RAGR: Review-Augmented Generative Recommendation

Yingyi Zhang, *Graduate Student Member, IEEE*, Junyi Li, Yejing Wang, Wenlin Zhang, Xiaowei Qian, Sheng Zhang, Yue Feng, Yichao Wang*, Yong Liu, Xiangyu Zhao*, *Member, IEEE*, and Xianneng Li*, *Member, IEEE*

***Abstract*—Sequential recommendation (SR) is traditionally formulated as next-item prediction over a chronological sequence of interacted items. Although recent generative recommendation (GR) methods introduce new machinery, such as semantic IDs, autoregressive decoding, and unified token spaces, they largely inherit the same item-only modeling assumption. We argue that this design constitutes a structural bottleneck, because user decision-making is not purely behavioral: while item interactions reveal what users choose, review feedback often explain why they choose it by exposing latent evaluative factors.**

Motivated by this observation, we propose *R*eview-*A*ugmented *G*enerative *R*ecommendation (RAGR), a novel GR framework that incorporates review feedback directly into the generative user sequence rather than treating reviews as auxiliary side information. Specifically, RAGR introduces a *Review-Augmented User Sequence Modeling* mechanism that interleaves item semantic IDs and review semantic IDs in chronological order to construct a mixed behavioral-semantic sequence, enabling review signals to participate directly in autoregressive next-token generation. To preserve the recommendation objective, we further introduce an *Item-Centric Task Generation Alignment* strategy based on direct preference optimization (DPO), which encourages the model to favor item tokens over review tokens at prediction positions. Experiments on three real-world datasets show that RAGR yields consistent and significant gains over strong GR backbones across all metrics. Our code and data are available at https://github.com/Zhang-Yingyi/TKDE_RAGR.

***Index Terms*—Generative Recommendation, Review-Augmented Recommendation, and Sequential Recommendation.**

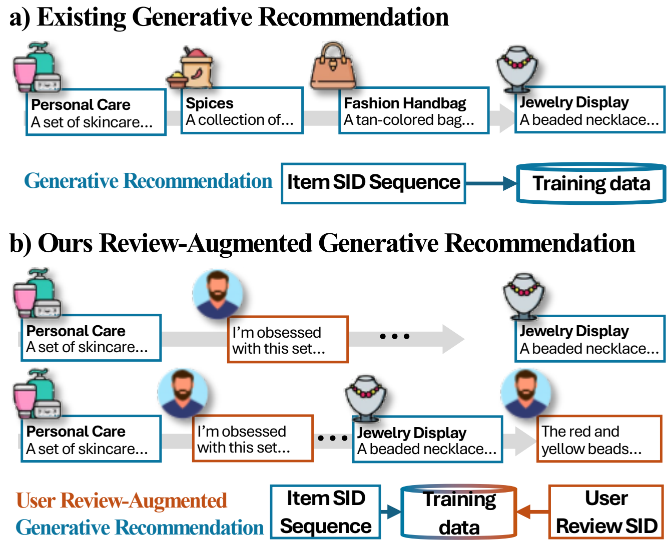


Fig. 1: Comparison Between Existing and Review-Augmented Generative Recommendation.

I. Introduction

RECOMMENDER systems are indispensable to modern e-commerce platforms [1], such as Amazon [2] and Alibaba [3], where they surface relevant items from catalogs of millions. Among various paradigms, sequential recommendation (**SR**) [4], [5] stands out because user preferences are inherently *dynamic*, shaped not only by static profiles but by the evolving temporal patterns of historical interactions [6], [7]. SR is predominantly cast as a *next-item prediction* task—given a chronological sequence of past interactions, the model forecasts the item a user will engage with next [8], [9]. This task formulation is tightly coupled with the representation mechanism of conventional models: architectures such as SASRec [4] and BERT4Rec [10] maintain an *item ID embedding table*, under which a user's history is encoded as an ID sequence and the learning objective *collapses* to predicting the next ID.

With the emergence of generative recommendation (**GR**) [11], [12], new machinery has been introduced—semantic ID (SID) [11], [13], [14], autoregressive decoding, unified token spaces—yet the underlying modeling assumption has been carried over almost unchanged: *the user sequence is still an item-only trajectory, and the objective is still to generate the next item* as shown in Fig. 1 a). We argue that this inherited assumption is not merely a missed opportunity but a *structural bottleneck*. In reality, user decision-making is inherently multi-faceted: purchases are often preceded by browsing and comparison, and followed by explicit feedback such as textual reviews [15], [16]. As McAuley and Leskovec [17] show, reviews expose *hidden evaluative dimensions*—such as quality, usability, and aesthetics—that shape a user's rating but remain invisible in interaction logs alone. An item-only sequence captures only the *outcome* of a decision—*what* the user chose—while omitting the explanatory signals that reveal *why* it was chosen. Without such signals, the model must infer preference

Yinyi Zhang is with the School of Economics and Management, Dalian University of Technology, China, and also with the Department of Data Science, City University of Hong Kong. E-mail: yingyizhang@mail.dlut.edu.cn.
Yue Feng is with the School of Economics and Management, Dalian University of Technology, China. E-mail: fy_0403@mail.dlut.edu.cn.
Xianneng Li is with the School of Economics and Management, Dalian University of Technology, China, and the National Key Laboratory of Maritime Decision Intelligence, China. E-mail: xianneng@dlut.edu.cn.
Yejing Wang, Junyi Li, Wenlin Zhang, Xiaowei Qian, Sheng Zhang and Xiangyu Zhao are with the Department of Data Science, City University of Hong Kong. E-mail: yejing.wang@my.cityu.edu.hk, wl.z@my.cityu.edu.hk, junyili@cityu.edu.hk, xiaowqian2-c@my.cityu.edu.hk, szhang844-c@my.cityu.edu.hk, xianzhao@cityu.edu.hk.
Yichao Wang and Yong Liu are with Huawei Technologies Ltd. E-mail: wangyichao5@huawei.com, liu.yong6@huawei.com.
*Corresponding author: Yichao Wang, Xiangyu Zhao, and Xianneng Li

dynamics solely from item co-occurrence patterns, resulting in a shallow and brittle understanding of user intent.

Notably, the generative framework already possesses the machinery needed to transcend this bottleneck. A distinguishing property of GR is its *unified tokenizer*, which projects item text content into a shared token space [12], [18], [19]. In current practice, however, this tokenizer is applied almost exclusively to items—mapping each item to SID [11], [13], [14] for autoregressive modeling—while the same mechanism could equally operate on other textual signals that users routinely produce, most notably post-interaction reviews. Put differently, existing research has substantially advanced the *expressiveness of item representation* yet left the *expressiveness of the user sequence* largely untouched. This asymmetry gives rise to our central research question: ***Can the unified tokenizer of GR be repurposed to encode user review feedback as semantic tokens and weave them into the interaction sequence, thereby enabling the model to capture not only behavioral outcomes but also the underlying preference rationale?*** Addressing this question entails two intertwined challenges: *(1) Heterogeneous sequence construction*—items and reviews differ in granularity and functional role; projecting both through a shared tokenizer into a coherent sequence requires careful design to avoid introducing noise. *(2) Recommendation-objective preservation*—once review tokens enter the generative sequence, a principled alignment mechanism is needed to ensure that they serve as supporting evidence for next-item prediction rather than a competing generation target.

To address these challenges, we propose ***R**eview-**A**ugmented **G**enerative **R**ecommendation* (**RAGR**), as illustrated in Fig. 1b. The central idea of RAGR is to move beyond the conventional treatment of reviews as auxiliary features in discriminative recommenders. Instead, we bring review feedback directly into the GR paradigm by encoding reviews as tokens within the user sequence itself—extending the item-only behavioral trajectory into a review-augmented one. Specifically, RAGR consists of two complementary components. **(i) Review-Augmented User Sequence Modeling.** Item SIDs and review SIDs are interleaved chronologically to form a mixed-signal user sequence, over which the GR backbone is trained to predict the next SID—allowing review context to directly participate in the autoregressive generation process. **(ii) Item-Centric Task Generation Alignment.** Because the mixed sequence may cause the model to devote generation capacity to reviews rather than items, we apply a direct preference optimization (DPO) [20] alignment mechanism that teaches the model to *prefer* generating item SIDs over review SIDs at prediction positions, ensuring that the learning objective remains anchored to next-item recommendation. Our main contributions are summarized as follows:

- We identify a structural limitation of existing GR methods—their confinement to item-only sequences—and propose RAGR, to our knowledge the first framework to elevate reviews from auxiliary features in discriminative recommenders to tokens within the GR paradigm, extending user sequences from purely behavioral to mixed behavioral-semantic.
- We design a *Review-Augmented User Sequence Modeling* mechanism that interleaves item and review SIDs chronologically, enabling review context to participate directly in the autoregressive generation process and enrich the preference signal available at each prediction step.
- We propose an *Item-Centric Task Generation Alignment* strategy based on DPO, which steers the model to prefer item SIDs over review SIDs at prediction positions, ensuring that reviews inform—but never displace—the next-item recommendation objective.
- We conduct extensive experiments on three real-world datasets, demonstrating that RAGR yields consistent and significant gains when applied to multiple GR backbones. Ablation studies further confirm that review augmentation and task alignment are both indispensable.

The remainder of this paper is structured as follows. Section II defines the review-augmented generative recommendation problem. Section III introduces the proposed approach, which is evaluated in Section IV. Then, Section V summarizes the recent development of sequential recommendation and generative recommendation. Finally, conclusions are drawn in Section VI.

II. Problem Formulation

Let $\mathcal{U}$ and $\mathcal{I}$ denote the user set and item set, respectively. For each user $u \in \mathcal{U}$, we observe a chronological interaction history

$$\mathcal{S}_u = \big[(i_1, r_1), (i_2, r_2), \dots, (i_T, r_T)\big], \tag{1}$$

where $i_t \in \mathcal{I}$ denotes the interacted item at step t, and r_t denotes the associated textual feedback, such as a review. Different from conventional sequential recommendation, which models only the item sequence $(i_1, i_2, \dots, i_T)$, we consider a review-augmented user sequence in which item interactions and review feedback are jointly modeled. Given the historical sequence up to step $t-1$,

$$\mathcal{S}_u^{<t} = \big[(i_1, r_1), (i_2, r_2), \dots, (i_{t-1}, r_{t-1})\big], \tag{2}$$

our goal is to predict the next target item i_t. Formally, we aim to learn a generative recommendation model

$$f_\theta : \mathcal{S}_u^{<t} \mapsto i_t, \tag{3}$$

where θ denotes model parameters.

To support generative modeling, we represent both items and reviews in a unified token space. Let $\mathbf{z}(i_t)$ and $\mathbf{z}(r_t)$ denote the tokenized representations of item i_t and review r_t, respectively. Then the historical sequence can be rewritten as

$$\widetilde{\mathcal{S}}_u^{<t} = \big[\mathbf{z}(i_1), \mathbf{z}(r_1), \mathbf{z}(i_2), \mathbf{z}(r_2), \dots, \mathbf{z}(i_{t-1}), \mathbf{z}(r_{t-1})\big]. \tag{4}$$

Accordingly, the recommendation objective is to generate the tokenized representation of the next target item:

$$p_\theta\big(\mathbf{z}(i_t) \mid \widetilde{\mathcal{S}}_u^{<t}\big). \tag{5}$$

Based on this formulation, we construct the training set as

$$\mathcal{D} = \Big\{ \big(\widetilde{\mathcal{S}}_u^{<t}, \mathbf{z}(i_t), \mathbf{z}(r_t)\big) \,\Big|\, u \in \mathcal{U},\ 2 \le t \le T \Big\}, \tag{6}$$

where $\mathbf{z}(i_t)$ is the tokenized target item and $\mathbf{z}(r_t)$ is the corresponding review of the user. The learning problem in this work is therefore to perform next-item generation over

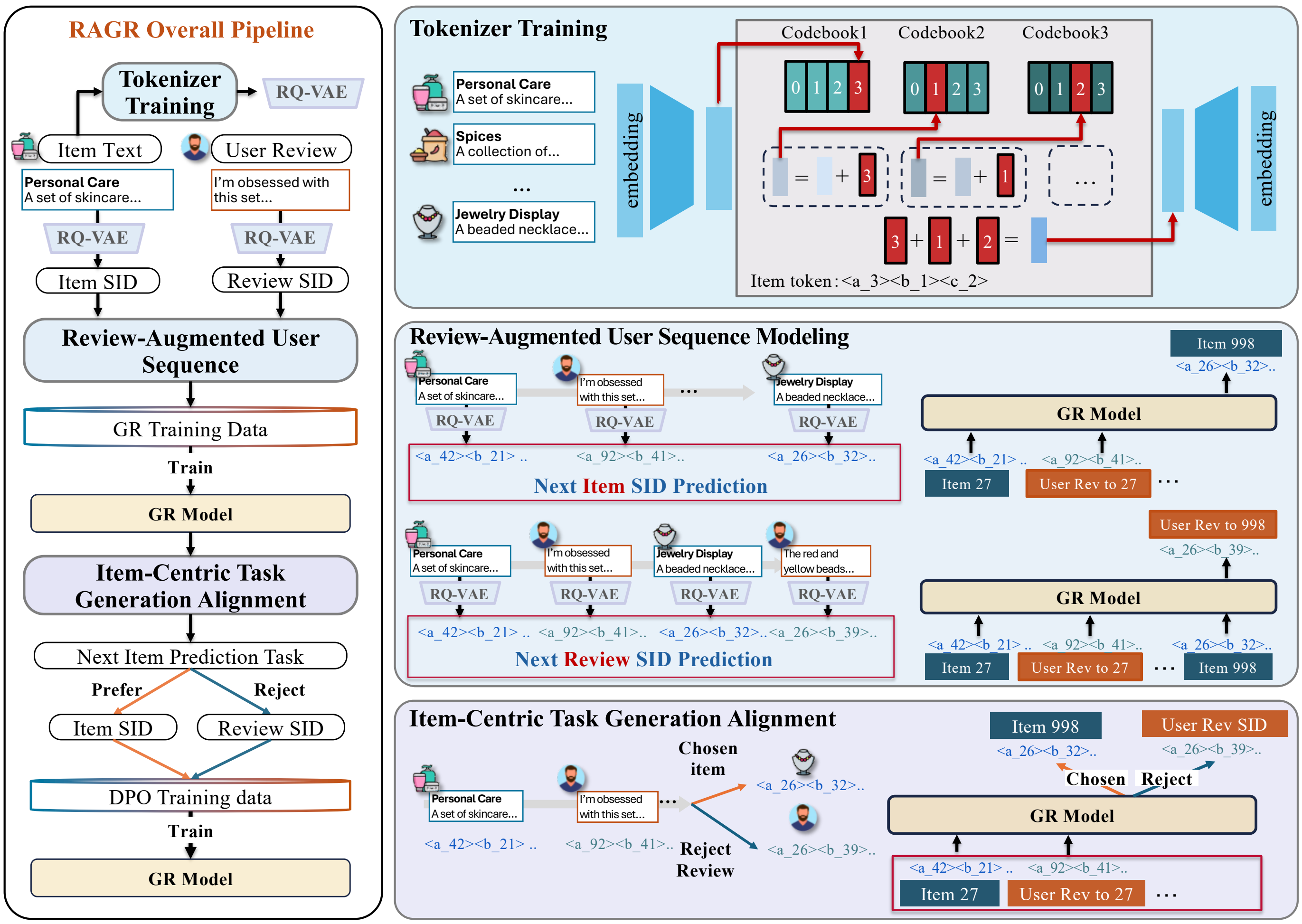


Fig. 2: The overall framework of the proposed **RAGR**, which consists of three main stages: Tokenizer Training, Review-Augmented User Sequence Modeling, and Item-Centric Task Generation Alignment.

review-augmented user sequences while preserving the item-centric recommendation objective. *However, incorporating review tokens into the unified generative space introduces ambiguity in the prediction target, since both item sequences and review sequences become plausible continuations under the same context.* To resolve this ambiguity, the proposed RAGR framework addresses two sub-problems in sequence: (1) how to train a generative model over the mixed item–review SID sequence so that review context enriches preference modeling (Section III-C), and (2) how to align the generation objective so that the model favors item SIDs over review SIDs at prediction positions (Section III-D).

III. The Proposed RAGR Method

A. Overview of RAGR

As illustrated in Fig. 2, RAGR consists of three stages: *Tokenizer Training*, *Review-Augmented User Sequence Modeling*, and *Item-Centric Task Generation Alignment*.

In the first stage, we **train a tokenizer** based on RQ-VAE [11] to map both items and reviews into a unified semantic ID space. This stage produces discrete token representations required for subsequent generative modeling. In the second stage, we construct **review-augmented user sequences** by interleaving item interactions with their associated review feedback. Based on such sequences, the generative recommender is trained for next item generation, so that recommendation is conditioned not only on item interactions but also on semantic evidence derived from review feedback. In the third stage, we introduce **item-centric task generation alignment** to preserve the task boundary of next item recommendation. Specifically, we employ DPO-based preference alignment to bias generation toward target items rather than review text, so that review feedback serves as evidence for item selection instead of becoming a competing generation target.

B. Tokenizer Training

To address the need for a unified discrete token space for both item interactions and review feedback, we first train a semantic tokenizer based on item text. As shown in Fig. 2, we first encode item text into dense semantic representations with an LLM-based text encoder, and then train an RQ-VAE [11] tokenizer to quantize such representations into multi-level discrete codes. In this way, each item can be represented as a semantic ID sequence that serves as the basis for subsequent review-augmented generative recommendation.

Formally, let x_i denote the textual content associated with item $i \in \mathcal{I}$, such as its title, category, or description. We first employ a pretrained LLM-based text encoder to obtain a dense embedding

$$\mathbf{e}_i = E(x_i), \tag{7}$$

where $E(\cdot)$ denotes the text encoder and $\mathbf{e}_i \in \mathbb{R}^d$ is the semantic embedding of item i. Compared with raw item IDs, such text-informed representations provide richer semantic information and serve as the basis for tokenizer learning.

Based on these item embeddings, we train a residual quantization variational autoencoder (RQ-VAE) to discretize the continuous semantic space into a sequence of code indices. Specifically, given an item embedding $\mathbf{e}_i$, the encoder of RQ-VAE first maps it into a latent representation

$$\mathbf{h}_i = g_{\text{enc}}(\mathbf{e}_i), \tag{8}$$

where $g_{\text{enc}}(\cdot)$ denotes the encoder network. We then apply residual quantization over M codebooks $\{\mathcal{C}^{(1)}, \mathcal{C}^{(2)}, \dots, \mathcal{C}^{(M)}\}$ in a coarse-to-fine manner. At the m-th level, the residual representation is quantized by selecting the nearest codeword from the corresponding codebook:

$$q_i^{(m)} = \arg\min_k \left\| \mathbf{r}_i^{(m-1)} - \mathbf{c}_k^{(m)} \right\|_2^2, \tag{9}$$

where $\mathbf{c}_k^{(m)} \in \mathcal{C}^{(m)}$ denotes the k-th codeword in the m-th codebook, and $\mathbf{r}_i^{(m-1)}$ is the residual to be quantized at level m. The residual is updated as

$$\mathbf{r}_i^{(m)} = \mathbf{r}_i^{(m-1)} - \mathbf{c}_{q_i^{(m)}}^{(m)}, \tag{10}$$

with $\mathbf{r}_i^{(0)} = \mathbf{h}_i$. After M levels of quantization, the item representation is approximated by the sum of selected codewords:

$$\hat{\mathbf{h}}_i = \sum_{m=1}^{M} \mathbf{c}_{q_i^{(m)}}^{(m)}. \tag{11}$$

Accordingly, each item i is represented as a multi-level discrete semantic ID

$$\mathbf{z}(i) = \left[q_i^{(1)}, q_i^{(2)}, \dots, q_i^{(M)}\right], \tag{12}$$

which serves as its tokenized representation in the unified generative space. Intuitively, earlier codebooks capture coarse-grained semantics, while later codebooks progressively refine the representation with finer-grained details. Following standard residual quantization objectives [21], we train the RQ-VAE by minimizing the discrepancy between the latent representation and its quantized approximation, together with codebook and commitment regularization:

$$\begin{aligned}\mathcal{L}_{\text{tok}} = \; & \underbrace{\left\| \mathbf{h}_i - \hat{\mathbf{h}}_i \right\|_2^2}_{\mathcal{L}_{\text{rec}}} \\ & + \underbrace{\sum_{m=1}^{M} \left\| \text{sg}\left[\mathbf{r}_i^{(m-1)}\right] - \mathbf{c}_{q_i^{(m)}}^{(m)} \right\|_2^2}_{\mathcal{L}_{\text{code}}} \\ & + \beta \underbrace{\sum_{m=1}^{M} \left\| \mathbf{r}_i^{(m-1)} - \text{sg}\left[\mathbf{c}_{q_i^{(m)}}^{(m)}\right] \right\|_2^2}_{\mathcal{L}_{\text{commit}}}. \end{aligned} \tag{13}$$

where $\text{sg}[\cdot]$ denotes the stop-gradient operator and β is the commitment coefficient.

C. Review-Augmented User Sequence Modeling

To address the limitation that existing generative recommendation models primarily capture what users selected but not why they selected it, we augment the conventional item-only interaction sequence with review feedback, so that each historical interaction is represented by both its item semantic ID and its corresponding review semantic ID. As illustrated in Fig. 2, instead of organizing user behavior as a pure item trajectory, we interleave item and review tokens in chronological order, allowing the generative recommender to condition sequence modeling on both behavioral outcomes and semantic feedback evidence.

Let $\mathbf{z}(i_t)$ and $\mathbf{z}(r_t)$ denote the semantic IDs of item i_t and review r_t, respectively, obtained from the tokenizer trained in the first stage (Section III-B). For each user u, we construct the review-augmented sequence as

$$\widetilde{\mathcal{S}}_u = \left[\mathbf{z}(i_1), \mathbf{z}(r_1), \mathbf{z}(i_2), \mathbf{z}(r_2), \dots, \mathbf{z}(i_T), \mathbf{z}(r_T)\right]. \tag{14}$$

Compared with conventional item-only sequences, this formulation allows the model to observe not only what the user interacted with, but also how the user described or evaluated those interactions.

Based on the review-augmented sequence, we construct a unified textual generation task by serializing item SIDs and review SIDs into autoregressive training instances. Specifically, we generate two types of sequence-to-sequence samples from the same review-augmented interaction history. The first type is *next-item SID prediction*, where the input consists of the historical item-review sequence up to step $t-1$, and the target is the semantic ID sequence of the next item $\mathbf{z}(i_t)$. The second type is *next-review SID prediction*, where the input additionally includes the target item $\mathbf{z}(i_t)$, and the target is the corresponding review semantic ID sequence $\mathbf{z}(r_t)$. In both cases, the model is trained as a standard generative recommender over textualized semantic IDs, rather than as two separate task-specific predictors.

Formally, let $(\mathbf{x}, \mathbf{y})$ denote a generic training instance constructed from the review-augmented sequence, where $\mathbf{x}$ is the serialized input sequence and $\mathbf{y}$ is the target semantic ID sequence to be generated. Then the GR model is optimized with a unified autoregressive generation objective:

$$\mathcal{L}_{\text{seq}} = -\sum_{(\mathbf{x}, \mathbf{y}) \in \mathcal{D}_{\text{seq}}} \log p_\theta(\mathbf{y} \mid \mathbf{x}), \tag{15}$$

where $\mathcal{D}_{\text{seq}}$ denotes the collection of all serialized next-item and next-review training instances. Under this formulation, item prediction and review prediction are not treated as separate optimization objectives, but as two forms of textual sequence generation derived from the same review-augmented user behavior.

This unified sequence modeling strategy provides two benefits. First, it exposes the model to a finer-grained generative sequence that interleaves item interactions and review feedback, thereby enriching user sequence modeling beyond pure item transitions. Second, it allows the GR model to learn the semantic dependency between items and their associated feedback within a shared generative space, which lays the foundation for the subsequent item-centric task alignment stage.

D. Item-Centric Task Generation Alignment

To address the risk that review-augmented sequence modeling may blur the task boundary of next-item recommendation and inadvertently drive the model to generate review semantic IDs, we further introduce an item-centric task alignment stage that explicitly aligns the GR model toward target item generation rather than review generation. As illustrated in Fig. 2, the key idea is to treat the target item semantic ID as the *preferred* output and the corresponding review semantic ID as the *rejected* output under the same context, so that review feedback remains informative evidence for item selection instead of becoming a competing generation target.

Formally, let $\mathbf{x}_t$ denote the serialized review-augmented context constructed from the historical sequence before the target interaction, i.e.,

$$\mathbf{x}_t = \big[\mathbf{z}(i_1), \mathbf{z}(r_1), \dots, \mathbf{z}(i_{t-1}), \mathbf{z}(r_{t-1})\big]. \tag{16}$$

For each target interaction at step t, we form a preference pair

$$\big(\mathbf{x}_t,\ \mathbf{y}_t^+,\ \mathbf{y}_t^-\big), \tag{17}$$

where

$$\mathbf{y}_t^+ = \mathbf{z}(i_t), \qquad \mathbf{y}_t^- = \mathbf{z}(r_t). \tag{18}$$

Here, $\mathbf{y}_t^+$ is the preferred output because it corresponds to the next target item that should be recommended, while $\mathbf{y}_t^-$ is the rejected output because it corresponds to review feedback, which should inform recommendation but should not replace the recommendation target itself. Based on these pairs, we construct a training set

$$\mathcal{D}_{\text{dpo}} = \left\{ \big(\mathbf{x}_t, \mathbf{y}_t^+, \mathbf{y}_t^-\big) \,\middle|\, u \in \mathcal{U},\ 2 \le t \le T \right\}. \tag{19}$$

We then apply DPO [20] to align the GR model toward the preferred item output. Let π_θ denote the current GR model and π_{ref} denote a reference model. To improve readability, we first define the relative preference score between the target item and the review signal under the same context as

$$\Delta_\theta(\mathbf{x}, \mathbf{y}^+, \mathbf{y}^-) = \log \frac{\pi_\theta(\mathbf{y}^+ \mid \mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}^+ \mid \mathbf{x})} - \log \frac{\pi_\theta(\mathbf{y}^- \mid \mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}^- \mid \mathbf{x})}. \tag{20}$$

For each preference triple $(\mathbf{x}, \mathbf{y}^+, \mathbf{y}^-) \in \mathcal{D}_{\text{dpo}}$, the DPO loss is then written as

$$\ell_{\text{DPO}}(\mathbf{x}, \mathbf{y}^+, \mathbf{y}^-) = -\log \sigma\big(\beta\, \Delta_\theta(\mathbf{x}, \mathbf{y}^+, \mathbf{y}^-)\big), \tag{21}$$

and the overall objective is

$$\mathcal{L}_{\text{DPO}} = \mathbb{E}_{(\mathbf{x}, \mathbf{y}^+, \mathbf{y}^-) \sim \mathcal{D}_{\text{dpo}}} \left[\ell_{\text{DPO}}(\mathbf{x}, \mathbf{y}^+, \mathbf{y}^-)\right]. \tag{22}$$

Here, $\sigma(\cdot)$ denotes the sigmoid function, and β is a temperature coefficient that controls the sharpness of optimization.

This objective explicitly encourages the GR model to assign a higher conditional likelihood to the target item semantic ID than to the review semantic ID under the same review-augmented context. In this way, review feedback is not removed from the generative sequence; instead, it is retained as contextual evidence while the model is aligned to preserve the item-centric recommendation objective. Consequently, the task alignment stage complements the previous sequence modeling stage: the latter enriches the user sequence with semantic feedback, whereas the former ensures that such feedback improves item generation without shifting the task from *next-item prediction* to *review generation*.

IV. Experiment

In this section, we conduct comprehensive experiments to evaluate the proposed **RAGR** framework. The experiments are designed to answer the following five research questions:

- **RQ1:** To what extent can RAGR improve the performance of existing generative recommendation backbones on next-item recommendation?
- **RQ2:** How effective are the two key components of RAGR, namely *Review-Augmented User Sequence Modeling* and *Item-Centric Task Generation Alignment*?
- **RQ3:** Which tokenizer training strategy is most effective for RAGR: training RQ-VAE on item text only, review text only, or both item and review text?
- **RQ4:** How sensitive is RAGR to the number of semantic ID tokens used in the tokenizer?
- **RQ5:** How sensitive is RAGR to the hyperparameter settings of DPO-based task alignment?

We first present the experimental settings, and then answer the above research questions through a series of experiments.

A. Experimental Settings

TABLE I: Statistics of the datasets after preprocessing.

Datasets	#Users	#Items	#Inter.	#RAGR Train	#Val.	#Test
Beauty	22,363	12,101	198,502	285,189	22,363	22,363
Toys	19,412	11,924	167,597	238,134	19,412	19,412
Sport	35,598	18,357	296,337	414,684	35,598	35,598

represents the number of users, items, interactions, and samples.

*1) **Datasets**:* To evaluate the effectiveness of **RAGR**, we conduct experiments on three benchmark datasets from the Amazon review datasets [2]: Amazon-Beauty (Beauty for short), Amazon-Sports and Outdoors (Sport for short), and Amazon-Toys and Games (Toys for short).[1] These datasets contain both user–item interaction records and user-written reviews, making them particularly suitable for evaluating review-augmented generative recommendation. For each dataset, we organize user behaviors into chronological interaction sequences, where each interaction is paired with its corresponding review text. Following common practice in sequential recommendation, we remove users and items with fewer than five interactions to ensure sufficient behavioral context. We then sort all interactions by timestamp and adopt the leave-one-out evaluation protocol [4], [11]: for each user, the last interaction is used for testing, the second last interaction is used for validation, and the remaining interactions are used for training. In our setting, the review text associated with each historical interaction is retained, so that both item interactions and post-decision feedback can be incorporated into the review-augmented user sequence. Table I summarizes the statistics of the three datasets.

[1] https://cseweb.ucsd.edu/~jmcauley/datasets/amazon/links.html

*2) **Baselines**:* To comprehensively evaluate the proposed method, we compare RAGR with two groups of baselines, including representative sequential recommenders and recent generative recommendation models.

Sequential recommenders.

- **GRU4Rec** [22]: GRU4Rec is a classical sequential recommender that models user interaction sequences with gated recurrent units (GRUs). By recurrently updating hidden states along the interaction trajectory, it captures short-term sequential dependency and predicts the next item based on the encoded user state.
- **BERT4Rec** [10]: BERT4Rec extends bidirectional Transformer modeling to sequential recommendation. Instead of relying on unidirectional next-step prediction, it adopts an objective to learn bidirectional contextual representations over user interaction sequences, thereby capturing richer dependency patterns among historical items.
- **SASRec** [4]: SASRec is a self-attention-based sequential recommender that uses Transformer-style attention blocks to model long-range dependencies in user interaction histories. By selectively attending to relevant past items, it effectively captures both short-term and long-term sequential signals for next-item prediction.
- **S^3-Rec** [23]: S^3-Rec enhances sequential recommendation through self-supervised representation learning. It introduces multiple auxiliary pretraining tasks over item attributes, item co-occurrence, and sequence context, so that the model can learn more informative sequence representations for next-item prediction.

Generative recommenders.

- **TIGER** [11]: TIGER is a representative generative recommender that tokenizes each item into a multi-level semantic ID and reformulates recommendation as an autoregressive generation problem. Instead of predicting over a fixed item vocabulary, it generates the semantic ID sequence of the next target item conditioned on the historical item sequence, thereby bridging generative retrieval and sequential recommendation.
- **LETTER** [13]: LETTER further improves generative recommendation by learning more effective semantic tokenization for items. It enhances the quality of item semantic IDs and the corresponding generative retrieval process, leading to stronger next-item prediction performance under the generative recommendation paradigm.

*3) **Evaluation Metrics**:* Following common practice in sequential recommendation [4], [11], we evaluate all methods with top-K ranking metrics, including **HIT@K** and **NDCG@K**, where $K \in \{5, 10, 20\}$.

*4) **Implementation Details**:* To verify the effectiveness and generality of RAGR across different GR backbones, we instantiate RAGR on two representative GR models, resulting in **TIGER+RAGR** and **LETTER+RAGR**. For a fair comparison, all baselines are trained and evaluated under the same data split and evaluation protocol. For conventional sequential recommenders, only item interaction sequences are used as input, following their original settings. For generative recommenders, we adopt their standard item tokenization and autoregressive prediction settings, while our method further incorporates review feedback into the unified generative sequence.

For the **tokenizer training** stage, we use T5[2] as a text encoder to obtain embeddings for items and reviews. For the RQ-VAE architecture, we use a six-layer encoder/decoder MLP with hidden dimensions $[2048, 1024, 512, 256, 128, 64]$. The number of residual codebooks is set to 4, and each codebook contains 256 codewords, and the codeword dimension set to 32. We enable k-means initialization for the codebooks with 100 iterations. The number of clusters is set to 10. In our implementation, the RQ-VAE is optimized with the AdamW optimizer, using a learning rate of 1×10^{-3}, a batch size of 2048, and a weight decay of 1×10^{-4}. The tokenizer is trained for at most 2000 epochs, with evaluation of the collision rate.

For **review-augmented user sequence modeling**, we interleave item semantic IDs and review semantic IDs in chronological order to construct the input sequence. The maximum history length is set to 20, and longer interaction histories are truncated from the left. For the GR backbone, we use T5 [24] as the base generative model. The model is trained for 200 epochs using the AdamW optimizer, with a learning rate of 1×10^{-3}, a per-device batch size of 256, gradient accumulation steps of 2, and a weight decay of 0.01. We adopt a cosine learning rate scheduler with a warmup ratio of 0.01.

For **item-centric task generation alignment**, we initialize the aligned model from the GR model trained in the previous stage, and use it as the policy model in DPO. The β in Eq. (21) is set to 0.5 to 0.7 with sensitivity analysis in Sections IV-F), and the reference model is fixed during alignment training. The model is trained using the AdamW optimizer, with a learning rate of 1×10^{-6}, a per-device batch size of 256.

All experiments were conducted on an Ubuntu server equipped with eight NVIDIA RTX PRO 6000 GPUs (96 GB). For the main performance experiment, all results are the average of three runs with different random seeds: 42, 43, and 44.

B. Overall Performance (RQ1)

To answer RQ1, we conducted a comprehensive performance comparison against SR baselines. Table II reports the overall performance comparison on Beauty, Toys, and Sport. Several observations can be drawn.

First, RAGR consistently improves both GR backbones, namely TIGER and LETTER, across all datasets and all evaluation metrics. For *TIGER*, the relative improvements range from *7%* to *26%*. In particular, on Beauty, TIGER+RAGR improves TIGER by *13%/15%* on HIT@5/NDCG@5 and by *9%/11%* on HIT@20/NDCG@20. On Toys, the gains are even larger, reaching *24%* on HIT@5 and *26%* on NDCG@5, while remaining consistently above *20%* on most metrics. On Sport, although the absolute gains are relatively smaller, RAGR still achieves stable improvements, ranging from *8%* to *19%*. For *LETTER*, RAGR also brings consistent improvements on all three datasets, with relative gains ranging from *7%* to *20%*. These results demonstrate that the proposed framework is not tied to a specific GR backbone, but can be effectively generalized across different generative recommendation architectures.

[2]https://huggingface.co/sentence-transformers/sentence-t5-base

TABLE II: Performance comparison on Beauty, Toys, and Sport. *Imp.* denotes the relative improvement of RAGR over its corresponding backbone model in each metric.

Dataset	Metric	GRU4Rec	BERT4Rec	SASRec	S^3-Rec	TIGER	TIGER+RAGR	Imp.	LETTER	LETTER+RAGR	Imp.
Beauty	HIT@5	0.0174	0.0202	0.0377	0.0361	0.0386	**0.0435***	***13%***	0.0371	**0.0446***	***20%***
	NDCG@5	0.0108	0.0111	0.0220	0.0228	0.0254	**0.0292***	***15%***	0.0253	**0.0294***	***16%***
	HIT@10	0.0323	0.0311	0.0605	0.0598	0.0607	**0.0649***	***7%***	0.0582	**0.0677***	***16%***
	NDCG@10	0.0156	0.0155	0.0303	0.0305	0.0325	**0.0361***	***11%***	0.0321	**0.0370***	***15%***
	HIT@20	0.0513	0.0518	0.0933	0.0940	0.0866	**0.0944***	***9%***	0.0885	**0.1019***	***15%***
	NDCG@20	0.0204	0.0207	0.0385	0.0391	0.0391	**0.0435***	***11%***	0.0397	**0.0455***	***15%***
Toys	HIT@5	0.0147	0.0140	0.0369	0.0376	0.0331	**0.0410***	***24%***	0.0321	**0.0386***	***20%***
	NDCG@5	0.0095	0.0082	0.0217	0.0238	0.0206	**0.0259***	***26%***	0.0210	**0.0243***	***16%***
	HIT@10	0.0257	0.0177	0.0591	0.0604	0.0526	**0.0630***	***20%***	0.0512	**0.0604***	***18%***
	NDCG@10	0.0130	0.0090	0.0289	0.0311	0.0269	**0.0329***	***22%***	0.0272	**0.0312***	***15%***
	HIT@20	0.0420	0.0333	0.0860	0.0910	0.0781	**0.0934***	***20%***	0.0770	**0.0902***	***17%***
	NDCG@20	0.0172	0.0104	0.0356	0.0388	0.0333	**0.0406***	***22%***	0.0337	**0.0387***	***15%***
Sport	HIT@5	0.0152	0.0114	0.0212	0.0204	0.0231	**0.0267***	***16%***	0.0240	**0.0264***	***10%***
	NDCG@5	0.0102	0.0085	0.0138	0.0136	0.0148	**0.0176***	***19%***	0.0156	**0.0172***	***10%***
	HIT@10	0.0231	0.0224	0.0311	0.0306	0.0385	**0.0415***	***8%***	0.0403	**0.0433***	***7%***
	NDCG@10	0.0127	0.0091	0.0177	0.0169	0.0197	**0.0224***	***14%***	0.0209	**0.0227***	***9%***
	HIT@20	0.0359	0.0300	0.0506	0.0483	0.0564	**0.0619***	***10%***	0.0612	**0.0661***	***8%***
	NDCG@20	0.0159	0.0080	0.0220	0.0213	0.0242	**0.0274***	***13%***	0.0262	**0.0284***	***9%***

"*" indicates statistically significant improvements (two-sided t-test with $p < 0.05$) over the corresponding backbone model.

Second, RAGR not only improves strong generative baselines, but also achieves the best overall performance against conventional sequential recommenders. On all three datasets, both TIGER+RAGR and LETTER+RAGR outperform GRU4Rec, BERT4Rec, SASRec, and S^3-Rec by clear margins. This suggests that enriching the user sequence with review feedback provides additional semantic evidence beyond item-only interaction modeling, which cannot be fully captured by conventional sequential recommenders.

Third, review-augmented sequence modeling appears to be especially beneficial in scenarios where textual feedback carries stronger preference signals. This pattern is most evident on Toys, where the gains of RAGR are consistently larger than those on Beauty and Sport. Moreover, the gains are consistently observed on both HIT@K and NDCG@K, indicating that RAGR improves not only recall performance but also the ranking quality of the ground-truth next item.

C. Ablation Study (RQ2)

To answer RQ2 and understand the contribution of each component in RAGR, we conduct an ablation study on two GR backbones. As illustrated in Fig. 3, we progressively extend the training paradigm from the original item-only generative recommendation to the full RAGR framework. Specifically, the compared variants are defined as follows:

- **+Input**: The original item-only input sequence is augmented with review semantic IDs, while the original next-item prediction target remains unchanged.
- **+Task**: Review prediction is further introduced into the textual generation process, so that both next-item SID prediction and next-review SID prediction are jointly modeled within the same unified sequence generation framework.
- **+RAGR**: The proposed item-centric task alignment is further incorporated, where the target item SID is treated as the chosen output and the corresponding review SID is treated as the rejected output in DPO training.

The results are reported in Table III, from which several observations can be made. **First, simply expanding the input sequence with review signals (+Input) does not improve performance; instead, it consistently degrades both TIGER and LETTER on all three datasets.** For example, on Beauty, TIGER drops from 0.0386 to 0.0260 in HIT@5, and LETTER drops from 0.0371 to 0.0311. Similar degradation is observed on Toys and Sport. This suggests that merely injecting review semantic IDs into the input sequence is insufficient, and may even introduce noise when the model is still optimized under the original item-only training objective. **Second, after extending the training objective to include review prediction (+Task), performance improves substantially over both the original backbone and the +Input variant.** For instance, on Toys, TIGER+Task reaches 0.0405/0.0258 on HIT@5/NDCG@5, clearly outperforming both TIGER and TIGER+Input; similarly, LETTER+Task also achieves large gains over the corresponding backbone. These results indicate that review SID becomes beneficial only when it is not merely appended to the input, but is explicitly incorporated into the unified sequence generation task. **Third, adding the proposed item-centric task alignment further improves performance in most settings and consistently yields the best overall results**. On Beauty, TIGER+RAGR improves over TIGER+Task from 0.0427 to 0.0435 in HIT@5 and from 0.0434 to 0.0435 in NDCG@20, while LETTER+RAGR further improves over LETTER+Task from 0.0421 to 0.0446 in HIT@5 and from 0.0971 to 0.1019 in HIT@20. On Toys, the improvement from +Task to +RAGR is also consistent for both backbones. On Sport, the gains are relatively smaller but remain stable. This verifies that DPO-based task alignment is effective in preserving the item-centric recommendation objective after review prediction is introduced, allowing review signals to serve as evidence rather than becoming competing generation targets.

To further understand why review augmentation improves recommendation performance, we compare the SID frequency

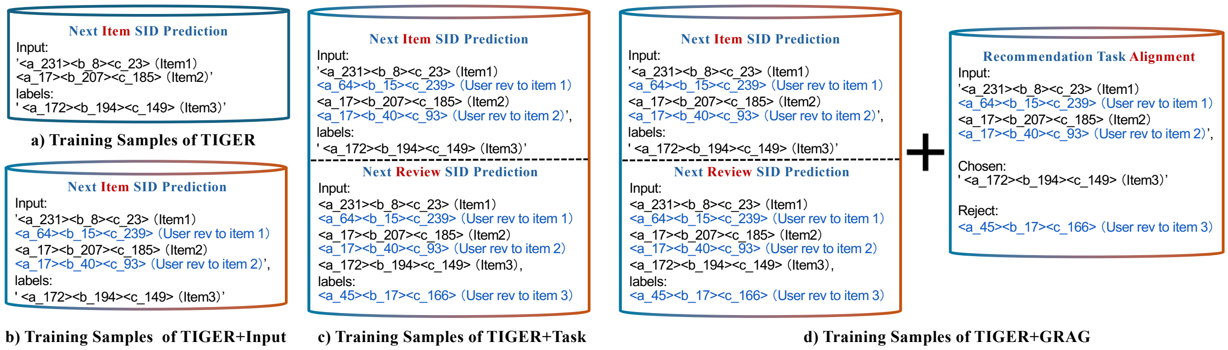


Fig. 3: Illustration of the progressively enhanced training paradigms in the ablation study. Starting from the original item-only training of TIGER, we gradually introduce review-augmented input (**+Input**), review-augmented task modeling (**+Task**), and finally the full RAGR framework with item-centric DPO alignment (**+RAGR**).

TABLE III: Ablation study of input expansion, task expansion, and task alignment on Beauty, Toys, and Sport.

Method	Beauty						Toys						Sport					
	H@5	N@5	H@10	N@10	H@20	N@20	H@5	N@5	H@10	N@10	H@20	N@20	H@5	N@5	H@10	N@10	H@20	N@20
TIGER	0.0386	0.0254	0.0607	0.0325	0.0866	0.0391	0.0331	0.0206	0.0526	0.0269	0.0781	0.0333	0.0231	0.0148	0.0385	0.0197	0.0564	0.0242
+Input	0.0260	0.0164	0.0417	0.0215	0.0619	0.0265	0.0273	0.0174	0.0424	0.0223	0.0676	0.0287	0.0191	0.0120	0.0317	0.0161	0.0482	0.0202
+Task	0.0427	0.0289	0.0649	0.0361	0.0938	0.0434	0.0405	0.0258	0.0624	0.0327	0.0930	0.0402	0.0266	0.0170	0.0413	0.0223	0.0606	0.0271
+RAGR	**0.0435**	**0.0292**	**0.0649**	**0.0361**	**0.0944**	**0.0435**	**0.0410**	**0.0259**	**0.0630**	**0.0329**	**0.0934**	**0.0406**	**0.0267**	**0.0176**	**0.0415**	**0.0224**	**0.0619**	**0.0274**
LETTER	0.0371	0.0253	0.0582	0.0321	0.0885	0.0397	0.0321	0.0210	0.0512	0.0272	0.0770	0.0337	0.0240	0.0156	0.0403	0.0209	0.0612	0.0262
+Input	0.0311	0.0205	0.0511	0.0269	0.0778	0.0336	0.0273	0.0174	0.0424	0.0223	0.0676	0.0287	0.0194	0.0126	0.0310	0.0164	0.0484	0.0208
+Task	0.0421	0.0280	0.0634	0.0353	0.0971	0.0435	0.0382	0.0240	0.0600	0.0310	0.0901	0.0386	0.0263	0.0172	0.0430	0.0226	0.0658	0.0284
+RAGR	**0.0446**	**0.0294**	**0.0677**	**0.0370**	**0.1019**	**0.0455**	**0.0386**	**0.0243**	**0.0604**	**0.0312**	**0.0902**	**0.0387**	**0.0264**	**0.0172**	**0.0433**	**0.0227**	**0.0661**	**0.0284**

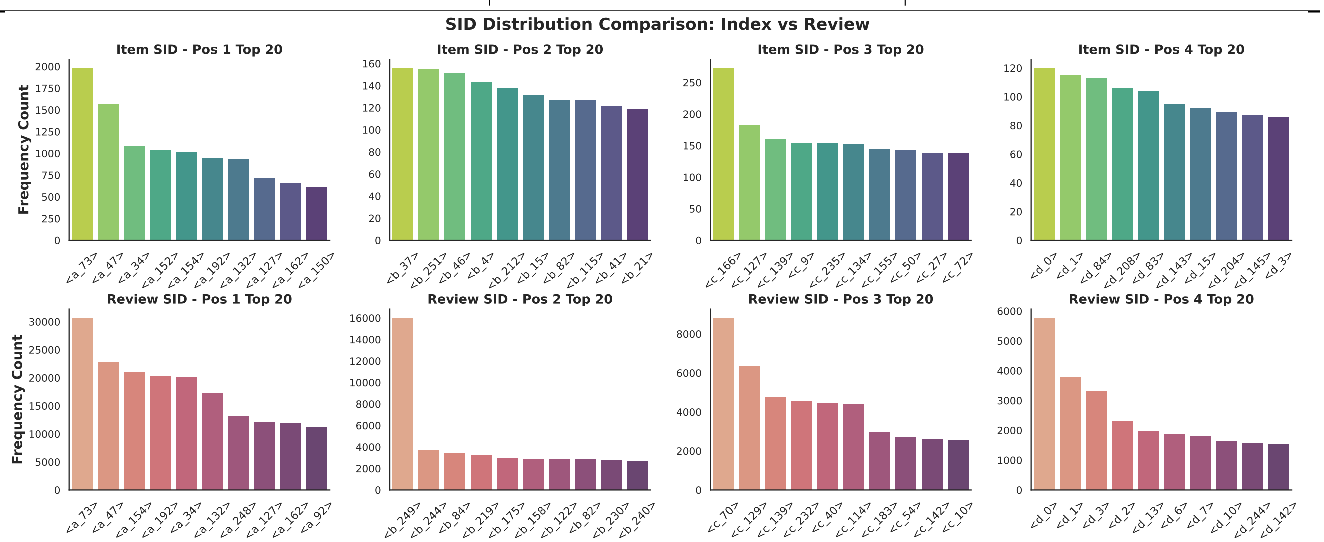


Fig. 4: Comparison of Top-10 SID frequency distributions between item and review.

distributions of item and review, as shown in Fig. 4. We observe that, after introducing review feedback, the SID distribution becomes substantially more diverse across different token positions. Compared with the item-only SID distribution, review-augmented sequences activate a broader set of semantic IDs and expose more tokens to effective training signals, rather than concentrating supervision on a small number of dominant item-side SIDs. In this sense, review augmentation not only injects additional preference semantics into the sequence, but also improves the utilization of the unified SID space during training, allowing more semantic IDs to learn meaningful representations. This provides a plausible explanation for the gains of **+Task** and **+RAGR** in Table III.

D. Tokenlizer Experiment (RQ3)

To answer RQ3, we consider three tokenizer training strategies, as illustrated in Fig. 5. Since the proposed framework relies on a unified semantic ID space for both item interactions and review feedback, the quality of the tokenizer is critical to the effectiveness of subsequent review-augmented sequence modeling and item-centric task alignment.

- **Item Text-Only**: the RQ-VAE tokenizer is trained only on item text embeddings.
- **Review Text-Only**: the RQ-VAE tokenizer is trained only on review text embeddings.
- **Item and Review Text**: the RQ-VAE tokenizer is trained on

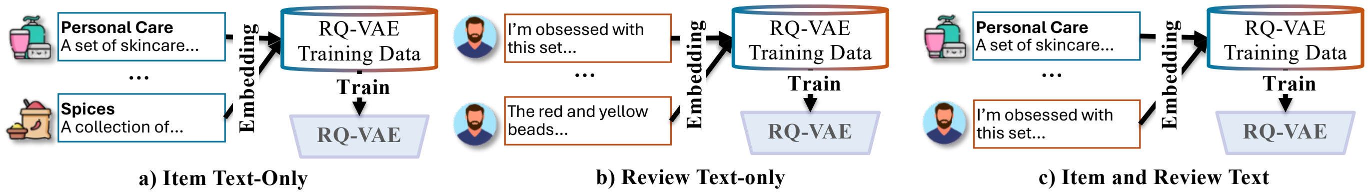


Fig. 5: Illustration of the three tokenizer: training RQ-VAE on item text only, review text only, and both item and review text.

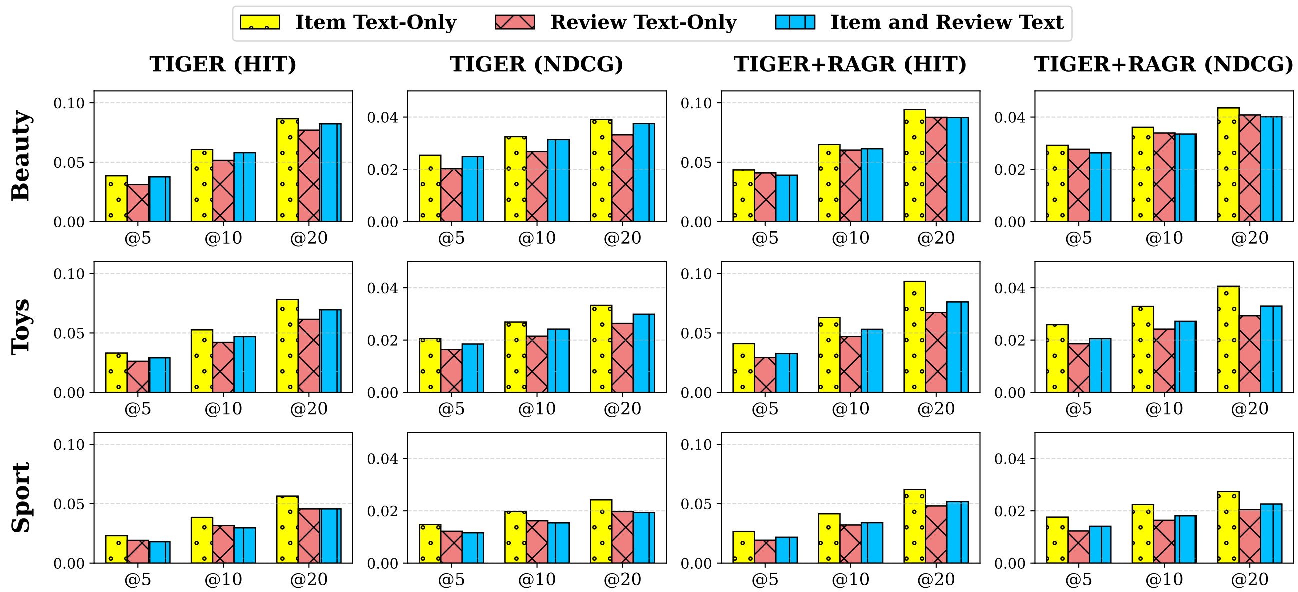


Fig. 6: Impact of different tokenizer training strategies on downstream recommendation performance. We compare tokenizers trained on *Item Text-Only*, *Review Text-Only*, and *Item and Review Text* under both TIGER and TIGER+RAGR.

the combination of item text and review text embeddings. To provide a more direct comparison, Table IV summarizes the training efficiency and collision rates of the three tokenizer training strategies across the three datasets.

TABLE IV: Comparison of different tokenizer training strategies in terms of training efficiency and collision rate (Col.).

Tokenizer	Training Time	Beauty Col.	Toys Col.	Sport Col.
Item Text-Only	2s/epoch	0.0009	0.0011	0.0025
Review Text-Only	29s/epoch	0.0009	0.0012	0.0024
Item and Review Text	33s/epoch	0.0008	0.0044	0.0005

Overall, the tokenizer experiment leads to three main observations. First, **Item Text-Only** consistently achieves the strongest downstream recommendation performance in most settings, for both the original TIGER backbone and the review-augmented TIGER+RAGR variant. As shown in Fig. 6, the item-text tokenizer yields the highest HIT and NDCG values on Beauty, Toys, and Sport in the majority of cases. This suggests that, for generative recommendation, learning semantic IDs from item text provides a more stable and effective token space for next-item prediction.

Second, **Review Text-Only** generally performs the worst among the three strategies. Although the review text contains rich user preference signals, using it alone to train the tokenizer appears to shift the item sid distribution, which weakens the quality of item representations and harms downstream recommendation performance. In contrast, **Item and Review Text** often performs better than **Review Text-Only**, but still does not consistently surpass **Item Text-Only**. This indicates that directly mixing review text into tokenizer training does not necessarily lead to a better semantic ID space, even though review feedback is beneficial at the sequence modeling stage.

Third, Table IV shows that **Item Text-Only** is also substantially more efficient, requiring only 2s per epoch, compared with 29s and 33s for **Review Text-Only** and **Item and Review Text**, respectively. Although the collision rates do not always perfectly align with downstream recommendation performance, the item-text tokenizer provides the best overall trade-off between efficiency and effectiveness. Therefore, in the remaining experiments, we adopt **Item Text-Only** as the default tokenizer training strategy.

E. SID Sensitivity Experiment (RQ4)

To answer **RQ4**, we investigate how the number of semantic ID tokens affects the effectiveness of RAGR. Intuitively, the number of semantic ID tokens controls the granularity of the discrete semantic space. Fewer tokens lead to a smaller semantic space and thus a higher risk of collisions, whereas more tokens improve semantic distinctiveness but may increase the difficulty of autoregressive generation. Therefore, studying the sensitivity of RAGR to the SID number is important for balancing representation quality and recommendation performance.

We vary the SID number from 3 to 5 and report both the tokenizer collision rate and the downstream recommendation performance. Table V summarizes the collision rate under different SID numbers on Beauty, Toys, and Sport. As expected,

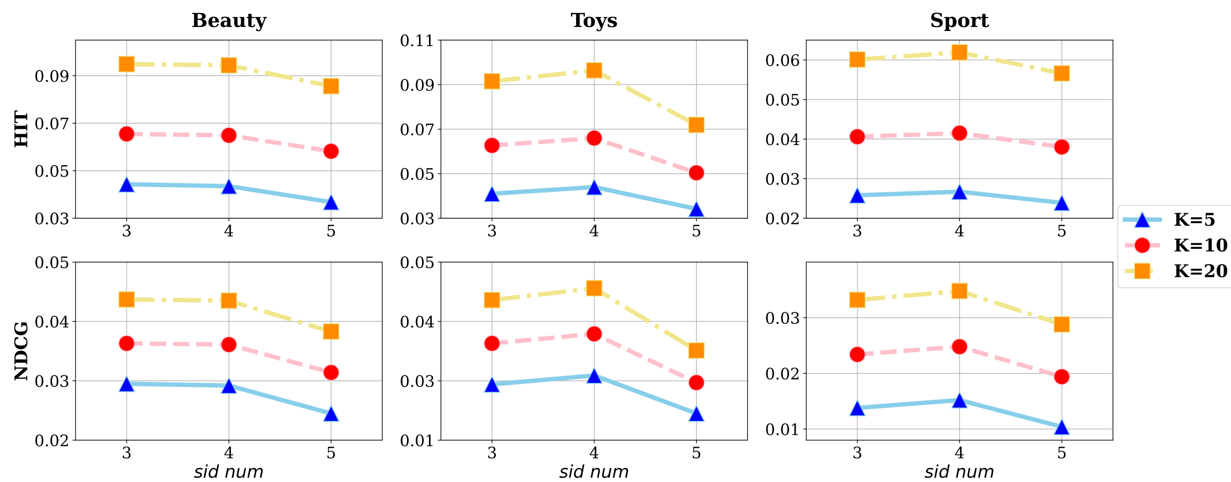


Fig. 7: Impact of the number of semantic ID tokens on recommendation performance. We report HIT@K and NDCG@K ($K \in \{5, 10, 20\}$) on Beauty, Toys, and Sport under different SID numbers.

TABLE V: Collision rates under different SID numbers on the three datasets.

SID Num	Beauty Col.	Toys Col.	Sport Col.
3	0.0121	0.0017	0.0045
4	0.0009	0.0012	0.0026
5	0.0007	0.0003	0.0018

increasing the SID number consistently reduces the collision rate across all datasets. For example, on Beauty, the collision rate decreases from 0.0121 when using 3 semantic ID tokens to 0.0009 and 0.0007 when using 4 and 5 tokens, respectively. Similar trends are observed on Toys and Sport. This confirms that longer semantic IDs indeed improve token distinctiveness and reduce representation collisions. However, a lower collision rate does not necessarily translate into better recommendation performance. Figure 7 reports the downstream results, from which several observations can be drawn.

First, using 4 semantic ID tokens achieves the best overall performance across the three datasets and both metrics. On Beauty, Toys, and Sport, SID number 4 consistently gives the strongest HIT and NDCG results, indicating that it provides the most effective balance between semantic distinctiveness and generation difficulty. Second, the relationship between SID length and performance is clearly non-monotonic. When the SID number is too small, the collision rate increases, leading to multiple items sharing the same or highly similar semantic IDs. In this case, the tokenizer loses item-level discriminability, and the model can only learn a coarse semantic partition of the item space. For example, on Beauty, SID number 3 still yields relatively competitive results, even though its collision rate is much higher than that of SID number 4. A plausible explanation is that the higher collision rate causes semantically similar or high-frequency items to be merged into shared semantic IDs, which introduces a smoothing effect and partially alleviates data sparsity in generation. However, such gains come at the cost of reduced item distinguishability, which limits further performance improvement. Third, increasing the SID number from 4 to 5 further reduces collision rates, but consistently hurts recommendation performance. This suggests that once semantic collisions are sufficiently controlled, making the semantic ID sequence even longer mainly enlarges the generation space and increases autoregressive decoding difficulty. As a result, the model faces a harder generation problem without receiving commensurate gains in representation quality.

Overall, the results suggest that SID length controls a trade-off between semantic uniqueness and generation complexity. Too few tokens lead to excessive collisions and insufficient item discrimination, while too many tokens make next-item generation unnecessarily sparse and difficult. Therefore, a moderate SID length is most desirable in practice. Based on these results, we set the SID number to 4 as the default configuration of RAGR.

F. DPO Sensitivity Experiment (RQ5)

To answer **RQ5**, we investigate how sensitive RAGR is to the hyperparameter settings of DPO-based task alignment. In particular, we focus on the preference coefficient β, which controls the sharpness of preference optimization, and the training epoch, which determines the extent to which the GR backbone is further aligned toward item-centric generation. Since DPO is introduced to preserve the task boundary of *next-item recommendation* after review signals are incorporated, understanding its hyperparameter sensitivity is important for assessing the stability and robustness of the proposed alignment strategy. Figure 8 visualizes the performance landscape under different combinations of β and training epoch in terms of HIT@$\{5, 10, 20\}$ and NDCG@$\{5, 10, 20\}$. Several observations can be made.

First, the performance surface is generally smooth rather than highly irregular, indicating that the proposed DPO-based alignment is reasonably stable over a broad range of hyperparameter settings. Second, for most evaluation metrics, moderate values of β consistently lead to better performance than overly small or overly large values. This suggests that if β is too small, the preference signal becomes too weak to effectively enforce item-centric alignment; in contrast, if β is

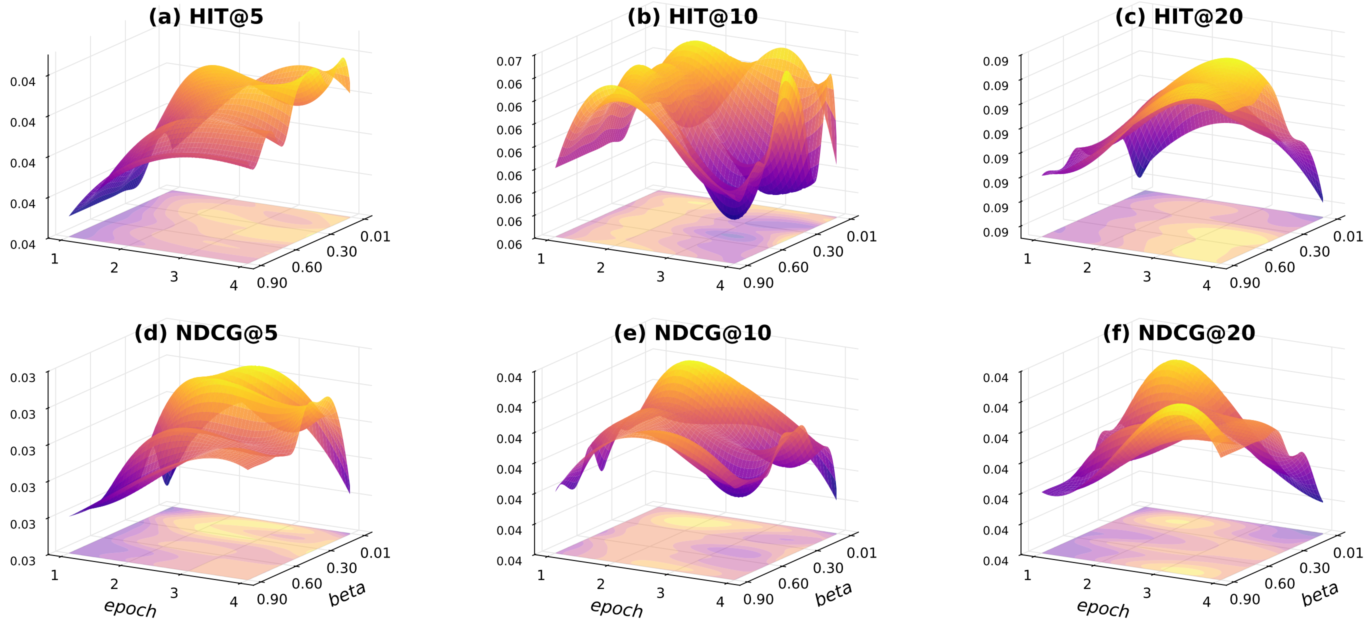


Fig. 8: Sensitivity analysis of the DPO-based task alignment with respect to the preference coefficient β and training epoch. We report HIT@$\{5, 10, 20\}$ and NDCG@$\{5, 10, 20\}$ under different hyperparameter combinations.

too large, the model may overemphasize the chosen–rejected contrast and thus harm the original generative recommendation capability. Third, the interaction between β and training epoch further reveals that the best performance is usually achieved within a moderate training range, rather than at the earliest or latest alignment epochs. This indicates that DPO-based task alignment should be sufficiently strong to shift the model toward target item generation, but not so strong as to overfit the preference pairs. In other words, the sensitivity analysis confirms that the effectiveness of RAGR does not rely on an extremely narrow hyperparameter region, while also showing that proper tuning of β and training epoch is beneficial for obtaining the best performance.

Overall, the results demonstrate that the proposed item-centric task alignment is robust to hyperparameter variation and can stably improve recommendation performance across a reasonably wide range of settings. Based on the validation results, we select the corresponding β and epoch combination that achieves the best trade-off between performance and stability as the default configuration in our experiments.

V. Related work

A. Sequential Recommendation

SR aims to model users' evolving preferences from their historical interaction sequences and predict the next item they are likely to engage with. As one of the most fundamental paradigms in recommender systems [8], it has been extensively studied under the *next-item prediction* formulation.

Existing sequential recommendation methods mainly differ in how they encode the interaction sequence. Early studies relied on recurrent architectures, such as GRU4Rec [22], to capture sequential dependency. Later, Transformer-based models became dominant, including SASRec [4] and BERT4Rec [10], which leverage self-attention and bidirectional contextual modeling for sequence representation learning. To further improve robustness and generalization, self-supervised methods such as S^3-Rec [23] and contrastive learning approaches [5] were introduced. More recently, temporal modeling [25], dynamic graph modeling [26], hyper adapter [27], and LLM-enhanced representations [19], [28] have further enriched sequential recommendation architectures.

Despite these advances, most existing SR methods still model user behavior as an item-only interaction sequence and optimize for next-item prediction. **As a result, they mainly capture what users selected, while overlooking richer behavior signals such as reviews and feedback that may explain why users selected or rejected items.** In contrast, our work moves beyond item-only sequence modeling by incorporating review feedback into the generative user sequence.

B. Generative Recommendation

GR reformulates recommendation as a generation problem, where the model produces the target item through autoregressive decoding rather than ranking over a fixed item vocabulary. Recent surveys have highlighted this paradigm as a rapidly growing direction in recommender systems, with progress mainly driven by three aspects: tokenization, generative architecture, and optimization [12], [18], [29].

Existing generative recommendation methods mainly focus on how to represent items as generatable tokens and how to improve the efficiency and effectiveness of generation. A representative line of work learns *semantic IDs* for items and performs next-item generation over these discrete identifiers, such as TIGER [11] and LETTER [13]. More recent studies further investigate tokenization and decoding strategies for generative recommendation, including ID tokenization for LLM-based recommendation [14], [30], parallel decoding of long semantic IDs [31], contextual tokenization of action sequences [32], unified sparse-dense generative recommendation [33], and acceleration-oriented decoding architectures [34]. Overall, these studies have substantially advanced the tokenization and architectural design of generative recommendation.

Despite these advances, existing generative recommendation methods still largely remain within the traditional next-item prediction paradigm. **That is, the user sequence is still predominantly modeled as an item-only interaction sequence, and the generation target remains the next item.** This holds even though they replace raw item IDs with SIDs and cast recommendation into an autoregressive generation process. In contrast, our work moves beyond item-only generative recommendation by incorporating review feedback into the unified generative sequence and further introducing item-centric task alignment.

C. Review-Aware Recommendation

Review-aware recommendation aims to leverage user-generated reviews to enrich preference modeling beyond pure interaction signals. As discussed in recent surveys [35], [36], reviews have been widely recognized as a valuable source of semantic evidence for recommendation, because they contain rich information about user opinions, product attributes, and decision rationale.

A large body of prior work incorporates reviews as auxiliary textual features for matching or rating prediction. Early methods jointly modeled users and items with review text to improve recommendation quality [15], while subsequent studies explored aspect-aware modeling [37], [38], review-level explanation and interpretability [39], [40], review property modeling [41], and graph-based or hypergraph-based semantic interaction modeling [42], [43]. At the same time, several studies have critically examined the actual utility of reviews in recommendation and discussed their limitations and possible improvements [44]. More recently, with the rise of large language models, another line of work has attempted to enhance recommendation by exploiting the text understanding and reasoning capability of LLMs, such as recommendation-as-language-processing [45], prompt-based personalized recommendation [46], rationale-enhanced LLM recommendation [47], and review-analysis-based feature enhancement [48].

Despite their differences, most existing review-aware recommendation methods still treat reviews as auxiliary features and side information outside the core recommendation sequence. **In other words, reviews are typically used to enhance user/item representations or improve matching, rather than being incorporated into a unified generative sequence together with item interactions.** In contrast, our work explicitly maps review feedback into the same semantic ID space as items, integrates reviews into the generative user sequence itself, and further introduces item-centric task alignment to ensure that review signals enhance next-item generation without becoming competing generation targets.

VI. Conclusion

In this paper, we take a first step toward rethinking generative recommendation beyond the conventional item-only paradigm. We argue that the longstanding assumption of modeling user sequences solely as item trajectories constitutes a fundamental limitation, as it captures what users choose but overlooks why they choose it. To our knowledge, this work is the first to bring review feedback directly into the generative recommendation paradigm by encoding reviews as tokens within the user sequence itself. Based on this idea, we propose **R**eview-**A**ugmented **G**enerative **R**ecommendation (**RAGR**), which augments item-only behavioral sequences with review semantics while preserving the next-item recommendation objective through item-centric alignment. Extensive experiments on three real-world datasets demonstrate the effectiveness and generality of our framework. We hope this work can inspire future research on enriching generative recommendation with more expressive user-side signals.

Acknowledgments

This research was supported by the Science Challenge Project, No.TZ2025005 and the National Natural Science Foundation of China (NSFC) under Grants 72071029, 72231010, 62502404. This research was partially supported by Hong Kong Research Grants Council (Research Impact Fund No.R1015-23, Collaborative Research Fund No.C1043-24GF, General Research Fund No.11218325), Institute of Digital Medicine of City University of Hong Kong (No.9229503), Huawei (Huawei Innovation Research Program), and the Graduate Research Fund of the School of Economics and Management of Dalian University of Technology (No. DUTSEMDRFKO1).

References

[1] Z. Zhao, W. Fan, J. Li, Y. Liu, X. Mei, Y. Wang, Z. Wen, F. Wang, X. Zhao, J. Tang *et al.*, "Recommender systems in the era of large language models (llms)," *IEEE Transactions on Knowledge and Data Engineering*, vol. 36, no. 11, pp. 6889–6907, 2024. 1

[2] R. He and J. McAuley, "Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering," in *proceedings of the 25th international conference on world wide web*, 2016, pp. 507–517. 1, 5

[3] J. Wang, P. Huang, H. Zhao, Z. Zhang, B. Zhao, and D. L. Lee, "Billion-scale commodity embedding for e-commerce recommendation in alibaba," in *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 2018, pp. 839–848. 1

[4] W.-C. Kang and J. McAuley, "Self-attentive sequential recommendation," in *2018 IEEE international conference on data mining (ICDM)*. IEEE, 2018, pp. 197–206. 1, 5, 6, 11

[5] X. Xie, F. Sun, Z. Liu, S. Wu, J. Gao, J. Zhang, B. Ding, and B. Cui, "Contrastive learning for sequential recommendation," in *2022 IEEE 38th international conference on data engineering (ICDE)*. IEEE, 2022, pp. 1259–1273. 1, 11

[6] G. Zhou, X. Zhu, C. Song, Y. Fan, H. Zhu, X. Ma, Y. Yan, J. Jin, H. Li, and K. Gai, "Deep interest network for click-through rate prediction," in *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 2018, pp. 1059–1068. 1

[7] G. Zhou, N. Mou, Y. Fan, Q. Pi, W. Bian, C. Zhou, X. Zhu, and K. Gai, "Deep interest evolution network for click-through rate prediction," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 5941–5948. 1

[8] L.-W. Pan, W.-K. Pan, M.-Y. Wei, H.-Z. Yin, and Z. Ming, "A survey on sequential recommendation," *Frontiers of Computer Science*, vol. 20, no. 3, p. 2003606, 2026. 1, 11

[9] H. Fang, D. Zhang, Y. Shu, and G. Guo, "Deep learning for sequential recommendation: Algorithms, influential factors, and evaluations," *ACM Transactions on Information Systems (TOIS)*, vol. 39, no. 1, pp. 1–42, 2020. 1

[10] F. Sun, J. Liu, J. Wu, C. Pei, X. Lin, W. Ou, and P. Jiang, "Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer," in *Proceedings of the 28th ACM international conference on information and knowledge management*, 2019, pp. 1441–1450. 1, 6, 11

[11] S. Rajput, N. Mehta, A. Singh, R. Hulikal Keshavan, T. Vu, L. Heldt, L. Hong, Y. Tay, V. Tran, J. Samost *et al.*, "Recommender systems with generative retrieval," *Advances in Neural Information Processing Systems*, vol. 36, pp. 10 299–10 315, 2023. 1, 2, 3, 5, 6, 11

[12] X. Li, B. Chen, J. She, S. Cao, Y. Wang, Q. Jia, H. He, Z. Zhou, Z. Liu, J. Liu *et al.*, "A survey of generative recommendation from a tri-decoupled perspective: Tokenization, architecture, and optimization," 2025. 1, 2, 11

[13] W. Wang, H. Bao, X. Lin, J. Zhang, Y. Li, F. Feng, S.-K. Ng, and T.-S. Chua, "Learnable item tokenization for generative recommendation," in *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, 2024, pp. 2400–2409. 1, 2, 6, 11

[14] H. Qu, W. Fan, Z. Zhao, and Q. Li, "Tokenrec: Learning to tokenize id for llm-based generative recommendations," *IEEE Transactions on Knowledge and Data Engineering*, 2025. 1, 2, 11

[15] L. Zheng, V. Noroozi, and P. S. Yu, "Joint deep modeling of users and items using reviews for recommendation," in *Proceedings of the tenth ACM international conference on web search and data mining*, 2017, pp. 425–434. 1, 12

[16] Z. Hai, K. Chang, J.-J. Kim, and C. C. Yang, "Identifying features in opinion mining via intrinsic and extrinsic domain relevance," *IEEE transactions on knowledge and data engineering*, vol. 26, no. 3, pp. 623–634, 2013. 1

[17] J. McAuley and J. Leskovec, "Hidden factors and hidden topics: understanding rating dimensions with review text," in *Proceedings of the 7th ACM conference on Recommender systems*, 2013, pp. 165–172. 1

[18] Y. Deldjoo, Z. He, J. McAuley, A. Korikov, S. Sanner, A. Ramisa, R. Vidal, M. Sathiamoorthy, A. Kasirzadeh, and S. Milano, "A review of modern recommender systems using generative models (gen-recsys)," in *Proceedings of the 30th ACM SIGKDD conference on Knowledge Discovery and Data Mining*, 2024, pp. 6448–6458. 2, 11

[19] H. Wang, X. Liu, W. Fan, X. Zhao, V. Kini, D. P. Yadav, F. Wang, Z. Wen, and H. Liu, "Rethinking large language model architectures for sequential recommendations," in *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, 2025, pp. 3376–3391. 2, 11

[20] R. Rafailov, A. Sharma, E. Mitchell, C. D. Manning, S. Ermon, and C. Finn, "Direct preference optimization: Your language model is secretly a reward model," *Advances in neural information processing systems*, vol. 36, pp. 53 728–53 741, 2023. 2, 5

[21] D. Lee, C. Kim, S. Kim, M. Cho, and W.-S. Han, "Autoregressive image generation using residual quantization," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 11 523–11 532. 4

[22] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, "Session-based recommendations with recurrent neural networks," *arXiv preprint arXiv:1511.06939*, 2015. 6, 11

[23] K. Zhou, H. Wang, W. X. Zhao, Y. Zhu, S. Wang, F. Zhang, Z. Wang, and J.-R. Wen, "S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization," in *Proceedings of the 29th ACM international conference on information & knowledge management*, 2020, pp. 1893–1902. 6, 11

[24] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of machine learning research*, vol. 21, no. 140, pp. 1–67, 2020. 6

[25] W. Ye, S. Wang, X. Chen, X. Wang, Z. Qin, and D. Yin, "Time matters: Sequential recommendation with complex temporal information," in *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*, 2020, pp. 1459–1468. 11

[26] M. Zhang, S. Wu, X. Yu, Q. Liu, and L. Wang, "Dynamic graph neural networks for sequential recommendation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 5, pp. 4741–4753, 2022. 11

[27] X. Li, F. Yan, X. Zhao, Y. Wang, B. Chen, H. Guo, and R. Tang, "Hamur: Hyper adapter for multi-domain recommendation," in *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, 2023, pp. 1268–1277. 11

[28] Q. Liu, X. Wu, W. Wang, Y. Wang, Y. Zhu, X. Zhao, F. Tian, and Y. Zheng, "Llmemb: Large language model can be a good embedding generator for sequential recommendation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 11, 2025, pp. 12 183–12 191. 11

[29] Y. Hou, A. Zhang, L. Sheng, J. Wu, X. Wang, T.-S. Chua, and J. McAuley, "Towards large generative recommendation: A tokenization perspective," in *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, 2025, pp. 6821–6824. 11

[30] C. M. Ju, L. Collins, L. Neves, B. Kumar, L. Y. Wang, T. Zhao, and N. Shah, "Generative recommendation with semantic ids: A practitioner's handbook," in *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, 2025, pp. 6420–6425. 11

[31] Y. Hou, J. Li, A. Shin, J. Jeon, A. Santhanam, W. Shao, K. Hassani, N. Yao, and J. McAuley, "Generating long semantic ids in parallel for recommendation," in *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, 2025, pp. 956–966. 11

[32] Y. Hou, J. Ni, Z. He, N. Sachdeva, W.-C. Kang, E. H. Chi, J. McAuley, and D. Z. Cheng, "Actionpiece: Contextually tokenizing action sequences for generative recommendation," in *Forty-second International Conference on Machine Learning*, 2025. 12

[33] Y. Yang, Z. Ji, Z. Li, Y. LI, Z. Mo, Y. Ding, K. Chen, Z. Zhang, J. Li, shuanglong li, and L. LIN, "Sparse meets dense: Unified generative recommendations with cascaded sparse-dense representations," in *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. 12

[34] Y. Wang, S. Zhou, J. Lu, Z. Liu, L. Liu, M. Wang, W. Zhang, F. Li, W. Su, P. Wang *et al.*, "Nezha: A zero-sacrifice and hyperspeed decoding architecture for generative recommendations," *arXiv preprint arXiv:2511.18793*, 2025. 12

[35] L. Wu, X. He, X. Wang, K. Zhang, and M. Wang, "A survey on accuracy-oriented neural recommendation: From collaborative filtering to information-rich recommendation," *IEEE transactions on knowledge and data engineering*, vol. 35, no. 5, pp. 4425–4445, 2022. 12

[36] E. Hasan, M. Rahman, C. Ding, J. X. Huang, and S. Raza, "Review-based recommender systems: A survey of approaches, challenges and future perspectives," *ACM Comput. Surv.*, vol. 58, no. 1, Sep. 2025. 12

[37] Z. Cheng, Y. Ding, L. Zhu, and M. Kankanhalli, "Aspect-aware latent factor model: Rating prediction with ratings and reviews," in *Proceedings of the 2018 world wide web conference*, 2018, pp. 639–648. 12

[38] Z. Nie and P. Sun, "Hadsf: Aspect aware semantic control for explainable recommendation," in *Proceedings of the Nineteenth ACM International Conference on Web Search and Data Mining*, 2026, pp. 509–519. 12

[39] C. Chen, M. Zhang, Y. Liu, and S. Ma, "Neural attentional rating regression with review-level explanations," in *Proceedings of the 2018 world wide web conference*, 2018, pp. 1583–1592. 12

[40] X. Dong, J. Ni, W. Cheng, Z. Chen, B. Zong, D. Song, Y. Liu, H. Chen, and G. De Melo, "Asymmetrical hierarchical networks with attentive interactions for interpretable review-based recommendation," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 05, 2020, pp. 7667–7674. 12

[41] X. Wang, I. Ounis, and C. Macdonald, "Leveraging review properties for effective recommendation," in *Proceedings of the Web Conference 2021*, 2021, pp. 2209–2219. 12

[42] C. Shi, B. Hu, W. X. Zhao, and P. S. Yu, "Heterogeneous information network embedding for recommendation," *IEEE transactions on knowledge and data engineering*, vol. 31, no. 2, pp. 357–370, 2018. 12

[43] D. Liu, J. Wu, J. Li, B. Du, J. Chang, and X. Li, "Adaptive hierarchical attention-enhanced gated network integrating reviews for item recommendation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 5, pp. 2076–2090, 2020. 12

[44] N. Sachdeva and J. McAuley, "How useful are reviews for recommendation? a critical review and potential improvements," in *proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*, 2020, pp. 1845–1848. 12

[45] S. Geng, S. Liu, Z. Fu, Y. Ge, and Y. Zhang, "Recommendation as language processing (rlp): A unified pretrain, personalized prompt & predict paradigm (p5)," in *Proceedings of the 16th ACM conference on recommender systems*, 2022, pp. 299–315. 12

[46] H. Lyu, S. Jiang, H. Zeng, Y. Xia, Q. Wang, S. Zhang, R. Chen, C. Leung, J. Tang, and J. Luo, "Llm-rec: Personalized recommendation via prompting large language models," in *Findings of the Association for Computational Linguistics: NAACL 2024*, 2024, pp. 583–612. 12

[47] X. Wang, J. Cui, Y. Suzuki, and F. Fukumoto, "Rdrec: Rationale distillation for llm-based recommendation," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2024, pp. 65–74. 12

[48] M. Assi, S. Hassan, and Y. Zou, "Llm-cure: Llm-based competitor user review analysis for feature enhancement," *ACM Transactions on Software Engineering and Methodology*, 2024. 12

Yingyi Zhang is currently a PhD candidate at the joint program between Dalian University of Technology and City University of Hong Kong . He earned his B.Eng. in Information Management and Information Systems from DUT in 2020. His research focuses on Personalization in RecSys, RAG, and LLMs, and he has published over 20 papers in top-tier venues (*e.g.*, ICLR, NeurIPS, and KDD). He has deployed recommender systems at Meituan, driving real-world impact. More information about him can be found at https://zhang-yingyi.github.io/.

Junyi Li is currently a postdoctoral researcher at the Department of Data Science, City University of Hong Kong. He received his Ph.D. degree from Université de Montréal and Renmin University of China in 2024. His research focuses on trustworthy, reasoning-capable, and evolving AI agents. He has published extensively in leading venues such as NeurIPS, ICLR, and ACL, receiving over 11,000 citations and an h-index of 24. More information about him can be found at https://lijunyi.tech/.

Yejing Wang is currently a Data Science Ph.D. candidate at City University of Hong Kong. He has published over 20 papers in top venues (*e.g.*, WWW, KDD, SIGIR, ICDM), and has deployed systems at scale at Taobao and Xiaohongshu to serve hundreds of millions of users and drive measurable business impact. His work has garnered over 700 citations. He co-organizes tutorials on KDD'25, WWW'25, WSDM'23, WWW'22 about RecSys. More information about him can be found at https://dave-adamswang.github.io/.

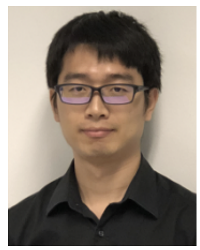
Wenlin Zhang is currently a Data Science Ph.D. candidate at City University of Hong Kong (CityU). He received the B.Eng. and M.Eng. degrees in Computer Science and Computer Technology from Northwestern Polytechnical University in 2020 and 2023, respectively. His research interests include large language models, search agents, and recommender systems. He has published several papers in top-tier venues, including NeurIPS, KDD, WWW, and ICLR. More information can be found at https://wlzhang2020.github.io/.

Xiaowei Qian is currently a Data Science Ph.D. candidate at City University of Hong Kong. He received a B.Eng. degree in Software Engineering from the University of Electronic Science and Technology of China in 2024. His current research interests include LLM Agents and Retrieval-augmented Generation. He has published several papers in conferences, such as ICML, AAAI, and KDD. More information about him can be found at https://xweiq.github.io/.

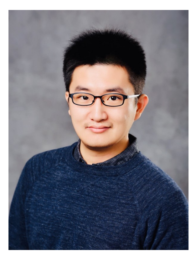
Sheng Zhang is currently a Data Science Ph.D. candidate at City University of Hong Kong. He received his M.Sc. in Data Science from City University of Hong Kong in 2024 and served as an Assistant Engineer at the Hefei Institute of Physical Science from 2024 to 2025. He has published 7 papers in top-tier conferences (e.g., KDD, SIGIR, and AAAI). His research interests include efficient sequential recommendation and agent memory. More information about him can be found at https://szhang-cityu.github.io/.

Yue Feng is currently a Ph.D. student of the School of Economics and Management at Dalian University of Technology. Her research interests primarily focus on emergency scenario-response, emergency resource scheduling and collaboration response modeling. She has published several papers in journals, such as Expert Systems with Applications, Computers & Industrial Engineering, and Systems.

Yichao Wang is a principal researcher at Singapore Search & Recommendation Lab, Huawei. He earned his MS in Software Engineering from Peking University (PKU) in 2018. His research focuses on Retrieval-Augmented Generation, Personalized Agents, and LLMs for Recommendation. He has published over 40 papers in top-tier conferences and journals (ICLR, KDD, AAAI, SIGIR, NeurIPS, WWW, CIKM, WSDM, COLING, TOIS, etc.) and served as a tutor at IJCAI'23, WWW'25.



Yong Liu is currently the Director of Singpoare Search & Recommendation Lab, Huawei. Prior to joining Huawei, he was a Senior Research Scientist at Nanyang Technological University (NTU), a Data Scientist at NTUC Enterprise, and a Research Scientist at Institute for Infocomm Research (I2R), A*STAR, Singapore. He received his Ph.D. degree in Computer Engineering from NTU in 2016 and B.S. degree from University of Science and Technology of China (USTC) in 2008. His current research interests include LLM Agents, RAG, Generative Search & Recommendation. He has been invited as a Area Chair/(Senior) PC member of major conferences, such as ICLR, NeurIPS, KDD, WWW, SIGIR, and reviewer for IEEE/ACM transactions. More information about him can be found at https://stephenliu0423.github.io/.



Xiangyu Zhao is an associate professor of the Department of Data Science at City University of Hong Kong. Prior to CityU, he completed his PhD in 2021 at MSU, MS in 2017 at USTC and BEng in 2014 at UESTC. His current research interests include data mining and machine learning. He has published more than 100 papers in top conferences (*e.g.*, KDD, AAAI, NeurIPS, ICLR) and journals (*e.g.*, Nature npj Digital Medicine, IEEE TKDE). His research has been awarded KDD'25 Best Paper Award Runner Up, ICDM'22 and ICDM'21 Best-ranked Papers, Joint AAAI/ACM SIGAI Doctoral Dissertation Award Nomination, Global Chinese AI Rising Stars (Top 25 in Data Mining). He guided the students to win the NeurIPS Competition 2024 Champion (1/1500+ global teams), KDD Cup 2024 Runner Up (2/500+ global teams), International Exhibition of Inventions Geneva 2023 Silver Medal, and iCAN 2022 Gold Medal. More information about him can be found at https://zhaoxyai.github.io/.

Xianneng Li is currently a Professor at the School of Economics and Management, Dalian University of Technology, where he also serves as the Associate Dean of the School, and Director of the Institute of Systems Engineering. He received the B.E. degree from Nanjing University, China, and the Ph.D. degree from Waseda University, Japan. Previously, he was a JSPS Postdoctoral Fellow and faculty member at Waseda University. His current research interests include machine intelligence and computational intelligence. He has published over 70 papers in top venues, including Marketing Science, IEEE TVEC, ICLR, KDD, and AAAI, etc. He serves on the Editorial Board of Applied Soft Computing and as a PC member for several conferences, including AAAI, GECCO, IEEE CEC, etc. More information about him can be found at https://faculty.dlut.edu.cn/li/en/index.htm.