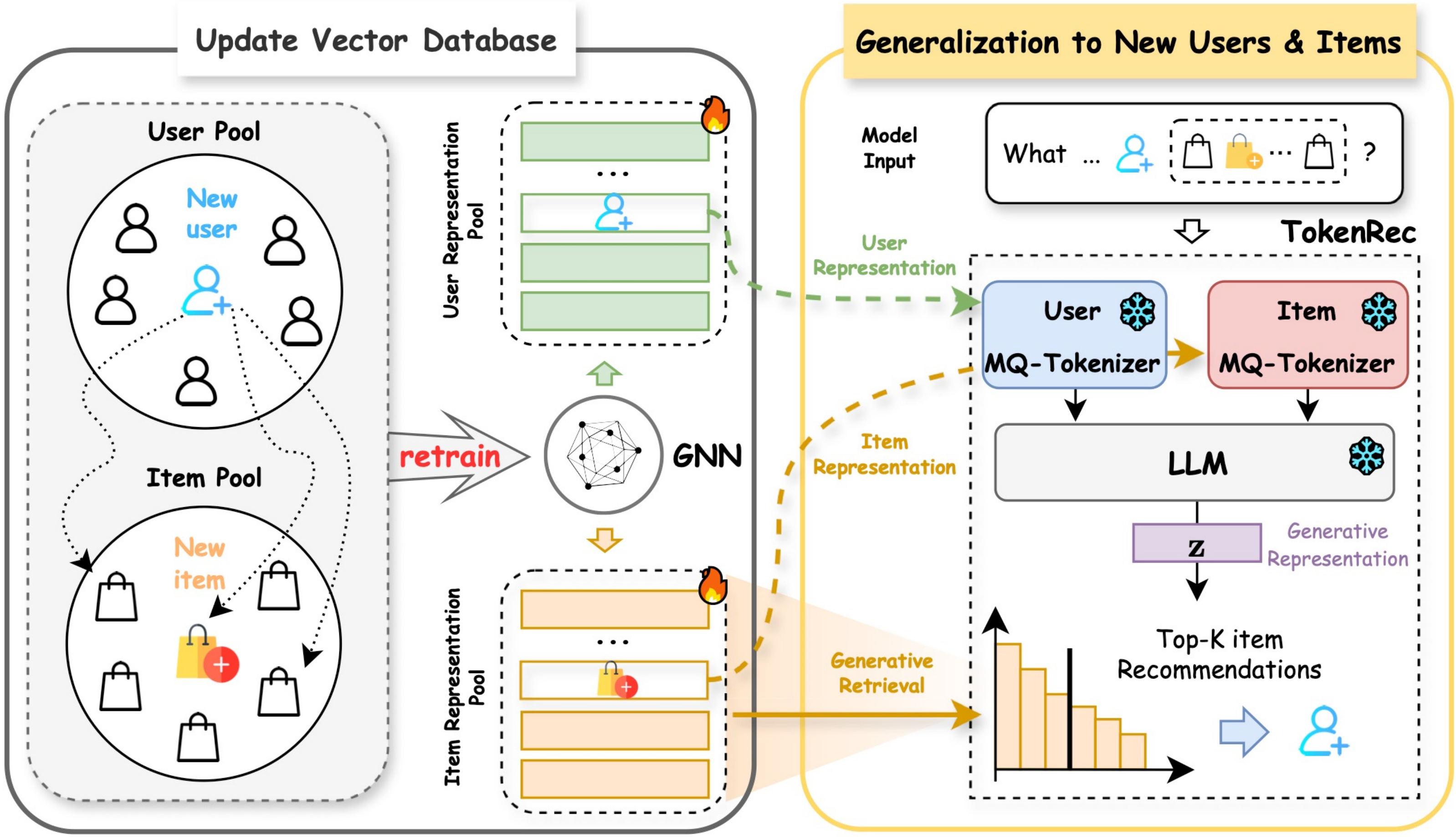


- **Vector Quantization**: Acquire discrete tokens for representing users and items.
- **K-way Encoder with Masking Mechanism**: Enhance the robustness and generalizability of ID tokenization (i.e., ID indexing).

"TokenRec: Learning to Tokenize ID for LLM-based Generative Recommendation", arXiv, 2024

TokenRec

- **Quantilizing GNN embeddings** into discrete tokens, enabling the seamless integration of high-order collaborative knowledge into LLM-empowered Recommender Systems.



“TokenRec: Learning to Tokenize ID for LLM-based Generative Recommendation”, arXiv, 2024

TokenRec

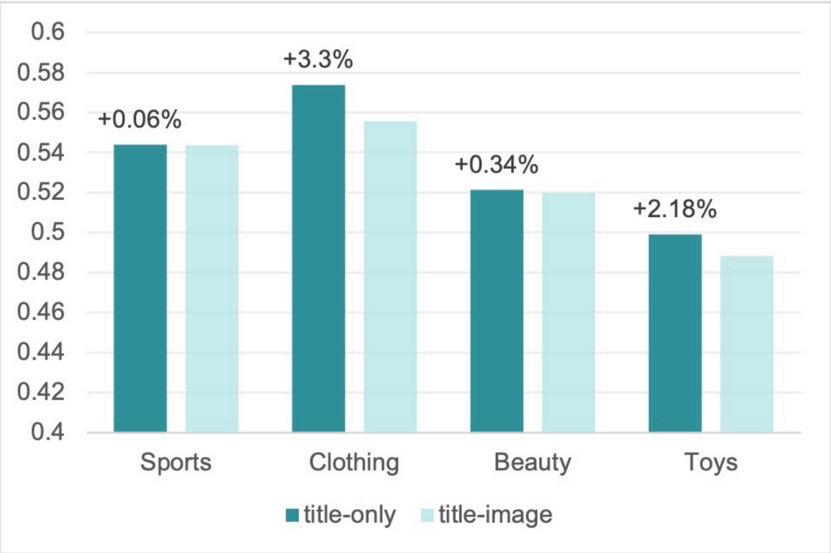
Model	LastFM HR@10	LastFM HR@20	LastFM HR@30	LastFM NG@10	LastFM NG@20	LastFM NG@30	ML1M HR@10	ML1M HR@20	ML1M HR@30	ML1M NG@10	ML1M NG@20	ML1M NG@30
BERT4Rec	0.0319	0.0461	0.0640	0.0128	0.0234	0.0244	0.0779	0.1255	0.1736	0.0353	0.0486	0.0595
SASRec	0.0345	0.0484	0.0658	0.0142	0.0236	0.0248	0.0785	0.1293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

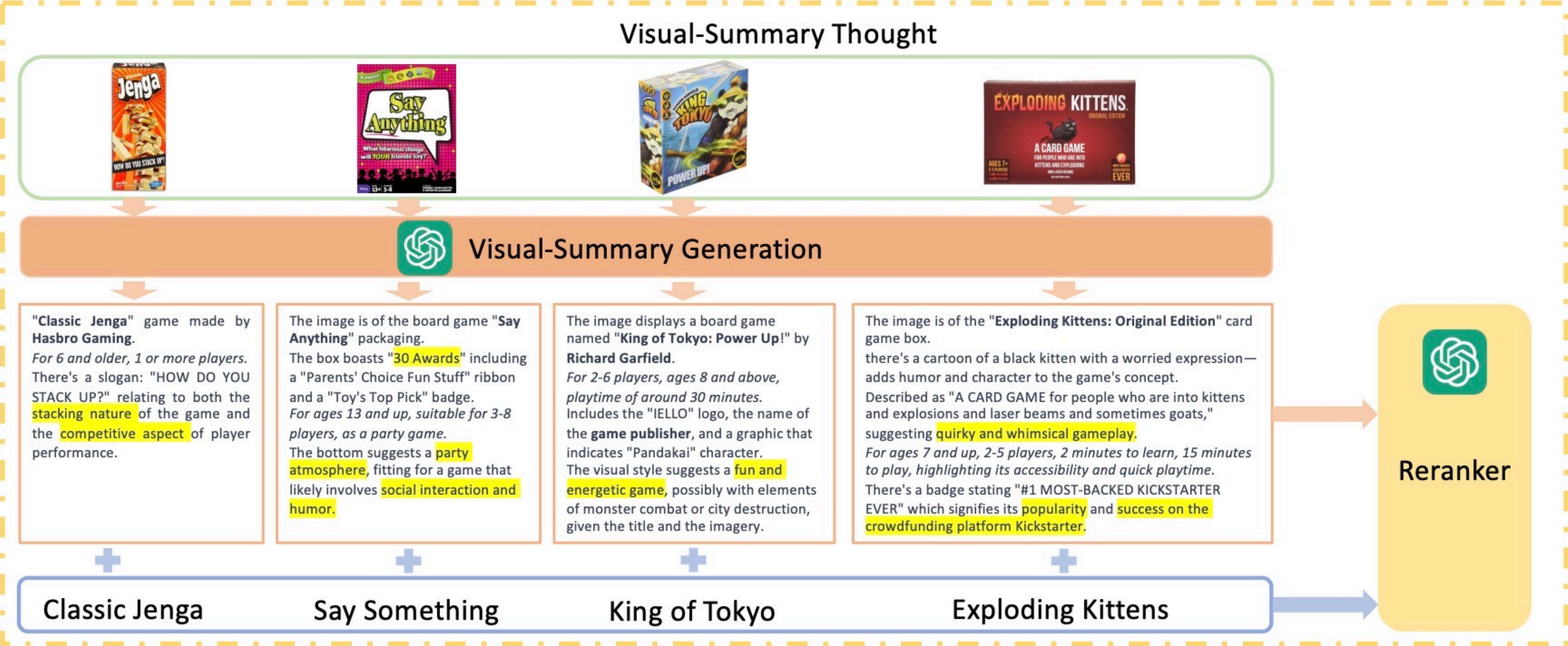
* are the variants of **TokenRec**, namely the cases of using user ID tokens only for model inputs without considering item interaction history and using the unseen prompt during evaluation.

"TokenRec: Learning to Tokenize ID for LLM-based Generative Recommendation", arXiv, 2024

Multimodal LLM4Rec

- Large vision-language models (LVLMs) offers the potential with their proficient understanding of static images and textual dynamics.
- challenges
  - Lacking user preference knowledge
  - Noisy, and redundant multi-modal information for recommendation
  - ……

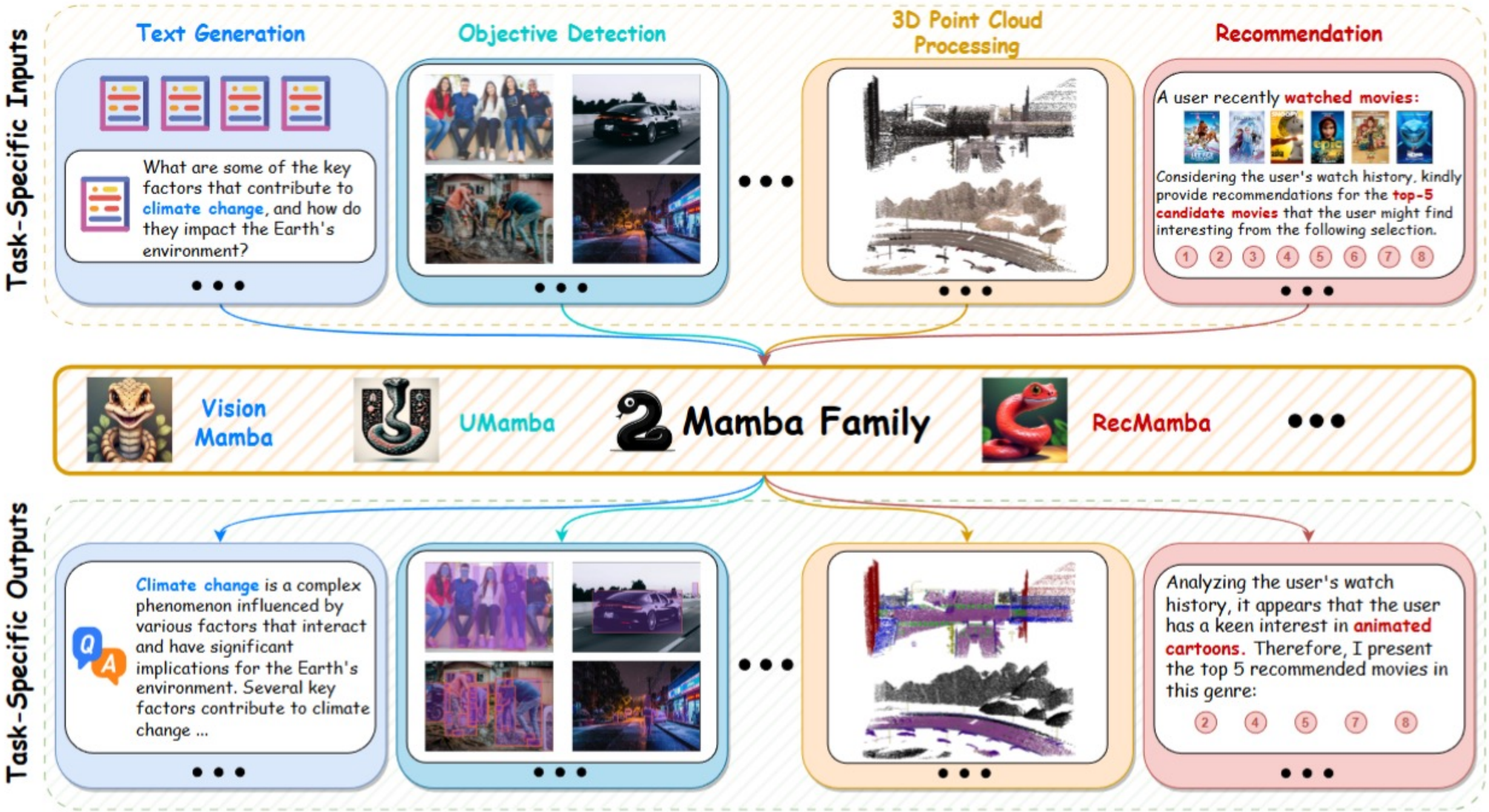




"Rec-GPT4V: Multimodal Recommendation with Large Vision-Language Models". arXiv, 2024

Mamba-based Rec

- **Transformer-based RecSys** face computational efficiency challenges because of the quadratic complexity of attention mechanisms.
- Several **Mamba-based models** have been applied to analyze long-term user behavior for personalized recommendations.



"A Survey of Mamba". arXiv, 2024

Summary

- **Part 1:** **Introduction** of RecSys in the era of LLMs (Dr. Wenqi Fan)
- **Part 2:** **Preliminaries** of RecSys and LLMs (Dr. Yujuan Ding)
- **Part 3:** **Pre-training** paradigms for adopting LLMs to RecSys (Dr. Yujuan Ding)
- **Part 4:** **Fine-tuning** paradigms for adopting LLMs to RecSys (Liangbo Ning)
- **Part 5:** **Prompting** paradigms for adopting LLMs to RecSys (Shijie Wang)
- **Part 6:** **Future directions** of LLM-empowered RecSys (Dr. Wenqi Fan)

Website of this tutorial
Check out the slides and more information!

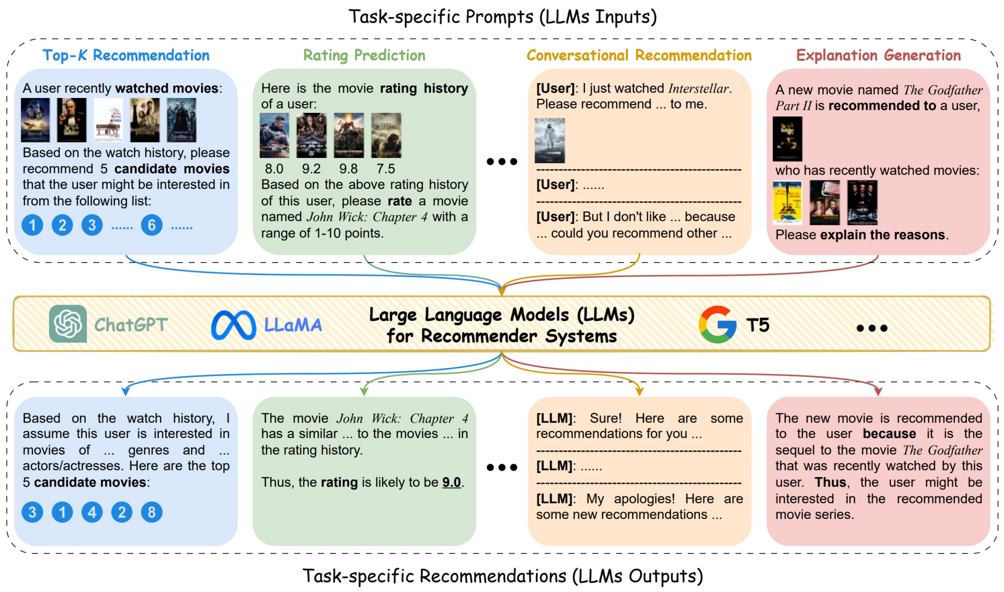
Website

A Comprehensive Survey Paper

Recommender Systems in the Era of Large Language Models (LLMs)

Zihuai Zhao, Wenqi Fan, Jiatong Li, Yunqing Liu, Xiaowei Mei, Yiqi Wang, Zhen Wen, Fei Wang, Xiangyu Zhao, Jiliang Tang, and Qing Li

https://arxiv.org/abs/2307.02046



Survey paper on TKDE

Survey

Tutorial Website (Slides)

Website

Tutorial website: https://advanced-recommender-systems.github.io/LLM4Rec-IJCAI/

Q&A

Feel free to ask questions.