DOES POSITIONAL ENCODING MATTER FOR SEMANTIC ID-BASED GENERATIVE RECOMMENDATION?

Anonymous authors
Paper under double-blind review

ABSTRACT

Semantic IDs have emerged as an effective representation for generative recommendation, where each item is represented by multiple discrete codes. These codes are typically serialized along the user interaction sequence into a flattened token sequence for generative modeling. However, such a formulation couples two structurally different sources of positional variation: the sequential order of user interactions and the hierarchical order induced by residual quantization. We argue that these two structures should be modeled on their respective positional axes. Based on this insight, we propose SIMPOS, a simple structure-aware positional modeling method that replaces the Transformer backbone's positional encoding with separate interaction-level and residual-level attention biases, requiring no additional learnable parameters. Experiments on four real-world recommendation datasets with different backbones and residual tokenizers show that SIMPOS consistently outperforms existing positional encoding methods, with relative gains of up to $+25.25\%$ in Recall@5. These results indicate that positional encoding matters in semantic ID-based generative recommendation, and that directly inheriting it from the Transformer backbone is suboptimal. Our code and data are available at `https://anonymous.4open.science/r/SimPos-8CFC/`.

1 INTRODUCTION

Recent advances in generative recommendation formulate next-item recommendation as a sequence generation problem (Geng et al., 2022; Hou et al., 2024; Zhai et al., 2024; Bao et al., 2023; Wang et al., 2024b). A key challenge in this paradigm is how to represent discrete items in a form that is both semantically meaningful and suitable for generation. Semantic IDs (SIDs) have emerged as a promising solution by representing each item with a short sequence of discrete codes derived from its semantic representation (Rajput et al., 2023; Hua et al., 2023; Hou et al., 2025; Fu et al., 2026). A typical SID-based generative recommender first encodes item content into continuous semantic embeddings, quantizes them into discrete semantic IDs, and serializes the resulting codes into the user interaction sequence for generative modeling. Among different tokenization schemes, residual quantization has been widely adopted (Lee et al., 2022; Ju et al., 2025; Zheng et al., 2024; Wei et al., 2026), constructing semantic IDs through successive residual refinement.

Despite substantial progress in learning effective semantic IDs, how these multi-code item representations should be positionally modeled in user interaction sequences remains largely overlooked. As illustrated in Figure 1, the residual codes of interacted items are serialized into a single token sequence, with each code assigned an index along the flattened sequence. This serialization, however, maps two distinct positional structures onto a single token-level positional axis. The first is the ***interaction position***, which specifies where an item occurs in the interaction sequence, capturing the sequential order across items. The second is the ***residual position***, which indicates the quantization level of a code, reflecting coarse-to-fine semantic refinement. Consequently, while each semantic code carries both an interaction-level and a residual-level coordinate, conventional positional

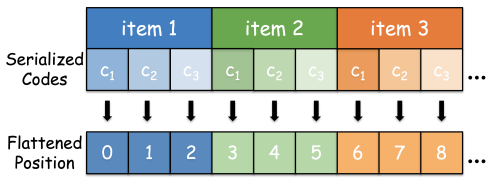


Figure 1: Flattened positional indexing for multi-code semantic IDs. Each code has an interaction position and a residual position, but receives a single index along the flattened sequence.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

modeling treats positions along the flattened sequence uniformly, without explicitly distinguishing these two structural roles.

This distinction becomes particularly important when considering relative positions between semantic codes. On the one hand, codes from the same pair of adjacent items can be one to several positions apart in the flattened sequence, depending on their residual levels. On the other hand, the last code of one item and the first code of the next are one position apart, just like two adjacent codes within the same item. A single token-level position, therefore, cannot tell these structurally different relations apart. These observations suggest that interaction order and residual order should both be preserved, but modeled on separate positional axes rather than conflated into a single token-level position.

Motivated by this insight, we propose SIMPOS, a simple structure-aware positional modeling method for semantic ID-based generative recommendation. Rather than relying on offsets within the flattened one-dimensional sequence, SIMPOS replaces the backbone's positional encoding with two additive attention biases. They penalize the distance between items in the interaction sequence and the distance between residual levels, respectively. Moreover, SIMPOS introduces no additional learnable parameters and can be plugged into both encoder–decoder and decoder-only Transformers without other architectural changes. Experiments on four datasets with two backbones and two residual tokenizers show that SIMPOS consistently outperforms existing positional encoding methods, with relative gains of up to $+25.25\%$ in Recall@5. These results indicate that positional encoding is an important yet overlooked factor in semantic ID-based generative recommendation. Our main contributions are summarized as follows:

- We identify a previously overlooked positional encoding issue in residual SID-based generative recommendation. Flattening multi-code items places interaction order and residual order on a single token-level positional axis, making structurally different relations indistinguishable.
- We propose SIMPOS, a plug-and-play structure-aware positional modeling method that encodes interaction-level and residual-level distances as two separate additive attention biases, without additional learnable parameters.
- We conduct extensive experiments on four real-world datasets using different Transformer backbones and residual tokenizers. SIMPOS consistently outperforms existing positional encoding methods under diverse settings, demonstrating its effectiveness and general applicability.

2 RELATED WORK

2.1 SEMANTIC ID-BASED GENERATIVE RECOMMENDATION

Unlike traditional atomic ID-based recommendation (Kang & McAuley, 2018; Sun et al., 2019; He et al., 2017; Chen et al., 2022), generative recommendation formulates recommendation as a sequence generation task (Li et al., 2024; Hou et al., 2024; Zhai et al., 2024; Li et al., 2023; Wang et al., 2024b). A representative framework is TIGER (Rajput et al., 2023), which represents each item as a sequence of discrete codes learned through residual quantization and autoregressively generates the SID of the next item. Following this paradigm, subsequent studies have explored diverse SID construction methods and generative modeling strategies. On the construction side, LETTER (Wang et al., 2024a) incorporates collaborative signals and code-assignment diversity into tokenization. DIGER (Fu et al., 2026) introduces differentiable semantic indexing to better align tokenization with recommendation objectives. On the generation side, recent methods move beyond autoregressive decoding through parallel decoding (Hou et al., 2025) or diffusion-based generation (Shi et al., 2025; Liu et al., 2026; Mu et al., 2026). While these studies advance SID construction and generation, the positional encoding of the flattened code sequence remains overlooked and is typically inherited from standard Transformer positional encodings. As a result, interaction order and residual order share a one-dimensional token-level positional axis, which motivates us to rethink positional encoding for SID-based generative recommendation.

2.2 POSITIONAL ENCODING IN TRANSFORMERS

Positional encodings in Transformers can be broadly grouped into absolute and relative ones. Absolute positional encodings assign each token a position-specific representation and inject it into the input embeddings (Vaswani et al., 2017; Devlin et al., 2019). In contrast, relative positional encodings model

positional relationships between tokens and have been extensively studied in various forms (Shaw et al., 2018; Dai et al., 2019; Raffel et al., 2020; He et al., 2021; Ke et al., 2021). Specifically, Vaswani et al. (2017) use a fixed, sinusoidal positional encoding composed of alternating sine and cosine functions of varying frequencies. Devlin et al. (2019) utilize learned absolute positional embeddings, adding them directly to the token and segment embeddings before feeding into the Transformer layers. Moreover, Shaw et al. (2018) introduce learned representations of relative positions into self-attention. RoPE (Su et al., 2024) encodes positional information through position-dependent rotations of query and key representations, while ALiBi (Press et al., 2022) directly introduces distance-dependent biases into attention scores. These methods define positions along a single token sequence, and how to model positional information for the multi-code item structure in SID-based generative recommendation remains underexplored. Our work addresses this gap by explicitly modeling both the interaction order across items and the residual order within each item.

3 METHODOLOGY

Let $\mathcal{U}$ and $\mathcal{I}$ denote the user set and item set, respectively. Given a user's chronologically ordered interaction sequence $\mathcal{S}_u = (i_1, i_2, \ldots, i_T)$, where $i_t \in \mathcal{I}$ denotes the t-th interacted item, our goal is to predict the next item i_{T+1}. Following prior work on SID-based generative recommendation (Rajput et al., 2023; Deng et al., 2025), we represent each item as a sequence of discrete semantic codes and formulate next-item prediction as autoregressive generation of the target item's SID.

3.1 PRELIMINARIES

Semantic ID Construction. We focus on residual-based semantic tokenization (e.g., RQ-VAE (Lee et al., 2022), RQ-KMeans (Luo et al., 2025)), which represents each item with an ordered sequence of discrete codes through residual vector quantization. Given an item $i \in \mathcal{I}$, we first obtain its dense semantic embedding $\mathbf{x}_i \in \mathbb{R}^d$ from its content information (e.g., title, category, description). A residual quantizer then hierarchically tokenizes the item embedding using K codebooks, each containing M codewords. The first codebook quantizes the original embedding, while each subsequent codebook quantizes the residual left by the previous levels. The resulting Semantic ID is $\mathbf{c}_i = (c_{i,1}, c_{i,2}, \ldots, c_{i,K})$, where $c_{i,k} \in \{1, \ldots, M\}$ denotes the selected index in the k-th codebook.

Interaction Sequence Representation. Given the interaction sequence $\mathcal{S}_u = (i_1, \ldots, i_T)$, we denote the k-th code of the t-th interacted item i_t as $c_{t,k}$ for simplicity. The semantic IDs of the historical items are then serialized into a single flattened token sequence:

$$\mathcal{C}_u = (c_{1,1}, \ldots, c_{1,K}, c_{2,1}, \ldots, c_{2,K}, \ldots, c_{T,1}, \ldots, c_{T,K}). \tag{1}$$

The resulting TK semantic codes then serve as the input to the Transformer model.

3.2 RETHINKING POSITIONAL STRUCTURE IN SEMANTIC IDS

The serialization in Eq. (1) maps each code $c_{t,k}$, indexed by its interaction position t and residual position k, onto a single position along the flattened sequence, given by $p(t,k) = (t-1)K + k$. For a query–key pair $(c_{t,k}, c_{s,j})$, we denote their relative position along the flattened sequence as

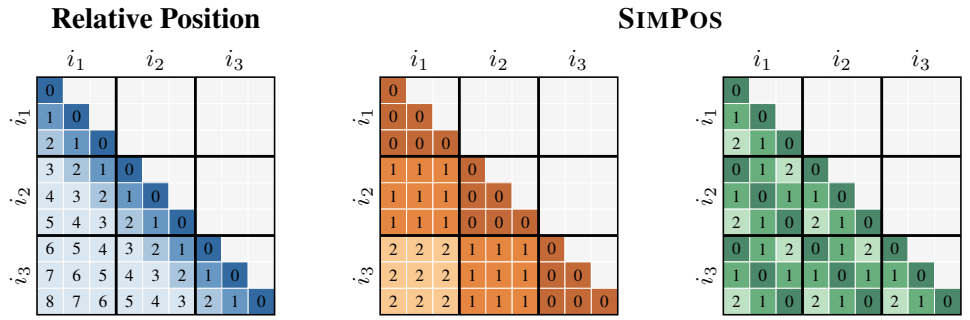
$$r(c_{t,k}, c_{s,j}) = p(t,k) - p(s,j) = (t-s)K + (k-j), \tag{2}$$

where $t - s$ and $k - j$ are their relative differences in interaction and residual positions, respectively.

Eq. (2) shows that flattening collapses two structurally different sources of positional variation into a single scalar. As illustrated in Figure 2(a), this leads to three observations:

(1) Different structural relations are represented on the same positional scale. With $K = 3$, two codes at the same residual level from adjacent items have $r = 3$ ($t - s = 1, k = j$), whereas two adjacent residual codes within the same item have $r = 1$ ($t = s, k - j = 1$). Consequently, one step in interaction position spans K token positions, whereas one step in residual position spans only one.

(2) One interaction-level relation maps to multiple flattened positions. For a pair of adjacent items, the interaction-position difference is fixed at $t - s = 1$. Their codes, however, yield values of r ranging from 1 to 5, depending on the residual levels being compared.



(a) Flattened Token Position (b) Interaction-level Position (c) Residual-level Position

Figure 2: Illustration of relative positional encoding and our proposed SIMPOS for semantic ID-based generative recommendation (e.g., a sequence of 3 items $[i_1, i_2, i_3]$, with each represented by 3 codes). (a) Relative positions are computed along the flattened sequence. SIMPOS explicitly distinguishes two positional structures: (b) interaction-level positions, which capture relative differences across items in the interaction sequence, and (c) residual-level positions, which capture relative differences across residual levels. Only the lower triangle (causal attention) is shown for clarity.

(3) Different structural relations map to the same flattened position. For example, $r(c_{1,2}, c_{1,1}) = r(c_{2,1}, c_{1,3}) = 1$, although the former pair lies within one item and the latter spans two adjacent items. A single token-level position, therefore, cannot tell these two relations apart.

These observations highlight that both interaction order and residual order carry meaningful positional information, but a single token-level position conflates them. The former captures the sequential order of user interactions, while the latter reflects the hierarchical refinement order induced by residual quantization. This motivates a structure-aware positional formulation that models interaction-level and residual-level relations separately.

3.3 SIMPOS: STRUCTURE-AWARE POSITIONAL ENCODING

Based on the above observations, we propose **SIMPOS**, a structure-aware positional encoding approach that decouples interaction-level and residual-level positional relations in SID sequences.

Interaction-level Position. We define the interaction-level positional bias of $(c_{t,k}, c_{s,j})$ as

$$B_{\text{int}}(c_{t,k}, c_{s,j}) = -\lambda_{\text{int}} \cdot |t - s|, \tag{3}$$

where $\lambda_{\text{int}} \geq 0$ controls the strength of the interaction position signal. As shown in Figure 2(b), all code pairs between the same two items receive the same interaction bias, regardless of residual levels. This bias encourages each code to attend more to codes from recent items, reflecting the recency prior commonly used in sequential recommendation (Liu et al., 2018; Wu et al., 2019).

Residual-level Position. Similarly, we define the residual-level positional bias of $(c_{t,k}, c_{s,j})$ as

$$B_{\text{res}}(c_{t,k}, c_{s,j}) = -\lambda_{\text{res}} \cdot |k - j|, \tag{4}$$

where $\lambda_{\text{res}} \geq 0$ controls the strength of the residual position signal. As illustrated in Figure 2(c), the bias depends only on the distance between residual levels, independent of their interaction positions. This bias encourages each code to attend more to codes at nearby residual levels, which capture semantics at similar granularity.

Unified Attention Formulation. When using SIMPOS, we do not add position embeddings in the neural networks. The only modification we apply is after the query–key dot product. Combining the two positional components, the attention logit for the query–key pair $(c_{t,k}, c_{s,j})$ is given by

$$\ell(c_{t,k}, c_{s,j}) = \frac{\mathbf{q}_{t,k}^{\top} \mathbf{k}_{s,j}}{\sqrt{d_h}} + B_{\text{int}}(c_{t,k}, c_{s,j}) + B_{\text{res}}(c_{t,k}, c_{s,j}). \tag{5}$$

where $\mathbf{q}_{t,k}$ and $\mathbf{k}_{s,j}$ denote the query and key representations, and d_h is the attention head dimension. By design, SIMPOS introduces *no additional learnable parameters*. The two coefficients are fixed and shared across all attention heads and layers. As SIMPOS operates directly on attention logits, it can be readily integrated into different Transformer architectures without modifying other model components or the training objective. In this way, SIMPOS measures interaction-level and residual-level distances on separate axes, each with its own scale. The bias between two codes is thus determined by their structural relation rather than their offset in the flattened sequence.

4 EXPERIMENTS

In this section, we conduct experiments to answer the following research questions: **RQ1**: Does SIMPOS outperform existing positional encodings across different datasets, Transformer backbones, and residual tokenizers? **RQ2**: How do the interaction-level and residual-level components each contribute to the performance of SIMPOS? **RQ3**: How sensitive is SIMPOS to the strengths of the two positional biases? **RQ4**: Does SIMPOS remain effective with different semantic embedding models?

4.1 EXPERIMENTAL SETUP

Datasets. We evaluate SIMPOS on four categories from the Amazon Reviews 2023[1] (Hou et al., 2026): "Beauty and Personal Care" (Beauty), "Toys and Games" (Toys), "Sports and Outdoors" (Sports), and "CDs and Vinyl" (CDs). Following prior work (Rajput et al., 2023; Hou et al., 2025), we treat each review as an interaction and sort each user's interactions chronologically. We remove duplicate user–item interactions and apply iterative k-core filtering to ensure sufficient behavioral signals. Table 1 summarizes the statistics of the processed datasets.

Table 1: Statistics of the processed datasets.

Datasets	k-core	# Users	# Items	# Interactions	Density	Average Sequence Length
Beauty	15	8,425	11,509	304,396	0.314%	36.13
Toys	10	41,744	31,132	728,215	0.056%	17.44
Sports	10	17,906	11,247	267,830	0.133%	14.96
CDs	10	22,170	20,089	519,283	0.117%	23.42

Backbones and Tokenizers. To examine the general applicability of SIMPOS, we consider two Transformer backbones and two residual tokenizers. For the encoder–decoder backbone, we adopt T5 (Raffel et al., 2020) following TIGER (Rajput et al., 2023), whose original positional encoding is the bucketed relative position bias. For the decoder-only backbone, we adopt GPT-2 (Radford et al., 2019), whose original positional encoding is the learnable absolute position embedding. For semantic ID construction, we use RQ-VAE (Lee et al., 2022) and RQ-KMeans (Luo et al., 2025). In all settings, we concatenate the title, description, and category of each item and encode them with `Qwen3-Embedding-0.6B`[2] into a 1024-dimensional embedding. The residual tokenizer then quantizes each embedding with $K = 4$ codebooks, each containing $M = 256$ codewords, so that every item is represented by a semantic ID of four codes.

Baselines. Since SIMPOS only modifies how positional information is modeled, we compare it with representative positional encodings (PEs). Within each backbone and tokenizer setting, all methods share the same semantic IDs and training configuration. Specifically, we consider **(1) LPE**, learnable absolute position embeddings added to the input embeddings, the default in GPT-2 (Radford et al., 2019); **(2) RelPE**, bucketed relative position biases added to attention logits, the default in T5 (Raffel et al., 2020); **(3) RoPE** (Su et al., 2024), which rotates query and key representations according to their positions; and **(4) ALiBi** (Press et al., 2022), which adds linear distance-based biases to attention logits. All baselines compute positions along the flattened sequence.

Evaluation. We adopt the widely used leave-last-out evaluation protocol (Kang & McAuley, 2018; Geng et al., 2022; Zheng et al., 2024). For each user, the last item is used for testing, the second-to-last

[1] https://amazon-reviews-2023.github.io/

[2] `https://huggingface.co/Qwen/Qwen3-Embedding-0.6B`

Table 2: Selected coefficients $(\lambda_{\text{int}}, \lambda_{\text{res}})$ of SIMPOS for each setting in Table 3.

Backbone	Tokenizer	Beauty		Toys		Sports		CDs	
		λ_{int}	λ_{res}	λ_{int}	λ_{res}	λ_{int}	λ_{res}	λ_{int}	λ_{res}
T5	RQ-VAE	0.8	0.2	0.8	0.1	0.6	0.2	0.6	0.2
	RQ-KMeans	0.6	0.1	0.8	0.1	0.8	0.1	0.6	0.1
GPT-2	RQ-VAE	0.6	0.2	0.8	0.2	0.6	0.1	0.6	0.1
	RQ-KMeans	0.6	0.1	0.8	0.2	0.6	0.2	0.6	0.1

Table 3: Performance of different positional encodings (PE) on four datasets with two backbones and two tokenizers. All settings use `Qwen3-Embedding-0.6B`[2] to encode item content and $K = 4$ codebooks. Methods within the same backbone and tokenizer group differ only in the PE. In each group, the best result is in **bold**, and the second best is underlined. *Improv.* is the relative gain of SIMPOS over the best baseline. SIMPOS achieves statistically significant improvements over the best baseline under a paired t-test ($p < 0.05$).

Tokenizer	PE	Beauty		Toys		Sports		CDs	
		R@5	N@5	R@5	N@5	R@5	N@5	R@5	N@5
T5 backbone (Original positional encoding is RelPE)									
RQ-VAE	LPE	0.0203	0.0129	0.0181	0.0109	0.0184	0.0122	0.0301	0.0182
	RelPE	0.0211	0.0134	0.0186	0.0118	0.0195	0.0126	0.0304	0.0186
	RoPE	0.0209	0.0132	0.0176	0.0103	0.0198	0.0129	0.0310	0.0192
	ALiBi	0.0213	0.0137	0.0189	0.0112	0.0188	0.0117	0.0310	0.0189
	SimPos	**0.0226**	**0.0151**	**0.0207**	**0.0134**	**0.0248**	**0.0152**	**0.0341**	**0.0213**
	Improv.	*+6.10%*	*+10.22%*	*+9.52%*	*+13.56%*	*+25.25%*	*+17.83%*	*+10.00%*	*+10.94%*
RQ-KMeans	LPE	0.0202	0.0127	0.0208	0.0133	0.0200	0.0130	0.0296	0.0181
	RelPE	0.0213	0.0135	0.0220	0.0142	0.0212	0.0138	0.0310	0.0193
	RoPE	0.0211	0.0138	0.0215	0.0139	0.0206	0.0134	0.0303	0.0200
	ALiBi	0.0216	0.0141	0.0211	0.0144	0.0219	0.0139	0.0306	0.0188
	SimPos	**0.0268**	**0.0170**	**0.0230**	**0.0158**	**0.0261**	**0.0171**	**0.0330**	**0.0213**
	Improv.	*+24.07%*	*+20.57%*	*+4.55%*	*+9.72%*	*+19.18%*	*+23.02%*	*+6.45%*	*+6.50%*
GPT-2 backbone (Original positional encoding is LPE)									
RQ-VAE	LPE	0.0198	0.0128	0.0170	0.0102	0.0212	0.0137	0.0285	0.0178
	RelPE	0.0198	0.0129	0.0164	0.0097	0.0210	0.0135	0.0277	0.0188
	RoPE	0.0204	0.0122	0.0162	0.0100	0.0203	0.0120	0.0275	0.0171
	ALiBi	0.0186	0.0130	0.0166	0.0099	0.0207	0.0128	0.0267	0.0169
	SimPos	**0.0230**	**0.0149**	**0.0184**	**0.0120**	**0.0231**	**0.0155**	**0.0318**	**0.0202**
	Improv.	*+12.75%*	*+14.62%*	*+8.24%*	*+17.65%*	*+8.96%*	*+13.14%*	*+11.58%*	*+7.45%*
RQ-KMeans	LPE	0.0206	0.0134	0.0195	0.0125	0.0215	0.0144	0.0302	0.0191
	RelPE	0.0198	0.0132	0.0200	0.0118	0.0214	0.0154	0.0289	0.0202
	RoPE	0.0208	0.0139	0.0190	0.0123	0.0209	0.0152	0.0283	0.0190
	ALiBi	0.0208	0.0131	0.0189	0.0123	0.0201	0.0136	0.0291	0.0195
	SimPos	**0.0227**	**0.0151**	**0.0216**	**0.0138**	**0.0252**	**0.0167**	**0.0336**	**0.0213**
	Improv.	*+9.13%*	*+8.63%*	*+8.00%*	*+10.40%*	*+17.21%*	*+8.44%*	*+11.26%*	*+5.45%*

item for validation, and the remaining interactions for training. We report Recall@5 (R@5) and NDCG@5 (N@5).

Implementation Details. For the backbones, we use a 4-layer, 6-head T5 encoder–decoder and a 4-layer, 4-head GPT-2 decoder. Both architectures share a hidden dimension of 128, and we keep the most recent 50 interactions of each user. All models are trained with AdamW at a learning rate of 3×10^{-4} and a batch size of 512 for up to 200 epochs, with early stopping based on validation N@5. For SIMPOS, λ_{int} and λ_{res} are searched over $\{0.4, 0.6, 0.8\}$ and $\{0.1, 0.2, 0.3\}$. Table 2 lists the coefficients selected for each setting. During inference, we use beam search with a beam size of 30. All experiments are conducted on a single NVIDIA A100 (80GB) GPU.

4.2 Overall Performance (RQ1)

Table 3 reports the performance of different positional encodings under two backbones and two tokenizers. We have the following observations.

First, SimPos consistently achieves the best performance across all settings. Across the 32 combinations of datasets, metrics, backbones, and tokenizers, SimPos outperforms every baseline PE. Compared with the strongest baseline in each group, SimPos achieves relative gains of 4.55% to 25.25% in R@5 and 5.45% to 23.02% in N@5, with average gains of 12.02% and 12.38%, respectively. Since the compared variants differ only in positional encoding, these improvements can be attributed to how positional information is modeled. Moreover, SimPos introduces no additional learnable parameters, so the gains do not stem from increased model capacity.

Second, among existing PEs, the best choice varies across settings. None of the four baseline PEs outperforms the others across all settings, and their ranking changes with the dataset, backbone, and tokenizer. Notably, the original PE of each backbone is often not the best baseline. Under the T5 backbone, its default RelPE is the best baseline in only 3 of 16 cases. Under the GPT-2 backbone, its default LPE is the best baseline in only 8 of 16 cases. We attribute this to a shared limitation of these methods. All of them define positions along the flattened token sequence, which conflates interaction order and residual order as discussed in Section 3.2. Therefore, replacing one flattened positional encoding with another yields limited and inconsistent changes. In contrast, SimPos models the two structures on separate positional axes and brings consistent improvements over all of them.

Third, SimPos generalizes across backbones and tokenizers. SimPos improves both the encoder–decoder T5 and the decoder-only GPT-2, with average relative gains of 13.59% and 10.81%, respectively. The gains are also comparable between RQ-VAE (12.36%) and RQ-KMeans (12.04%). These results indicate that the benefit of SimPos does not depend on a specific architecture or quantization method. This is consistent with our motivation, since both tokenizers produce semantic IDs with the same interaction-level and residual-level structure, which SimPos models regardless of the backbone.

4.3 Further Analysis

We further analyze SimPos from three aspects. We first examine the contribution of each positional component (RQ2), then study the sensitivity to the bias strengths (RQ3), and finally evaluate the robustness of SimPos to different item embedding models (RQ4).

4.3.1 Effect of Positional Components (RQ2)

To examine the contribution of each component, we remove one bias at a time. Setting $\lambda_{\text{int}} = 0$ keeps only the residual-level bias, and setting $\lambda_{\text{res}} = 0$ keeps only the interaction-level bias, which correspond to the leftmost points in Figures 3 and 4.

Each component is beneficial on its own. Both single-component variants outperform NoPE in all settings and with both tokenizers. Notably, the residual-only variant improves over NoPE even though it carries no information about interaction order. For example, on Toys with the T5 backbone, it raises R@5 from about 0.013 to 0.017 with RQ-VAE and from about 0.014 to 0.020 with RQ-KMeans. This suggests that the residual structure within semantic IDs provides useful positional information on its own, even without interaction order.

The two components are complementary. The best results are always achieved when both coefficients are positive, and they exceed either single-component variant. This is consistent with our analysis in Section 3.2, since interaction order and residual order describe different structures of the flattened sequence and neither can replace the other.

Positional modeling matters more for the encoder–decoder backbone. The gap between NoPE and SimPos is considerably larger on T5 than on GPT-2. For example, on Toys with RQ-VAE, NoPE reaches about 0.013 in R@5 on T5, compared with about 0.022 for the best SimPos configuration, whereas the corresponding values on GPT-2 are about 0.015 and 0.019. A likely reason is that the bidirectional encoder of T5 is insensitive to input order without positional encoding, while the causal attention of GPT-2 implicitly captures part of the order information (Haviv et al., 2022). Even so, SimPos still brings clear gains over NoPE on GPT-2, indicating that implicit order information is insufficient to capture the structure of semantic IDs.

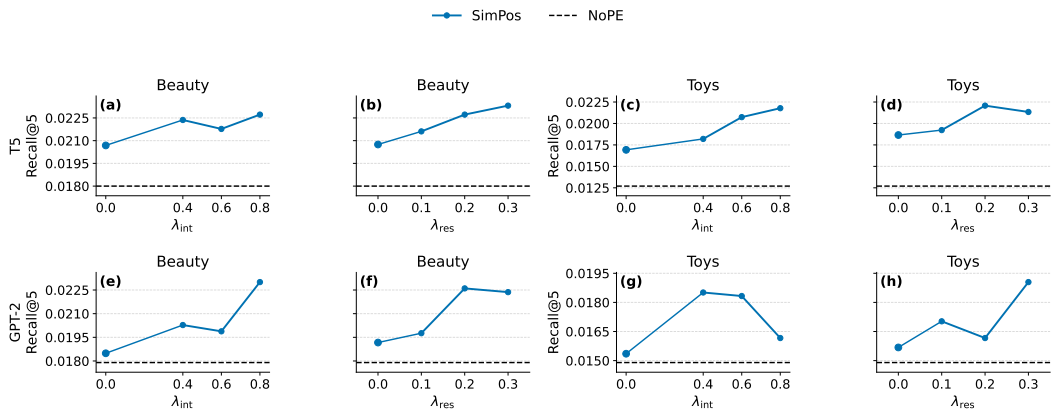


Figure 3: Effect of the interaction-level strength λ_{int} and the residual-level strength λ_{res} on R@5 on Beauty and Toys with RQ-VAE. We vary $\lambda_{\text{int}} \in \{0, 0.4, 0.6, 0.8\}$ with $\lambda_{\text{res}} = 0.1$, and $\lambda_{\text{res}} \in \{0, 0.1, 0.2, 0.3\}$ with $\lambda_{\text{int}} = 0.8$. Setting either coefficient to 0 removes the corresponding component. NoPE sets $\lambda_{\text{int}} = \lambda_{\text{res}} = 0$, which removes all positional encodings.

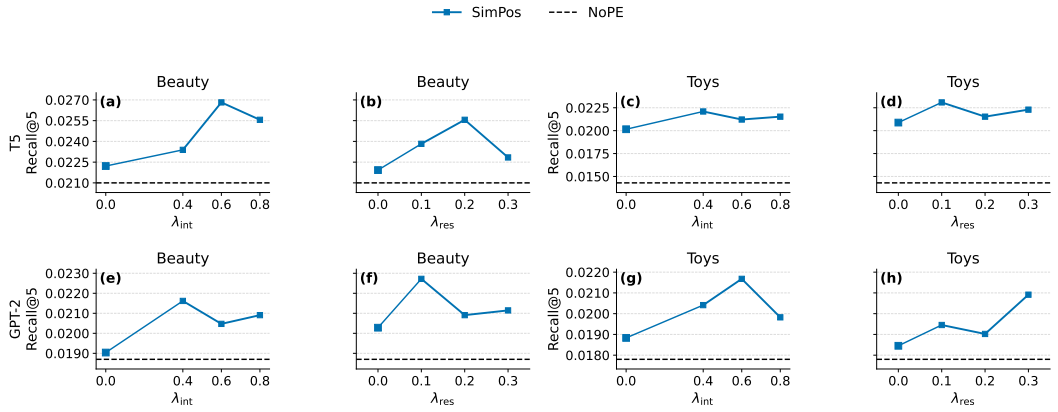


Figure 4: Effect of the interaction-level strength λ_{int} and the residual-level strength λ_{res} on R@5 on Beauty and Toys with RQ-KMeans. We vary $\lambda_{\text{int}} \in \{0, 0.4, 0.6, 0.8\}$ with $\lambda_{\text{res}} = 0.1$, and $\lambda_{\text{res}} \in \{0, 0.1, 0.2, 0.3\}$ with $\lambda_{\text{int}} = 0.8$. Setting either coefficient to 0 removes the corresponding component. NoPE sets $\lambda_{\text{int}} = \lambda_{\text{res}} = 0$, which removes all positional encodings.

4.3.2 Sensitivity to Positional Bias Strengths (RQ3)

Figures 3 and 4 also show how performance varies with the strength of each bias.

Performance peaks at moderate coefficients. With both tokenizers, performance generally improves as either coefficient increases from zero and peaks at moderate values, with $\lambda_{\text{int}} \in [0.4, 0.8]$ and $\lambda_{\text{res}} \in [0.1, 0.3]$. In some settings, an overly large coefficient reduces performance, possibly because a strong bias suppresses attention to distant items or residual levels. For example, with RQ-KMeans on Beauty, R@5 on T5 decreases when λ_{res} increases from 0.2 to 0.3. Nevertheless, all configurations with positive coefficients outperform NoPE across both backbones, tokenizers, and datasets.

A strong recency preference is beneficial. The preferred values of λ_{int} are relatively large. With $\lambda_{\text{int}} = 0.8$, the attention logit to a code five items earlier is reduced by 4, so attention concentrates on the most recent interactions. The effectiveness of such a strong bias suggests that next-item prediction on these datasets relies mainly on recent interactions, which SimPos encodes explicitly.

The two coefficients operate at different scales. The effective range of λ_{res} is smaller than that of λ_{int}, since residual distances are bounded by the number of codes per item, whereas interaction distances grow with the sequence length.

Table 4: Performance of different positional encodings (PE) with the T5 backbone, using `sentence-T5-XL` (Ni et al., 2022) as the item embedding model. All other settings are the same as in Table 3. In each group, the best result is in **bold**, and the second best is underlined. *Improv.* is the relative gain of SIMPOS over the best baseline.

Tokenizer	PE	Beauty		Toys		Sports		CDs	
		R@5	N@5	R@5	N@5	R@5	N@5	R@5	N@5
T5 backbone (Original positional encoding is RelPE)									
RQ-VAE	LPE	0.0211	0.0132	0.0194	0.0125	0.0201	0.0131	0.0278	0.0175
	RelPE	0.0217	0.0137	0.0200	0.0128	0.0208	0.0135	0.0291	0.0182
	RoPE	0.0204	0.0129	0.0189	0.0120	0.0195	0.0127	0.0274	0.0171
	ALiBi	0.0213	0.0130	0.0196	0.0122	0.0213	0.0137	0.0282	0.0187
	SimPos	**0.0234**	**0.0152**	**0.0212**	**0.0138**	**0.0236**	**0.0156**	**0.0351**	**0.0225**
	Improv.	*+7.83%*	*+10.95%*	*+6.00%*	*+7.81%*	*+10.80%*	*+13.87%*	*+20.62%*	*+20.32%*
RQ-KMeans	LPE	0.0208	0.0131	0.0197	0.0126	0.0199	0.0133	0.0326	0.0203
	RelPE	0.0215	0.0135	0.0203	0.0130	0.0207	0.0137	0.0342	0.0210
	RoPE	0.0202	0.0127	0.0191	0.0122	0.0194	0.0129	0.0321	0.0197
	ALiBi	0.0220	0.0139	0.0186	0.0119	0.0214	0.0141	0.0332	0.0208
	SimPos	**0.0242**	**0.0151**	**0.0231**	**0.0150**	**0.0252**	**0.0166**	**0.0363**	**0.0228**
	Improv.	*+10.00%*	*+8.63%*	*+13.79%*	*+15.38%*	*+17.76%*	*+17.73%*	*+6.14%*	*+8.57%*

4.3.3 EFFECT OF SEMANTIC EMBEDDING MODELS (RQ4)

To examine whether the gains of SIMPOS depend on a specific embedding model, we replace `Qwen3-Embedding-0.6B`[2] with `sentence-T5-XL`[3], the item encoder adopted by TIGER (Rajput et al., 2023), and keep all other settings unchanged. Table 4 reports the results with the T5 backbone, which can be compared with the T5 results in Table 3.

SIMPOS remains the best positional encoding with a different embedding model. It achieves average relative gains of 12.28% with RQ-VAE and 12.25% with RQ-KMeans, which are close to those obtained with `Qwen3-Embedding-0.6B` (12.93% and 14.26%, respectively).

Neither embedding model is consistently better across datasets. For example, with RQ-KMeans, `Qwen3-Embedding-0.6B` yields higher R@5 on Beauty (0.0268 vs. 0.0242), whereas `sentence-T5-XL` yields higher R@5 on CDs (0.0363 vs. 0.0330). In contrast, the gains of SIMPOS remain stable, since the embedding model determines the semantic content of the codes, whereas SIMPOS addresses how the codes are positionally modeled. Therefore, the gains of SIMPOS do not depend on a particular embedding model.

5 CONCLUSION

In this paper, we study positional encoding in semantic ID-based generative recommendation. We show that serializing multi-code items into a flattened sequence places interaction order and residual order on a single token-level positional axis, making structurally different relations indistinguishable. To address this issue, we propose SIMPOS, a plug-and-play positional modeling method that encodes interaction-level and residual-level distances as two separate additive attention biases, without additional learnable parameters. Experiments on four real-world datasets with two Transformer backbones and two residual tokenizers show that SIMPOS consistently outperforms existing positional encoding methods. These findings suggest that positional encoding deserves explicit design in semantic ID-based generative recommendation, rather than being directly inherited from the backbone.

Limitations and Future Work. SIMPOS is designed for residual semantic IDs, whose codes follow a coarse-to-fine order. A natural next step is to extend structure-aware positional modeling to non-hierarchical semantic IDs, such as product-quantized IDs (Hou et al., 2025). In addition, the two bias coefficients are fixed hyperparameters shared across attention heads and layers. Learning these coefficients or making them head-specific may further improve flexibility. Finally, evaluating SIMPOS on industrial-scale data and larger backbones remains future work.

[3] https://huggingface.co/sentence-transformers/sentence-t5-xl

AI USE STATEMENT

Generative AI tools were used to assist with literature review and the refinement of experimental interpretation and presentation. They were also used to assist with drafting and editing parts of the manuscript for clarity, organization, and readability. All AI-assisted content, including technical statements, literature references, and interpretations of experimental results, was reviewed and verified by the authors. The research methodology, experiments, and final conclusions were determined and validated by the authors. We take full responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the model architecture, data preprocessing, training procedure, and evaluation protocol in § 3 and § 4.1. Code is also released to further facilitate reproducibility.

REFERENCES

Keqin Bao, Jizhi Zhang, Yang Zhang, Wenjie Wang, Fuli Feng, and Xiangnan He. Tallrec: An effective and efficient tuning framework to align large language model with recommendation. In *Proceedings of the 17th ACM conference on recommender systems*, pp. 1007–1014, 2023.

Huiyuan Chen, Yusan Lin, Menghai Pan, Lan Wang, Chin-Chia Michael Yeh, Xiaoting Li, Yan Zheng, Fei Wang, and Hao Yang. Denoising self-attentive sequential recommendation. In *Proceedings of the 16th ACM conference on recommender systems*, pp. 92–101, 2022.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G Carbonell, Quoc Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pp. 2978–2988, 2019.

Jiaxin Deng, Shiyao Wang, Kuo Cai, Lejian Ren, Qigen Hu, Weifeng Ding, Qiang Luo, and Guorui Zhou. Onerec: Unifying retrieve and rank with generative recommender and iterative preference alignment. *arXiv preprint arXiv:2502.18965*, 2025.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pp. 4171–4186, 2019.

Junchen Fu, Xuri Ge, Alexandros Karatzoglou, Ioannis Arapakis, Suzan Verberne, Joemon M. Jose, and Zhaochun Ren. Differentiable semantic id for generative recommendation. In *Proceedings of the 49th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 369–379, 2026.

Shijie Geng, Shuchang Liu, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. Recommendation as language processing (rlp): A unified pretrain, personalized prompt & predict paradigm (p5). In *Proceedings of the 16th ACM conference on recommender systems*, pp. 299–315, 2022.

Adi Haviv, Ori Ram, Ofir Press, Peter Izsak, and Omer Levy. Transformer language models without positional encodings still learn positional information. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 1382–1390, 2022.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=XPZIaotutsD`.

Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*, pp. 173–182, 2017.

Yupeng Hou, Junjie Zhang, Zihan Lin, Hongyu Lu, Ruobing Xie, Julian McAuley, and Wayne Xin Zhao. Large language models are zero-shot rankers for recommender systems. In *European conference on information retrieval*, pp. 364–381. Springer, 2024.

Yupeng Hou, Jiacheng Li, Ashley Shin, Jinsung Jeon, Abhishek Santhanam, Wei Shao, Kaveh Hassani, Ning Yao, and Julian 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*, pp. 956–966, 2025.

Yupeng Hou, Jiacheng Li, Xiangjun Fu, Zhankui He, An Yan, Xiusi Chen, and Julian McAuley. Bridging language and items for retrieval and recommendation: Benchmarking llms as semantic encoders. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3251–3265, 2026.

Wenyue Hua, Shuyuan Xu, Yingqiang Ge, and Yongfeng Zhang. How to index item ids for recommendation foundation models. In *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region*, pp. 195–204, 2023.

Clark Mingxuan Ju, Liam Collins, Leonardo Neves, Bhuvesh Kumar, Louis Yufeng Wang, Tong Zhao, and Neil Shah. Generative recommendation with semantic ids: a practitioner's handbook. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, pp. 6420–6425, 2025.

Wang-Cheng Kang and Julian McAuley. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*, pp. 197–206. IEEE, 2018.

Guolin Ke, Di He, and Tie-Yan Liu. Rethinking positional encoding in language pre-training. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=09-528y2Fgf.

Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and Wook-Shin Han. Autoregressive image generation using residual quantization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11523–11532, 2022.

Jinming Li, Wentao Zhang, Tian Wang, Guanglei Xiong, Alan Lu, and Gerard Medioni. Gpt4rec: A generative framework for personalized recommendation and user interests interpretation. *arXiv preprint arXiv:2304.03879*, 2023.

Lei Li, Yongfeng Zhang, Dugang Liu, and Li Chen. Large language models for generative recommendation: A survey and visionary discussions. In *Proceedings of the 2024 joint international conference on computational linguistics, language resources and evaluation (LREC-COLING 2024)*, pp. 10146–10159, 2024.

Qiao Liu, Yifu Zeng, Refuoe Mokhosi, and Haibin Zhang. Stamp: short-term attention/memory priority model for session-based recommendation. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 1831–1839, 2018.

Zhao Liu, Yichen Zhu, Yiqing Yang, Xiao Lv, Guoping Tang, Rui Huang, Qiang Luo, Ruiming Tang, and Guorui Zhou. Diffgrm: Diffusion-based generative recommendation model. In *Proceedings of the ACM Web Conference 2026*, pp. 5853–5864, 2026.

Xinchen Luo, Jiangxia Cao, Tianyu Sun, Jinkai Yu, Rui Huang, Wei Yuan, Hezheng Lin, Yichen Zheng, Shiyao Wang, Qigen Hu, et al. Qarm: Quantitative alignment multi-modal recommendation at kuaishou. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, pp. 5915–5922, 2025.

Lingyu Mu, Hao Deng, Haibo Xing, Jinxin Hu, Yu Zhang, Xiaoyi Zeng, and Jing Zhang. Masked diffusion generative recommendation. *arXiv preprint arXiv:2601.19501*, 2026.

Jianmo Ni, Gustavo Hernandez Abrego, Noah Constant, Ji Ma, Keith Hall, Daniel Cer, and Yinfei Yang. Sentence-t5: Scalable sentence encoders from pre-trained text-to-text models. In *Findings of the association for computational linguistics: ACL 2022*, pp. 1864–1874, 2022.

Ofir Press, Noah Smith, and Mike Lewis. Train short, test long: Attention with linear biases enables input length extrapolation. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=R8sQPpGCv0.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan Hulikal Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Tran, Jonah Samost, et al. Recommender systems with generative retrieval. *Advances in Neural Information Processing Systems*, 36:10299–10315, 2023.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pp. 464–468, 2018.

Teng Shi, Chenglei Shen, Weijie Yu, Shen Nie, Chongxuan Li, Xiao Zhang, Ming He, Yan Han, and Jun Xu. Llada-rec: Discrete diffusion for parallel semantic id generation in generative recommendation. *arXiv preprint arXiv:2511.06254*, 2025.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pp. 1441–1450, 2019.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Wenjie Wang, Honghui Bao, Xinyu Lin, Jizhi Zhang, Yongqi Li, Fuli Feng, See-Kiong Ng, and Tat-Seng Chua. Learnable item tokenization for generative recommendation. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pp. 2400–2409, 2024a.

Ye Wang, Jiahao Xun, Minjie Hong, Jieming Zhu, Tao Jin, Wang Lin, Haoyuan Li, Linjun Li, Yan Xia, Zhou Zhao, et al. Eager: Two-stream generative recommender with behavior-semantic collaboration. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 3245–3254, 2024b.

Yibiao Wei, Jie Zou, Pengfei Zhang, Xiao Ao, Weikang Guo, Zeyu Ma, and Yang Yang. CARD: non-uniform quantization of visual semantic unit for generative recommendation. In *Proceedings of the 49th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2026*, pp. 1959–1970. ACM, 2026.

Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. Session-based recommendation with graph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 346–353, 2019.

Jiaqi Zhai, Lucy Liao, Xing Liu, Yueming Wang, Rui Li, Xuan Cao, Leon Gao, Zhaojie Gong, Fangda Gu, Michael He, et al. Actions speak louder than words: trillion-parameter sequential transducers for generative recommendations. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 58484–58509, 2024.

Bowen Zheng, Yupeng Hou, Hongyu Lu, Yu Chen, Wayne Xin Zhao, Ming Chen, and Ji-Rong Wen. Adapting large language models by integrating collaborative semantics for recommendation. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*, pp. 1435–1448. IEEE, 2024.