TRAIN FINE, SERVE COARSE: TOKENIZER-AGNOSTIC ITEM-LEVEL ALIGNMENT FOR GENERATIVE RECOMMENDATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Generative recommendation models commonly undergo post-training to align their outputs with application-specific objectives such as relevance or user preference. The rewards used for alignment are typically defined on individual items, whereas policy optimization operates on the semantic IDs that encode these items. This creates a mismatch between the granularity of reward evaluation and policy optimization when multiple items share a semantic ID, or *node*. Items within the same node can receive different rewards for the same input, yet their rewards affect a single shared node likelihood, limiting the policy's ability to learn from these distinctions. We propose **Depth-Extended Alignment (DEA)**, which introduces a training-time refinement of the semantic-ID hierarchy to enable policy optimization at the granularity of individual items. We derive a hierarchical alignment objective that accounts for reward differences across nodes and among items within each node, while shared representations enable item-specific feedback to update the generative model. At inference, DEA preserves the original node-level decoding process without additional item-level computation. Offline experiments on public and real-world industrial datasets demonstrate substantial performance improvements with both RQ-Kmeans and RQ-VAE, two widely used tokenization methods. Validated through rigorous online A/B testing, DEA now operates in full production on a large-scale e-commerce platform serving hundreds of millions of active users.

1 INTRODUCTION

Generative recommendation uses a tokenizer to encode items as short semantic-ID sequences and retrieves candidates by autoregressively generating these sequences (Rajput et al., 2023; Deng et al., 2025; Xue et al., 2026; Jiang et al., 2026). Post-training then aligns the generative model with application-specific objectives such as relevance or user preference (Deng et al., 2025; Zhou et al., 2025; Chen et al., 2026). Typical approaches obtain reward or preference signals for recommendation outputs and use them to update generation probabilities: OneRec constructs preference pairs with a reward model for direct preference optimization, while OneRec-V2 uses interaction-derived rewards for reinforcement learning (Deng et al., 2025; Zhou et al., 2025). Such rewards can evaluate individual items (Zhou et al., 2025; Jiang et al., 2026), as with OneRec-V2's duration-aware reward for a target video, whereas policy optimization acts on generated semantic IDs. We call a complete semantic ID a *node*. When multiple items share a node, whether through quantization collisions or deliberate grouping for efficient retrieval (Rajput et al., 2023; Zhu et al., 2026a), reward evaluation and policy optimization operate at different granularities. Items receiving different rewards for the same input nevertheless contribute through a single node likelihood, limiting the policy's ability to learn from these within-node distinctions.

Industrial systems can train shared-SID generation policies by selecting a representative item from each bucket or mapping item-level behavioral feedback to SID-level rewards. For example, UniVA selects the highest-bid eligible advertisement from a complete SID bucket and computes the terminal reward for that selected item (Zhang et al., 2026). The retrieval framework using CQ-SID instead assigns rewards according to whether a generated SID belongs to the sets associated with purchased, clicked, or exposed items (Zhu et al., 2026a). These mechanisms provide training signals for gen-

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.

erated nodes, but the signals themselves do not explicitly distinguish the utilities of different items sharing a complete SID. When feedback enters policy optimization only through the shared node likelihood, it lacks an item-dependent update channel (Cor. 1). Alternatively, TIGER appends a disambiguation token to make each complete semantic ID unique to an item (Rajput et al., 2023). This removes the granularity mismatch but requires decoding more nodes to retrieve a candidate set of comparable size, increasing beam-search cost relative to multi-item nodes (Zhu et al., 2026a). Even when a tokenizer groups items accurately along its chosen semantic dimensions, business applications may require alignment with multiple objectives that depend on different item attributes. Semantic coherence within a node therefore does not ensure reward agreement across these objectives. This tension can become more pronounced in scenarios with fine-grained contextual constraints. For example, search imposes query-specific relevance requirements, demanding that item encodings preserve distinctions beyond broad semantic similarity (Zhu et al., 2026a). A node containing both Nikon and Canon cameras may include relevant and irrelevant items for the query "Nikon camera," while both may be relevant to "DSLR camera." Thus, a fixed semantic grouping need not preserve reward agreement across queries. In our industrial search setting, 26.5% of evaluated query–node pairs with multi-item buckets contain mixed relevance labels (§5.1). The challenge is therefore to exploit item-level reward differences during alignment without giving up the retrieval efficiency of shared-node generation.

We propose **Depth-Extended Alignment (DEA)**, which refines the semantic-ID hierarchy during training so that policy optimization can distinguish individual items within each node. This training-time refinement brings optimization to the same granularity as reward evaluation, while leaving the existing tokenizer and node-level output space unchanged. The item-level refinement shares the decoder backbone with the node policy, allowing item-level optimization to update the model used for node-level inference. We decompose changes in the joint training objective into cross-node and within-node terms (Prop. 1) to organize the hierarchical alignment surrogate. At inference, DEA retains the original node-level decoding process without evaluating the item-level extension, preserving serving efficiency (§4.1).

Offline experiments on public and real-world industrial datasets demonstrate substantial performance improvements with both RQ-Kmeans (Deng et al., 2025) and RQ-VAE (Rajput et al., 2023), two widely used tokenization methods. Controlled comparisons and ablations examine the contribution of item-level refinement and the hierarchical objective while preserving the same node-level serving protocol. Following rigorous online A/B evaluation, DEA has been fully deployed on a large-scale e-commerce platform serving hundreds of millions of active users.

Our main contributions are summarized as follows:

- We identify the granularity gap between item-level rewards and node-level policy optimization, and characterize the limitation of a shared node likelihood in exploiting within-node reward differences.
- We introduce DEA with a hierarchical training objective and characterize how item-specific feedback updates the shared backbone retained for node-level inference.
- We demonstrate DEA's effectiveness through offline experiments on industrial and public datasets and online A/B testing, showing improvements in retrieval relevance and platform revenue in production.

2 Related Work

Tokenization for generative recommendation. TIGER quantizes item representations into semantic IDs and appends a token to disambiguate collisions (Rajput et al., 2023), while LC-Rec combines quantized item identifiers with tuning to align language and collaborative semantics (Zheng et al., 2024). Subsequent work develops learnable tokenizers (Wang et al., 2024; Liu et al., 2025) and multi-identifier pre-training (Zheng et al., 2025). In contrast to unique item identifiers, CQ-SID explicitly groups items into semantic clusters to reduce the beam-search budget in e-commerce search (Zhu et al., 2026a). These approaches shape the representation and retrieval granularity through item encoding. DEA instead refines the alignment granularity under a fixed tokenizer.

Reward alignment and credit assignment. OneRec and OneRec-V2 align generative recommenders through iterative preference alignment and interaction-based rewards, respectively (Deng et al., 2025; Zhou et al., 2025); S-DPO incorporates multiple negatives into preference optimization (Chen et al., 2024). For shared semantic IDs, UniVA selects the highest-bid eligible advertisement within a complete SID bucket and evaluates its terminal reward (Zhang et al., 2026), while the CQ-SID framework assigns rewards according to membership in SID sets associated with purchased, clicked, or exposed items (Zhu et al., 2026a). Other work refines credit assignment along different dimensions: Rank-GRPO uses rank-wise advantages and importance ratios for generated recommendation lists (Zhu et al., 2026b); RAD-DPO detaches negative-sample log-likelihood terms on shared SID prefixes (Chen et al., 2026); F-GRPO assigns separate advantages to candidate generation and ordering using coverage and ranking rewards (Surana et al., 2026). DEA addresses reward differences among items sharing a *complete* SID. It introduces an item-dependent update channel and derives cross-node and within-node advantages from the same item reward, rather than assigning credit to list positions, partially shared prefixes, or generation and ranking stages.

Item-level modeling and training-only refinement. Gryphon and Gryphon-v2 combine semantic-ID generation with item-level scoring that remains active at inference (Tikhonovich et al., 2026; Lipkina et al., 2026). Training-only objectives offer another way to enrich shared representations, as in auxiliary learning and deep supervision (Jaderberg et al., 2016; Lee et al., 2015). In generative recommendation, OxygenREC-v2 distills a teacher conditioned on future interactions into a student without that privileged input (Tang et al., 2026). DEA instead introduces a conditional item policy within each node during alignment. This refinement participates in the joint-policy objective and updates the shared generative model, while inference retains only the original node-level decoding process without item-level scoring.

3 PRELIMINARIES

Let $\mathcal{I}$ denote the item catalog. A fixed tokenizer $\mathcal{T}$ maps each item $c \in \mathcal{I}$ to an L-token semantic ID $y_{1:L} = (y_1, \ldots, y_L)$, with y_ℓ drawn from a codebook of size V_ℓ. Given an input context x, an autoregressive policy factorizes as

$$\pi_\theta(y_{1:L} \mid x) = \prod_{\ell=1}^{L} \pi_\theta(y_\ell \mid x, y_{1:\ell-1}). \tag{1}$$

We call a complete semantic ID a *node* $n = y_{1:L}$ and write its probability as $\pi_\theta(n \mid x)$. Each occupied node corresponds to a *bucket* $S_n = \{c \in \mathcal{I} : \mathcal{T}(c) = n\}$ of $K_n = |S_n|$ items. For alignment, π_θ denotes the autoregressive policy whose token probabilities are normalized over valid continuations of occupied-node prefixes (Appendix B). At inference, a retrieved node n is expanded into its candidate items S_n.

Let $r(x, c)$ denote the reward assigned to item c for context x. Item-level interaction feedback provides alignment rewards in generative recommendation, as exemplified by the duration-aware watch feedback in OneRec-V2 (Zhou et al., 2025). In our alignment procedure, rewards are evaluated for items within sampled nodes. Node-level alignment converts item rewards into training signals for $\pi_\theta(n \mid x)$. For fixed (x, n) and reward-based coefficients, contributions through this shared likelihood remain scalar multiples of the same node-score direction (Cor. 1).

4 METHOD: DEPTH-EXTENDED ALIGNMENT

Figure 1 presents an overview of DEA. DEA enables item-level policy optimization during alignment while retaining node-level inference. It introduces a conditional item policy within each node's bucket and parameterizes it using a hidden state shared with the generative model (§4.1). We decompose the change in the joint policy's expected reward into cross-node and within-node terms (§4.2), providing the basis for the practical alignment surrogate in §4.3.

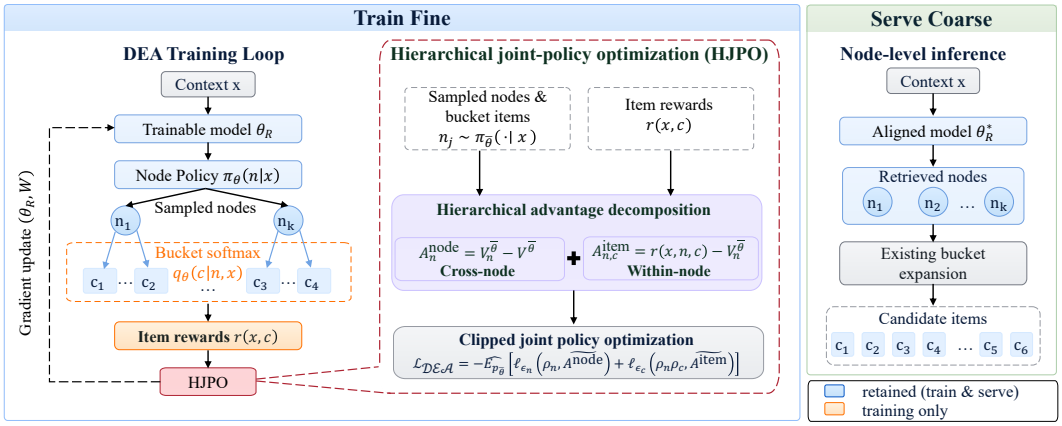

Figure 1: Overview of Depth-Extended Alignment (DEA). In the compact loss, $\widehat{\mathbb{E}}_{p_{\bar{\theta}}}$ denotes an average over sampled nodes and enumeration of bucket items weighted by the fixed $q_{\bar{\theta}}(c \mid n, x)$. Since these weights sum to one within each bucket, this notation is equivalent to Eq. 14. The normalized advantages $\widetilde{A}$ are defined in Eq. 12.

4.1 Training-time item-level refinement

With the tokenizer fixed, DEA refines each node $n = y_{1:L}$ into its constituent items during training. The item policy reuses a hidden state computed by the generative model. Let $q_\theta(c \mid n, x)$ be a conditional distribution over the nonempty bucket S_n. The training policy is

$$p_\theta(n, c \mid x) = \pi_\theta(n \mid x) q_\theta(c \mid n, x), \qquad \sum_{c \in S_n} q_\theta(c \mid n, x) = 1. \tag{2}$$

The joint action (n, c) receives the item reward $r(x, n, c) \equiv r(x, c)$.

Let θ_R denote the original node-model parameters retained at inference, including the decoder backbone parameters θ_B and the original L-level output heads. DEA preserves the original node-head architecture. The final node-token logits are

$$\mathbf{z} = \text{Head}^{\text{orig}}(h_{L-1}), \qquad h_{L-1} = h_{\theta_B}(x, y_{1:L-1}) \in \mathbb{R}^{d_h}. \tag{3}$$

The item policy reuses this hidden state. For a frozen item embedding $e_c \in \mathbb{R}^d$, it introduces a trainable projection $W \in \mathbb{R}^{d \times d_h}$ used only during alignment:

$$\ell_c = e_c^\top W \, h_{L-1} / \tau, \qquad q_\theta(c \mid n, x) = \frac{\exp(\ell_c)}{\sum_{c' \in S_n} \exp(\ell_{c'})}, \tag{4}$$

with fixed temperature $\tau > 0$ and full training parameters $\theta = (\theta_R, W)$. The node policy depends only on θ_R, whereas the item policy depends on θ_B and W. We take e_c to be the frozen continuous item representation entering the tokenizer's first quantization stage, before codeword assignment. This is the quantization input for RQ-Kmeans and the encoder latent for RQ-VAE, with the tokenizer's existing preprocessing retained (Appendix B).

The completed node selects S_n, while its item logits reuse h_{L-1} without detaching it from the backbone. Computing them requires additional training-time scoring but no additional decoder step. The item log-probability gradient updates W and reaches the retained model through θ_B; W itself is not a deployed parameter. Thus DEA adds training-time parameters but no inference-time parameters or item-level computation.

Node-level inference. Normalization within each bucket gives

$$\sum_{c \in S_n} p_\theta(n, c \mid x) = \pi_\theta(n \mid x) \sum_{c \in S_n} q_\theta(c \mid n, x) = \pi_\theta(n \mid x). \tag{5}$$

Inference discards the item branch, including W, and uses only the aligned backbone and original node output heads, without computing q_θ. The tokenizer, output-head architecture, decoding structure, and node-to-item serving protocol remain unchanged.

4.2 HIERARCHICAL TRAINING OBJECTIVE

For a fixed context x, we use the expected item reward under the joint policy, $J(\theta) = \mathbb{E}_{p_\theta(\cdot|x)}[r(x, n, c)]$, as a training objective and average it over contexts. Deployment instead uses the node policy with the fixed node-to-item expansion protocol. Rewards and buckets are fixed during an update. The analysis takes π_θ to be a normalized policy on occupied nodes, with target support contained in the behavior support. The prefix-masked policy in Appendix B supplies this normalization; its masked token probabilities define both the rollout distribution and the full-node importance ratios.

Let $p_{\bar\theta} = \pi_{\bar\theta} q_{\bar\theta}$ be a fixed behavior snapshot. Its node and context values are

$$V_n^{\bar\theta}(x) = \sum_{c \in S_n} q_{\bar\theta}(c \mid n, x) r(x, n, c), \qquad V^{\bar\theta}(x) = \sum_n \pi_{\bar\theta}(n \mid x) V_n^{\bar\theta}(x). \tag{6}$$

Suppressing x where unambiguous, define

$$\begin{aligned} A_n^{\mathrm{node}} &= V_n^{\bar\theta} - V^{\bar\theta}, \qquad A_{n,c}^{\mathrm{item}} = r(x, n, c) - V_n^{\bar\theta}, \\ r(x, n, c) - V^{\bar\theta} &= A_n^{\mathrm{node}} + A_{n,c}^{\mathrm{item}}. \end{aligned} \tag{7}$$

The first advantage compares a node with the behavior policy's average return; the second compares an item with its node value. Both are derived from the same item reward. Write $\rho_n = \pi_\theta(n \mid x)/\pi_{\bar\theta}(n \mid x)$ and $\rho_c = q_\theta(c \mid n, x)/q_{\bar\theta}(c \mid n, x)$.

The following identity separates changes in the joint training objective into cross-node reallocation and within-node refinement, motivating the two components of our alignment surrogate.

Proposition 1 (Joint-policy return decomposition)**.** *Under the support conditions above, with $\bar\theta$ held fixed,*

$$J(\theta) - J(\bar\theta) = \mathbb{E}_{(n,c)\sim p_{\bar\theta}}\big[\, \rho_n \, A_n^{\mathrm{node}} + \rho_n \rho_c \, A_{n,c}^{\mathrm{item}} \,\big]. \tag{8}$$

The within-node term uses the joint ratio $\rho_n \rho_c$. This identity holds even when the two policy factors share parameters (proof in Appendix A). The decomposition concerns the joint training objective and does not guarantee improved node-level retrieval under the serving protocol.

To connect the decomposition to parameter updates, define the scores $g_n = \nabla_\theta \log \pi_\theta(n \mid x)|_{\bar\theta}$ and $u_{n,c} = \nabla_\theta \log q_\theta(c \mid n, x)|_{\bar\theta}$ (Williams, 1992). At the on-policy point,

$$\nabla_\theta J(\theta)|_{\bar\theta} = \mathbb{E}_{n\sim\pi_{\bar\theta}}[A_n^{\mathrm{node}} g_n] + \mathbb{E}_{(n,c)\sim p_{\bar\theta}}[A_{n,c}^{\mathrm{item}} u_{n,c}]. \tag{9}$$

Here $\mathbb{E}_{q_{\bar\theta}}[A_{n,c}^{\mathrm{item}} \mid n] = 0$, so the within-node contribution along g_n vanishes. Away from this on-policy point, differentiating the within-node term in Eq. 8 generally involves both policy scores.

Corollary 1 (Within-node shared-direction property)**.** *For fixed (x, n), if item rewards enter an update only through the node likelihood with coefficients held fixed during differentiation, all item contributions are scalar multiples of the same node-score direction. The item policy instead has the hidden-state score*

$$\nabla_h \log q_\theta(c \mid n, x) = \frac{W^\top}{\tau}\left(e_c - \sum_{c' \in S_n} q_\theta(c' \mid n, x)\, e_{c'} \right), \qquad h = h_{L-1}. \tag{10}$$

By the chain rule, its backbone gradient is $(D_{\theta_B} h)^\top \nabla_h \log q_\theta(c \mid n, x)$, providing an item-dependent update channel to the shared decoder.

The item term provides an item-dependent gradient to the shared backbone retained at inference. We evaluate its effect on retrieval using only the node policy under the unchanged serving protocol.

4.3 PRACTICAL ALIGNMENT OBJECTIVE

We instantiate the two terms in Eq. 8 as a GRPO-style alignment objective.

For each context x, we independently sample $M \geq 2$ occupied nodes with replacement from $\pi_{\bar\theta}(\cdot \mid x)$ and enumerate each bucket S_{n_j}. Sampling and probability bookkeeping follow Appendix B. The

behavior snapshot, rewards, and computed advantages remain fixed during updates. The node value and its leave-one-out baseline (Ahmadian et al., 2024) are

$$\widehat{V}_j = \sum_{c \in S_{n_j}} \mathrm{sg}\big(q_{\bar{\theta}}(c \mid n_j, x)\big)\, r(x, n_j, c), \qquad \widehat{V}_{-j} = \tfrac{1}{M-1} \sum_{k \neq j} \widehat{V}_k, \tag{11}$$

where sg denotes stop-gradient.

The practical advantages are

$$\widetilde{A}^{\mathrm{node}}_j = \mathrm{sg}\left[\tfrac{\widehat{V}_j - \widehat{V}_{-j}}{s}\right], \qquad \widetilde{A}^{\mathrm{item}}_{j,c} = \mathrm{sg}\left[\tfrac{r(x,n_j,c) - \widehat{V}_j}{s}\right], \tag{12}$$

where s is the standard deviation of pooled per-item rewards $\{r(x, n_j, c) : j, c\}$ in the sampled query group, floored at $\varepsilon = 10^{-6}$. A common scale preserves the relative scaling of the two components within a group.

The node ratio uses the probability of the full semantic ID, treated as one action:

$$\rho^{(j)}_n = \frac{\pi_\theta(n_j \mid x)}{\pi_{\bar{\theta}}(n_j \mid x)}, \qquad \rho^{(j)}_c = \frac{q_\theta(c \mid n_j, x)}{q_{\bar{\theta}}(c \mid n_j, x)}. \tag{13}$$

Using the PPO-style operator (Schulman et al., 2017) $\ell_\epsilon(\rho, A) = \min(\rho A, \mathrm{clip}(\rho, 1-\epsilon, 1+\epsilon)A)$, we minimize

$$\mathcal{L}_{\mathrm{DEA}}(x) = -\frac{1}{M} \sum_{j=1}^{M} \Big[\ell_{\epsilon_n}\big(\rho^{(j)}_n,\, \widetilde{A}^{\mathrm{node}}_j\big) + \sum_{c \in S_{n_j}} \mathrm{sg}\big(q_{\bar{\theta}}(c \mid n_j, x)\big)\, \ell_{\epsilon_c}\big(\rho^{(j)}_n \rho^{(j)}_c,\, \widetilde{A}^{\mathrm{item}}_{j,c}\big) \Big], \tag{14}$$

averaged over contexts. The node term is clipped on ρ_n, and the item term on $\rho_n \rho_c$; the main experiments use $\epsilon_n = \epsilon_c$. We compare this clipping organization with flat joint clipping in §5.3.

Common-scale normalization and separate clipping define a practical optimization surrogate. The surrogate optimizes the joint training policy; experiments evaluate the resulting node policy under the fixed serving protocol.

5 Experiments

5.1 Experimental setup

Datasets. We evaluate DEA on a proprietary industrial dataset and the public KuaiRec dataset. The industrial benchmark uses anonymized user–ad interaction logs from consecutive days for alignment training and evaluation, with query–title relevance scores as rewards. Among evaluated query–node pairs with multi-item buckets, 26.5% contain mixed relevance labels; the audit definition and bucket statistics are given in Appendix C. KuaiRec provides a supplementary short-video recommendation task with observed watch feedback (Gao et al., 2022); data construction and reward details are given in Appendix C.2.

Backbone. All methods use Qwen3-0.6B (Team, 2025) as the generative recommendation backbone, initialized from the same SFT checkpoint for each task and tokenizer. The 0.6B size is fixed across all methods in this study, constrained by the serving cost and latency budget of the production generative retrieval branch. Both tasks use fixed RQ-Kmeans and RQ-VAE tokenizers and node-level inference. For the industrial GRPO-based comparisons, we match candidate enumeration, optimizer steps, query batch size, and sampled-node count per query. Sampling, item representations, and industrial training settings are detailed in Appendix B.

Baselines. We compare DEA with SFT (base), DPO (Rafailov et al., 2023), and two node-level GRPO (Shao et al., 2024; Liu et al., 2024) variants. GRPO-Agg (aggregate) adapts the candidate-list reward aggregation in CRS (Liu et al., 2026) to within-node aggregation. In our industrial setting, its node reward is a weighted relevance rate: the proportions of relevant, partially relevant, and irrelevant items are weighted by 1.0, 0.5, and 0, respectively. This is the current production alignment baseline. GRPO-Max draws on UniVA's representative-item selection strategy (Zhang et al., 2026), selecting by task reward rather than advertisement bid and using the highest item reward as the node reward. Baseline objectives are detailed in Appendix C.

Table 1: Main results on industrial search and KuaiRec (%). Best values are in **bold**.

	Industrial search				KuaiRec	
	RQ-Kmeans		RQ-VAE		RQ-Kmeans	RQ-VAE
Method	Relevance	HR@500	Relevance	HR@500	Watch ratio	Watch ratio
SFT (base)	55.90 ± 0.11	65.47 ± 0.15	51.67 ± 0.10	62.27 ± 0.14	52.97 ± 0.12	49.57 ± 0.13
DPO	57.23 ± 0.14	68.55 ± 0.17	53.92 ± 0.11	63.04 ± 0.15	55.26 ± 0.13	53.80 ± 0.16
GRPO-Agg	57.54 ± 0.12	68.88 ± 0.16	53.55 ± 0.10	64.95 ± 0.14	56.25 ± 0.13	52.44 ± 0.12
GRPO-Max	58.09 ± 0.15	67.14 ± 0.14	52.51 ± 0.12	63.44 ± 0.11	55.87 ± 0.14	53.94 ± 0.15
DEA	$\mathbf{61.50} \pm 0.16$	$\mathbf{70.22} \pm 0.18$	$\mathbf{55.64} \pm 0.13$	$\mathbf{65.73} \pm 0.17$	$\mathbf{58.26} \pm 0.15$	$\mathbf{55.64} \pm 0.14$

Metrics. For industrial search, relevance is the proportion of relevant items among 1,500 candidates expanded from retrieved nodes. Hit rate (HR@500) measures hits against held-out interaction targets when retrieved nodes are expanded into items and the candidate list is truncated to 500 items. Relevance is the primary industrial metric; HR@500 serves as a guardrail to check that alignment does not degrade retrieval of held-out interaction targets. KuaiRec uses normalized clipped watch ratio. Performance metrics in Tables 1–3 are reported as mean $\pm$ standard deviation over three random seeds; Δ_{IR} reports the difference in mean relevance. Evaluation details are given in Appendix C.

5.2 Main results

Table 1 evaluates whether item-level alignment improves retrieval over node-level preference optimization and reward aggregation across tasks and tokenizers, under the same node-level serving protocol.

On industrial search, DEA achieves the highest relevance with both tokenizers, improving over GRPO-Agg by 3.96 percentage points with RQ-Kmeans and 2.09 points with RQ-VAE. HR@500 does not decrease relative to GRPO-Agg under either tokenizer, satisfying the guardrail criterion. DEA also outperforms GRPO-Max, supporting the effectiveness of item-level alignment beyond changing the scalar node-reward aggregation rule.

On KuaiRec, DEA improves watch ratio over the strongest baseline for each tokenizer by 2.01 percentage points with RQ-Kmeans and 1.70 points with RQ-VAE. Together with the industrial results, these gains show that DEA's benefits extend across distinct reward signals and tokenization methods.

5.3 Ablation study

Table 2 compares DEA with a fixed-reference centered auxiliary control and two component variants on industrial search with both tokenizers. The controls retain DEA's initialization, training budget, and common-scale advantage normalization. Appendix C specifies their objectives.

The fixed-reference centered auxiliary control supplements node-level alignment with a centered, signed-weight item-level surrogate under a fixed uniform reference, using the same backbone and training-only item-branch architecture as DEA. DEA achieves higher relevance and HR@500 with both tokenizers (Table 2). This comparison evaluates the complete alignment formulations, which differ in item reference distributions, node valuation, importance weighting, and item clipping.

We consider the following component variants:

- **DEA without item-level refinement.** Remove the within-node optimization term and use a fixed uniform distribution within each node to compute its value. This variant retains DEA's node-level estimator but removes item-dependent policy optimization.
- **DEA with flat joint clipping.** Retain the joint node–item policy, advantages, common scale, and behavior-policy enumeration weights, but apply a single joint clipping operation to the summed advantage.

Both component variants yield lower relevance and HR@500 than full DEA with both tokenizers (Table 2). Without item-level refinement, the retained node-level estimator improves both metrics

Table 2: Auxiliary and component ablations on industrial search (%).

Method	RQ-Kmeans		RQ-VAE	
	Relevance	HR@500	Relevance	HR@500
Fixed-reference auxiliary	60.67 ± 0.14	69.12 ± 0.15	54.39 ± 0.12	64.32 ± 0.13
DEA w/o IR	59.21 ± 0.15	69.71 ± 0.16	53.51 ± 0.10	64.72 ± 0.14
DEA with flat clipping	59.83 ± 0.13	68.65 ± 0.14	53.74 ± 0.10	64.01 ± 0.12
DEA	$\mathbf{61.50} \pm 0.16$	$\mathbf{70.22} \pm 0.18$	$\mathbf{55.64} \pm 0.13$	$\mathbf{65.73} \pm 0.17$

Table 3: Relevance by mixed-label node ratio with RQ-Kmeans (%). Query shares are among queries included in the stratification; Δ_{IR} denotes the mean relevance gain of DEA over DEA w/o IR in percentage points.

Group	Queries (%)	GRPO-Agg	DEA w/o IR	DEA	Δ_{IR}
Low ($C < 0.33$)	53.2	62.05 ± 0.07	62.27 ± 0.14	$\mathbf{62.54} \pm 0.12$	$+0.27$
Medium ($0.33 \leq C < 0.66$)	30.7	59.83 ± 0.13	60.09 ± 0.12	$\mathbf{61.42} \pm 0.14$	$+1.33$
High ($C \geq 0.66$)	16.1	56.13 ± 0.13	57.74 ± 0.15	$\mathbf{59.32} \pm 0.14$	$+1.58$

over GRPO-Agg with RQ-Kmeans, but neither with RQ-VAE. The flat-clipping comparison evaluates separate versus joint clipping under the same four-update behavior-snapshot schedule; the unclipped objectives remain equivalent (Appendix C.3).

Effect of within-node reward disagreement. We examine whether item-level refinement provides greater benefits when within-node rewards disagree more frequently. On a separate industrial evaluation dataset with RQ-Kmeans, we group queries by $C(x)$, the fraction of audited multi-item nodes with mixed relevance labels among the pre-alignment retriever's top-500 nodes (Appendix C). The fixed groups in Table 3 are shared across methods and include queries with at least one audited multi-item node; each method is evaluated on its own top-1,500 expanded items. DEA outperforms both controls in all groups. Its relevance gain over DEA w/o IR rises from 0.27 percentage points in the low-disagreement group to 1.33 and 1.58 points in the medium- and high-disagreement groups, respectively. The larger gains on queries with more mixed-label nodes support the value of item-level refinement where a shared node reward obscures differences among its items.

5.4 Training dynamics

Figure 2 tracks deployed node policy relevance during industrial alignment with RQ-Kmeans. We evaluate the top-1,500 expanded items on a fixed held-out query set every 500 optimizer steps. Curves show means over three random seeds, with error bars indicating standard deviations. DEA establishes an early advantage and maintains higher relevance than both controls across checkpoints. GRPO-Agg largely plateaus after approximately 2,000–2,500 steps, whereas DEA without item-level refinement (DEA w/o IR) continues to improve but remains below full DEA. At 5,000 steps, DEA reaches 61.50% relevance, compared with 59.21% for DEA w/o IR and 57.54% for GRPO-Agg. The persistent advantage over DEA w/o IR supports item-level refinement beyond the retained node-level estimator and normalization. All checkpoints use node-level inference.

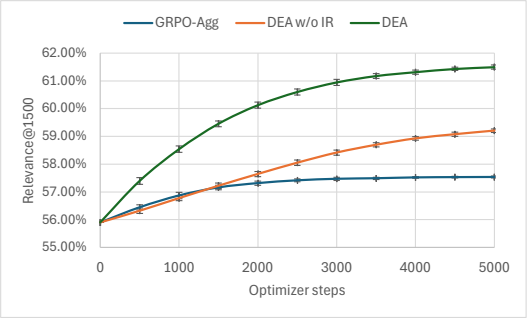

Figure 2: Training dynamics on industrial search with RQ-Kmeans.

Table 4: Training time and inference latency.

Method	Training time (h)	Inference latency (s)
GRPO-Agg	8.5	0.157 ± 0.002
GRPO-Max	8.3	0.157 ± 0.002
DEA	10.0	0.157 ± 0.002

Table 5: Online A/B results of DEA.

Metric	DEA
Relevance	$+1.64\%$
Clicks	$+0.72\%$
Revenue	$+1.34\%$

5.5 Training and inference cost

Table 4 compares the training time and inference latency of DEA with GRPO-Agg and GRPO-Max using RQ-Kmeans. All methods are trained on eight NVIDIA B200 GPUs with 192 GB of memory per GPU under the configuration in Table 6. Inference is benchmarked on an NVIDIA GeForce RTX 4090 GPU with 24 GB of memory, using a batch size of 256 and FP16 precision for all methods. Latency measures the time to complete one candidate-retrieval request for a single query. DEA takes 10.0 hours to train, compared with 8.5 hours for GRPO-Agg and 8.3 hours for GRPO-Max. Its training-time refinement incurs a relative cost of $1.18\times$ compared with GRPO-Agg, while all three methods have a reported inference latency of 0.157 ± 0.002 seconds, where $\pm$ denotes the observed variation range across repeated runs. This is consistent with DEA retaining node-level inference without evaluating the training-only item branch. This trade-off is useful when serving latency is more constrained than training compute.

5.6 Online A/B test

We launched DEA in a large-scale e-commerce platform and conducted a one-week online A/B test covering 5% of traffic. The control group uses the highly optimized production GRPO-Agg baseline described in the experimental setup. Table 5 reports relative gains over this baseline: 1.64% in relevance, 0.72% in clicks, and 1.34% in revenue. These results are consistent with DEA's fine-grained item-level alignment improving the relevance of retrieved items and contributing to higher platform revenue. All three metrics improve simultaneously, demonstrating the practical effectiveness of DEA in production.

6 Limitations

We deploy the node marginal, so inference emits a node, not an item within it; DEA shapes the node policy but cannot restore within-node item selection at serving. This boundary follows from the retained node-level serving interface: DEA focuses on improving alignment within that interface, while finer-grained item selection at inference is a complementary direction.

7 Conclusion

We address a structural granularity mismatch in generative recommendation post-training: rewards distinguish individual items, while policy optimization operates on semantic IDs that can be shared by multiple items. A shared node likelihood limits how optimization can exploit reward differences within a node. DEA bridges this gap through a training-time refinement of the semantic-ID hierarchy and a hierarchical objective that accounts for reward differences both across and within nodes. Shared representations allow item-specific feedback to update the generative model, while the tokenizer and node-level decoding process remain unchanged at inference. Offline experiments on public and industrial datasets with RQ-Kmeans and RQ-VAE demonstrate consistent retrieval improvements, and online A/B testing shows gains in retrieved-item quality and platform revenue. These findings support the central principle of training at item granularity while serving at node granularity: fine-grained alignment can improve generative recommendation without sacrificing the retrieval efficiency of shared semantic IDs.

AI USE STATEMENT

During the preparation of this work, the authors used AI assistant to assist with language polishing, improving readability, and enhancing grammatical correctness. All research ideas, the method design and the experimental protocol were conceived by the authors. The authors have reviewed all AI-assisted content and take responsibility for the final text, claims, and associated artifacts.

ETHICS STATEMENT

We use the KuaiRec dataset (Gao et al., 2022) strictly in accordance with its license and terms of use. Our industrial experiments use anonymized interaction logs, and the use of industrial data, online A/B testing, and production deployment comply with applicable ethical and privacy requirements.

REPRODUCIBILITY STATEMENT

Proofs are provided in Appendix A. Appendix B specifies occupied-node sampling, item representations and initialization, and the industrial training configuration. Appendix C describes dataset construction, reward and evaluation definitions, and the baseline and ablation objectives. Training hardware and measured computational costs are reported in Section 5.5.

REFERENCES

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting reinforce-style optimization for learning from human feedback in llms. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12248–12267, 2024.

Yuxin Chen, Junfei Tan, An Zhang, Zhengyi Yang, Leheng Sheng, Enzhi Zhang, Xiang Wang, and Tat-Seng Chua. On softmax direct preference optimization for recommendation. *Advances in Neural Information Processing Systems*, 37:27463–27489, 2024.

Zhiguo Chen, Guohao Sun, Yiming Qiu, Xingzhi Yao, Mingming Li, Huimu Wang, Yangqi Zhang, Songlin Wang, and Sulong Xu. Rad-dpo: Robust adaptive denoising direct preference optimization for generative retrieval in e-commerce. *arXiv preprint arXiv:2602.23964*, 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, 2022.

Max Jaderberg, Volodymyr Mnih, Wojciech Marian Czarnecki, Tom Schaul, Joel Z Leibo, David Silver, and Koray Kavukcuoglu. Reinforcement learning with unsupervised auxiliary tasks. *arXiv preprint arXiv:1611.05397*, 2016.

Jie Jiang, Jun Zhang, Yi Li, Yue Liu, Changping Wang, Yuan Wang, Yuling Xiong, Xun Liu, Haiyang Wu, Qian Li, et al. Gpr: Towards a generative pre-trained one-model paradigm for large-scale advertising recommendation. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 7477–7488, 2026.

Chen-Yu Lee, Saining Xie, Patrick Gallagher, Zhengyou Zhang, and Zhuowen Tu. Deeply-supervised nets. In *Artificial intelligence and statistics*, pp. 562–570. Pmlr, 2015.

Anna Lipkina, Daria Tikhonovich, Viktor Yanush, Mariia Ulianova, Oleg Sorokin, Vladislav Dodonov, Ilya Murzin, Denis Burshtein, and Nikolay Savushkin. Gryphon-v2: One model in place of a cascade-generate-and-rank recommender with rollout distillation. *arXiv preprint arXiv:2608.06213*, 2026.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

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, 2025.

Zhiding Liu, Ben Chen, Mingyue Cheng, Enhong Chen, Li Li, Chenyi Lei, Wenwu Ou, Han Li, and Kun Gai. Towards context-aware reasoning-enhanced generative searching in e-commerce. In *Proceedings of the ACM Web Conference 2026*, pp. 6551–6561, 2026.

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.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Rohan Surana, Gagan Mundada, Junda Wu, Xintong Li, Yizhu Jiao, Bowen Jin, Sizhe Zhou, Tong Yu, Ritwik Sinha, Jiawei Han, et al. F-grpo: Factorized group-relative policy optimization for unified candidate generation and ranking. *arXiv preprint arXiv:2605.12995*, 2026.

Guo Tang, Hanye Wu, Changjiang Han, Qingyang Li, Ming Zhang, Xiangyu Qian, Yanchen Qiao, Huanjie Wang, Zhi Ma, Zhen Li, et al. Oxygenrec-v2: Internalizing discrimination into generative recommendation. *arXiv preprint arXiv:2607.24255*, 2026.

Qwen Team. Qwen3 technical report, 2025. URL `https://arxiv.org/abs/2505.09388`.

Daria Tikhonovich, Oleg Sorokin, Vladislav Dodonov, Mariia Ulianova, and Ilya Murzin. Gryphon: A unified architecture for semantic-id generation and item-level scoring in industrial recommendations. *arXiv preprint arXiv:2606.08604*, 2026.

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.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256, 1992.

Ben Xue, Dan Liu, Lixiang Wang, Mingjie Sun, Peng Wang, Pengfei Zhang, Shaoyun Shi, Tianyu Xu, Yunhao Sha, Zhiqiang Liu, et al. Generative recommendation for large-scale advertising. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 8346–8357, 2026.

Xinxun Zhang, Yuling Xiong, Jiale Zhou, Zhengkai Guo, Zhennan Pang, Junbang Huo, Jingwen Wang, Xuyang Sun, Enming Zhang, Jiaguang Jin, et al. Unified value alignment for generative recommendation in industrial advertising. *arXiv preprint arXiv:2605.05803*, 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.

Bowen Zheng, Enze Liu, Zhongfu Chen, Zhongrui Ma, Yue Wang, Wayne Xin Zhao, and Ji-Rong Wen. Pre-training generative recommender with multi-identifier item tokenization. *arXiv preprint arXiv:2504.04400*, 2025.

Guorui Zhou, Hengrui Hu, Hongtao Cheng, Huanjie Wang, Jiaxin Deng, Jinghao Zhang, Kuo Cai, Lejian Ren, Lu Ren, Liao Yu, Pengfei Zheng, Qiang Luo, Qianqian Wang, Qigen Hu, Rui Huang, Ruiming Tang, Shiyao Wang, Shujie Yang, Tao Wu, Wuchao Li, Xinchen Luo, Xingmei Wang, Yi Su, Yunfan Wu, Zexuan Cheng, Zhanyu Liu, Zixing Zhang, Bin Zhang, Boxuan Wang, Chaoyi Ma, Chengru Song, Chenhui Wang, Chenglong Chu, Di Wang, Dongxue Meng, Dunju Zang, Fan Yang, Fangyu Zhang, Feng Jiang, Fuxing Zhang, Gang Wang, Guowang Zhang, Han Li, Honghui Bao, Hongyang Cao, Jiaming Huang, Jiapeng Chen, Jiaqiang Liu, Jinghui Jia, Kun Gai, Lantao Hu, Liang Zeng, Qiang Wang, Qidong Zhou, Rongzhou Zhang, Shengzhe Wang, Shihui He, Shuang Yang, Siyang Mao, Sui Huang, Tiantian He, Tingting Gao, Wei Yuan, Xiao Liang, Xiaoxiao Xu, Xugang Liu, Yan Wang, Yang Zhou, Yi Wang, Yiwu Liu, Yue Song, Yufei Zhang, Yunfeng Zhao, Zhixin Ling, and Ziming Li. Onerec-v2 technical report, 2025. URL https://arxiv.org/abs/2508.20900.

Jianbo Zhu, Xing Fang, Jing Wang, Mingmin Jin, Bokang Wang, Guangxin Song, Zhenyu Xie, and Junjie Bai. Efficient generative retrieval for e-commerce search with semantic cluster ids and expert-guided rl. *arXiv preprint arXiv:2605.14434*, 2026a.

Yaochen Zhu, Harald Steck, Dawen Liang, Yinhan He, Vito Ostuni, Jundong Li, and Nathan Kallus. Rank-grpo: Training llm-based conversational recommender systems with reinforcement learning. In *International Conference on Learning Representations*, volume 2026, pp. 155483–155512, 2026b.

A PROOFS

We fix the context, rewards, buckets, and behavior snapshot, and use the normalized policies and fixed support specified in §4.2.

Proof of Proposition 1. Write $V_n^\theta = \sum_{c\in S_n} q_\theta(c \mid n,x)r(x,n,c)$. Adding and subtracting $\sum_n \pi_\theta(n \mid x)V_n^{\bar\theta}$ gives

$$\begin{aligned} J(\theta) - J(\bar\theta) &= \sum_n \pi_\theta(n \mid x)\big(V_n^{\bar\theta} - V^{\bar\theta}\big) + \sum_n \pi_\theta(n \mid x)\big(V_n^\theta - V_n^{\bar\theta}\big) \\ &= \mathbb{E}_{p_{\bar\theta}}\big[\rho_n A_n^{\text{node}} + \rho_n\rho_c A_{n,c}^{\text{item}}\big]. \end{aligned} \tag{15}$$

The second equality follows from normalization and importance weighting, regardless of parameter sharing. Differentiating at $\theta = \bar\theta$ gives Eq. 9, since $\mathbb{E}_{q_{\bar\theta}}[A^{\text{item}} \mid n] = 0$. □

Proof of Corollary 1. For fixed (x,n) and coefficients a_c held constant during differentiation,

$$\sum_{c\in S_n} a_c \nabla_\theta \log \pi_\theta(n \mid x) = \left(\sum_{c\in S_n} a_c\right) \nabla_\theta \log \pi_\theta(n \mid x). \tag{16}$$

Item rewards thus change only the coefficient of the shared node-score direction; centered weights $a_c = q_{\bar\theta}(c \mid n,x)A_{n,c}^{\text{item}}$ give zero. Differentiating the item softmax in Eq. 4 with respect to h_{L-1} gives the stated hidden-state score, and the chain rule gives its backbone gradient. □

B IMPLEMENTATION DETAILS

Occupied-node sampling. For prefix s, let $\mathcal{A}(s)$ be the next tokens that extend to an occupied node in the fixed catalog trie. With decoder logits $z_{\theta,a}(x,s)$, the alignment policy uses

$$\pi_\theta(a \mid x,s) = \begin{cases} \dfrac{\exp\big(z_{\theta,a}(x,s)/\tau_{\text{roll}}\big)}{\sum_{b\in\mathcal{A}(s)} \exp\big(z_{\theta,b}(x,s)/\tau_{\text{roll}}\big)}, & a \in \mathcal{A}(s), \\ 0, & a \notin \mathcal{A}(s). \end{cases} \tag{17}$$

We draw M independent length-L paths with replacement at $\tau_{\text{roll}} = 1.0$, without top-k/top-p truncation, and retain repeated nodes. Full-node log-probabilities sum the masked token log-probabilities; current probabilities are recomputed on the recorded prefixes with the same masks and temperature. Node and bucket-normalized item behavior log-probabilities are cached and held fixed throughout the rollout's updates, as are the catalog, buckets, and trie. The ratios in Eq. 14 use these behavior probabilities and differentiable current probabilities, including their normalization.

Item representations and initialization. For RQ-Kmeans, e_c is the input to the first clustering stage (Deng et al., 2025); for RQ-VAE, it is the encoder latent before the first codeword assignment (Rajput et al., 2023). Both use continuous pre-quantization representations rather than selected codeword sums. We retain the tokenizer's preprocessing and normalization, add no further embedding normalization, and freeze e_c during alignment. The projection $W \in \mathbb{R}^{d \times d_h}$ uses Xavier uniform initialization with unit gain and no additional item bias. All controls retain the original node head and matched node-model initialization; DEA and the fixed-reference centered auxiliary use matched item-branch architectures, embedding extraction, and projection initialization.

Training configuration. Table 6 gives the industrial settings. Batch sizes count global query groups, and optimizer batch size includes gradient accumulation. Each rollout uses a fixed behavior snapshot for four optimizer steps before refreshing it, with items enumerated within sampled nodes. Training time and inference latency are reported in §5.5.

Table 6: Industrial training configuration.

Setting	Value
Rollout / optimizer batch size	$256/64$
Micro-batch size / accumulation steps	$16/4$
Epochs / optimizer steps per rollout	$1/4$
Total optimizer steps	5000
Behavior refresh interval	Every rollout batch
Sampled nodes per query M	128
Learning rate	1.12×10^{-6}
Clip range $\epsilon_n = \epsilon_c$ / normalizer floor ε	$0.1/10^{-6}$
Rollout / item temperature (τ_{roll}/τ)	$1.0/0.8$
Random seeds	3

C Experimental details

C.1 Industrial benchmark and online evaluation

The industrial generative retriever uses Qwen3-0.6B. All alignment methods start from the same pre-alignment node-model checkpoint for a given tokenizer. Table 7 summarizes the industrial data and both tokenizer configurations; the tokenizer and node-to-item serving protocol remain fixed during alignment.

The platform's query–title relevance service provides both alignment rewards $r(x, c) \in [0, 1]$ and evaluation labels. For each held-out query $x \in \mathcal{Q}$, we expand retrieved nodes into items and truncate the candidate list to an item budget $B = 1{,}500$. Let $I_B(x)$ be the retained list and $z(x, c) \in \{0, 1\}$ its item relevance labels. We report

$$\text{Rel} = \frac{1}{|\mathcal{Q}|} \sum_{x \in \mathcal{Q}} \frac{\sum_{c \in I_B(x)} z(x, c)}{|I_B(x)|}. \tag{18}$$

For HR@500, we expand retrieved nodes into their constituent items until the candidate list reaches 500 items, truncating it at that item budget. We then measure the hit rate against held-out interaction targets in this list. The cutoff counts expanded items, not nodes.

The 26.5% statistic is computed over query–node pairs from multi-item nodes among the pre-alignment retriever's top-500 beam-search nodes on held-out queries. A pair has mixed labels if

Table 7: Industrial data and tokenizer configuration.

Quantity	Value
Generative retriever	Qwen3-0.6B
Available alignment query pool	$\sim$6.5M
Held-out evaluation queries $\lvert\mathcal{Q}\rvert$ (next day)	tens of millions
Catalog items N_{item}	>300M
Occupied nodes N_{node}	$\approx$40M
Multi-item node ratio $N_{\text{node}}^{\text{multi}}/N_{\text{node}}$	$\approx$80%
Mean bucket size over multi-item nodes $\overline{K}_{\text{multi}}$	$\approx$9
RQ-Kmeans (L, codebook sizes)	2, 16384×8192
RQ-VAE (L, codebook sizes)	3, $1024 \times 512 \times 256$

its items include relevance scores both below 0.5 and at least 0.5; the denominator is the number of evaluated query–node pairs.

Online evaluation. DEA is evaluated in a one-week A/B test covering 5% of traffic in the platform's recommendation advertising system. Table 5 reports relative changes in relevance, clicks, and revenue against the highly optimized production GRPO-Agg baseline, which aggregates the proportions of relevant, partially relevant, and irrelevant items within each node with weights 1.0, 0.5, and 0, respectively.

C.2 KuaiRec benchmark

We retain $2{,}062$ videos observed by all $1{,}411$ users, with 412 videos used to construct contexts and $1{,}650$ forming the candidate catalog. The user split is $987/141/283$ for training/validation/test. Contexts combine frozen video-content embeddings with observed feedback. Both the alignment reward and the reported metric use $\text{clip}(\texttt{watch_ratio}, 0, 2)/2$, where $\texttt{watch_ratio} =$ play duration/video duration, averaged over repeated user–video records. Rewards for alignment come directly from training users' observations. RQ-Kmeans and RQ-VAE are trained using KuaiRec data and remain fixed during alignment; all compared methods use node-level inference. Results are averaged over three random seeds.

C.3 Baseline and ablation objectives

DPO preference pairs. For industrial search, we construct preference pairs using the online relevance model: items that pass its relevance filter serve as positive samples, while those that do not serve as negative samples. If the sampled positive and negative items map to the same node under the fixed tokenizer, we discard the pair and resample, so each retained pair contains distinct node IDs.

Node-level GRPO baselines. In the industrial setting, GRPO-Agg uses the weighted relevance rate

$$R_{\text{agg}}(x, n) = \frac{1.0\, N_{\text{rel}}(x, n) + 0.5\, N_{\text{partial}}(x, n)}{N_{\text{rel}}(x, n) + N_{\text{partial}}(x, n) + N_{\text{irrel}}(x, n)}, \tag{19}$$

where N_{rel}, N_{partial}, and N_{irrel} count relevant, partially relevant, and irrelevant items in S_n for query x, respectively. Irrelevant items contribute zero to the numerator but remain in the denominator. On KuaiRec, GRPO-Agg uses $R_{\text{agg}}(x, n) = |S_n|^{-1} \sum_{c \in S_n} r(x, c)$ with the watch-feedback reward defined above. GRPO-Max uses $R_{\max}(x, n) = \max_{c \in S_n} r(x, c)$. Both baselines use standard GRPO group-normalized advantages, rather than DEA's leave-one-out/common-scale estimator.

Fixed-reference centered auxiliary. With the same backbone, item branch, and additional parameter budget as DEA, this control minimizes the negative expected node-level surrogate $-\mathbb{E}_{n \sim \pi_{\tilde{\theta}}}[\ell_{\epsilon_n}(\rho_n, \widetilde{A}^{\text{node}})]$, valued under a fixed uniform q_{ref} on S_n, plus the unit-weight auxiliary

loss

$$-\mathbb{E}_{n\sim\pi_{\bar\theta},\, c\sim q_{\mathrm{ref}}}\big[\widetilde{A}^{\mathrm{aux}}_{n,c}\log q^{\mathrm{aux}}_{\theta}(c\mid n,x)\big],\qquad \widetilde{A}^{\mathrm{aux}}_{n,c}=\mathrm{sg}\left[\frac{r_{n,c}-\mathbb{E}_{c'\sim q_{\mathrm{ref}}}[r_{n,c'}]}{s}\right], \tag{20}$$

where s is DEA's common advantage scale. Centered signed weights cancel the softmax log-normalizer; this is a specific fixed-reference auxiliary surrogate, whose item factor is used only in the auxiliary term.

DEA without item-level refinement. This variant retains only $\ell_{\epsilon_n}(\rho_n, \widetilde{A}^{\mathrm{node}})$ and computes node values under a fixed uniform q_{ref} on S_n. Unlike standard node-level GRPO, it retains DEA's leave-one-out estimator and common-scale normalization.

Flat joint clipping. The flat control retains the joint policy and uses

$$\begin{aligned}\mathcal{L}_{\mathrm{flat}}(x) = -\frac{1}{M}\sum_{j=1}^{M}\sum_{c\in S_{n_j}} &\mathrm{sg}\big(q_{\bar\theta}(c\mid n_j,x)\big)\\ &\cdot \ell_{\epsilon}\Big(\rho_n^{(j)}\rho_c^{(j)},\ \widetilde{A}^{\mathrm{node}}_{j}+\widetilde{A}^{\mathrm{item}}_{j,c}\Big),\end{aligned} \tag{21}$$

with $\epsilon=\epsilon_n=\epsilon_c$ and the same fixed advantages, common scale, and behavior weights as DEA. Writing these weights as $\bar q_c$ and advantages as a_n, a_c, exact enumeration gives $\sum_c \bar q_c\rho_c = 1$, hence

$$\sum_c \bar q_c\rho_n\rho_c(a_n+a_c)=\rho_n a_n+\sum_c \bar q_c\rho_n\rho_c a_c. \tag{22}$$

The unclipped objectives and gradients are therefore identical for any θ; the comparison tests clipping organization rather than distinct unclipped objectives.