中国科学院计算技术研究所
INSTITUTE OF COMPUTING TECHNOLOGY, CHINESE ACADEMY OF SCIENCES

OneRec - V1

OneRec Technical Report

Guorui Zhou, Jiaxin Deng, Jinghao Zhang, Kuo Cai, Lejian Ren, Qiang Luo, Qianqian Wang, Qigen Hu, Rui Huang, Shiyao Wang, Weifeng Ding, Wuchao Li, Xinchen Luo, Xingmei Wang, Zexuan Cheng, Zixing Zhang, Bin Zhang, Boxuan Wang, Chaoyi Ma, Chengru Song, Chenhui Wang, Di Wang, Dongxue Meng, Fan Yang, Fangyu Zhang, Feng Jiang, Fuxing Zhang, Gang Wang, Guowang Zhang, Han Li, Hengrui Hu, Hezheng Lin, Hongtao Cheng, Hongyang Cao, Huanjie Wang, Jiaming Huang, Jiapeng Chen, Jiaqiang Liu, Jinghui Jia, Kun Gai, Lantao Hu, Liang Zeng, Liao Yu, Qiang Wang, Qidong Zhou, Shengzhe Wang, Shihui He, Shuang Yang, Shujie Yang, Sui Huang, Tao Wu, Tiantian He, Tingting Gao, Wei Yuan, Xiao Liang, Xiaoxiao Xu, Xugang Liu, Yan Wang, Yi Wang, Yiwu Liu, Yue Song, Yufei Zhang, Yunfan Wu, Yunfeng Zhao, Zhanyu Liu

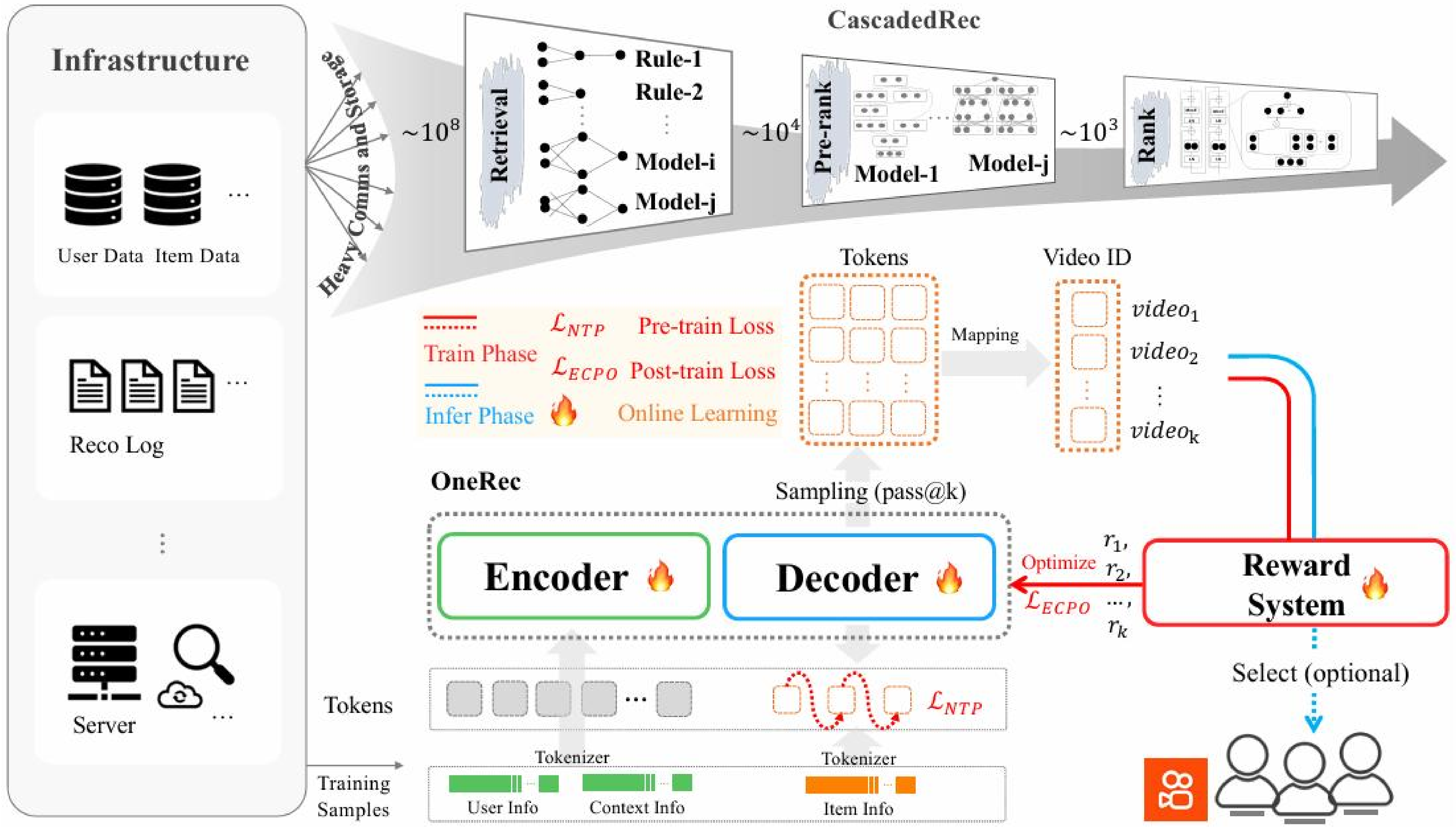
arxiv: 25.06 => 25.09

于璐璐 25.10.28

» Motivation

Recommendation System: Cascaded => End-to-End (integrating retrieval and ranking)

- **Fragmented Compute**
  - low computational efficiency
- **Objective Collision**
  - cross-stage modeling conflicts
- **Lag Behind AI Evolution**
  - remarkable progress in LLM and VLM domains
    - scaling laws



» Architecture (generative)

Tokenizer: item representation => coarse-to-fine semantic IDs

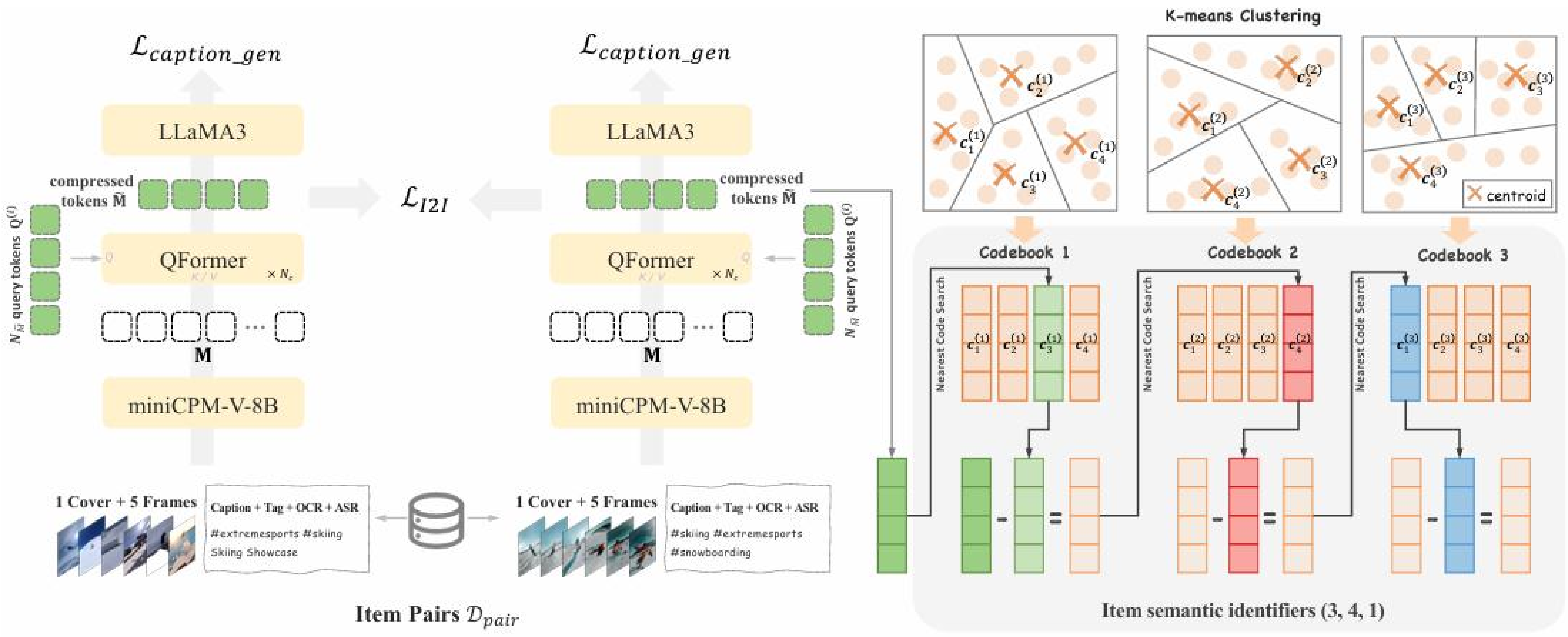


Figure 3 | Illustration of our tokenizer implementation. We first align multimodal representations of item pairs with high collaborative similarity to obtain collaborative multimodal representations, then tokenize these representations into discrete semantic IDs using RQ-Kmeans.

- **Aligned Collaborative-Aware Multimodal Representation**
  - **Prior Work:** multimodal representation (**context features**) => semantic IDs
    - ➢ neglecting **collaborative signals**

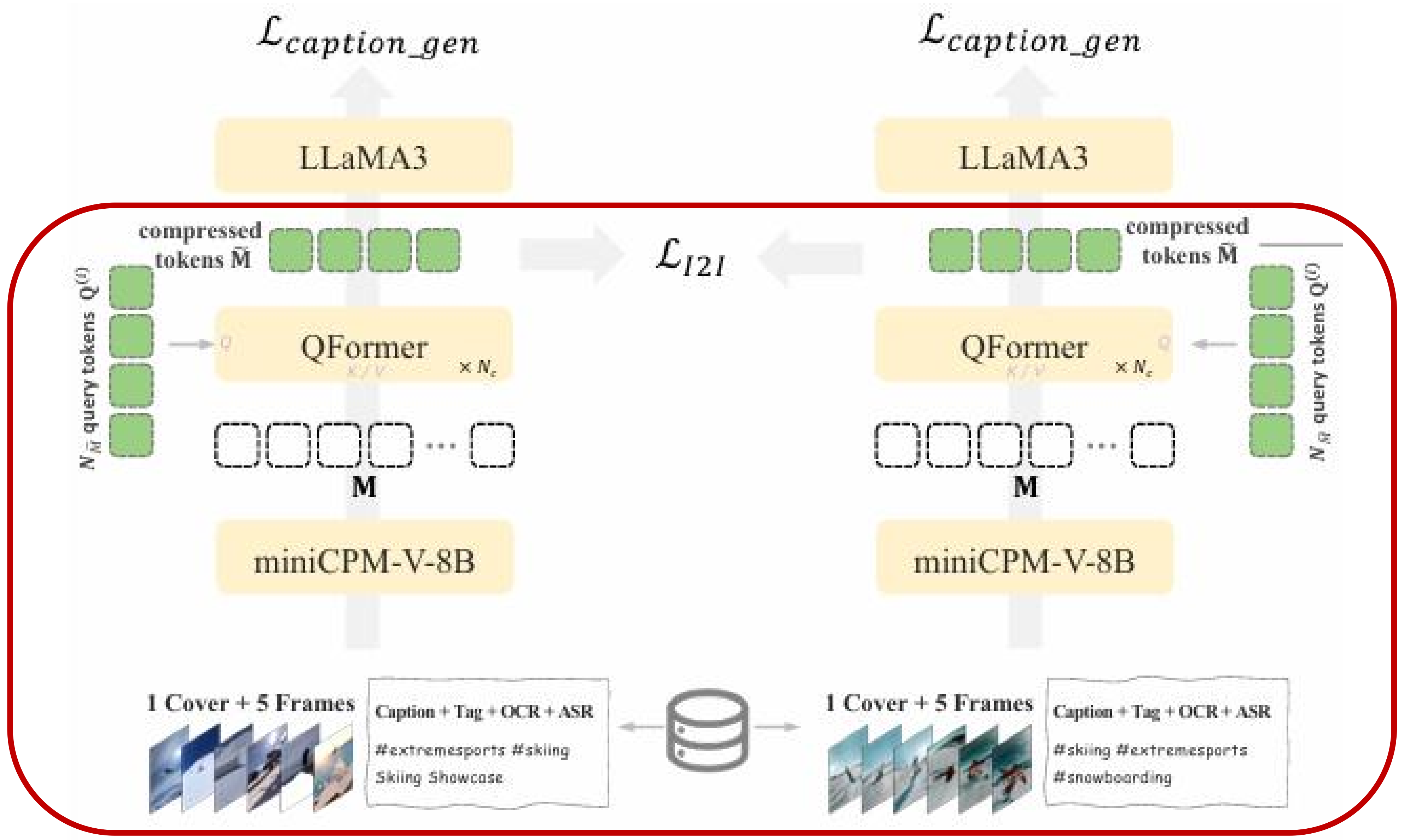
» Architecture (generative)

Tokenizer: item representation => coarse-to-fine semantic IDs

- **Aligned Collaborative-Aware Multimodal Representation**

➢ align multimodal representations of **collaboratively similar item pairs**

  - **Multimodal Rpresentations**



Item Pairs $\mathcal{D}_{pair}$

Architecture (generative)

Tokenizer: item representation => coarse-to-fine semantic IDs

- **Aligned Collaborative-Aware Multimodal Representation**
  - **Item Pairs**
    - **User-to-Item**

      (positively target item, most collaboratively similar item from historical positives)
    - **Item-to-Item:** high similarity scores, e.g., Swing similarity

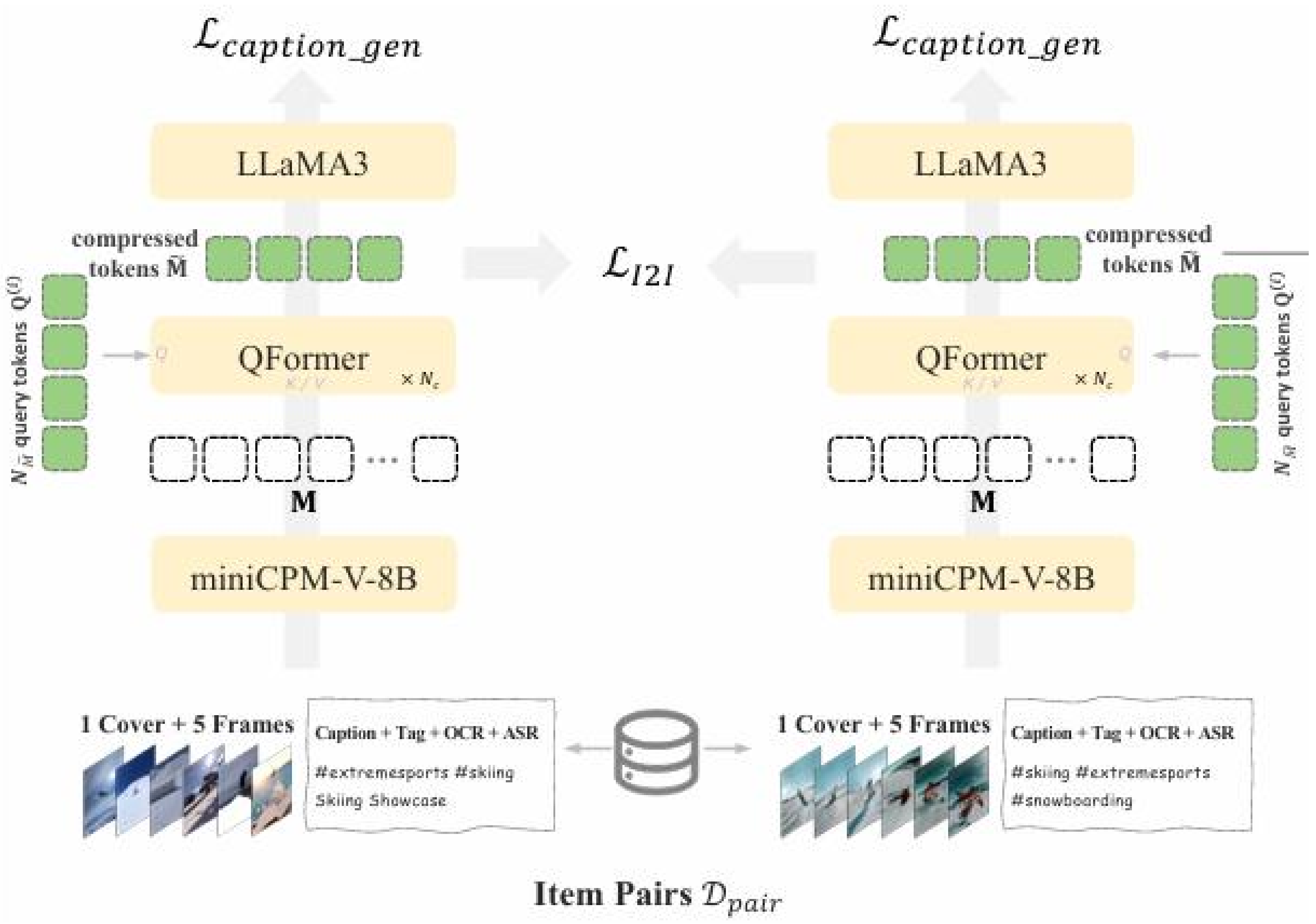
» Architecture (generative)

Tokenizer: item representation => coarse-to-fine semantic IDs

- **Aligned Collaborative-Aware Multimodal Representation**
  - **Item-to-Item Loss and Caption Loss**
    - **Item-to-Item Loss:** align representations of collaboratively similar pairs
    - **Caption Loss:** preserve content understanding capabilities

$$\mathcal{L}_{I2I} = -\frac{1}{|\mathcal{B}|} \sum_{(i,j)\in\mathcal{B}} \log \frac{\exp\left(\mathrm{sim}\left(\tilde{\mathbf{M}}_i, \tilde{\mathbf{M}}_j\right)/\tau\right)}{\sum_{(i',j')\in\mathcal{B}} \exp\left(\mathrm{sim}\left(\tilde{\mathbf{M}}_i, \tilde{\mathbf{M}}_{j'}\right)/\tau\right)}$$

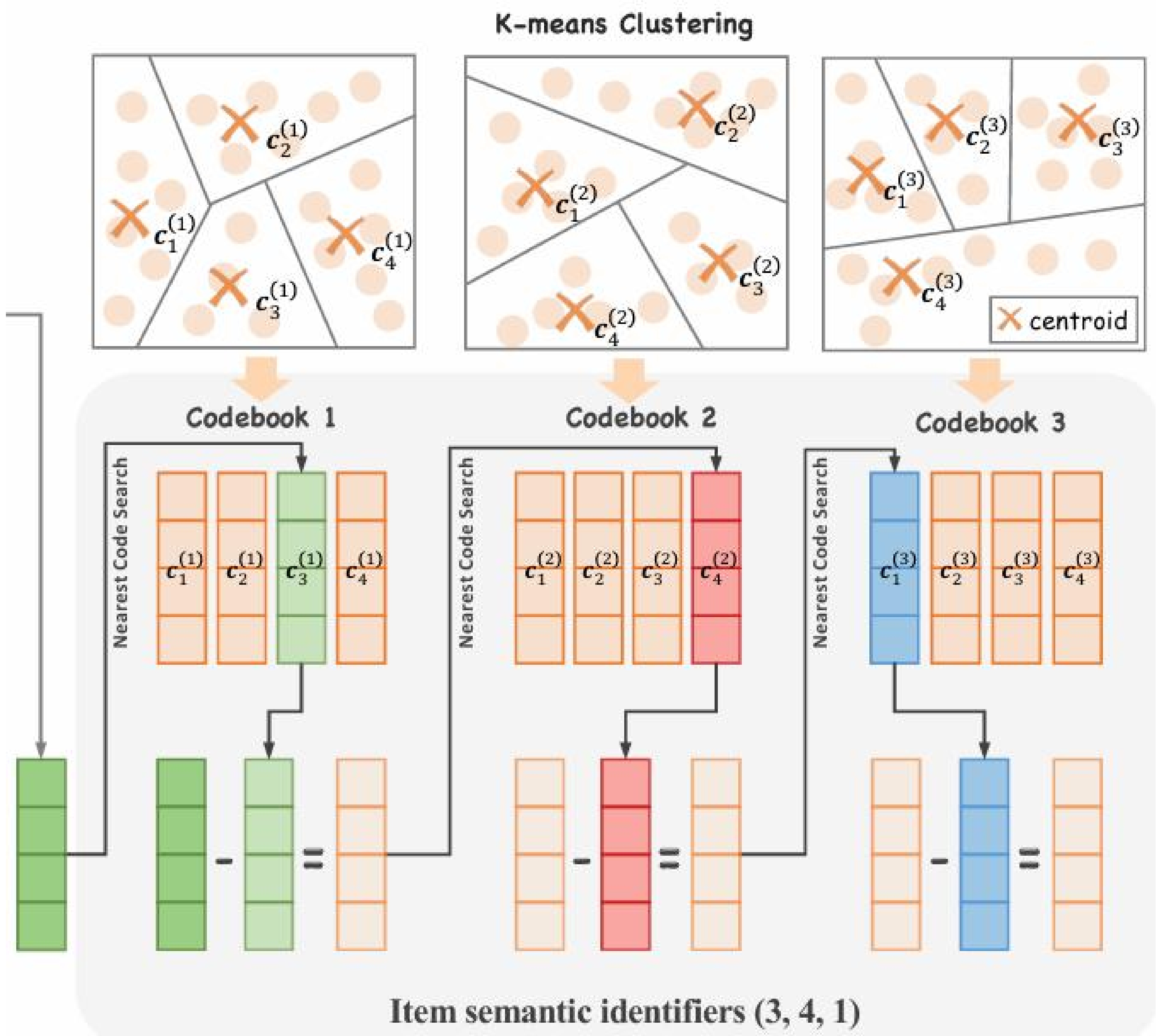
$$\mathcal{L}_{caption\_gen} = -\sum_{k} \log P\left(t^{k+1} \mid \left[t^1, t^2, \cdots, t^k\right]\right)$$



Architecture (generative)

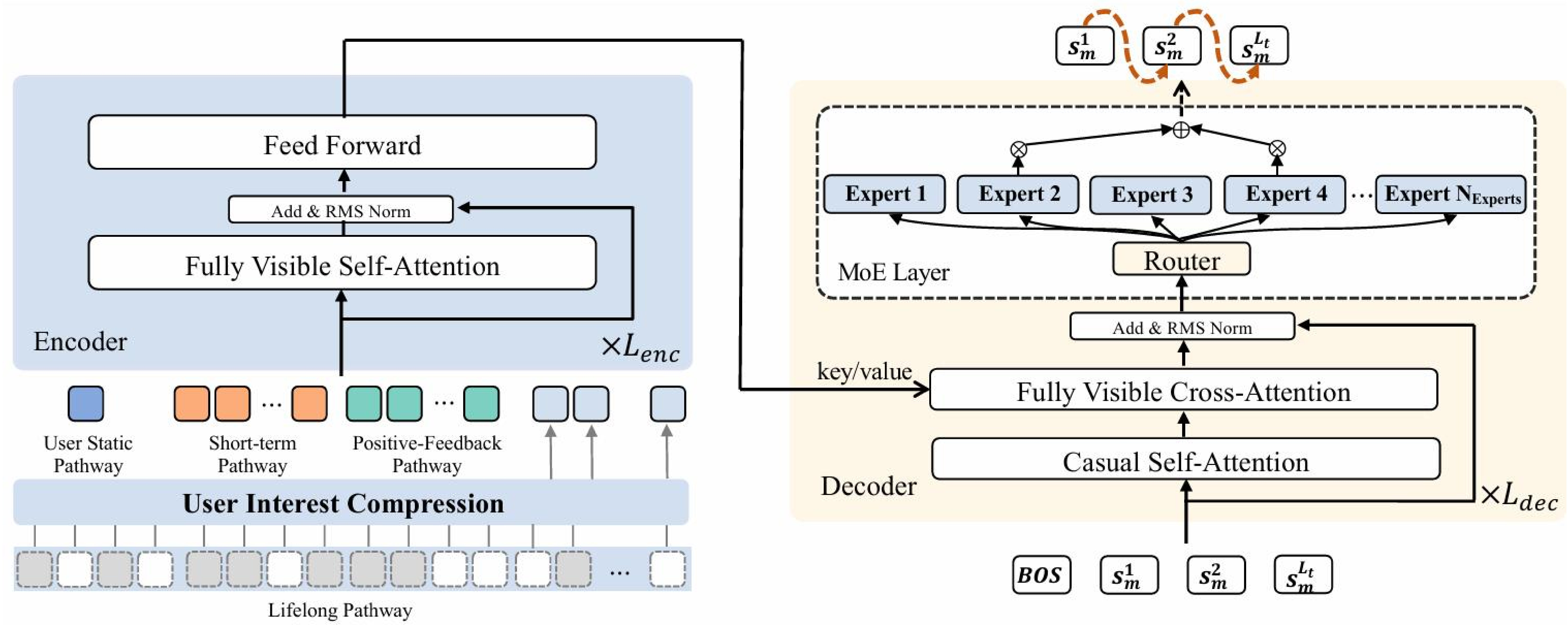
Tokenizer: item representation => coarse-to-fine semantic IDs

- **Tokenization: RQ-Kmeans**



» Architecture (generative)

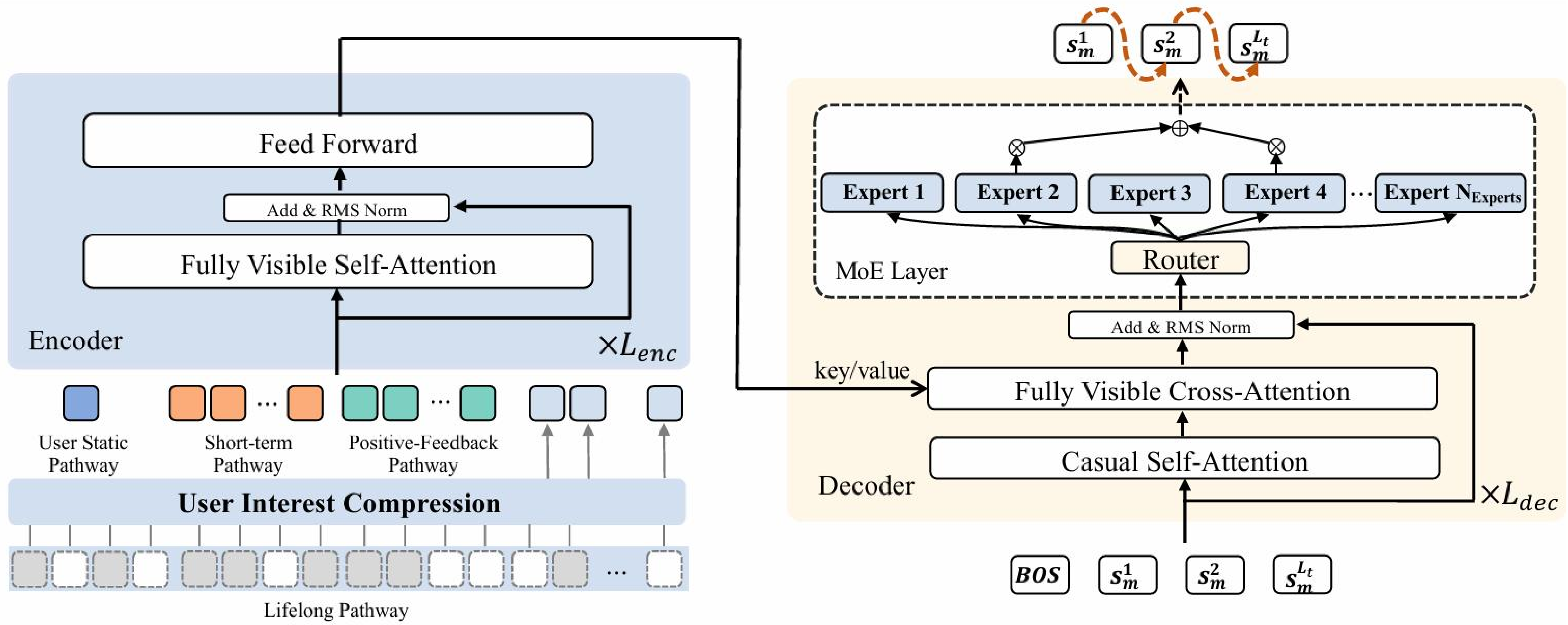
Encoder - Decoder



- **Encoder => Multi-Scale Feature Engineering**
  - **Lifelong Pathway:** up tp 100,000 items => two-stage hierachical compression strategy
    - **Behavior Compression:** hierachical K-means clustering => item closet to the centroid
    - **Feature Aggregation:** categorical features; continuous features: average

» Architecture (generative)

Encoder - Decoder



- **Decoder:** pointwise generation

The model is trained using cross-entropy loss for next-token prediction on the semantic identifiers of target video m:

$$\mathcal{L}_{\text{NTP}} = -\sum_{j=0}^{L_t-1} \log P\left(s_m^{j+1} \mid \left[s_{[\text{BOS}]}, s_m^1, s_m^2, \cdots, s_m^j\right]\right)$$

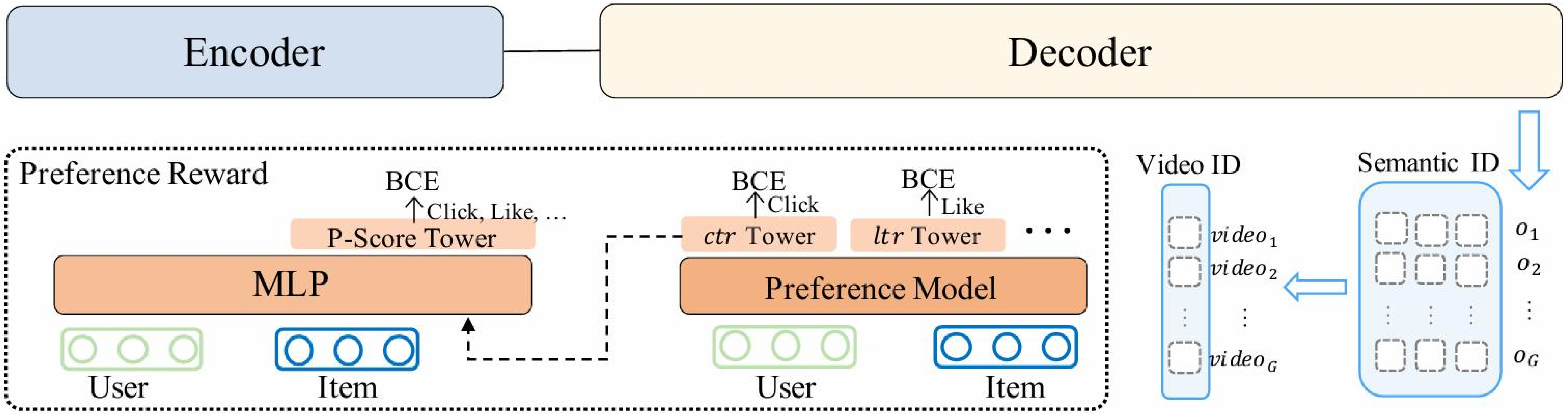
» Architecture (generative)

Reward System

- The pre-trained model only fits the distribution of the **exposed item space**. (obtained from the past traditional system) ==> **preference alignment**

➢ User Preference Alignment

- Defining a "good recommendation" is challenging, including multiple objectives $\xrightarrow[\textbf{fusion}]{\textbf{weighted}}$ score
- **Reward: P-Score,** a neural network to **learn a personalized fusion socre**



Architecture (generative)

Reward System => User Preference Alignment

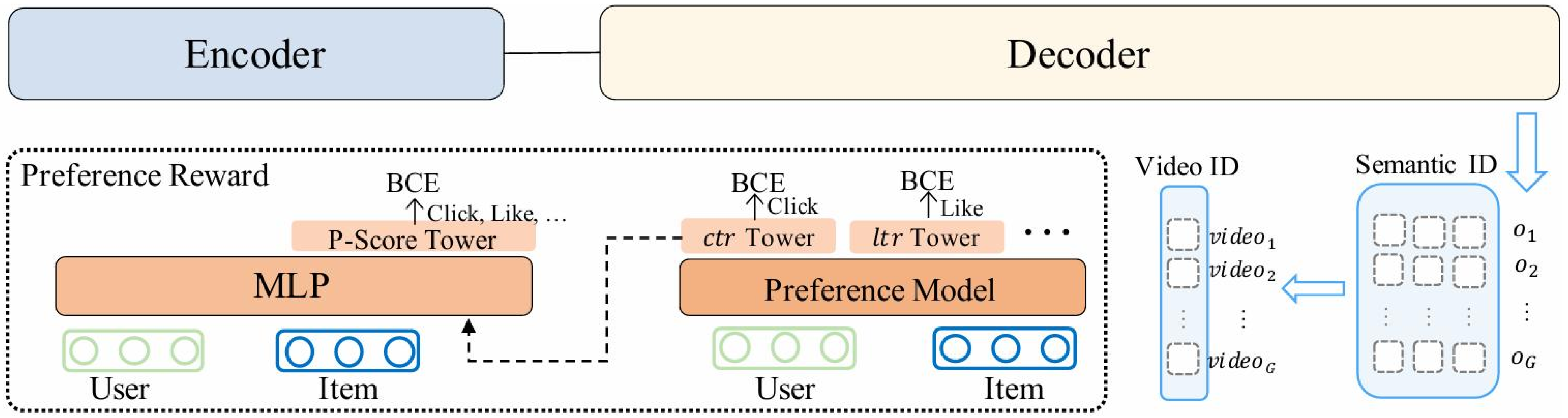
- **P-Score,** a neural network to **learn a personalized fusion socre**
  - **Multiple towers**: to learn specific objectives (BCE Loss)

=> fed into the final MLP -> P-Score Tower -> P-Score:

$$\mathcal{L}_{\text{P-Score}} = \sum_{xtr \in S_o} w^{\text{xtr}} \mathcal{L}^{\text{xtr}}_{\text{P-Score}}$$

$$\mathcal{L}^{\text{xtr}}_{\text{P-Score}} = -(y^{\text{xtr}} \log p + (1 - y^{\text{xtr}}) \log (1 - p)),$$

$$S_o = \{\text{ctr}, \text{lvtr}, \text{ltr}, \text{vtr}, \ldots\}$$



» Architecture (generative)

Reward System => User Preference Alignment

- **Early Clipped GRPO:** use the P-Score to align user preferences
  - G items: generated by the old policy model; r_i: P-Score of item i

$$\mathcal{J}_{ECPO}(\theta) = \mathbb{E}_{u\sim P(U),\{o_i\}_{i=1}^{G}\sim\pi_{\theta_{old}}}\left[\frac{1}{G}\sum_{i=1}^{G}\min\left(\frac{\pi_\theta(o_i|u)}{\pi'_{\theta_{old}}(o_i|u)}A_i, \text{clip}\left(\frac{\pi_\theta(o_i|u)}{\pi'_{\theta_{old}}(o_i|u)}, 1-\epsilon, 1+\epsilon\right)A_i\right)\right]$$

$$A_i = \frac{r_i - \text{mean}(\{r_1, r_2, \ldots, r_G\})}{\text{std}(\{r_1, r_2, \ldots, r_G\})}$$

$$\boxed{\pi'_{\theta_{old}}(o_i|u) = \max\left(\frac{\text{sg}(\pi_\theta(o_i|u))}{1+\epsilon+\delta}, \pi_{\theta_{old}}(o_i|u)\right), \quad \delta > 0}$$

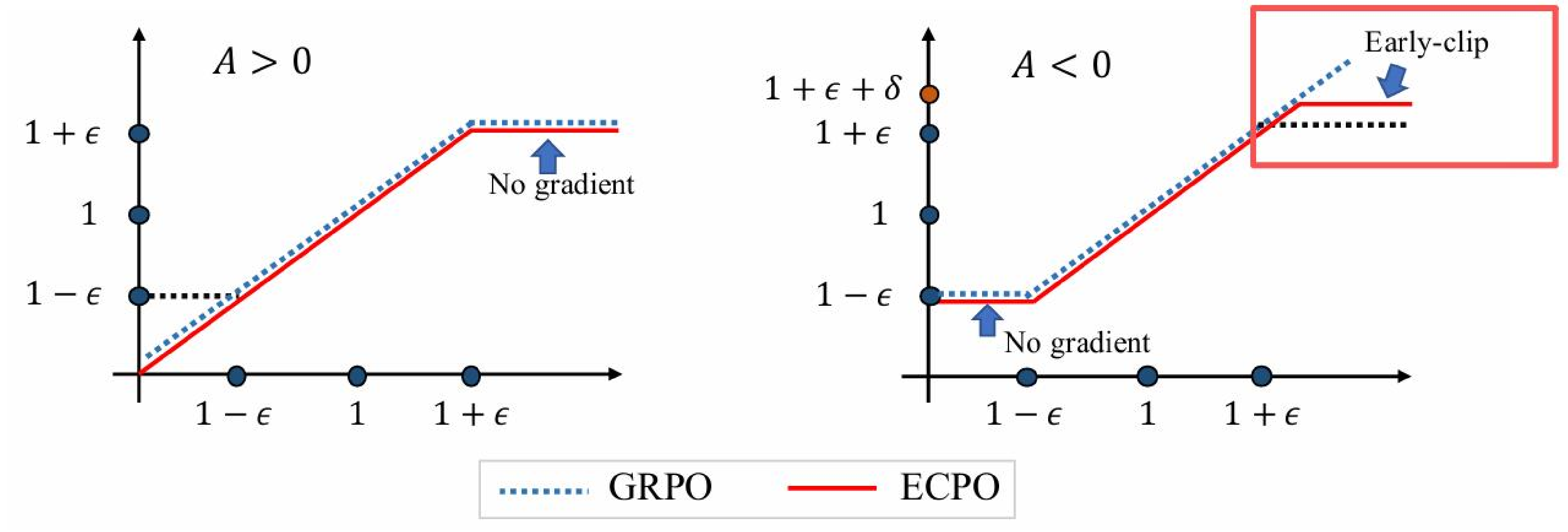


Figure 6 | Illustration of ECPO. The x-axis is $\pi_\theta/\pi_{\theta_{old}}$ and the y-axis is the clipped $\pi_\theta/\pi_{\theta_{old}}$. Items with $A > 0$ are processed in the same way as the original GRPO, while items with $A < 0$ are constrained by early-clipping to limit the maximum ratio.

» Architecture (generative)

Reward System => Generation Format Regularization

- **ECPO significantly increases the generation of illegal outputs**

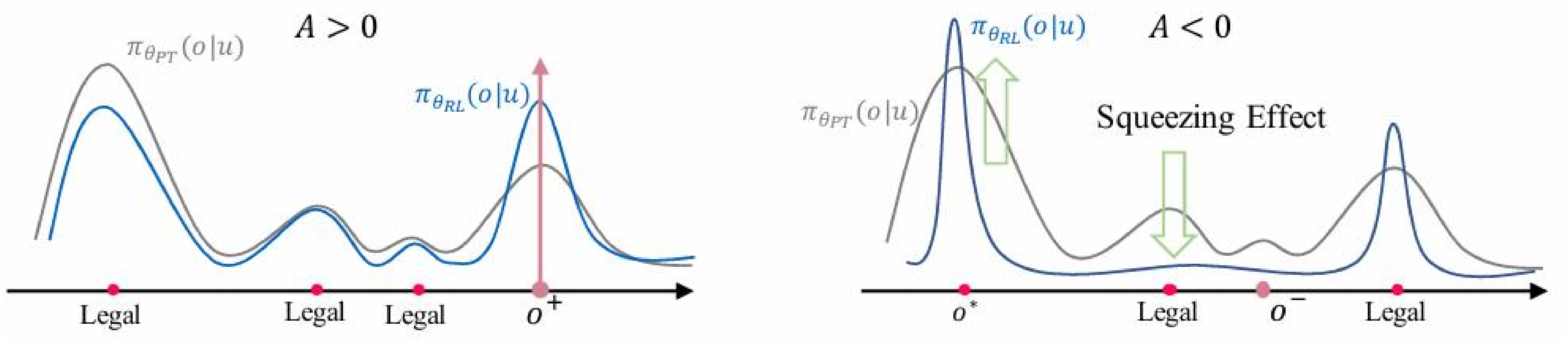


Figure 7 | Illustration of squeezing effect. $\pi_{\theta_{PT}}$ represents the pre-trained model, while $\pi_{\theta_{RL}}$ represents the model trained with ECPO. o^+ refers to videos with positive advantages, while o^- refers to those with negative advantages.

- **Generation Format Regularization**
  - K samples from the G samples & $A_i = \begin{cases} 1 & \text{if } o_i \in I_{\text{legal}} \\ 0 & \text{if } o_i \notin I_{\text{legal}} \end{cases}$

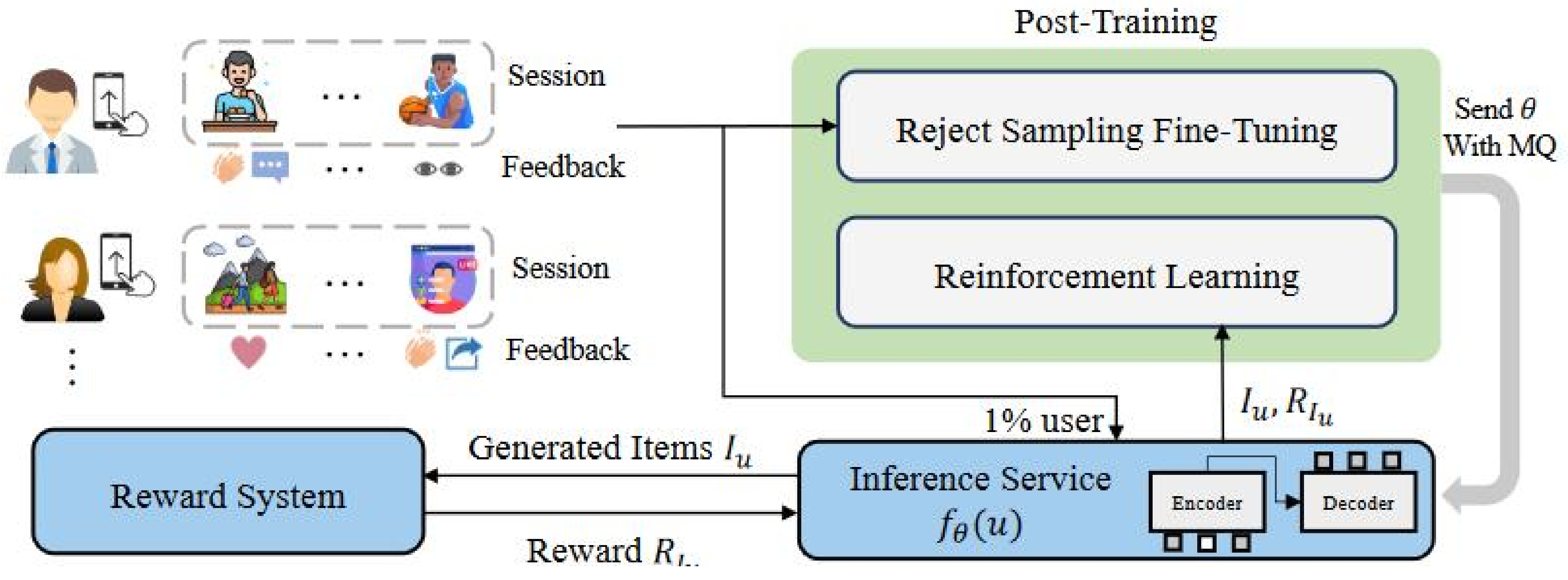
» Architecture (generative)

Recommendation System: Cascaded => End-to-End (integrating retrieval and ranking)

- **Improve the Model FLOPs Utilization (MFU)**
  - **Training:** 4.6% -> 23.7%; **Inference:** 11.2% -> 28.8%
- **Training Framework**
  - **Concrete Model Architecture:**
  - **Pre-Training:** next token prediction
  - **Post-Training**
    - **RSFT (continual next token prediction)**
      - filter out the bottom 50% of exposure sessions
    - **RL**
      - randomly select 1% of users from the RSFT

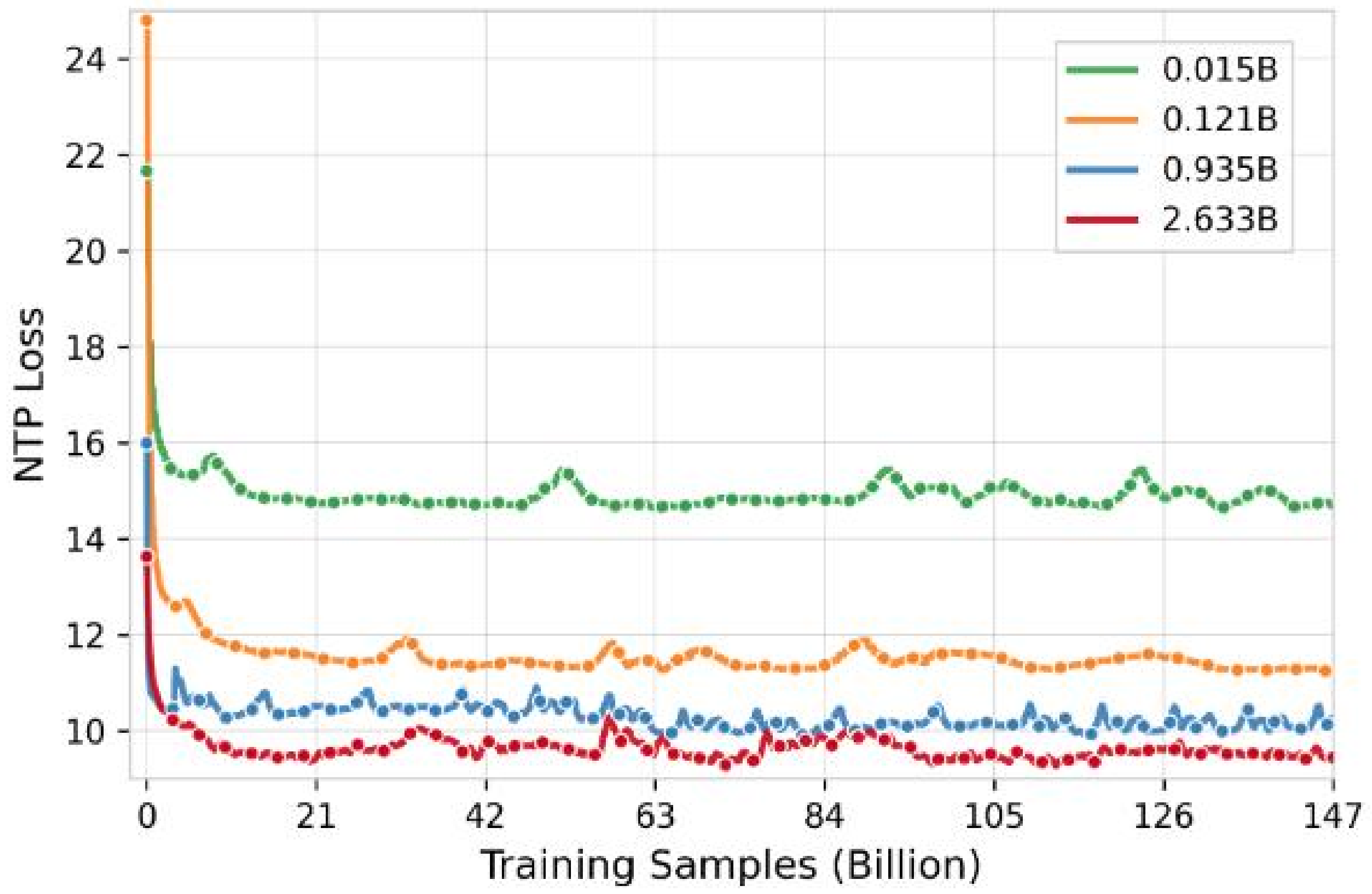
Table 1 | OneRec model architectures. "Layers" = #Encoder + #Decoder. "FFN Hid. Dim" is FFNs' intermediate size or MoEs' intermediate expert size.

Model	Layers	Hid. Dim	FFN Hid. Dim	Attn. Heads	Experts (Tot/Act)	MoE Loc.
OneRec-0.015B (Dense)	4	128	256	4	N/A	N/A
OneRec-0.121B (Dense)	8	1024	2048	8	N/A	N/A
OneRec-0.935B (MoE)	8	1024	2048	8	24 / 2	Decoder
OneRec-2.633B (MoE)	24	1024	2048	8	24 / 4	Enc & Dec



» Scaling Experiments

- **Parameters Scaling**



- **Codebook Scaling**

Metric	Size=8K	Size=32K	Impr.
lvtr	0.5118	0.5245	2.48%
vtr	0.9384	0.9491	1.14%
ltr	0.0298	0.0299	0.34%
wtr	0.0153	0.0154	0.65%
cmtr	0.0650	0.0664	2.15%
P-score	0.2516	0.2635	4.75%

Table 3 | Codebook Scaling.

- **lvtr** (Long View Through Rate): Predicted probability of significant video viewing
- **vtr** (View Through Rate): Predicted probability of video viewing
- **ltr** (Like Through Rate): Predicted probability of video liking
- **wtr** (Follow Through Rate): Predicted probability of the creator following
- **cmtr** (Comment Through Rate): Predicted probability of video commenting

OneRec - V2

OneRec-V2 Technical Report

Guorui Zhou, Hengrui Hu, Hongtao Cheng, Huanjie Wang, Jiaxin Deng, Jinghao Zhang, Kuo Cai, Lejian Ren, Lu Ren, Liao Yu, Pengfei Zheng, Qiang Luo, Qianqian Wang, Qigen Hu, Rui Huang, Ruiming Tang, Shiyao Wang, Shujie Yang, Tao Wu, Wuchao Li, Xinchen Luo, Xingmei Wang, Yi Su, Yunfan Wu, Zexuan Cheng, Zhanyu Liu, Zixing Zhang, Bin Zhang, Boxuan Wang, Chaoyi Ma, Chengru Song, Chenhui Wang, Chenglong Chu, Di Wang, Dongxue Meng, Dunju Zang, Fan Yang, Fangyu Zhang, Feng Jiang, Fuxing Zhang, Gang Wang, Guowang Zhang, Han Li, Honghui Bao, Hongyang Cao, Jiaming Huang, Jiapeng Chen, Jiaqiang Liu, Jinghui Jia, Kun Gai, Lantao Hu, Liang Zeng, Qiang Wang, Qidong Zhou, Rongzhou Zhang, Shengzhe Wang, Shihui He, Shuang Yang, Siyang Mao, Sui Huang, Tiantian He, Tingting Gao, Wei Yuan, Xiao Liang, Xiaoxiao Xu, Xugang Liu, Yan Wang, Yang Zhou, Yi Wang, Yiwu Liu, Yue Song, Yufei Zhang, Yunfeng Zhao, Zhixin Ling, Ziming Li

arxiv: 25.09

» Motivation

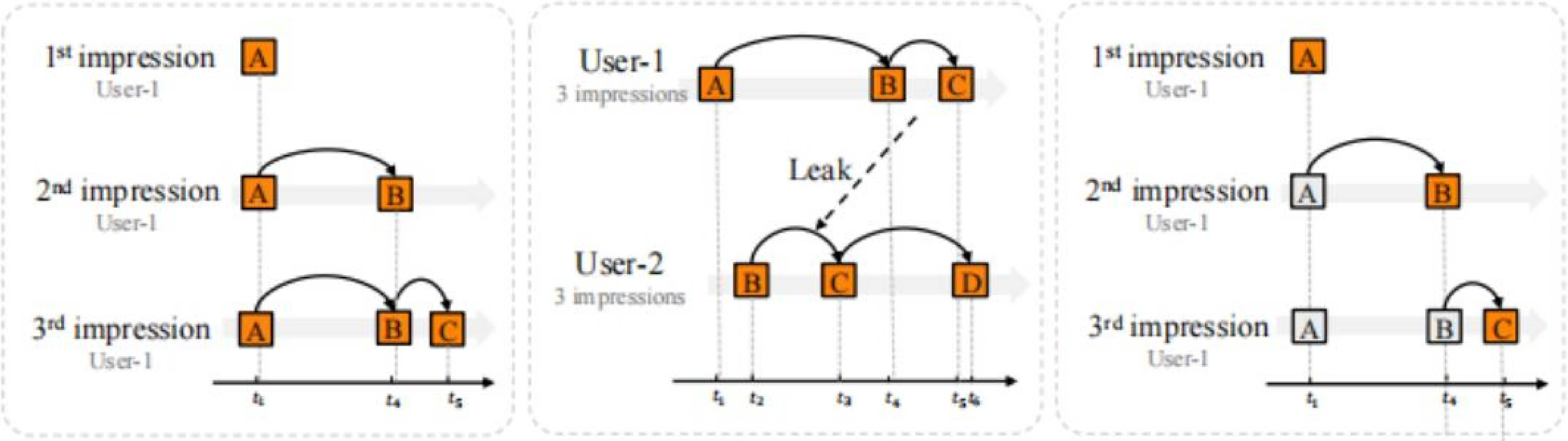
Two critical challenges hinder the scalability and performance of OneRec-V1

- **Inefficient computational allocation in encoder-decoder architecture**
  - 97.66% of resources are consumed by sequence encoding context encoding rather than generation
  - ➢ limits model scalability
- **Limitations in reinforcement learning that relies solely on reward models**
  - inefficient sampling and potential reward hacking due to proxy reward signals

» Lazy Decoder-Only Architecture

Design Principles

- **Data Organization**



(a) Naive Impression Organization (b) User-Centric Organization (c) New Impression Only Organization

Figure 3 | **Naive Impression Organization**: The pattern A→B is redundantly trained across multiple impressions. **User-Centric Organization**: When training on User-2's data at time t_3, the model has already learned the pattern B→C from User-1's future interactions at t_4. **New Impression Only Organization**: It trains only on the newest impression.

» Lazy Decoder-Only Architecture

Design Principles

- **Analysis of the computation details**
  - **Context Encoding**
    - context transformation operations in the encoder
    - context projection operations in the cross-attention of the decoder
  - **Target Decoding**

Context Length N	512	3000
Encoder-Decoder (0.5B:0.5B)		
Total Computation (GFLOPs)	346	1988
Context Encoding (GFLOPs)	338	1980
Target Decoding (GFLOPs)	8.1	8.1
Target Proportion	**2.34%**	**0.41%**
Naive Decoder-Only (1B)		
Total Computation (GFLOPs)	632	3618
Context Encoding (GFLOPs)	614	3600
Target Decoding (GFLOPs)	18	18
Target Proportion	**2.85%**	**0.49%**
Lazy Decoder-Only (1B)		
Total Computation (GFLOPs)	18	18
Target Proportion	**≈ 100%**	**≈ 100%**

» Lazy Decoder-Only Architecture

Overall Architecture

- **Context:** static conditioning information
  - Context = $[\boldsymbol{C}_0, \boldsymbol{C}_1, \ldots, \boldsymbol{C}_{S_{kv} \cdot L_{kv}-1}]$ $\quad \mathbf{C}_{S_{\text{kv}} \cdot L_{\text{kv}}-1} \in \mathbb{R}^{G_{\text{kv}} \cdot d_{\text{head}}}$

$$\mathbf{k}_l = \text{RMSNorm}_{k,l}(\mathbf{C}_{l \cdot S_{\text{kv}}}),$$

$$\mathbf{v}_l = \begin{cases} \text{RMSNorm}_{v,l}(\mathbf{C}_{l \cdot S_{\text{kv}}+1}), & \text{if } S_{\text{kv}} = 2 \text{ (separated key-value)} \\ \mathbf{k}_l, & \text{if } S_{\text{kv}} = 1 \text{ (shared representation)} \end{cases}$$

  - G_{kv}: the number of groups; L_{kv}: the numer of k-v layers
- **Lazy cross-attention mechanism:** w/o k-v projections
- **Grouped Query Attention (GQA)**

➢ **Context Processor =>** $(\mathbf{k}_0, \mathbf{v}_0), \ldots, (\mathbf{k}_{L_{\text{kv}}-1}, \mathbf{v}_{L_{\text{kv}}-1})\}$

  - **KV-Sharing**: block-wise layer-share

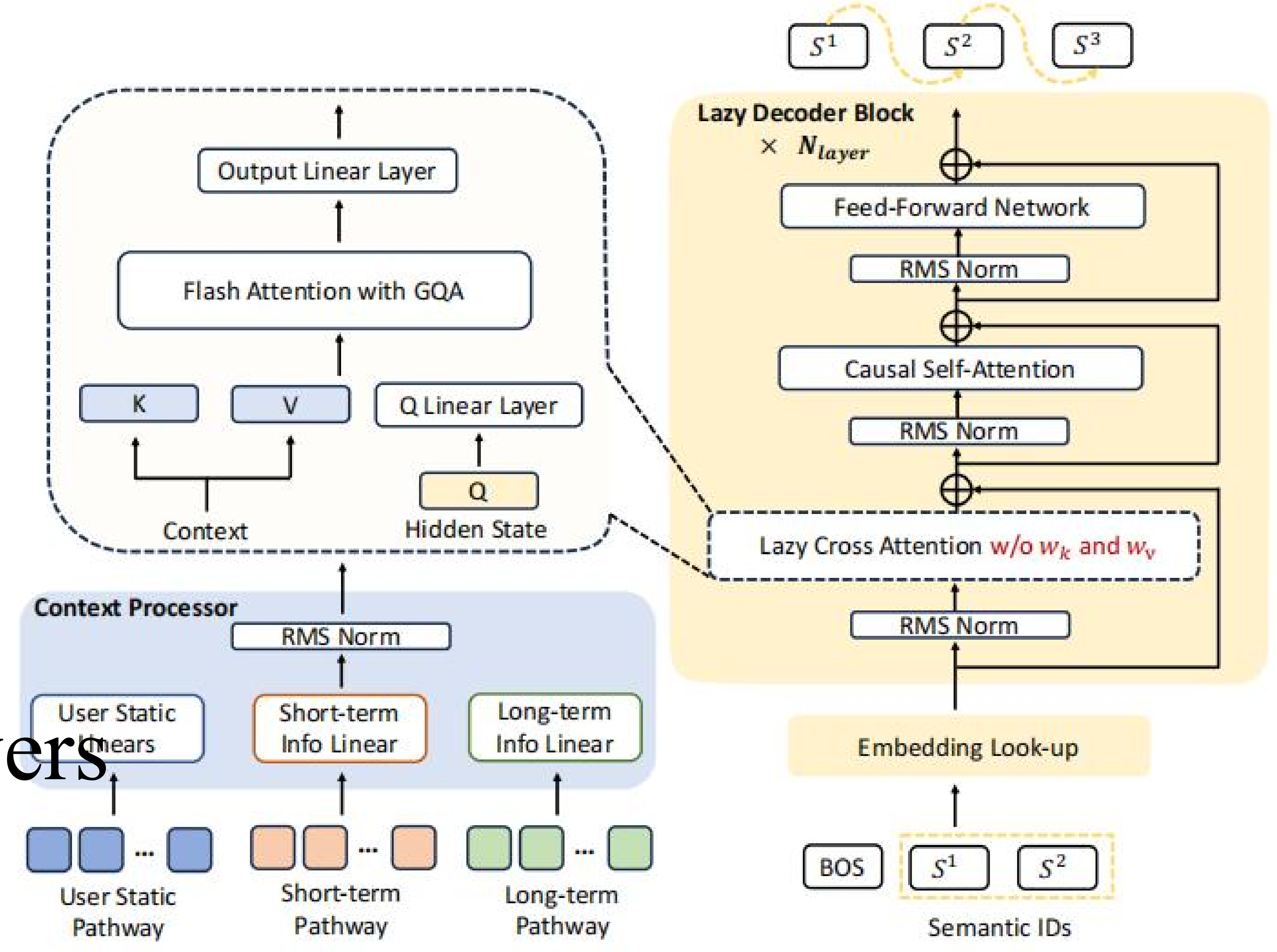


Figure 4 | Architecture of the proposed lazy decoder-only generative recommender. The Context Processor transforms heterogeneous user feature pathways into unified context representations, which are then normalized to produce layer-shared key-value pairs for cross-attention. The Lazy Decoder processes BOS token and tokenized semantic IDs of the target item through stacked transformer blocks. Each block comprises: (1) lazy cross-attention without key-value projections enabling Grouped Query Attention (GQA); (2) causal self-attention; and (3) a feed-forward network. The final representations are projected to predict semantic IDs for next-item recommendation.

Lazy Decoder-Only Architecture

Efficiency

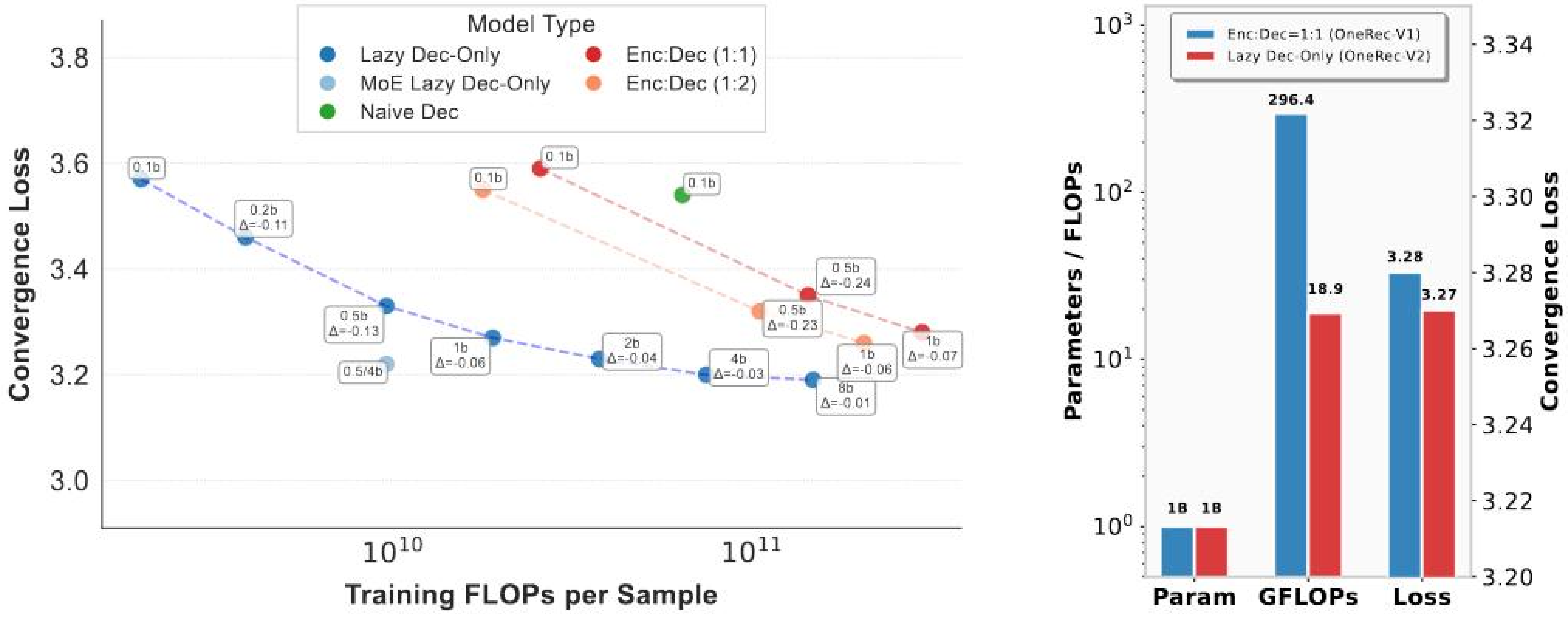


Figure 1 | Left: Scaling curves for various model architectures from 0.1B to 8B parameters, among which Lazy Decoder-only models demonstrate best scaling efficiency. Right: OneRec-V1 v.s. OneRec-V2 at 1B parameters.

Preference Alignment with Real-World User Interactions

Reinforcement Learning with User Feedback Signals

- **Duration-aware reward shaping**
  - Duration follows a long-tailed distribution

  => partition items into buckets with a **logarithmic** strategy

$$P_{u,b} = \{p_j \mid (d_j, p_j) \in H_u, \mathcal{F}(d_j) = b\}$$

bucket

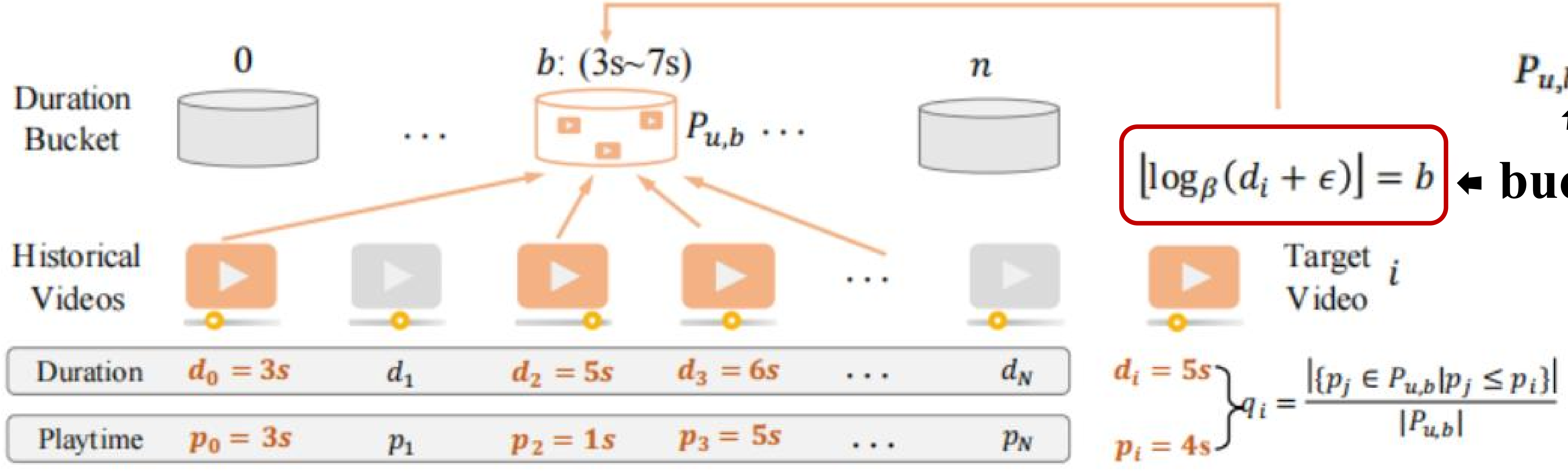


Figure 7 | Illustration of the Duration-Aware Reward Shaping. The videos in a user's watch history are bucketed according to the durations, and for a target video, the quantile of its playing time within the corresponding bucket is computed as the user's preference score.

Preference Alignment with Real-World User Interactions

Reinforcement Learning with User Feedback Signals

- **Duration-aware reward shaping**
  - **percentile rank** of p_i within the user's historical distribution

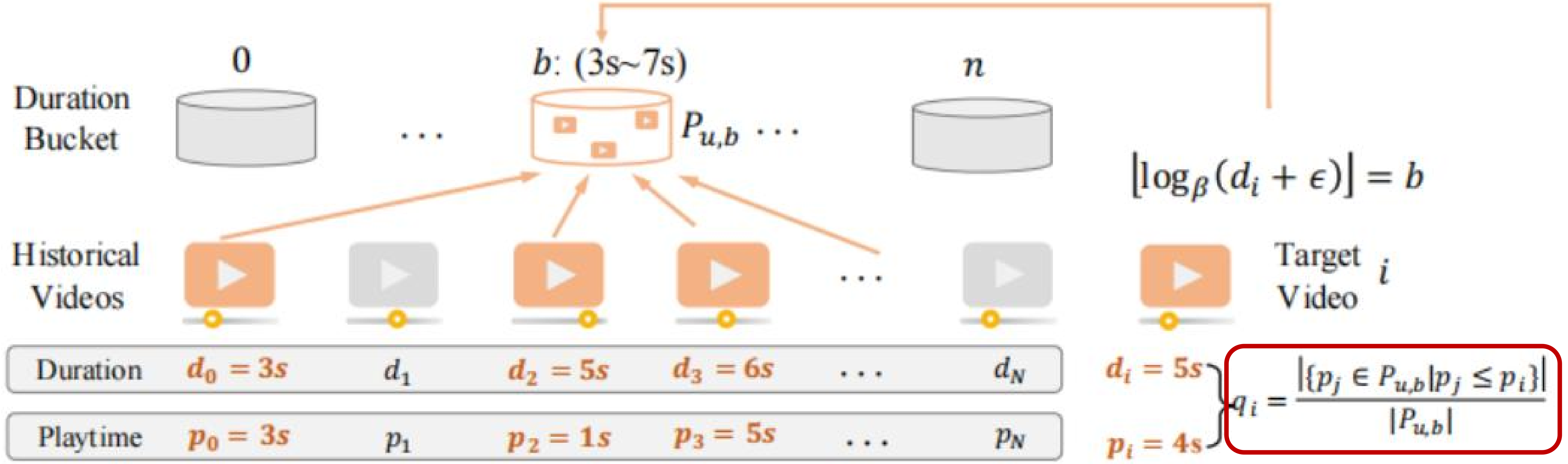


Figure 7 | Illustration of the Duration-Aware Reward Shaping. The videos in a user's watch history are bucketed according to the durations, and for a target video, the quantile of its playing time within the corresponding bucket is computed as the user's preference score.

Preference Alignment with Real-World User Interactions

Reinforcement Learning with User Feedback Signals

- **Duration-aware reward shaping**
  - neg_i: explicit negative feedback

$$A_i = \begin{cases} +1, & q_i > \tau_B \text{ and } neg_i = 0, \\ -1, & neg_i = 1, \\ 0, & \text{otherwise.} \end{cases}$$

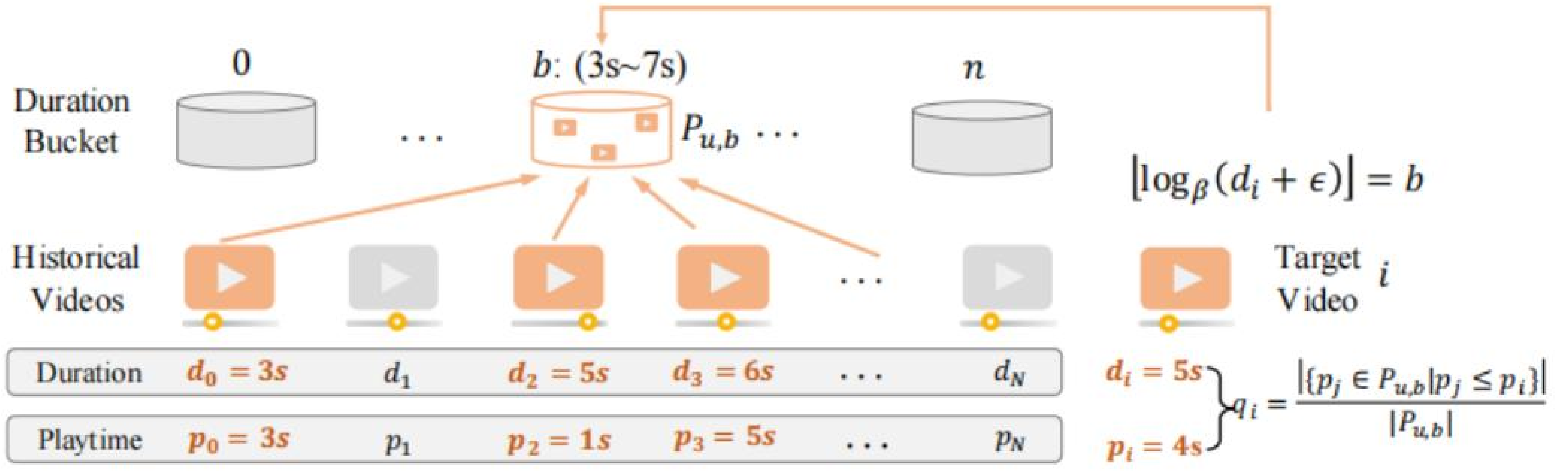


Figure 7 | Illustration of the Duration-Aware Reward Shaping. The videos in a user's watch history are bucketed according to the durations, and for a target video, the quantile of its playing time within the corresponding bucket is computed as the user's preference score.

Preference Alignment with Real-World User Interactions

Reinforcement Learning with User Feedback Signals

- **Issue of Early-Clipped GRPO (ECPO)**
  - **gradient explosion**, induced by negative samples
    - gradient analysis:

$$\mathcal{J}^i_{ECPO}(\theta) = -A_i \cdot \frac{\pi_\theta}{sg(\pi_\theta)}, \longrightarrow \frac{\partial \mathcal{J}^i_{ECPO}(\theta)}{\partial \theta} = -A_i \cdot \frac{1}{\pi_\theta \downarrow} \frac{\partial \pi_\theta}{\partial \theta} \uparrow \longrightarrow$$

overfitting or even collapse

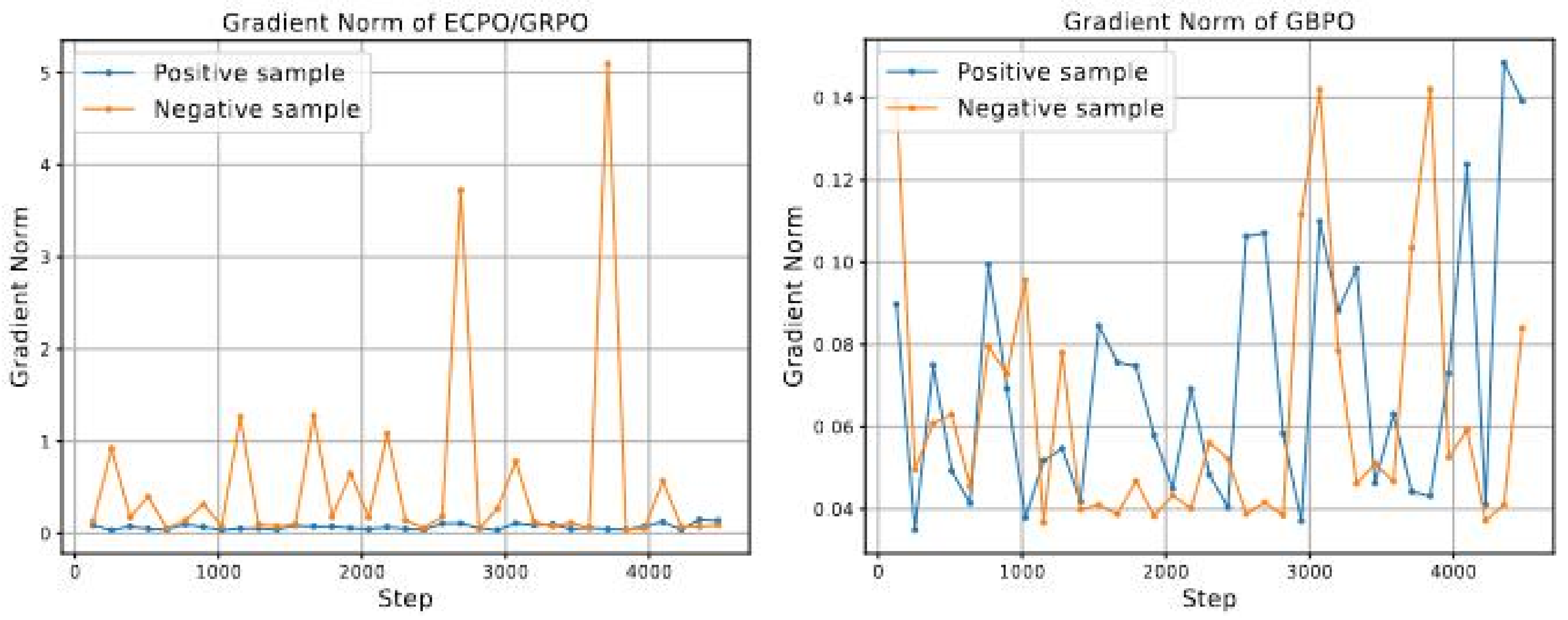


Figure 8 | Gradient comparison between GBPO and traditional ratio-clipping methods. In training of negative samples, GBPO exhibits significantly more stable gradients.

Reinforcement Learning with User Feedback Signals

- **Reinforcement Learning -> Gradient-Bounded Policy Optimization (GBPO)**

$$\mathcal{J}_{GBPO}(\theta) = -\mathbb{E}_{u\sim P(U),\{o_i\}_{i=1}^{G}\sim\pi_{\theta_{old}}}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{\pi_\theta(o_i|u)}{\pi'_{\theta_{old}}(o_i|u)}\cdot A_i\right],$$

- introduce a **dynamic bound on $\pi_{\theta_{old}}$**

$$\pi'_{\theta_{old}}(o_i|u) = \begin{cases}\max(\pi_{\theta_{old}}, sg(\pi_\theta)), & A_i \geq 0,\\ \max(\pi_{\theta_{old}}, 1 - sg(\pi_\theta)), & A_i < 0.\end{cases}$$

- **based on** the BCE Loss: $\mathcal{L}_{BCE}(y, p_\theta) = -[y\cdot\log(p_\theta) + (1-y)\cdot\log(1-p_\theta)],$

- remove the clipping operation on the ratio

$$\frac{\partial\mathcal{L}_{BCE}}{\partial\theta} = \begin{cases}-\dfrac{1}{p_\theta}\dfrac{\partial p_\theta}{\partial\theta}, & y = 1,\\ \dfrac{1}{1-p_\theta}\dfrac{\partial p_\theta}{\partial\theta}, & y = 0.\end{cases}$$

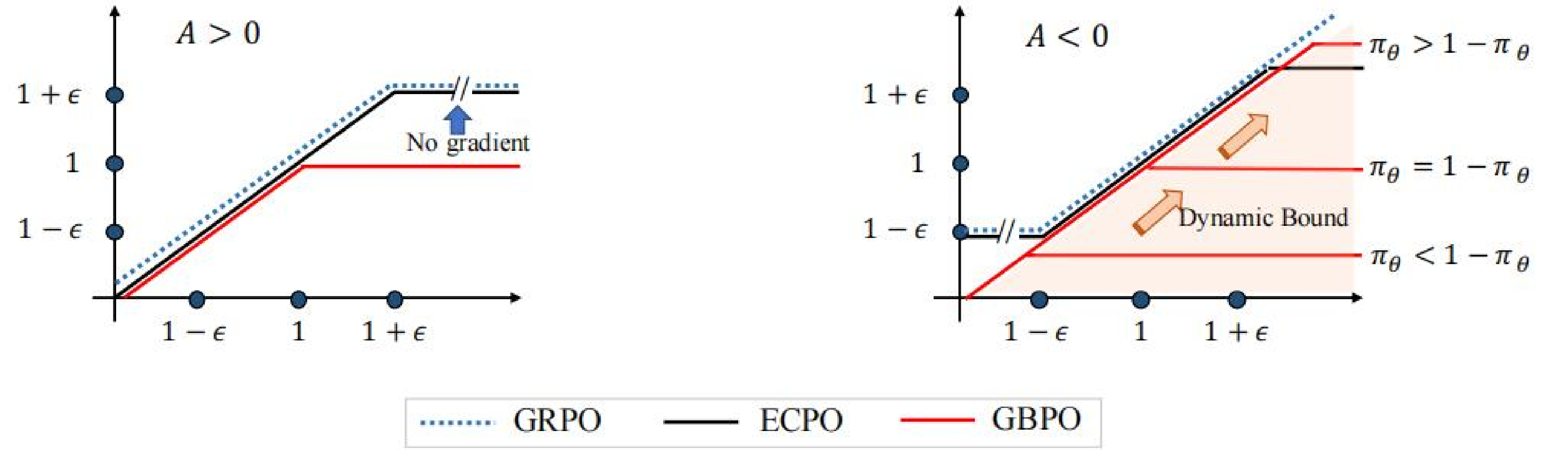


Figure 9 | Illustration of GBPO. The x-axis is $\pi_\theta/\pi_{\theta_{old}}$ and the y-axis is the clipped $\pi_\theta/\pi_{\theta_{old}}$. "//" means "No gradient". Compared with traditional ratio-clipping methods, the main differences of GBPO are: 1. It does not discard the gradients of any samples. 2. For negative samples, the bounding of the ratio is based on a dynamic bound related to π_θ.

中国科学院计算技术研究所
INSTITUTE OF COMPUTING TECHNOLOGY, CHINESE ACADEMY OF SCIENCES

OneRec - Think

OneRec-Think: In-Text Reasoning for Generative Recommendation

Zhanyu Liu, Shiyao Wang, Xingmei Wang, Rongzhou Zhang, Jiaxin Deng, Honghui Bao, Jinghao Zhang, Wuchao Li, Pengfei Zheng, Xiangyu Wu, Yifei Hu, Qigen Hu, Xinchen Luo, Lejian Ren, Zixing Zhang, Qianqian Wang, Kuo Cai, Yunfan Wu, Hongtao Cheng, Zexuan Cheng, Lu Ren, Huanjie Wang, Yi Su, Ruiming Tang, Kun Gai, Guorui Zhou

arxiv: 25.10

» Motivation

- Existing generative models (e.g., OneRec) operate as **implicit predictors**, critically **lacking** the capacity for **explicit and controllable reasoning**.

➢ **OneRec-Think**: a unified framework that seamlessly **integrates** dialogue, reasoning, and personalized recommendation.

Itemic Alignment through Multi-Task Pre-training

- **Definitions**
  - **Itemic token:** constitues semantic ID of the item v, $\boldsymbol{s}_v = (s_v^1, \ldots, s_v^L)$ => **expand** the vocabulary

➢ Unify reasoning and recommendation in a single autoregressive pass:

- given user's **interaction history sequence**, $S_u = (\boldsymbol{s}_{v_1}, \ldots, \boldsymbol{s}_{v_n})$

=> **generate reasoning sequence**, e.g, analysis of user interest, $\boldsymbol{\tau} = (r_1, \ldots, r_M)$

- **given S$_u$ and** τ => **generate next target item**

$$\boldsymbol{\tau} \sim P\left(\cdot \mid \mathcal{P}(\boldsymbol{s}_{v_1}, \ldots, \boldsymbol{s}_{v_n}); \theta\right)$$
$$\boldsymbol{s}_{v_{n+1}} \sim P\left(\cdot \mid \mathcal{P}(\boldsymbol{s}_{v_1}, \ldots, \boldsymbol{s}_{v_n}), \boldsymbol{\tau}; \theta\right)$$

Methodology

Itemic Alignment through Multi-Task Pre-training

- **Interleaved User Persona Grounding**
  - **Interleave** the itemic tokens and text tokens of User Persona.
  - ➢ **Next Token Prediction Loss**

Task 1: Interleaved User Persona Grounding

The user is a 25-30 year old software Engineer. He recently liked video <item_a_1123><item_b_5813><item_c_4212>, Captioned "Exploring the Andromeda Galaxy with the James Webb Telescope." and video <item_a_234><item_b_167><item_c_332>, captioned ...

» Methodology

Itemic Alignment through Multi-Task Pre-training

- **Sequential Preference Modeling**
  - To **predict subsequent item** from chronological user histories.
    - given a sequence of a user's recent interactions => predict the next item
  - ➢ **Next Token Prediction Loss,** computed on the tokens of the target itemic token

Sequential Preference Modeling data

```
<|im_start|>system
You are a sequential recommendation
engine. Your task is to analyze the
provided sequence of user-item
interactions and predict the single next
item the user is most likely to engage
with.
<|im_end|>
<|im_start|>user
User interaction history: <|item_begin|><
item_a_1024><item_b_2048><item_c_4096><|
item_end|>...<|item_begin|><item_a_1234><
item_b_5678><item_c_5876><|item_end|>.
Predict the next item.
<|im_end|>
<|im_start|>assistant
The next recommended item is <|item_begin
|><item_a_5555><item_b_6666><item_c_7777
><|item_end|>.
<|im_end|>
```

Itemic Alignment through Multi-Task Pre-training

- **Item Dense Captioning**
  - To understand the semantic characteristics represented by item combinations.
    - ask the model to generate a textual description of a given itemic token

➢ **Next Token Prediction Loss,** computed on the tokens of the target textual description

Itemic Dense Captioning data

```
<|im_start|>system
You are an expert content analyst. Given
an itemic token, generate a concise and
accurate textual description of its
content.
<|im_end|>
<|im_start|>user
Provide a description for the itemic
token: <|item_begin|><item_a_1357><
item_b_2468><item_c_7753><|item_end|>.
<|im_end|>
<|im_start|>assistant
This video showcases a local food stall,
highlighting traditional cooking methods
and signature dishes like grilled skewers
```

» Methodology

Itemic Alignment through Multi-Task Pre-training

- **General Language Modeling**
  - To **maintain** the base ability of the language model.
    - contain the pretraining and instruction fine-tuning data of the general corpus
  - ➢ **Next Token Prediction Loss**

Task Type	Data Percentage
Interleaved User Persona Grounding	24.30%
Sequential Preference Modeling	65.73%
Itemic Dense Captioning	4.94%
General Language Modeling	5.03%

- **Training Strategy**
  - **Token warm-up**: only train **itemic token embeddings** on the **Interleaved User Persona Grounding** task
  - **Multi-task integration**: jointly optimize **all parameters** on the **combined task** using a designed ratio

» Methodology

Reasoning Activation

- Due to the noisy and lengthy nature of real-world user behavior sequences, direct application to industrial recommendation scenarios often **fails** to yield effective CoT reasoning.

➢ **Supervised fine-tuning framework**

- **Bootstrapping with Pruned Contexts**
  - target item $s_{v_{n+1}}$, $S_u = (s_{v_1}, \ldots, s_{v_n}) \rightarrow$ retrieve **top-k most relevant items**:

$$g((s_{v_1}, \ldots, s_{v_n}), s_{v_{n+1}}) = (s_{w_1}, \ldots, s_{w_k})$$

  - query the pre-aligned model to **generate a rationale τ**, explaining the target item:

$$\tau \sim P\left(\cdot \mid \mathcal{P}_r((s_{w_1}, \ldots, s_{w_k}), s_{v_{n+1}}); \theta\right)$$

- **Learning to Reason from Noisy Sequences**

Methodology

Reasoning Activation

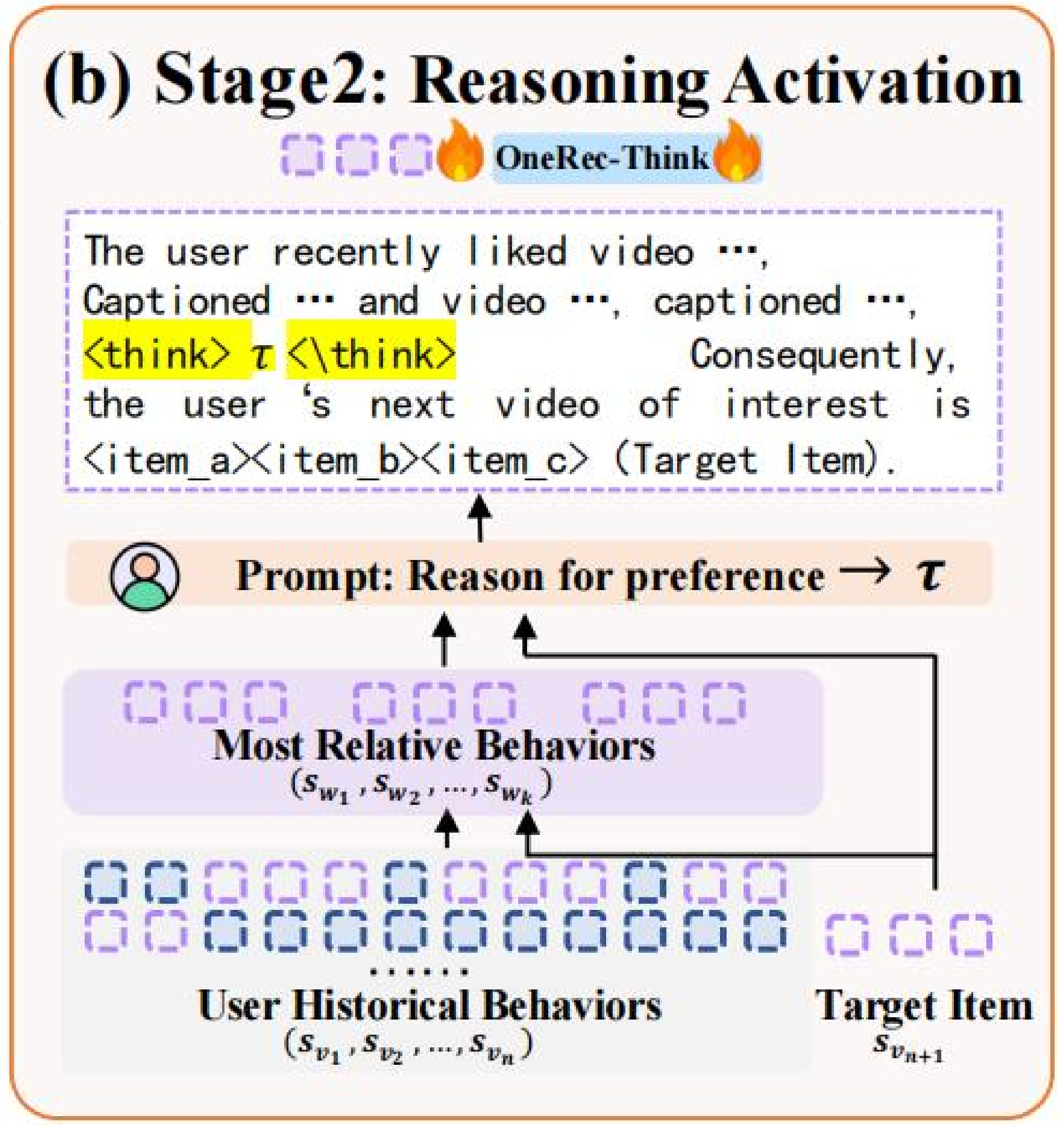
- Due to the noisy and lengthy nature of real-world user behavior sequences, direct application to industrial recommendation scenarios often **fails** to yield effective CoT reasoning.

➢ Supervised fine-tuning framework

- **Learning to Reason from Noisy Sequences**
  - The **rationales** serve as **supervision** for learning to reason from **raw sequences**.

$$\mathcal{L}_{\mathrm{RA}} = -\left(\sum_{i=1}^{M} \log P(r_i|\mathcal{P}(s_{v_1},\ldots,s_{v_n}), r_{<i};\theta) + \sum_{j=1}^{L} \log P(s_{v_{n+1}}^{j}|\mathcal{P}(s_{v_1},\ldots,s_{v_n}), \tau, s_{v_{n+1}}^{<j};\theta)\right)$$

where $\tau = \{r_1,\ldots,r_M\}$ represents the rationale tokens and $s_{v_{n+1}} = \{s_{v_{n+1}}^1,\ldots,s_{v_{n+1}}^L\}$ denotes the target item tokens.



» Methodology

Reasoning Enhancement

- To refine the recommendation accuracy using a novel reward mechanism.
- **Beam Candidate Reward Maximization**
  - Most reasoning rollouts **fail to hit the target item** and consequently yield identical zero rewards
  - **Beam search with width K** => items with **top-K** probability in the beam search
  - **Optimize** the model using GRPO based on:

$$\mathcal{R}_{\text{Rollout-Beam}} = \max_{\hat{s}_{v_{n+1}} \in \mathcal{B}} \sum_{l=1}^{L} \mathbb{I}(\hat{s}^{l}_{v_{n+1}} = s^{l}_{v_{n+1}})$$

Methodology

Industrial Deployment: A "Think-Ahead" Architecture

- Decouple the model's inference into two stages
  - **First Stage:** the full OneRec-Think model → reasoning path and the initial item-tokens
    - Sample **T** diverse reasoning paths: $\boldsymbol{\tau}^{(i)} \sim P(\cdot \mid H_u; \theta)$

➢ $\mathcal{A}_u^{(i)} = \text{BeamSearch}\Big(P(\hat{s}^1_{v_{n+1}}, \hat{s}^2_{v_{n+1}} \mid H_u, \boldsymbol{\tau}^{(i)}; \theta), m\Big)$, m candidate item prefixes

➢ personalized candidate **space**: $\mathcal{C}_u = \bigcup_{i=1}^{T} \mathcal{A}_u^{(i)}$, **T*m** candidate item prefixes

  - **Second Stage:** real-time updated OneRec model → produce the final itemic token

=> items with top-K probability

$$\hat{\boldsymbol{s}}_{v_{n+1}} = \arg\max_{\boldsymbol{s}_{v_{n+1}}} P_{h_{\text{online}}}\left(\boldsymbol{s}_{v_{n+1}} \,\middle|\, \boldsymbol{s}_{v_1}, \ldots, \boldsymbol{s}_{v_n}\right)$$

$$\text{s.t.} \quad (\hat{s}^1_{v_{n+1}}, \hat{s}^2_{v_{n+1}}) \in \mathcal{C}_u$$

Experiments

Experimental Settings

- **Datasets**: Beauty, Toys, and Sports from the popular Amazon review benchmark
- **Baselines**
  - **Classic sequential methods**: BERT4Rec, GRU4Rec, SASRec
  - **Generative Recommendation Models**: TIGER, HSTU, ReaRec
    - **ReaRec**: enhances user representations through **implicit multi-step reasoning**
- **Metrics**: Recall@K, NDCG@K, K=5, 10
- **Backbone model:** Qwen3-1.7B => **Industrial Settings:** Qwen-8B
- **Four-level hierarchical**, 256 tokens per level => **Industrial Settings:** 8192

Overall Performance

Table 1: Overall performance comparison between the baselines and OneRec-Think on three datasets. The bold results highlight the best results, while the second-best ones are underlined.

Dataset	Method	BERT4Rec	HGN	GRU4Rec	SASRec	TIGER	HSTU	ReaRec	OneRec-Think
Beauty	R@5	0.0232	0.0319	0.0395	0.0402	0.0405	0.0424	<u>0.0450</u>	**0.0563**
	R@10	0.0396	0.0536	0.0584	0.0607	0.0623	0.0652	<u>0.0704</u>	**0.0791**
	N@5	0.0146	0.0196	0.0265	0.0254	0.0267	<u>0.0280</u>	0.0262	**0.0398**
	N@10	0.0199	0.0266	0.0326	0.0320	0.0337	<u>0.0353</u>	0.0344	**0.0471**
Sports	R@5	0.0102	0.0183	0.0190	0.0199	0.0215	<u>0.0268</u>	0.0214	**0.0288**
	R@10	0.0175	0.0313	0.0312	0.0301	<u>0.0347</u>	0.0343	0.0332	**0.0412**
	N@5	0.0065	0.0109	0.0122	0.0106	0.0137	<u>0.0173</u>	0.0116	**0.0199**
	N@10	0.0088	0.0150	0.0161	0.0141	0.0179	<u>0.0226</u>	0.0154	**0.0239**
Toys	R@5	0.0215	0.0326	0.0330	0.0448	0.0337	0.0366	<u>0.0523</u>	**0.0579**
	R@10	0.0332	0.0517	0.0490	0.0626	0.0547	0.0566	<u>0.0764</u>	**0.0797**
	N@5	0.0131	0.0192	0.0228	<u>0.0300</u>	0.0209	0.0245	0.0298	**0.0412**
	N@10	0.0168	0.0254	0.0279	0.0358	0.0276	0.0309	<u>0.0376</u>	**0.0482**

» Experiments

Ablation Study

- **Base:** tuned by the raw itemic token sequence
- **IA:** itemic alignment
- **R:** enhanced reasoning mechanism

Table 2: Ablation Study of different variants of OneRec-Think on Beauty dataset.

Training Method	R@5	R@10	N@5	N@10
Base	0.0460	0.0654	0.0314	0.0377
Base+IA	0.0532	0.0735	0.0342	0.0402
Base+IA+R	**0.0563**	**0.0791**	**0.0398**	**0.0471**

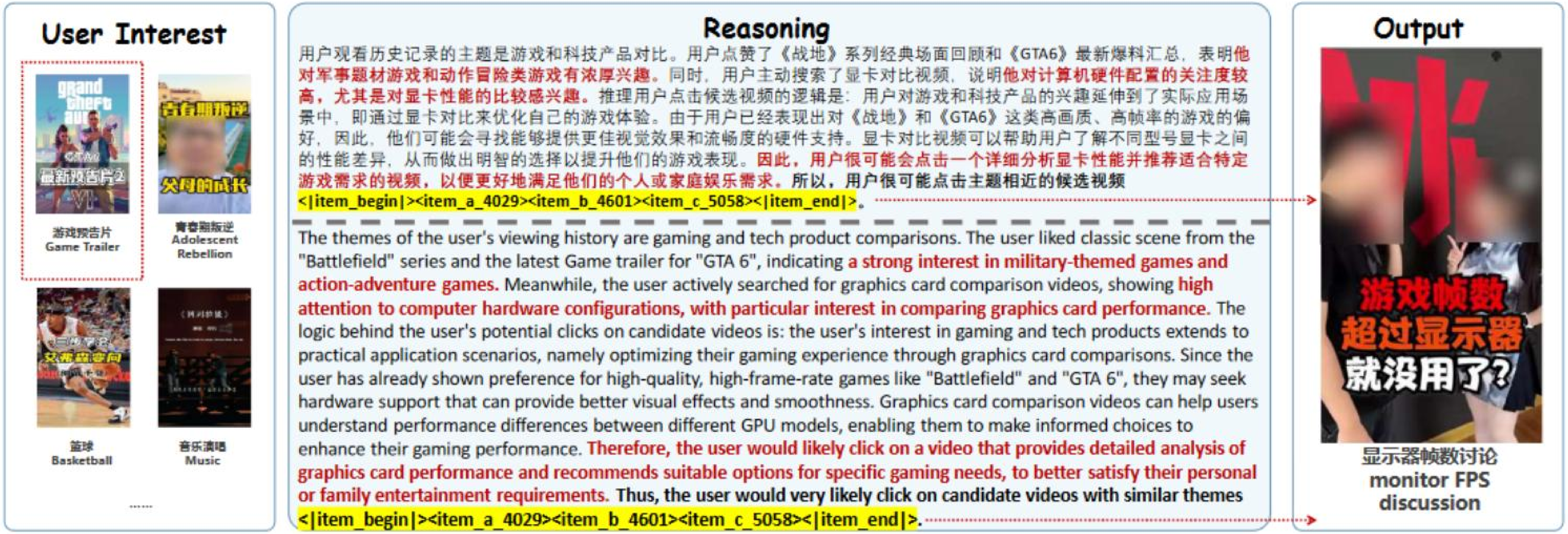
» Experiments

Case Study

- **Conservation Settings**

Case Study

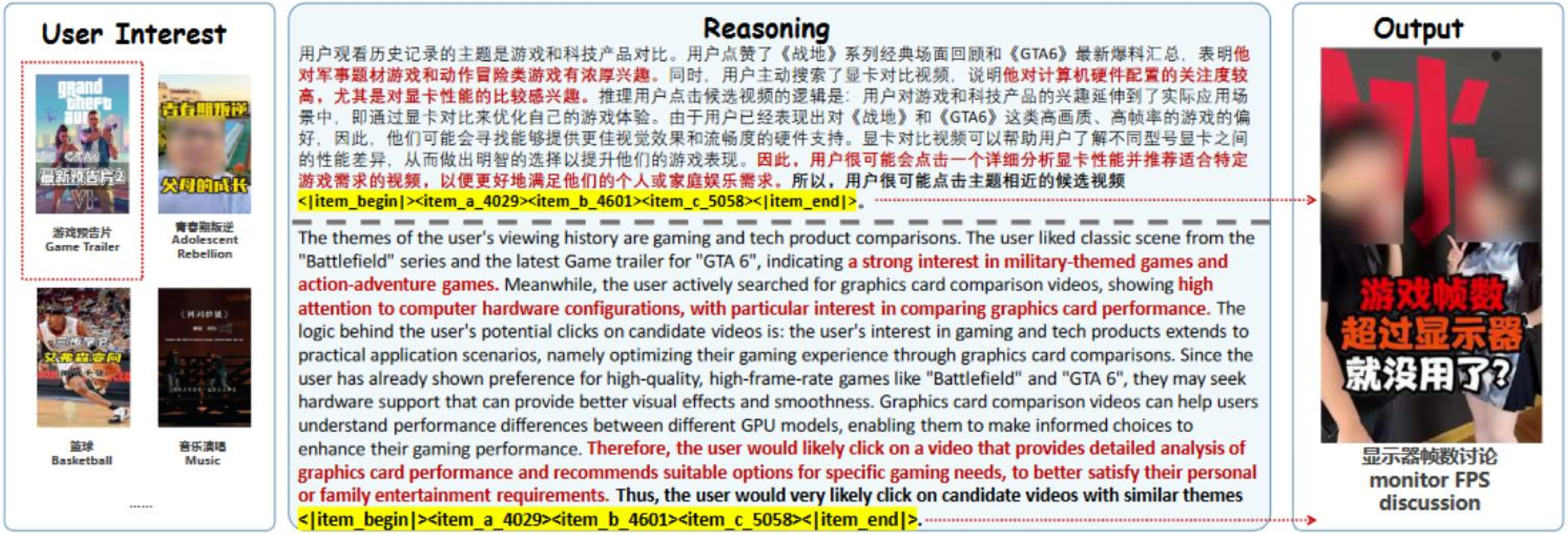
- **Reasoning Settings**



» Experiments

Case Study

- **Reasoning Settings**



感谢大家耐心倾听！