ALIGNING WHERE IT MATTERS: CRITICAL-BRANCH PREFERENCE OPTIMIZATION FOR LLM-BASED GENERATIVE RECOMMENDATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Generative recommendation increasingly represents items with hierarchical Semantic IDs (SIDs) and formulates next-item prediction as autoregressive token generation. While pretrained large language models (LLMs) have shown strong potential for SID-based recommendation, how they exploit the internal structure of SIDs remains insufficiently understood. In this work, we conduct a fine-grained analysis of hierarchical SIDs in LLM-based generative recommendation. We find that the advantage of pretrained LLMs is highly concentrated at early SID levels, which are both sufficient and necessary for recovering item semantics and are more critical for preserving behavior-relevant information. This observation motivates us to explicitly optimize early SID decisions through preference learning. However, we show that standard Direct Preference Optimization (DPO) is mismatched with autoregressive SID generation: suffix tokens introduce both margin-scale and gradient-direction interference, such that improvement of the critical decision cannot be guaranteed. To address this issue, we propose Critical-Branch Direct Preference Optimization (CB-DPO), which applies preference supervision directly at the first-divergence position of each preference pair, providing a cleaner optimization signal and guaranteeing an increase in the critical margin. Experiments on KuaiRec and KuaiRand-1K demonstrate the effectiveness of CB-DPO across both LLM-based and LLM-free generative recommendation models.[1]

1 INTRODUCTION

Generative recommendation has recently emerged as a promising paradigm for sequential recommendation (Rajput et al., 2023; Zhou et al., 2025b; Deng et al., 2025). Instead of ranking items from a predefined candidate set, it formulates next-item prediction as auto-regressive generation over discrete item identifiers. A particularly influential representation is the Semantic ID (SID), which represents each item as a short sequence of discrete codes obtained by progressively quantizing its semantic representation (Van Den Oord et al., 2017; Lee et al., 2022; Rajput et al., 2023). Compared with atomic item IDs, hierarchical SIDs provide a structured and compact generation space and enable semantic sharing among related items. As a result, SID-based item generation has been increasingly adopted in generative recommendation, providing a natural interface for modeling user behavior and item prediction within a unified token-generation framework.

Meanwhile, the rapid development of large language models (LLMs) has motivated a growing line of work that uses pretrained LLMs as generative recommenders (Zheng et al., 2024; Wu et al., 2024; Qu et al., 2025), aiming to transfer their rich semantic knowledge and strong sequence modeling capabilities to recommendation. Recent studies have explored incorporating recommendation-specific item identifiers into pretrained LLMs and aligning these discrete tokens with semantic and collaborative information (Lin et al., 2025; Ye et al., 2026; Lin et al., 2026; He et al., 2026a). Despite these advances, hierarchical SIDs are still largely treated as complete item representations, leaving *how pretrained LLMs interact with their internal structure* insufficiently understood. In particular, it remains unclear whether different SID levels play comparable roles in LLM-based recommendation, or whether the model relies more heavily on certain hierarchical decisions.

[1] Anonymous code: `https://anonymous.4open.science/r/iclr27_submit_code-C647`

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.

In this work, we conduct a fine-grained study of hierarchical SIDs from the perspective of LLM-based generative recommendation. We first adapt a pretrained Qwen3 model (Yang et al., 2025) to newly introduced SID tokens through semantic warm-up and subsequently train it for next-item generation. Compared with the LLM-free TIGER baseline, the LLM-based model achieves clearly stronger performance. More importantly, we find that this advantage is *highly non-uniform across SID levels*: the gains are concentrated primarily at the early SID positions. We further investigate this asymmetry from both semantic reconstruction and sequential recommendation perspectives, showing that early SID levels are both sufficient and necessary for recovering item semantics and are more critical for preserving behavior-relevant information while later SIDs contribute more to specifying the exact item and can be reliably recovered once the preceding prefix is correctly predicted.

The above observations inspire us that the key to further improving recommendation performance lie in improving the early SID decisions. However, the standard next-token prediction objective provides no explicit supervision for distinguishing preferred tokens from undesirable alternatives at these critical positions. A natural approach is to construct preference pairs that diverge at early SID levels and apply Direct Preference Optimization (DPO) (Rafailov et al., 2023) to explicitly favor the preferred branches. Unfortunately, we find that the mismatch between the DPO objective and auto-regressive SID generation introduces both *margin-scale* and *gradient-direction interference* into the increase of the critical margin, resulting in that optimization at the critical decision cannot be guaranteed.

To address this issue, we propose **Critical-Branch Direct Preference Optimization (CB-DPO)**, which applies preference supervision directly at the first-divergence position of each preference pair. By removing suffix margins from the preference objective, CB-DPO provides a cleaner optimization signal targeted at the critical branching decision, thereby guaranteeing an increase in the critical margin. In practice, we focus more on preference pairs whose first divergence occurs within the first two SID levels and adopt a stratified sampling strategy to maintain stable proportions of different divergence levels within each mini-batch, promoting more stable training. Figure 1 provides an overview of our key observations and the motivation and design of CB-DPO.

Our contributions are summarized as follows:

- We provide a fine-grained analysis of hierarchical SIDs in LLM-based generative recommendation, revealing that the advantage of pretrained LLMs is concentrated at early SID levels, which carry substantially more semantic and behavior-relevant information.
- We identify a mismatch between standard DPO and auto-regressive SID generation, and theoretically show that suffix tokens introduce both margin-scale and gradient-direction interference, preventing guaranteed improvement of the critical margin.
- We propose Critical-Branch Direct Preference Optimization (CB-DPO), which directly optimizes the first-divergence decision and guarantees an increase in the critical margin. Experiments on KuaiRec and KuaiRand-1K demonstrate its effectiveness across both LLM-based and LLM-free generative recommendation models.

2 Non-uniform SID-Level Roles in LLM-based Recommendation

Recent studies have adapted pretrained LLMs to SID-based generative recommendation by incorporating recommendation-specific SID tokens into their representation and generation space (Zheng et al., 2024; Lin et al., 2025; Ye et al., 2026). However, *how pretrained LLMs utilize the hierarchical structure of these SIDs* remains insufficiently understood. In this section, we conduct a fine-grained analysis to characterize the roles of different SID levels in LLM-based generative recommendation from both semantic and behavioral perspectives.

2.1 LLM Advantages Concentrate at Early SID Levels

To investigate the extent to which a pretrained LLM can acquire semantic understanding of SID tokens at different hierarchical levels, we adopt Qwen3-1.7B-Base (Yang et al., 2025) as the backbone and adapt it to the newly introduced SID tokens before modeling user behavior sequences. We first construct hierarchical SIDs for items using RQ-VAE (Lee et al., 2022; Rajput et al., 2023) and denote an SID sequence as $\boldsymbol{s} = [\mathrm{sid}_0, \mathrm{sid}_1, \ldots, \mathrm{sid}_{L-1}]$, where $\mathrm{sid}_i \in \mathcal{C}_i$ is the selected codeword

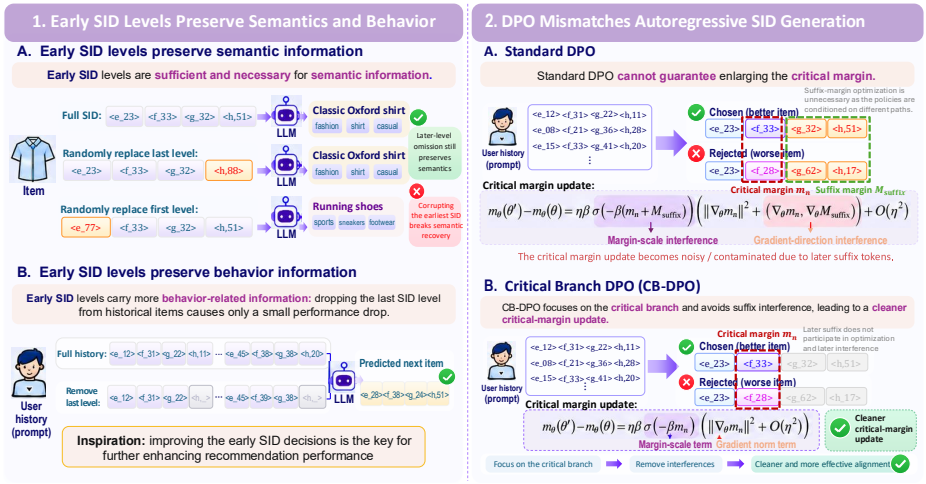


Figure 1: Overview of our main findings and the motivation of the proposed CB-DPO.

from the i-th codebook $\mathcal{C}_i$. To accommodate the learned SIDs, we augment the Qwen3 tokenizer vocabulary with dedicated tokens for all SID codes, together with two special item-boundary tokens (Zheng et al., 2024). This introduces a total of $\sum_{i=0}^{L-1} |\mathcal{C}_i| + 2$ new tokens. The corresponding newly added embedding rows are initialized according to the mean and variance of the original token embedding matrix. During subsequent training, we freeze the original token embedding rows and the pretrained backbone parameters, while optimizing the newly introduced SID embeddings together with LoRA adapters (Hu et al., 2021).

Before training on user behavior sequences, we further introduce a two-phase semantic warm-up to provide the model with an initial understanding of the newly added SID vocabulary:

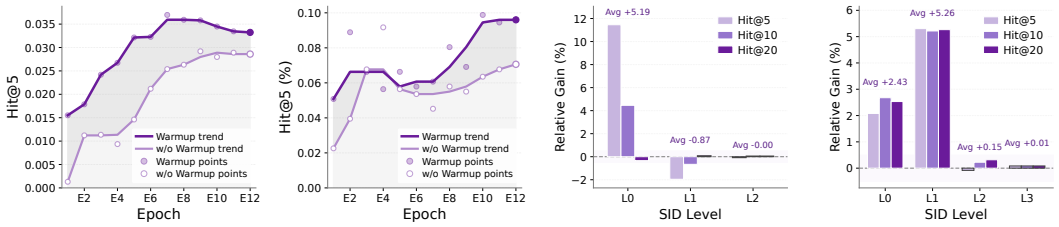
Phase I: Context2SID Warm-up. We first perform Context2SID training task, where the model is given the semantic $\mathbf{context}$ of an item and learns to generate its SID sequence $\boldsymbol{s}$. For each short-video item, the context consists of the hierarchical category path together with a caption augmented with the associated tags, thereby covering both coarse-grained categorical information and finer-grained semantic descriptions. This phase provides the model with its first explicit exposure to the newly introduced SID tokens and establishes an initial association between natural-language item semantics and the discrete SID space. Examples are shown in Section D.

Phase II: SID2Context Warm-up. We then introduce the inverse SID2Context task, where the model is given $\boldsymbol{s}$ and trained to reconstruct the corresponding semantic context. In contrast to Context2SID, which primarily teaches the model how semantic information maps into the discrete SID space, SID2Context encourages the model to associate the SID sequence itself with recoverable item semantics. To mitigate forgetting of the learned mapping learned in the previous phase, we mix the SID2Context data with the Context2SID data at an equal ratio within each training batch.

The warm-up stage itself does not directly optimize the recommendation objective, instead it provides the pretrained model with an initial semantic awareness of the newly introduced SID tokens. Thus we further train the model on user interaction sequences for next-item generation.

Phase III: Seq2Item Training. We formulate the sequential recommendation task as Seq2Item training. Specifically, given a sequence of SIDs corresponding to a user's historical effective interactions, the model is trained to predict the SID of the next item that receives high-value positive feedback. Unlike the preceding warm-up phases, Seq2Item introduces user behavioral signals and requires the model to capture sequential preference patterns over the SID space.

In addition to training the pretrained Qwen3 model through the above stages, we adopt the classical TIGER model (Rajput et al., 2023) as an LLM-free baseline, which is trained with exactly the same SID codebook configuration and Seq2Item data for comparison. We conduct experiments on the preprocessed KuaiRec (Gao et al., 2022a) and KuaiRand-1K (Gao et al., 2022b) datasets. For



(a) KuaiRec (b) KuaiRand-1K (c) KuaiRec (d) KuaiRand-1K

Figure 2: Analysis of Qwen3 and TIGER on KuaiRec and KuaiRand-1K. (a–b) Hit@5 across Seq2Item training epochs for Qwen3 with and without semantic warm-up. (c–d) Relative improvement of Qwen3 over TIGER at different SID levels under teacher forcing, evaluated by Hit@5/10/20.

KuaiRec, we use codebooks with sizes $[256, 256, 256]$, while KuaiRand-1K uses four codebooks with sizes $[1024, 512, 256, 64]$.

Observation I. LLM-based models with semantic warm-up outperform the LLM-free baseline. The two Base rows in Table 1 show the performance difference between Qwen3 and TIGER after Seq2Item training. We can observe that Qwen3 consistently achieves substantially better performance than the LLM-free TIGER baseline across both datasets. For example, Qwen3 improves Hit@5 from 0.02576 to 0.03696 on KuaiRec and from 0.0564 to 0.0988 on KuaiRand-1K. Meanwhile, directly training Qwen3 on Seq2Item without the two-phase semantic warm-up also leads to a noticeable degradation. We further evaluate intermediate checkpoints throughout Seq2Item training. As shown in the left two panels of Figure 2, removing semantic warm-up leads to slower convergence and weaker performance throughout Seq2Item training on both datasets. This suggests that a better understanding of the semantic content encoded in SID tokens can facilitate subsequent SID-based sequential modeling.

Observation II. LLM-based models exhibit a highly non-uniform advantage across SID levels, with most gains concentrated at early levels. To understand where the advantage of Qwen3 over TIGER comes from, we conduct a level-wise comparison under teacher-forcing setting. For the i-th SID level, we provide the ground-truth prefix $\boldsymbol{s}_{<i} = [\mathrm{sid}_0, \ldots, \mathrm{sid}_{i-1}]$ and let the model predict the next SID token sid_i, with beam search used to compute Hit@K. This setting isolates the prediction quality at each SID level by eliminating error propagation from earlier positions. As shown in the right two panels of Figure 2, Qwen3's gains are highly concentrated at early SID levels: on KuaiRec, the improvement mainly appears at Level 0, while on KuaiRand-1K it is concentrated at Levels 0–1. In contrast, the gains at later levels are negligible. This indicates that the advantage of the LLM-based model primarily stems from better early SID decisions.

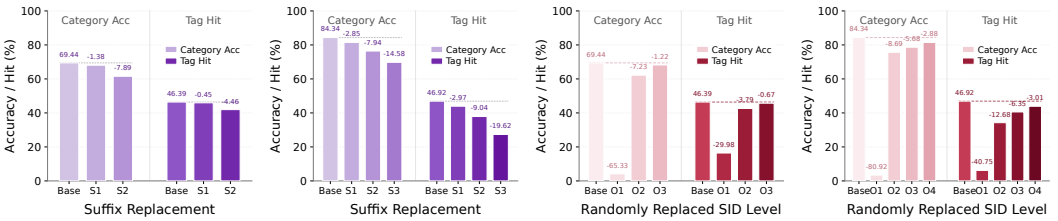
2.2 Early SIDs Carry More Semantic and Behavioral Information

The above **Observations I and II** suggest that the semantic information captured by different SID levels may contribute unequally to LLM-based recommendation. In particular, early SID levels may encode a larger share of the semantics that can be exploited by the pretrained LLM, and therefore play a more important role in subsequent sequential modeling. To further examine the semantic roles of different SID levels, we focus on two complementary perspectives:

- **semantic sufficiency:** *whether early SID levels alone preserve most of the item semantics;*
- **semantic necessity:** *whether perturbing early SID levels causes substantially greater semantic degradation than perturbing later ones.*

We evaluate both perspectives on the SID2Context task under controlled SID perturbations using two metrics: *(a) category accuracy*, which requires both the first- and second-level categories to be correctly recovered, and *(b) tag hit rate*, which measures whether any ground-truth item tag is correctly predicted.

Observation III. Early SID tokens are both semantically sufficient and necessary for LLM-based models. First, we progressively replace suffix SID tokens with randomly sampled codes from corresponding codebooks while keeping the earlier tokens unchanged. We use **Sk** to denote the setting where the **S**uffix spanning k SID levels, i.e., $[\mathrm{sid}_{L-k}, \ldots, \mathrm{sid}_{L-1}]$, is randomly replaced.



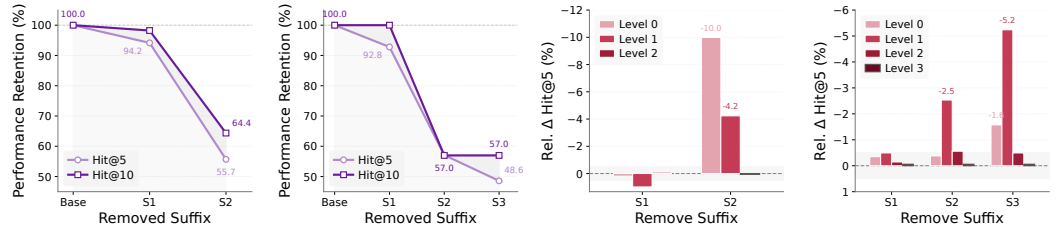
(a) KuaiRec (b) KuaiRand-1K (c) KuaiRec (d) KuaiRand-1K

Figure 3: Semantic sufficiency and necessity of hierarchical SIDs on KuaiRec and KuaiRand-1K. (a–b) Semantic sufficiency: progressively replacing suffix SID levels causes minor degradation when early SID levels are retained. (c–d) Semantic necessity: perturbing earlier levels independently leads to substantially larger degradation in category accuracy and tag hit rate.

As shown in the left two panels of Figure 3, compared with the unperturbed Base, randomly replacing only the later SID tokens causes relatively small degradation in both metrics, whereas the performance drops progressively as more suffix levels are replaced. This indicates that early SID tokens alone are sufficient to recover item semantics.

Then, we further examine semantic necessity by perturbing one SID level at a time while keeping all remaining levels unchanged. We use **Oi** to denote the setting where **O**nly the i-th SID token is randomly replaced. As shown in the right two panels of Figure 3, perturbing the earliest SID token leads to a drastic degradation in both semantic metrics, whereas the degradation becomes progressively smaller when later SID levels are replaced. These results support the semantic necessity of early SID tokens for preserving item semantics.

The previous analyses mainly focus on SID2Context and characterize how semantic information is distributed across SID levels. We next examine whether a similar positional asymmetry also appears in Seq2Item, where SIDs are used to model user behavior and predict the next item. Specifically, we progressively remove suffix SID tokens from each item in the user history while keeping the prediction target unchanged, and evaluate how recommendation performance changes as increasingly more suffix levels are removed.



(a) KuaiRec (b) KuaiRand-1K (c) KuaiRec (d) KuaiRand-1K

Figure 4: Seq2Item analysis under progressive suffix removal on KuaiRec and KuaiRand-1K. (a–b) Recommendation performance retention under suffix removal from user histories. (c–d) Relative changes in teacher-forced Hit@5 at each SID level under the same settings.

Observation IV. Early SID tokens are also more critical for preserving behavior-relevant information in Seq2Item. As shown in the left two panels of Figure 4, removing only the last SID level causes limited degradation, whereas removing additional suffix levels leads to substantially larger performance drops. Furthermore, we report the level-wise performance changes under the same teacher-forcing setting in the right two panels. It can be observed that such degradation is concentrated mainly at the earlier target SID levels, while later-level predictions remain relatively stable. Together with the Base results where the last two SID levels achieve Hit@5 above 70% and even above 90% under teacher forcing, this suggests that later SID tokens can be relatively reliably recovered from the preceding prefix. These results indicate that early SID tokens primarily carry semantic and behavior-relevant information, whereas later SID tokens mainly provide relatively stable specifications for identifying the target item.

3 PREFERENCE OPTIMIZATION AT CRITICAL SID BRANCHES

The observations in Section 2 inspire us to reconsider how supervision should be applied to hierarchical SID generation. Since early SID levels carry substantially more semantic and behavior-relevant information for LLM-based models, **improving these early decisions becomes the key for further enhancing recommendation performance**. However, standard next-token prediction objective only learns from positive targets and provides no explicit signal for distinguishing undesirable alternatives at these critical positions. A natural solution is therefore to construct preference pairs **whose chosen and rejected SID sequences diverge at early levels**, allowing the model to explicitly identify and correct undesirable decisions at these critical positions. Such pairwise preference signals can be naturally learned through Direct Preference Optimization (DPO) (Rafailov et al., 2023). Unfortunately, as we show next, standard DPO is insufficient to achieve this.

3.1 MISMATCH BETWEEN STANDARD DPO AND AUTOREGRESSIVE SID GENERATION

Given a user history $\boldsymbol{x}$, let $\boldsymbol{s}^+$ and $\boldsymbol{s}^-$ denote the SID sequences of a chosen and a rejected item respectively. Standard DPO optimizes the relative likelihood between the two sequences:

$$\mathcal{L}_{\text{DPO}} = -\log\sigma\left(\beta\left[\log\frac{\pi_\theta(\boldsymbol{s}^+\mid\boldsymbol{x})}{\pi_{\text{ref}}(\boldsymbol{s}^+\mid\boldsymbol{x})} - \log\frac{\pi_\theta(\boldsymbol{s}^-\mid\boldsymbol{x})}{\pi_{\text{ref}}(\boldsymbol{s}^-\mid\boldsymbol{x})}\right]\right), \tag{1}$$

where π_θ denotes the policy model, π_{ref} is the reference model, and β controls the strength of preference optimization. For autoregressive SID generation, the sequence likelihood is factorized over SID positions: $\log\pi_\theta(\boldsymbol{s}\mid\boldsymbol{x}) = \sum_{i=0}^{L-1}\log\pi_\theta(\boldsymbol{s}_i\mid\boldsymbol{x},\boldsymbol{s}_{<i})$. Therefore, standard DPO aggregates the preference signals from all SID positions into a single sequence-level margin.

We define the log-ratio at position i as $r_i(\boldsymbol{s}) = \log\pi_\theta(\boldsymbol{s}_i\mid\boldsymbol{x},\boldsymbol{s}_{<i}) - \log\pi_{\text{ref}}(\boldsymbol{s}_i\mid\boldsymbol{x},\boldsymbol{s}_{<i})$ and the corresponding preference margin as $m_i = r_i(\boldsymbol{s}^+) - r_i(\boldsymbol{s}^-)$. In addition, we denote the first position where the chosen and rejected SID sequences differ as $t = \min\left\{i \mid \boldsymbol{s}_i^+ \neq \boldsymbol{s}_i^-\right\}$. As motivated by the observations in Section 2, we focus more on preference pairs whose first divergence occurs at early SID levels, meaning that t is typically small, e.g., $t = 0$ or 1. Since SIDs are generated autoregressively one level at a time, an incorrect decision at early position places all subsequent predictions on the wrong generation path, making this first divergence particularly critical.

Therefore we refer to m_t as the **critical margin**, whose optimization directly encourages the model to correct erroneous choices at these early decisive positions. Noticing that $m_i = 0$ for all $i < t$, the DPO loss can be reformulated as

$$\mathcal{L}_{\text{DPO}} = -\log\sigma\left(\beta\left[\sum_{i=0}^{L-1} r_i(\boldsymbol{s}^+) - \sum_{i=0}^{L-1} r_i(\boldsymbol{s}^-)\right]\right) = -\log\sigma\left(\beta\left[m_t + \sum_{i=t+1}^{L-1} m_i\right]\right). \tag{2}$$

For simplicity, we use $M_{\text{suffix}} = \sum_{i=t+1}^{L-1} m_i$ and $M_{\text{seq}} = m_t + M_{\text{suffix}}$ to denote the suffix margin and total margin respectively. We next examine whether the DPO objective targeting M_{seq} can guarantee an increase in the critical margin m_t. The following theorem shows that this is not guaranteed.

Theorem 1. *DPO increases the total preference margin M_{seq} while it does not guarantee an increase in the critical margin m_t at the first-divergence position.*

Remark 1. The proof can be seen in Appendix B. It can be seen from Eq. 2 that the margin optimized by DPO consists of two components: the critical-branch margin m_t and the suffix margin M_{suffix}. At the first-divergence position t, the chosen and rejected SID tokens are compared under exactly the same prefix, $\boldsymbol{s}_{<t} = \boldsymbol{s}_{<t}^+ = \boldsymbol{s}_{<t}^-$. However, the suffix margins do not characterize the preference at the critical branch because for any position $i > t$, the two policies $\pi_\theta(\boldsymbol{s}_i^+ \mid \boldsymbol{x}, \boldsymbol{s}_{<t}, \boldsymbol{s}_{t:i-1}^+)$ and $\pi_\theta(\boldsymbol{s}_i^- \mid \boldsymbol{x}, \boldsymbol{s}_{<t}, \boldsymbol{s}_{t:i-1}^-)$ are already conditioned on different generation paths. Since SID prediction is performed autoregressively in a step-by-step manner, optimizing such margins after the prefixes have diverged is not well aligned with the actual decision process, leading to a mismatch between sequence-level preference optimization and SID generation.

Furthermore, this mismatch interferes with the optimization of the critical-branch margin m_t. Specifically, after one gradient-descent step with learning rate η, the change in m_t can be written as

$$m_t(\theta') - m_t(\theta) = \eta\beta\,\underbrace{\sigma(-\beta(m_t + M_{\text{suffix}}))}_{\text{Margin-scale interference}}(\|\nabla_\theta m_t\|^2 + \underbrace{\langle\nabla_\theta m_t, \nabla_\theta M_{\text{suffix}}\rangle}_{\text{Gradient-direction interference}}) + O(\eta^2). \tag{3}$$

As shown in Eq. 3, the suffix introduces two forms of interference into the optimization of m_t:

- **Margin-scale interference.** Increasing M_{suffix} will drives $\sigma(-\beta(m_t + M_{\text{suffix}}))$ toward zero, thereby slowing the increase of the critical margin m_t.
- **Gradient-direction interference.** The interaction term $\langle \nabla_\theta m_t, \nabla_\theta M_{\text{suffix}} \rangle$ introduces an uncertain effect on the update of the critical margin, potentially either amplifying or weakening its increase depending on the gradient alignment.

3.2 Critical-Branch Direct Preference Optimization

The above analysis shows that the mismatch between DPO objective and autoregressive SID generation causes suffix margins to interfere with the optimization of the critical margin. To address this issue, we propose **Critical-Branch DPO (CB-DPO)**, which performs preference optimization directly at the first-divergence position. Specifically, for a batch $\mathcal{B} = \{(\boldsymbol{x}^{(k)}, \boldsymbol{s}^{(k,+)}, \boldsymbol{s}^{(k,-)})\}_{k=1}^{|\mathcal{B}|}$, we denote by $m_i^{(k)}$ the preferene margin at position i for the k-th pair. The CB-DPO objective is then formulated as

$$\mathcal{L}_{\text{CB-DPO}} = \frac{1}{|\mathcal{B}|} \sum_{k=1}^{|\mathcal{B}|} \left[-\log \sigma \left(\beta \sum_{i=0}^{L-1} \mathbb{I}[i = t^{(k)}]\, m_i^{(k)} \right) - \lambda \log \pi_\theta \left(\boldsymbol{s}^{(k,+)} \mid \boldsymbol{x}^{(k)} \right) \right], \tag{4}$$

where $\mathbb{I}(\cdot)$ is the indicator function and $t^{(k)} = \min\{i \mid \boldsymbol{s}_i^{(k,+)} \neq \boldsymbol{s}_i^{(k,-)}\}$ is the first-divergence position for the k-th pair and λ controls the strength of the next token prediction (NTP) regularization on the chosen decision. In addition, we introduce an auxiliary next-token prediction loss on the chosen samples, which we find helps counteract the "negative squeezing" effect (Rafailov et al., 2023; Ren & Sutherland, 2025; Razin et al., 2025) and mitigate entropy collapse. We next establish that CB-DPO directly improves the critical margin.

Theorem 2. *CB-DPO guarantees the increase in the critical margin m_t at first divergence.*

Remark 2. It can be easily shown that for CB-DPO, after one gradient-descent step, the change in the critical margin becomes

$$m_t(\theta') - m_t(\theta) = \eta\beta\sigma(-\beta m_t)\|\nabla_\theta m_t\|^2 + O(\eta^2), \tag{5}$$

which guarantees a first-order increase in m_t whenever $\nabla_\theta m_t \neq 0$. Compared with Eq 3, by removing M_{suffix}, CB-DPO eliminates both the margin-scale and gradient-coupling interference introduced by suffix tokens. Thus, CB-DPO provides a cleaner and more direct optimization signal for correcting the critical branching decision.

In practice, when performing CB-DPO, we focus more on constructing pairs whose first divergence occurs within the first two SID levels, i.e., $t = 0$ or 1. This is sufficient as our observations in Section 2 show that early SID levels are more critical for improving performance, while later SID levels contribute more to specifying the exact item and can be reliably recovered once the preceding SID prefix is correctly predicted. In fact, we find that early-divergence pairs are easier to obtain, naturally aligning data availability with our optimization target.

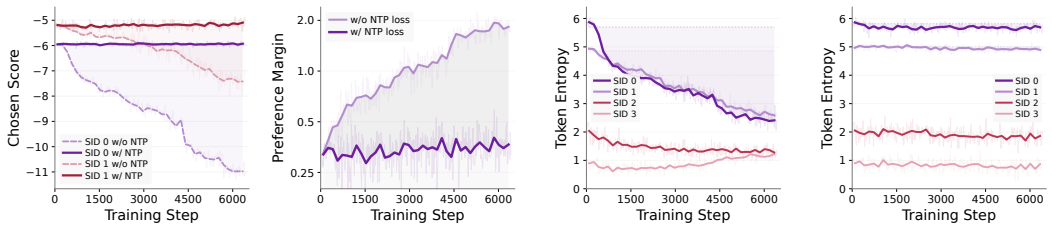
Moreover, to stabilize the training process, we adopt a **stratified sampling strategy** according to the first-divergence position. Specifically, we maintain approximately fixed proportions of preference pairs with different divergence levels in each mini-batch, preventing the optimization from being dominated by pairs from any particular SID level.

4 Experiments

Experimental Setup. We conduct experiments on two public datasets, KuaiRec (Gao et al., 2022a) and KuaiRand-1K (Gao et al., 2022b), using RQ-VAE (Lee et al., 2022) to construct hierarchical SIDs with codebook sizes $[256, 256, 256]$ and $[1024, 512, 256, 64]$, respectively. We adopt Qwen3-1.7B-Base (Yang et al., 2025) and TIGER (Rajput et al., 2023) as the LLM-based and LLM-free generative backbones, and use trie-constrained beam search for item generation during evaluation. For KuaiRec, interactions with a watch ratio greater than 1 are used for both user histories and

Table 1: Overall Hit@K and NDCG@K recommendation performance on KuaiRec and KuaiRand-1K. **Bold** and underlined values indicate the best and second-best results in each column, respectively. Rel. Gain denotes the relative improvement of CB-DPO over the corresponding Base model. Results on KuaiRand-1K are reported as 100%.

Methods	KuaiRec				KuaiRand-1K			
	H@5	H@10	N@5	N@10	H@5	H@10	N@5	N@10
TIGER (Encoder–Decoder)								
Base	0.02576	0.04574	0.01539	0.02179	0.0564	0.0988	0.0238	0.0325
DPO	0.02556	0.04612	0.01531	0.02187	0.0536	0.1044	0.0229	0.0326
CB-DPO	0.02587	0.04604	0.01542	0.02187	0.0635	0.1214	0.0251	0.0375
Rel. Gain	*+0.44%*	*+0.66%*	*+0.19%*	*+0.39%*	*+12.59%*	*+22.87%*	*+5.46%*	*+15.38%*
Qwen3-1.7B (Decoder-only LLM)								
Base	0.03696	**0.05666**	0.02515	0.03147	0.0988	0.1411	0.0482	0.0571
w/o Semantic Warmup	0.02972	0.04946	0.01968	0.02599	0.0917	0.1411	0.0458	0.0560
DPO	0.03674	0.05633	0.02530	0.03157	0.0847	0.1355	0.0408	0.0508
CB-DPO	**0.03724**	0.05655	**0.02550**	**0.03168**	**0.1129**	**0.1637**	**0.0505**	**0.0617**
Rel. Gain	*+0.76%*	*-0.20%*	*+1.38%*	*+0.67%*	*+14.27%*	*+16.02%*	*+4.77%*	*+8.06%*



(a) Chosen Score (b) Preference Margin (c) w/o NTP (d) w/ NTP

Figure 5: Effect of NTP regularization on preference optimization. (a) Chosen-token scores at the first two SID levels, which are stabilized by NTP. (b) Preference margin during training: without NTP, the margin exhibits an unconstrained and pathological increase, whereas with NTP it grows steadily. (c–d) Token-level entropy without and with NTP, respectively, showing substantially reduced entropy collapse under NTP regularization.

prediction targets, while preference negatives are preferentially selected from items with relatively high watch ratios but no larger than 0.8. For KuaiRand-1K, items receiving strong positive feedback, including long viewing and explicit interactions, are used as prediction targets, while clicked items are additionally included in user histories. When constructing preference pairs, we preferentially select clicked items without strong positive feedback as negatives, and use exposure-only items when such samples are unavailable. The maximum user history length is set to 20 for both datasets. More implementation details are provided in Appendix C.

Overall Recommendation Performance. Table 1 reports the overall top-K recommendation performance on KuaiRec and KuaiRand-1K. We first observe that the LLM-based Qwen3 model consistently outperforms the LLM-free TIGER baseline after standard Seq2Item training. This is consistent with our earlier observation that pretrained LLMs can better exploit the semantic information encoded in SIDs. We then compare different preference optimization strategies. CB-DPO improves over the corresponding Base model on most metrics across both datasets and both backbones. On KuaiRec, CB-DPO improves Qwen3 by 0.76%, 1.38%, and 0.67% on Hit@5, NDCG@5, and NDCG@10, respectively, while yielding consistent gains for TIGER across all four metrics. The gains of CB-DPO are more pronounced on KuaiRand-1K. For Qwen3, CB-DPO improves Hit@5 and Hit@10 by 14.27% and 16.02%, respectively, while the corresponding gains for TIGER reach 12.59% and 22.87%. We attribute this stronger improvement partly to the richer explicit feedback signals available in KuaiRand-1K, which allow us to construct cleaner and more informative preference pairs. In contrast, DPO provides less stable improvements and can even degrade recommendation performance. For instance, on KuaiRand-1K, DPO decreases Qwen3 Hit@5 from

Table 2: Teacher-forced level-wise Hit@K on KuaiRand-1K. All values are multiplied by 100%. Level 0 denotes the earliest SID level. Rel. Gain denotes the relative improvement of CB-DPO over the corresponding Base model.

Methods	Level 0			Level 1			Level 2			Level 3		
	Hit@5	Hit@10	Hit@20	Hit@5	Hit@10	Hit@20	Hit@5	Hit@10	Hit@20	Hit@5	Hit@10	Hit@20
TIGER (Encoder–Decoder)												
Base	19.0656	26.8733	37.0544	10.8244	16.3912	24.3345	72.8805	87.0756	95.3178	99.6150	99.9324	99.9976
DPO	18.9371	26.8225	36.9416	10.7429	16.3216	24.2934	72.9324	87.0324	95.3351	99.6166	99.9340	99.9976
CB-DPO	19.2067	27.2246	37.2703	11.1724	16.7969	24.7094	73.2334	87.2627	95.4192	99.6056	99.9285	99.9969
Rel. Gain	*+0.74%*	*+1.31%*	*+0.58%*	*+3.21%*	*+2.48%*	*+1.54%*	*+0.48%*	*+0.21%*	*+0.11%*	*-0.01%*	*0.00%*	*0.00%*
Qwen3-1.7B (Decoder-only LLM)												
Base	19.4621	27.5929	37.9928	11.3985	17.2461	25.6157	72.8121	87.2721	95.6219	99.6213	99.9442	99.9984
w/o Semantic Warmup	18.7650	26.2749	36.1937	10.0414	15.5189	23.1933	72.3980	87.0913	95.1763	99.5930	99.9387	99.9992
DPO	19.3167	27.4560	37.7177	11.1146	16.8602	25.1807	72.7532	87.2792	95.6046	99.6197	99.9403	99.9992
CB-DPO	19.6865	27.9401	38.1678	11.9213	17.6763	26.0815	72.9355	87.2972	95.6211	99.6181	99.9426	99.9984
Rel. Gain	*+1.15%*	*+1.26%*	*+0.46%*	*+4.59%*	*+2.49%*	*+1.82%*	*+0.17%*	*+0.03%*	*0.00%*	*0.00%*	*0.00%*	*0.00%*

0.0988 to 0.0847, whereas CB-DPO improves it to 0.1129. This contrast supports our motivation that directly optimizing the critical branch is better aligned with autoregressive SID generation than sequence-level preference optimization. Notably, although CB-DPO is motivated by our analysis of LLM-based SID generation, it also consistently improves the LLM-free TIGER model, suggesting that it captures a more general property of auto-regressive SID generation.

Level-wise Analysis of CB-DPO. To understand where the gains of CB-DPO come from, we evaluate each SID level under teacher forcing, where the ground-truth prefix is provided to isolate the prediction quality at the current level. As shown in Tables 2 and 4, CB-DPO mainly improves the early SID levels of Qwen3. For example, on KuaiRand-1K, the gains reach 1.15% and 4.59% at Levels 0 and 1, but quickly diminish at later levels. A similar trend is also observed on TIGER, suggesting that the benefit of critical-branch optimization extends beyond the LLM backbone. In contrast, standard DPO provides less consistent improvements at these critical positions. For example, on KuaiRand-1K, DPO decreases Qwen3's Level-1 Hit@5 from 11.3985 to 11.1146, whereas CB-DPO increases it to 11.9213. These results show that CB-DPO improves recommendation primarily by strengthening early SID predictions, supporting our motivation to directly optimize the critical branching decisions.

NTP Regularization for Mitigating Negative Squeezing. Preference optimization alone does not explicitly preserve the likelihood of the chosen token and may suffer from the "negative squeezing" effect, progressively sharpening the output distribution and even causing entropy collapse. We therefore introduce NTP regularization to mitigate this issue. As shown in Figure 5(a–b), with NTP regularization, the chosen-token scores remain stable while the preference margin increases steadily. In contrast, without NTP, the chosen scores are severely suppressed while the margin keeps expanding in an unconstrained manner. We further monitor the token entropy in Figure 5(c–d). Without NTP, the entropy collapses markedly, especially at the first two SID levels, whereas NTP keeps the entropy of all SID levels substantially more stable throughout training. These results show that NTP regularization effectively mitigates negative squeezing and stabilizes preference optimization.

5 Conclusion

In this work, we study the internal roles of hierarchical SIDs in LLM-based generative recommendation. We find that early SID levels carry more semantic and behavior-relevant information, while later levels mainly refine the prediction toward specific items. Motivated by this, we further analyze preference optimization for SID generation and identify a mismatch between standard DPO and the auto-regressive decision process: suffix margins can interfere with the optimization of the critical first-divergence decision through both margin-scale and gradient-level effects. To address this issue, we propose CB-DPO, which applies preference supervision directly at the first-divergence position and provides a cleaner optimization signal for critical early decisions. Experiments on KuaiRec and KuaiRand-1K demonstrate the effectiveness of our approach.

AI USE STATEMENT

In this work, we used generative AI tools to assist with writing portions of the code and to improve the readability and language of the manuscript. Generative AI tools were not used to generate research ideas, formulate the core methodological contributions, develop theoretical claims or proofs, design experiments, or interpret experimental results. All AI-assisted code was reviewed and tested by the authors for correctness, and all AI-assisted edits to the manuscript were carefully reviewed by the authors. We take responsibility for the final content of this work, including all text and code produced with the assistance of generative AI.

REFERENCES

Lei Cheng, Haonan Hu, Beibei Kong, Yudong Li, Zang Li, Yunsheng Pang, Hongyang Su, Jianchao Tu, Yunlong Wang, Bing Wen, et al. Tgr: Advancing industrial recommendation from generative-paradigm ranking toward unified generation and reasoning. *arXiv preprint arXiv:2609.00986*, 2026.

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.

Chongming Gao, Shijun Li, Wenqiang Lei, Jiawei Chen, Biao Li, Peng Jiang, Xiangnan He, Jiaxin Mao, and Tat-Seng Chua. Kuairec: A fully-observed dataset and insights for evaluating recommender systems. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pp. 540–550, 2022a.

Chongming Gao, Shijun Li, Yuan Zhang, Jiawei Chen, Biao Li, Wenqiang Lei, Peng Jiang, and Xiangnan He. Kuairand: An unbiased sequential recommendation dataset with randomly exposed videos. In *Proceedings of the 31st ACM international conference on information & knowledge management*, pp. 3953–3957, 2022b.

Yingzhi He, Yan Sun, Junfei Tan, Yuxin Chen, Xiaoyu Kong, Chunxu Shen, Xiang Wang, An Zhang, and Tat-Seng Chua. Reasoning over semantic ids enhances generative recommendation. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 1638–1649, 2026a.

Yinhan He, Liam Collins, Bhuvesh Kumar, Jundong Li, Neil Shah, and Donald Loveland. Implicit reasoning for large language model-based generative recommendation. *arXiv preprint arXiv:2606.14142*, 2026b.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Bowen Jin, Hansi Zeng, Guoyin Wang, Xiusi Chen, Tianxin Wei, Ruirui Li, Zhengyang Wang, Zheng Li, Yang Li, Hanqing Lu, et al. Language models as semantic indexers. *arXiv preprint arXiv:2310.07815*, 2023.

Xiaoyu Kong, Leheng Sheng, Junfei Tan, Yuxin Chen, Jiancan Wu, An Zhang, Xiang Wang, and Xiangnan He. Minionerec: An open-source framework for scaling generative recommendation. *arXiv preprint arXiv:2510.24431*, 2025.

Xiaoyu Kong, Junfei Tan, Bin Liu, Yanwen Huang, Ziru Xu, Han Zhu, Jiancan Wu, An Zhang, Xiang Wang, and Xiangnan He. SINGER: Leveraging semantic identifier hierarchies for generative recommendation, 2026. URL https://openreview.net/forum?id=zw0IMYPLQZ.

Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and Wook-Shin Han. Autoregressive image generation using residual quantization. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 11513–11522. IEEE, 2022.

Fake Lin, Binbin Hu, Zhi Zheng, Xi Zhu, Ziqi Liu, Zhiqiang Zhang, Jun Zhou, and Tong Xu. Token-level collaborative alignment for llm-based generative recommendation. In *Proceedings of the ACM Web Conference 2026*, pp. 6909–6919, 2026.

Xinyu Lin, Haihan Shi, Wenjie Wang, Fuli Feng, Qifan Wang, See-Kiong Ng, and Tat-Seng Chua. Order-agnostic identifier for large language model-based generative recommendation. In *Proceedings of the 48th international ACM SIGIR conference on research and development in information retrieval*, pp. 1923–1933, 2025.

Enze Liu, Bowen Zheng, Cheng Ling, Lantao Hu, Han Li, and Wayne Xin Zhao. Generative recommender with end-to-end learnable item tokenization. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 729–739, 2025a.

Jingzhe Liu, Liam Collins, Jiliang Tang, Tong Zhao, Neil Shah, and Mingxuan Ju. Understanding generative recommendation with semantic ids from a model-scaling view. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 3234–3245, 2026.

Zhanyu Liu, Shiyao Wang, Xingmei Wang, Rongzhou Zhang, Jiaxin Deng, Honghui Bao, Jinghao Zhang, Wuchao Li, Pengfei Zheng, Xiangyu Wu, et al. Onerec-think: In-text reasoning for generative recommendation. *arXiv preprint arXiv:2510.11639*, 2025b.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

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.

Yunsheng Pang, Zijian Liu, Yudong Li, Shaojie Zhu, Zijian Luo, Chenyun Yu, Sikai Wu, Shichen Shen, Cong Xu, Bin Wang, et al. Higr: Industrial-scale hierarchical generative slate recommendation framework in tencent. *arXiv preprint arXiv:2512.24787*, 2025.

Haohao Qu, Wenqi Fan, Zihuai Zhao, and Qing Li. Tokenrec: Learning to tokenize id for llm-based generative recommendations. *IEEE Transactions on Knowledge and Data Engineering*, 2025.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

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.

Noam Razin, Sadhika Malladi, Adithya Bhaskar, Danqi Chen, Sanjeev Arora, and Boris Hanin. Unintentional unalignment: Likelihood displacement in direct preference optimization. In *International Conference on Learning Representations*, volume 2025, pp. 24791–24834, 2025.

Yi Ren and Danica Sutherland. Learning dynamics of llm finetuning. In *International Conference on Learning Representations*, volume 2025, pp. 70523–70563, 2025.

Junfei Tan, Yuxin Chen, An Zhang, Junguang Jiang, Bin Liu, Ziru Xu, Han Zhu, Jian Xu, Bo Zheng, and Xiang Wang. Reinforced preference optimization for recommendation. *arXiv preprint arXiv:2510.12211*, 2025.

Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

Bin Wang and Zhengyu Zhang. Guiding the coarse levels of semantic ids makes the fine levels learnable, 2026. URL `https://arxiv.org/abs/2609.22227`.

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, 2024.

Likang Wu, Zhi Zheng, Zhaopeng Qiu, Hao Wang, Hongchao Gu, Tingjia Shen, Chuan Qin, Chen Zhu, Hengshu Zhu, Qi Liu, et al. A survey on large language models for recommendation. *World Wide Web*, 27(5):60, 2024.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. C-pack: Packaged resources to advance general chinese embedding, 2023.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Wencai Ye, Mingjie Sun, Shuhang Chen, Wenjin Wu, and Peng Jiang. Align^{3}gr: Unified multi-level alignment for llm-based generative recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 16154–16162, 2026.

Xin Yu, Stephen Li, Sina Aghaei, Zifan Zhu, Jiamu Bai, Guanjie Huang, Bo Peng, Yiyao Liu, and Lingzhou Xue. Difficulty-aware semantic-id optimization for generative recommendation. *arXiv preprint arXiv:2608.20611*, 2026a.

Yuanqing Yu, Yifan Wang, Weizhi Ma, Zhiqiang Guo, and Min Zhang. Apao: Bridging the training-inference gap in generative recommendation via adaptive prefix-aware optimization. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 6338–6349, 2026b.

Junchao Zeng, Junzhang Zhu, Junyang Chen, Yudong Li, Wei Liu, Chengxiang Zhuo, and Zang Li. Bridging the structural gap: Adapting autoregressive generation for recommendation. *arXiv preprint arXiv:2607.21028*, 2026.

Zhiwei Zhang, Qingyang Li, Ming Zhang, Yanchen Qiao, Zhen Wang, Ziyang Ji, Xiangyu Qian, Shijie Yang, Yanlong Zang, Weijie Ding, et al. Oxygenrec: An instruction-following generative framework for e-commerce recommendation. In *Proceedings of the 49th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 5087–5091, 2026.

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.

Guorui Zhou, Honghui Bao, Jiaming Huang, Jiaxin Deng, Jinghao Zhang, Junda She, Kuo Cai, Lejian Ren, Lu Ren, Qiang Luo, et al. Openonerec technical report. *arXiv preprint arXiv:2512.24762*, 2025a.

Guorui Zhou, Jiaxin Deng, Jinghao Zhang, Kuo Cai, Lejian Ren, Qiang Luo, Qianqian Wang, Qigen Hu, Rui Huang, Shiyao Wang, et al. Onerec technical report. *arXiv preprint arXiv:2506.13695*, 2025b.

Guorui Zhou, Hengrui Hu, Hongtao Cheng, Huanjie Wang, Jiaxin Deng, Jinghao Zhang, Kuo Cai, Lejian Ren, Lu Ren, Liao Yu, et al. Onerec-v2 technical report, 2025. *URL https://arxiv.org/abs/2508.20900*, 2025c.

A Related Work

Semantic-ID-based Generative Recommendation Semantic IDs (SIDs) turn item prediction into generation over a compact, structured code space (Van Den Oord et al., 2017; Lee et al., 2022; Luo et al., 2025), which has been an important component of recent generative recommendation methods (Zhou et al., 2025c; Pang et al., 2025). TIGER (Rajput et al., 2023) introduced a representative pipeline that quantizes item content into a short code sequence and auto-regressively predicts the next item's SID from interaction history. Subsequent work has improved the construction of these identifiers: for example, LMIndexer (Jin et al., 2023) learns semantic indices through progressive and self-supervised objectives; LETTER (Wang et al., 2024) integrates semantic reconstruction, collaborative information, and code-use diversity; and ETEGRec (Liu et al., 2025a) couples item tokenization with recommendation training. These studies primarily ask how to obtain better item codes. Recent analyses have also exposed limitations of hierarchical codes. A model-scaling study identifies SID information capacity as a potential bottleneck (Liu et al., 2026), while BARGE (Zeng et al., 2026) addresses item-boundary loss and hierarchical path drift during autoregressive generation. In addition, Guided SID explicitly assigns task-relevant attributes to coarse code levels to make early decisions semantically meaningful (Wang & Zhang, 2026). Our work uses a fixed hierarchical SID representation and instead measures how a pretrained LLM exploits each level relative to an LLM-free recommender, then optimizes the first preference-relevant branching decision. Thus, the contribution is a diagnosis and optimization of the decoder's hierarchical decision process, rather than a new SID tokenizer.

LLM-based Generative Recommendation with Semantic IDs Using pretrained LLMs for SID generation requires grounding newly introduced item tokens in both language and user behavior (Liu et al., 2025b; Zhou et al., 2025a; Zhang et al., 2026; Cheng et al., 2026). LC-Rec (Zheng et al., 2024) learns discrete item indices and alignment tasks that integrate language and collaborative semantics, whereas TokenRec (Qu et al., 2025) quantizes collaborative representations into tokens for LLM-based retrieval. SETRec (Lin et al., 2025) departs from ordered SID paths by using order-agnostic token sets to improve generation efficiency. More recent methods directly align LLM token predictions with collaborative scores (Lin et al., 2026) or release complete Qwen-based SID training and post-training pipelines (Kong et al., 2025). SIDReasoner (He et al., 2026a) and PauseRec (He et al., 2026b) study how SID grounding interacts with explicit and implicit LLM reasoning, respectively. Align^{3}GR (Ye et al., 2026) unifies token, behavior, and preference alignment, including progressive DPO. These efforts establish the value of semantic grounding and preference adaptation, but generally report item-level results or optimize all token positions as part of broader training pipelines. We complement them with level-wise comparisons, semantic reconstruction and perturbation tests, and a preference objective defined at the first position where preferred and rejected SIDs take different branches.

Preference Optimization for Generative Recommendation. Preference-based post-training has recently been introduced into generative recommendation to better align generated items with user preferences. OneRec (Deng et al., 2025) incorporates iterative DPO (Rafailov et al., 2023) into end-to-end generative recommendation, while OneRec-V2 (Zhou et al., 2025c) further leverages feedback from real user interactions. ReRe (Tan et al., 2025) improves reinforcement-based preference optimization through more effective negative sampling and finer-grained reward signals. More recent studies have begun to exploit the structured nature of SID generation. SINGER (Kong et al., 2026) introduces SID-level rewards and prefix-based curricula, while DASO (Yu et al., 2026a) uses prefix-match depth to identify difficult SID levels and adaptively allocate rollouts. APAO (Yu et al., 2026b) instead focuses on the training–inference gap in autoregressive recommendation, improving vulnerable prefixes that may be prematurely pruned during beam-search decoding. These methods demonstrate the importance of considering intermediate SID decisions, but mainly intervene through reward design, sampling, or prefix-aware training. Our work starts from a different motivation. Through the analysis of LLM-based generative recommendation, we find that early SID decisions play particularly critical semantic and behavioral roles. Based on this observation, we further identify a mismatch between DPO and autoregressive SID generation: after the chosen and rejected sequences first diverge, their suffix probabilities are conditioned on different generation paths, yet standard DPO aggregates these suffix margins into the same preference objective. We

therefore focus on the first-divergence decision made under the shared prefix and directly optimize its critical margin.

B Proof of Theorem

Theorem 3. *DPO increases the total preference margin M_{seq} while it does not guarantee an increase in the critical margin m_t at the first-divergence position.*

Proof. From Eq. 2, the DPO loss can be written as $\mathcal{L}_{\text{DPO}} = -\log\sigma(\beta M_{\text{seq}})$. Let $z = \beta M_{\text{seq}}(\theta)$. Since

$$-\log\sigma(z) = \log(1 + e^{-z}),$$

differentiation gives

$$\frac{d}{dz}\left[-\log\sigma(z)\right] = -\frac{1}{1+e^{z}} = -\sigma(-z).$$

Applying the chain rule with $z = \beta M_{\text{seq}}(\theta)$ yields

$$\nabla_\theta \mathcal{L}_{\text{DPO}}(\theta) = -\beta\sigma(-\beta M_{\text{seq}}(\theta))\nabla_\theta M_{\text{seq}}(\theta). \tag{6}$$

By definition, one gradient-descent step is

$$\theta' = \theta - \eta\nabla_\theta \mathcal{L}_{\text{DPO}}(\theta). \tag{7}$$

Substituting Eq. 6 into Eq. 7 gives

$$\theta' - \theta = \eta\beta\sigma(-\beta M_{\text{seq}}(\theta))\nabla_\theta M_{\text{seq}}(\theta). \tag{8}$$

We can see that the scalar factor $\beta\sigma(-\beta M_{\text{seq}})$ controls the magnitude of the update for the entire preference pair, whereas the direction of the update is exactly $\nabla_\theta M_{\text{seq}}$.

For any twice continuously differentiable scalar function $f(\theta)$, Taylor's theorem around θ gives

$$f(\theta') - f(\theta) = \nabla_\theta f(\theta)^\top(\theta' - \theta) + O(\|\theta' - \theta\|^2). \tag{9}$$

From Eq. 8, $\|\theta' - \theta\| = O(\eta)$ as long as the local gradient is finite. Hence the remainder in Eq. 9 is $O(\eta^2)$. Now set $f = M_{\text{seq}}$ in Eq. 9 and substitute Eq. 8 for $(\theta' - \theta)$. We obtain

$$\begin{aligned} M_{\text{seq}}(\theta') - M_{\text{seq}}(\theta) &= \nabla_\theta M_{\text{seq}}(\theta)^\top\left[\eta\beta\sigma(-\beta M_{\text{seq}}(\theta))\nabla_\theta M_{\text{seq}}(\theta)\right] + O(\eta^2) \\ &= \eta\beta\sigma(-\beta M_{\text{seq}}(\theta))\|\nabla_\theta M_{\text{seq}}(\theta)\|^2 + O(\eta^2). \end{aligned} \tag{10}$$

Since $\beta > 0$ and $\sigma(\cdot) > 0$, its first-order coefficient is strictly positive whenever $\nabla_\theta M_{\text{seq}} \neq 0$. Therefore, for sufficiently small η, the total margin increases.

Similarly, set $f = m_t$ in Eq. 9. Substituting Eq. 8 gives

$$\begin{aligned} m_t(\theta') - m_t(\theta) &= \nabla_\theta m_t(\theta)^\top\left[\eta\beta\sigma(-\beta M_{\text{seq}}(\theta))\nabla_\theta M_{\text{seq}}(\theta)\right] + O(\eta^2) \\ &= \eta\beta\sigma(-\beta M_{\text{seq}}(\theta))\left\langle\nabla_\theta m_t(\theta), \nabla_\theta M_{\text{seq}}(\theta)\right\rangle + O(\eta^2) \\ &= \eta\beta\sigma(-\beta(m_t + M_{\text{suffix}}))\left(\|\nabla_\theta m_t\|^2 + \langle\nabla_\theta m_t, \nabla_\theta M_{\text{suffix}}\rangle\right) + O(\eta^2) \end{aligned} \tag{11}$$

Unlike the squared norm in Eq. 10, the inner product in Eq. 11 may be positive, or negative. Hence, DPO guarantees first-order improvement of the sum M_{seq}, but not of the critical component m_t.

Furthermore, consider the case where m_t remains some value while $M_{\text{suffix}} \to +\infty$. Then

$$\mathcal{L}_{\text{DPO}} = -\log\sigma(\beta(m_t + M_{\text{suffix}})) \to 0,$$

showing that the DPO objective can be made arbitrarily small without increasing the critical margin. At the same time, $\sigma(-\beta(m_t + M_{\text{suffix}})) \to 0$. Hence, provided that the gradient terms in Eq. 11 remain bounded, the first-order change of m_t vanishes $m_t(\theta') - m_t(\theta) \approx 0$ with respect to the DPO scaling factor. Therefore, optimizing the suffix margin alone can saturate the DPO objective while providing almost no further optimization signal to the critical margin. □

Table 3: Basic statistics of the datasets after preprocessing.

Dataset	#Users	#Items	#Interactions	Sparsity
KuaiRec	7,176	10,728	12,530,806	83.7229%
KuaiRand-1K	1,000	1,885,630	6,223,803	99.6699%

C Experimental Setup

Datasets. We conduct experiments on two public datasets KuaiRec Gao et al. (2022a) and KuaiRand-1K Gao et al. (2022b). For KuaiRec, we use the data from the Big Matrix. For KuaiRand-1K, we observe that the original dataset contains a large number of items with weak semantic information, including empty descriptions, missing categories, duplicated descriptions, and platform-generated template text. These items are difficult to provide meaningful and discriminative semantic information. Therefore, we apply rule-based preprocessing to filter such items. The basic statistics of the two datasets after preprocessing are summarized in Table 3.

Semantic ID Construction. We construct hierarchical SIDs using an RQ-VAE (Lee et al., 2022). Item semantic embeddings are obtained from the BGE model (Xiao et al., 2023) and subsequently quantized into discrete code sequences. For KuaiRec, we use three codebooks with sizes $[256, 256, 256]$, resulting in an SID collision rate of 0.0613%. For KuaiRand-1K, we use four codebooks with sizes $[1024, 512, 256, 64]$, with a collision rate of 0.0441%.

Backbones and Semantic Warm-up Phase. We consider two representative generative recommendation backbones: Qwen3-1.7B-Base, a pretrained decoder-only LLM (Yang et al., 2025), and TIGER (Rajput et al., 2023), an encoder–decoder generative recommendation model with approximately 26.31M parameters. For Qwen3, we first perform the two-phase semantic warm-up introduced in Section 2 before Seq2Item training. When training Qwen3, we freeze the original token embedding rows and the pretrained backbone parameters, while optimizing the newly introduced SID embeddings together with LoRA adapters (Hu et al., 2021). We apply LoRA (Hu et al., 2021) with rank $r = 16$ and $\alpha = 32$, corresponding to a scaling factor of $\alpha/r = 2$. Unless otherwise specified, we adopt the same setting for all subsequent training stages. During semantic warm-up, we use AdamW (Loshchilov & Hutter, 2017) as the optimizer with a learning rate of 1×10^{-3}. We use a batch size of $\mathbf{40}$ with 8 gradient accumulation steps on each GPU. On KuaiRec, we use all available item–context pairs and reserve 1% for validation. On KuaiRand-1K, we randomly sample 500K item–context pairs and also reserve 1% for validation. The LoRA adapters contain approximately 17.43M trainable parameters. TIGER is trained directly on the same Seq2Item data and SID configuration without semantic warm-up. All experiments are conducted using four NVIDIA A100 GPUs, each with 40 GB of memory.

Seq2Item Training Phase For KuaiRec, we use a temporally continuous subset of the Big Matrix spanning from August 27 to September 5, 2020, to construct the sequential training and evaluation data. We leave the penultimate day for validation and the final day for testing. Interactions with a watch ratio greater than 1 are regarded as effective interactions and are used for both user histories and prediction targets. The maximum length of each user history is set to 20. For KuaiRand-1K, we use interactions from the entire available time span. We leave the final two days for testing and the preceding two days for validation, while all earlier interactions are used for training. Items receiving strong positive feedback, including long viewing, likes, follows, comments, forwards, and profile visits, are used as prediction targets while user histories, we additionally include clicked items to provide a broader behavioral context. The maximum history length is also set to 20. All training sequences are randomly shuffled. During evaluation, both Qwen3 and TIGER adopt trie-constrained decoding and use beam search to generate candidate items. When training Qwen3, we use AdamW (Loshchilov & Hutter, 2017) as the optimizer with a learning rate of 1×10^{-4} and we use a batch size of $\mathbf{24}$. When training TIGER, we use a 6-layer Transformer encoder and a 6-layer Transformer decoder with hidden size $d_{\text{model}} = 384$, 6 attention heads, and a feed-forward dimension of 1536, resulting in approximately 26.31M parameters. We optimize the model using AdamW with a batch size of 512 for each GPU and a learning rate of 1×10^{-4}.

Preference Optimization Phase For KuaiRec, when constructing preference data, we preferentially select items with relatively high watch ratios but no larger than 0.8 as negative samples. For pairs whose first SID differs, the positive and negative interactions are restricted to occur within 3 minutes of each other. For pairs whose second SID differs, the negative interaction is allowed to occur within the 2 hours preceding the positive exposure. The preference data are constructed using the last three days of the training split together with the validation split, while evaluation is still performed on the original Seq2Item test set. Each positive item is paired with at most one negative sample. For KuaiRand-1K, we preferentially select clicked items without strong positive feedback as negative samples, and use exposure-only items when such samples are unavailable. For pairs whose first SID differs, the positive and negative items are required to have the same exposure time. For pairs whose second SID differs, the negative interaction is allowed to occur within the 2 hours preceding the positive exposure. The preference data are constructed using the last three days of the training split together with the validation split. During DPO or CB-DPO training, we tune the hyperparameters over the following ranges. For Qwen3, the coefficient β is selected from $\{0.05, 0.1, 0.5, 1.0\}$, and the NTP regularization weight λ from $\{0, 1\}$; both reference-based and reference-free variants are considered. For TIGER, β is selected from $\{0.05, 0.1, 0.5, 1.0\}$, and λ from $\{0, 1, 3, 5\}$, again considering both reference-based and reference-free settings.

Table 4: Teacher-forced level-wise Hit@K on KuaiRec. Level 0 denotes the earliest SID level. Rel. Gain denotes the relative improvement of CB-DPO over the corresponding Base model.

Methods	Level 0			Level 1			Level 2		
	Hit@5	Hit@10	Hit@20	Hit@5	Hit@10	Hit@20	Hit@5	Hit@10	Hit@20
TIGER (Encoder–Decoder)									
Base	0.08850	0.15963	0.28208	0.74558	0.86037	0.92043	0.99964	1.00000	1.00000
DPO	0.08980	0.16114	0.28503	0.74775	0.86072	0.92082	0.99964	1.00000	1.00000
CB-DPO	0.09036	0.16185	0.28523	0.75490	0.86406	0.92105	0.99964	1.00000	1.00000
Rel. Gain	*+2.10%*	*+1.39%*	*+1.12%*	*+1.25%*	*+0.43%*	*+0.07%*	*0.00%*	*0.00%*	*0.00%*
Qwen3-1.7B (Decoder-only LLM)									
Base	0.09864	0.16675	0.28112	0.73098	0.85468	0.92062	0.99959	1.00000	1.00000
w/o Semantic Warmup	0.09115	0.15903	0.27118	0.74106	0.85388	0.91928	0.99925	1.00000	1.00000
DPO	0.10014	0.16857	0.28197	0.73481	0.85722	0.92108	0.99959	1.00000	1.00000
CB-DPO	0.09952	0.16744	0.28094	0.73655	0.85740	0.92148	0.99961	1.00000	1.00000
Rel. Gain	*+0.89%*	*+0.42%*	*-0.07%*	*+0.76%*	*+0.32%*	*+0.09%*	*0.00%*	*0.00%*	*0.00%*

D PROMPT EXAMPLES

Context2SID Example

Prompt: Generate the item SID according to its semantic information.
Category path: ACG → Animation → Teen Animation.
Item description: Po absorbs the chi of his clan, becomes the Dragon Warrior, and defeats Kai. #Animation #AnimeRecap #KungFuPanda.
Item SID:

Completion:
<BOI_ITEM> <SID_L0_217> <SID_L1_358> <SID_L2_111> <EOI_ITEM>

SID2Context Example

Prompt: Generate the semantic information of the item according to its SID.
Item SID: <BOI_ITEM> <SID_L0_558> <SID_L1_771> <SID_L2_805> <EOI_ITEM>.
Item semantic information:

Completion:
Category path: Real Estate & Home → Interior Design → Renovation Materials.
Item description: Do you know these gas stove repair techniques? #GasStoveRepairTips

Seq2Item Example

Prompt: Given the SID sequence of a user's historical effective interactions, predict the SID of the next item receiving high-value positive feedback. User history:
<BOI_ITEM> <SID_L0_405> <SID_L1_263> <SID_L2_116> <EOI_ITEM>...
<BOI_ITEM> <SID_L0_530> <SID_L1_371> <SID_L2_52> <EOI_ITEM>
Next high-value positive-feedback item SID:

Completion: <BOI_ITEM> <SID_L0_517> <SID_L1_411> <SID_L2_245> <EOI_ITEM>