RETHINKING EXPLORATION IN GENERATIVE RECOMMENDATION: FROM MORE CANDIDATES TO BETTER CANDIDATE SETS

Anonymous authors
Paper under double-blind review

ABSTRACT

Autoregressive generative recommendation uses reinforcement learning (RL) to align sequence generation with recommendation utility. Candidate groups determine experiences for policy learning and choices at inference. However, multi-candidate generation faces a structural exploration–quality trade-off: expanding the search space does not necessarily improve the candidate group. This manifests in two failures. (1) *Candidate homogenization*: probability-driven decoding repeatedly expands shared high-probability prefixes, spending budget on similar trajectories while leaving alternative decisions underexplored. (2) *Candidate-quality degradation*: stochastic exploration can discover a few high-reward outcomes while introducing many low-quality candidates, increasing Reward Max but decreasing Reward Mean. Candidate count and maximum reward alone do not characterize effective exploration. These failures motivate learning complementary exploration while controlling quality costs. We propose **Q-ROLE** (**Quality-Gated Role Exploration**), a lightweight framework with two coupled components. *Role-conditioned generation* introduces persistent Anchor and Explorer identities within a shared policy, making task-focused exploitation and complementary exploration learnable responsibilities. A *quality-gated coverage reward* grants a bonus for divergence from the Anchor only when an Explorer meets an Anchor-relative base-reward threshold; otherwise, it receives only its base reward. The learned roles are retained during inference, carrying this division of responsibilities into candidate generation. Q-ROLE preserves the model backbone and requires no auxiliary prefix-value model. Extensive experiments on public recommendation benchmarks demonstrate that Q-ROLE improves a range of strong recommendation baselines, delivering higher recommendation accuracy under matched candidate budgets, while online experiments in an industrial generative reranking system report improved ranking performance.

1 INTRODUCTION

Generative recommendation represents items as Semantic IDs (SIDs) or directly generates recommendation slates, casting recommendation as conditional sequence generation (Rajput et al., 2023; Hou et al., 2025; Song et al., 2026). Reinforcement learning further aligns the generation policy with task utility, often through group-relative comparisons among candidates generated for the same request. Candidate construction therefore determines both the outputs available for selection and the experiences available for learning (Wang et al., 2026). Yet decoding commonly prioritizes model likelihood, which need not agree with task reward. **A larger candidate budget consequently need not provide more effective exploration.**

Figure 1 illustrates two limitations. **First, candidates can remain concentrated on shared prefixes.** Probability-driven pruning favors similar early decisions and can discard alternative paths before their final utility is known (Yu et al., 2026) (panel a). **Second, better individual candidates can conceal poorer candidate groups.** Stochastic exploration may discover a high-reward outcome while introducing many low-quality candidates (panel b). Our pilot experiments in Section 3 exhibit both prefix concentration and a divergence between Reward Max and Reward Mean under a fixed candidate budget. These observations motivate evaluating coverage together with candidate quality.

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.

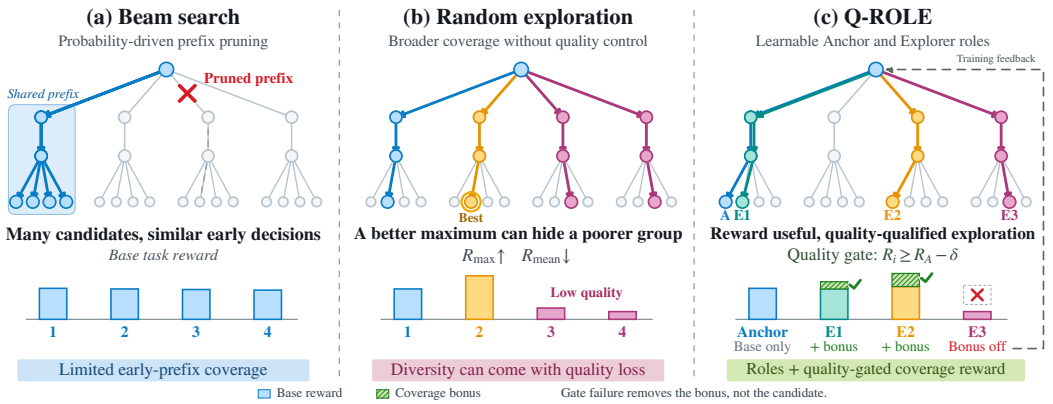


Figure 1: Comparison of exploration strategies. (a) Beam search concentrates candidates on shared prefixes. (b) Random exploration may increase Reward Max while reducing Reward Mean. (c) Q-ROLE assigns Anchor/Explorer roles and grants coverage bonuses only to quality-qualified Explorers.

Existing approaches address exploration through different mechanisms (Zhao et al., 2026). Increasing the search budget permits more candidates but may repeat high-probability paths. Stochastic sampling and diversity-aware decoding broaden coverage without explicitly conditioning diversity incentives on task quality (Holtzman et al., 2019; Vijayakumar et al., 2018). V-STAR uses prefix value and uncertainty to allocate search effort and tree-structured advantages to improve policy learning (Jiang et al., 2026a). We investigate a complementary question: **can the policy learn persistent responsibilities for reliable exploitation and quality-aware exploration?**

Our central insight is that **exploration should be assessed by the quality and complementarity of the candidate group, beyond its best outcome**. Under a limited budget, candidates can take different responsibilities: some focus on task utility, while others seek useful alternatives. Conventional group sampling provides no explicit role condition for learning this division. Introducing roles makes these responsibilities visible to the policy, while quality-dependent incentives distinguish useful exploration from arbitrary deviation.

We therefore propose **Q-ROLE**, a lightweight framework for **Quality-Gated Role Exploration** (Figure 1(c)). It combines two components. *Role-conditioned generation* assigns an Anchor and Explorers to a shared generator, enabling persistent specialization. The Anchor receives the base task reward; Explorers can additionally receive a *quality-gated coverage reward* for diverging from the Anchor while meeting a base-reward threshold. Thus, roles specify *who explores*, and the gate specifies *which exploration receives additional reinforcement*. The learned roles also condition inference-time decoding. Q-ROLE preserves the model backbone and underlying optimization framework, without requiring an additional prefix-value model.

Our contributions are threefold:

1. We diagnose prefix concentration and coverage–quality trade-offs, showing why candidate count and maximum reward alone do not characterize effective exploration.

2. We introduce role-conditioned generation and quality-gated coverage rewards, with consistent role conditioning during training and inference.

3. Extensive experiments on public benchmarks and an industrial system demonstrate consistent improvements in recommendation and ranking performance.

2 Related Work

2.1 Generative Recommendation

Generative recommendation models user preferences through conditional sequence generation (Geng et al., 2022; Liu et al., 2025; Lin et al., 2025). P5 adopts a unified text-to-text framework, while TIGER uses hierarchical Semantic IDs (SIDs) for autoregressive item retrieval (Rajput et al., 2023). Likelihood-driven decoding can concentrate candidates on shared prefixes. We study candidate quality and complementarity under limited budgets through coverage, mean reward, and maximum reward.

2.2 Decoding and Exploration

Standard sampling and diverse decoding broaden coverage without explicitly tying diversity incentives to task utility (Holtzman et al., 2019; Vijayakumar et al., 2018). Tree of Thoughts and V-STAR explore alternative branches; V-STAR further uses prefix values and uncertainty to allocate search effort, and tree-structured advantages for policy learning (Yao et al., 2023; Jiang et al., 2026b). Q-ROLE instead learns persistent Anchor/Explorer roles within a shared policy and rewards quality-qualified divergence without an auxiliary prefix-value model.

2.3 Reinforcement Learning for Generative Recommendation

Reinforcement learning aligns generation with task utility beyond likelihood (Cai et al., 2026; Damani et al., 2026). PPO stabilizes policy updates (Schulman et al., 2017), while GRPO uses group-relative advantages (Shao et al., 2024). For recommendation, UGR accounts for confidence and sample difficulty (Fan et al., 2026), HRPO decomposes item feedback into hierarchical residual token credits (Guo et al., 2026), and V-STAR's Sibling-GRPO computes sibling-relative advantages (Jiang et al., 2026b). Q-ROLE complements these reward-shaping and credit-assignment approaches (Fan et al., 2025) by learning complementary responsibilities within candidate groups. Its role conditioning relates to diverse skill learning such as DIAYN (Eysenbach et al., 2019), while an Anchor-relative gate controls exploration bonuses.

3 Motivation

Given context x, decoding produces a candidate group $\mathcal{C}_G(x) = (y^{(0)}, \ldots, y^{(G-1)})$. Let $R_i = R(x, y^{(i)})$ denote base task reward, excluding exploration bonuses. Reward Mean and Reward Max average $G^{-1}\sum_i R_i$ and $\max_i R_i$ over requests, respectively. Prefix coverage is $U_\ell(x) = |\{y^{(i)}_{1:\ell} : 0 \le i < G\}|$, where positions correspond to ad-ranking positions.

Figure 2 examines 431,778 industrial requests. Alongside standard beam decoding, we evaluate K1–K4: the first K positions use independent sampling, the remaining positions within the first four use deterministic selection, and subsequent completion is unchanged. Temperature is 1.5 and min-p is 0.05 throughout.

3.1 Insight 1: Candidate Count Does Not Imply Complementary Coverage

Standard beam decoding yields 11.49 distinct full sequences on average but only 1.33 distinct top-four prefixes; all 12 model candidates share one such prefix in 94.99% of requests (Figure 2(a)). Thus, distinct complete sequences can repeat nearly identical early decisions.

This redundancy persists among quality-qualified candidates. For this diagnostic, qualification requires $R_i \ge R_{\text{ref}}(x) - 10^{-4}$, where $R_{\text{ref}}(x)$ is the fixed standard-beam baseline's beam-0 reward. At K4, 9.88 qualified candidates cover only 3.25 distinct top-four prefixes on average; pooling across requests, 67.1% of qualified candidate slots repeat a prefix already represented within the same request (Figure 2(d)). Acceptable quality alone therefore does not ensure complementary coverage.

These observations motivate explicit generation responsibilities. Q-ROLE uses persistent Anchor and Explorer identities within a shared policy for task-focused exploitation and complementary ex-

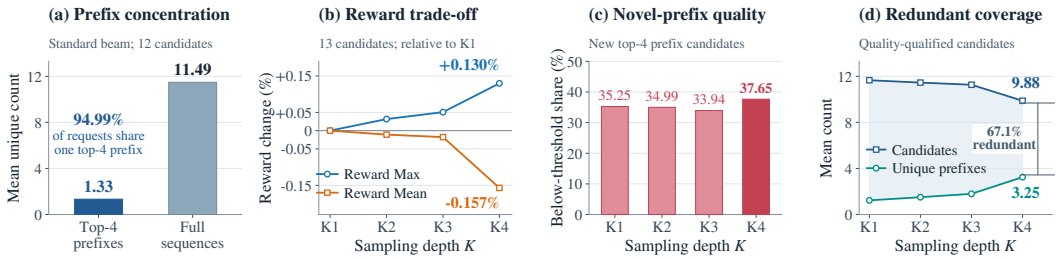


Figure 2: Pilot evidence for quality-aware exploration: (a) prefix concentration, (b) reward trade-off, (c) new-prefix quality, and (d) qualified-prefix redundancy. Panels (a,c,d) use 12 model candidates; (b) includes one additional fixed eCPM candidate.

ploration, respectively. Applying the same role conditioning during training and inference preserves this learned division of responsibilities.

3.2 Insight 2: Broader Coverage Can Reduce Average Quality

Broader exploration introduces a separate quality cost. With the candidate budget fixed, K4 increases predicted eCPM Reward Max by 0.130% relative to K1 but decreases Reward Mean by 0.157% (Figure 2(b)). A higher maximum reward can therefore coincide with lower average group quality.

A new-prefix candidate has a top-four prefix absent from the same request's standard-beam group of 12 model candidates. Across K1–K4, 33.94–37.65% of these candidates fall below the fixed qualification threshold, measured by pooling candidate counts across requests (Figure 2(c)). Thus, rewarding novelty without considering task reward can encourage low-quality exploration.

These observations motivate conditioning exploration incentives on task quality. Q-ROLE grants an Explorer a coverage bonus for divergence from the Anchor only when its base reward meets an Anchor-relative threshold; otherwise, it receives only its base reward. This ties the incentive for broader coverage to candidate quality.

4 Method

We propose **Q-ROLE** (**Quality-Gated Role Exploration**), which combines role-conditioned generation with quality-gated coverage rewards to learn high-quality, complementary exploration under a limited candidate budget. Each training iteration constructs role-conditioned candidates, computes exploration rewards, and updates the policy, as illustrated in the upper part of Figure 3. Inference retains the corresponding role conditioning and generation responsibilities, as illustrated in the lower part. Q-ROLE requires neither changes to the model backbone nor an additional prefix value model.

4.1 Role-Conditioned Candidate Generation

Given context x, we introduce a role identity z and incorporate its learnable embedding into the generation context:

$$\pi_\theta(y \mid x, z) = \prod_{\ell=1}^{L} \pi_\theta(y_\ell \mid x, y_{<\ell}, z), \tag{1}$$

where y is a length-L SID or recommendation sequence, and all roles share the same generator. For a candidate group $\mathcal{C}_G(x) = (y^{(0)}, \ldots, y^{(G-1)})$, we assign $z_0 = 0$ to the *Anchor* and assign *Explorer* roles to the remaining candidates.

The Anchor generates a high-probability reference trajectory, while Explorers sample other valid trajectories through controlled stochastic decoding. Role definitions remain consistent across requests and training iterations, and each candidate retains its role identity. The Anchor receives the base task reward, whereas Explorers additionally receive a quality-gated coverage bonus. These distinct incentives allow specialization to be learned jointly through the shared model and role embeddings.

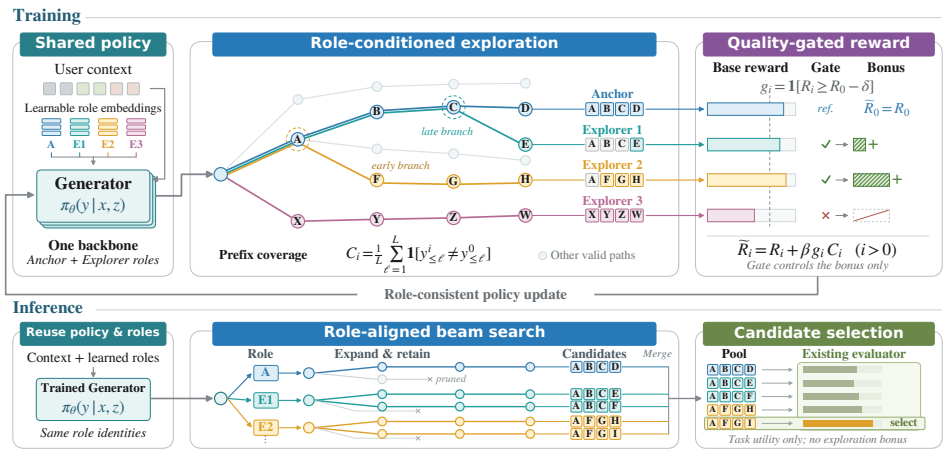


Figure 3: The framework of Q-ROLE. During training, learnable Anchor and Explorer roles condition a shared generator to produce complementary trajectories, while a quality gate applies prefix-coverage bonuses only to qualified Explorers. During inference, learned roles are retained in decoding, and role-aligned beam search pools diverse candidates for selection by the task evaluator.

4.2 Quality-Gated Coverage Reward

We reward complementary trajectories only when their base quality is acceptable. Relative to the Anchor, we define the prefix divergence and coverage score of Explorer i as

$$d_{i,\ell} = 1 - \prod_{t=1}^{\ell} \mathbf{1}[y_t^{(i)} = y_t^{(0)}], \qquad C_i = \frac{1}{L}\sum_{\ell=1}^{L} d_{i,\ell}. \tag{2}$$

Divergence is zero before the first branching point and remains one thereafter. Thus, $C_i \in [0, 1]$ measures trajectory differences relative to the Anchor and assigns greater credit to earlier divergence.

Let $R_i = R(x, y^{(i)})$ denote the base task reward, excluding the exploration bonus. We define the quality gate and training reward as

$$g_i = \mathbf{1}[R_i \ge R_0 - \delta], \qquad i > 0, \tag{3}$$

$$\widetilde{R}_i = \begin{cases} R_0, & i = 0, \\ R_i + \beta g_i C_i, & i > 0, \end{cases} \tag{4}$$

where $\delta \ge 0$ is the quality tolerance and $\beta \ge 0$ controls exploration strength. Candidates that fail the gate retain only their base reward, while eligible Explorers receive a coverage bonus bounded by β. Trajectory differences therefore determine the bonus magnitude, and base quality determines eligibility. In the upper-right panel of Figure 3, the qualifying Explorers receive coverage bonuses, whereas the low-quality Explorer's bonus is disabled while its base reward is retained.

4.3 Role-Consistent Policy Optimization

Q-ROLE preserves the underlying optimization form. Using sequence-level group-relative optimization as an example, the advantage is

$$A_i = \frac{\widetilde{R}_i - \mu_{\widetilde{R}}}{\sigma_{\widetilde{R}} + \epsilon}, \tag{5}$$

where $\mu_{\widetilde{R}}$ and $\sigma_{\widetilde{R}}$ are the within-group mean and standard deviation, and $\epsilon > 0$ ensures numerical stability. Rewards, gates, and advantages are detached during policy updates.

Let $h_{i,\ell} = (x, y^{(i)}_{<\ell}, z_i)$. Both the current and old policies retain the role conditioning used during generation, giving the token probability ratio

$$\rho_{i,\ell}(\theta) = \frac{\pi_\theta(y_\ell^{(i)} \mid h_{i,\ell})}{\pi_{\theta_{\text{old}}}(y_\ell^{(i)} \mid h_{i,\ell})}. \tag{6}$$

For candidate groups decoded from the old policy, we maximize the clipped surrogate objective

$$\mathcal{J}_{\text{policy}}(\theta) = \mathbb{E}_{x,\mathcal{C}_G}\left[\frac{1}{GL}\sum_{i=0}^{G-1}\sum_{\ell=1}^{L}\min\Big\{\rho_{i,\ell}A_i,\ \text{clip}(\rho_{i,\ell}, 1-\eta, 1+\eta)A_i\Big\}\right], \tag{7}$$

where η is the clipping threshold. Existing auxiliary losses and regularization terms remain unchanged. When reference-policy regularization is used, its probability evaluations retain the same role conditioning. Updates jointly optimize the shared model and role embeddings, and the resulting policy constructs the next candidate group. The feedback arrow beneath the training panels in Figure 3 depicts this role-consistent reward-to-policy update.

4.4 Training–Inference Alignment

Inference preserves the role identities and conditioning used during training, together with the Anchor–Explorer division of responsibilities. The lower part of Figure 3 illustrates a role-conditioned beam-search instance, where hypotheses retain their role identities through expansion and pruning. In the industrial sampling configuration described below, among G model-generated candidates, the Anchor uses greedy decoding, while the remaining candidates are sampled independently under their assigned Explorer roles.

For Explorer i, we apply temperature scaling over the valid output set $\mathcal{V}_{i,\ell}$:

$$q_{i,\ell}(v) \propto \pi_\theta(v \mid x, y^{(i)}_{<\ell}, z_i)^{1/T_i}, \qquad v \in \mathcal{V}_{i,\ell}. \tag{8}$$

We then apply min-p filtering:

$$\mathcal{S}_{i,\ell} = \left\{ v \in \mathcal{V}_{i,\ell} : q_{i,\ell}(v) \geq \alpha \max_{u \in \mathcal{V}_{i,\ell}} q_{i,\ell}(u) \right\}, \tag{9}$$

and sample after renormalization. Here, $T_i > 0$ controls stochasticity, and $\alpha \in [0, 1]$ removes relatively low-probability outputs. Larger candidate budgets retain the Anchor role and decoding rule while allocating additional generation opportunities to Explorers.

Consistent role conditioning carries the learned specialization into inference, while temperature and min-p control decoding stochasticity. In the industrial deployment, model-generated candidates and retained eCPM candidates are scored by the existing evaluator; the latter are counted separately in the total budget. This task-based scoring and selection stage corresponds to the lower-right panel of Figure 3. All reported metrics use the original task utility, excluding exploration bonuses.

5 Experiments

5.1 Experimental Setup

We evaluate UGR, HRPO, GFlowGR, GoalRank, and DEGR on Office Products and Beauty using chronological leave-one-out splits. Each request contains one held-out target and 999 negatives, and generation is restricted to this pool. Host–Q-ROLE pairs share initialization, data, evaluator, and training budget; slate hosts use adapted supervision and evaluators. Checkpoints and decoding settings are selected on validation data, and inference scoring does not access held-out labels.

We report HR@10 and NDCG@10 as percentages. The main UGR comparison uses 16 generated slots; Office diagnostics use $G = 12$. Duplicates consume slots without refilling. Given a fixed evaluation threshold $R_i \geq R_{\text{ref}}(x) - \delta_{\text{eval}}$, U_Q counts distinct qualified top-four prefixes. The reference and tolerance are shared across methods and separate from the training-time Anchor and gate. Reward Mean and Reward Max use base task rewards, excluding exploration bonuses. Explorer repetition is the fraction of the $G-1$ Explorer slots that duplicate a prefix beyond its first representative; Pool Hit is the percentage of requests whose target appears in the generated candidate pool.

Table 1: Shared-task performance (%): host → host + Q-ROLE. UGR uses 16 generated slots.

	Office Products		Beauty	
Host	HR@10	NDCG@10	HR@10	NDCG@10
UGR	$7.258 \rightarrow \mathbf{7.834}$	$3.868 \rightarrow \mathbf{4.132}$	$6.400 \rightarrow \mathbf{6.852}$	$3.500 \rightarrow \mathbf{3.713}$
HRPO	$6.600 \rightarrow \mathbf{7.037}$	$3.510 \rightarrow \mathbf{3.710}$	$5.800 \rightarrow \mathbf{6.158}$	$3.100 \rightarrow \mathbf{3.265}$
GFlowGR	$4.100 \rightarrow \mathbf{4.444}$	$2.280 \rightarrow \mathbf{2.445}$	$3.400 \rightarrow \mathbf{3.700}$	$1.984 \rightarrow \mathbf{2.135}$
GoalRank	$7.050 \rightarrow \mathbf{7.485}$	$3.720 \rightarrow \mathbf{3.918}$	$6.150 \rightarrow \mathbf{6.503}$	$3.250 \rightarrow \mathbf{3.411}$
DEGR	$6.800 \rightarrow \mathbf{7.280}$	$3.600 \rightarrow \mathbf{3.819}$	$5.950 \rightarrow \mathbf{6.344}$	$3.160 \rightarrow \mathbf{3.340}$

Table 2: Candidate quality and accuracy on Office ($G = 12$). Mean and Max are base task rewards.

Method	U_Q	Reward Mean	Reward Max	HR@10 (%)
Beam Search	1.33	0.755	0.812	7.180
Random Sampling	2.85	0.682	0.838	7.330
Diverse/Group Beam	2.42	0.743	0.825	7.425
APAO	2.95	0.764	0.832	7.610
V-STAR (+ VED)	3.65	0.760	0.838	7.680
Q-ROLE	**4.52**	**0.778**	**0.846**	**7.790**

5.2 Overall Recommendation Performance

Table 1 reports paired results across five hosts and both datasets. Q-ROLE improves both metrics in all ten host–dataset pairs, with relative HR@10 gains of 5.74%–8.82% under the shared task. A separate Office adaptation retaining HRPO's cohort supervision, prefix utilities, and token-level credit-to-go improves HR@10 from 6.720% to 7.215% (7.4% relative). This supports compatibility with HRPO's core optimization mechanisms under the adapted task, without reproducing its original KuaiRand/Kuai evaluation protocol.

5.3 Candidate Quality and Inference Cost

We compare candidate construction methods on Office Products using UGR and $G = 12$. Random Sampling uses temperature 1.5 and min-p 0.05. Table 2 reports qualified coverage, base reward, and recommendation accuracy; V-STAR uses Value-Guided Efficient Decoding (VED) in this comparison.

Random Sampling improves Reward Max over Beam Search while lowering Reward Mean. Q-ROLE improves qualified coverage, mean reward, and HR@10 over V-STAR with VED, illustrating the value of assessing candidate groups beyond their best outcome.

Inference cost. Table 3 compares both V-STAR decoding configurations with Q-ROLE on one NVIDIA A100 at batch size 64. Q-ROLE matches standard-beam V-STAR's 68 ms latency with higher HR@10. Relative to VED, it reduces decoding tokens by 14.3% and latency from 145 to 68 ms, without auxiliary prefix-value queries. Query counts exclude the final candidate selector.

5.4 Mechanism Analysis

Components and reward allocation. Table 4 uses a shared evaluation reference and $G = 12$. Base is the no-role/no-bonus reference for component ablations. Beam Search in Table 2 is the reference for candidate construction. Roles Only and Gated Bonus Only each improve over Base. Ungated bonuses increase U_Q from 4.52 to 4.78 but lower Reward Mean from 0.778 to 0.710 and HR@10 from 7.79% to 7.45%.

Full uses $\beta = 0.005$ and $\delta = 0.01$. Random Gating activates bonuses independently of quality at a target rate of 72%; the low-β control rewards every Explorer. Neither guarantees equal realized bonus magnitude. The Matched Random Gate additionally scales random bonuses to match

Table 3: Inference cost on Office ($G = 12$); one A100, batch size 64.

Method	Decoding tokens	Value calls/req.	Latency (ms)	HR@10 (%)
V-STAR (Standard Beam)	144 ± 0	0	68	7.450
V-STAR (+ VED)	168 ± 12	12	145	7.680
Q-ROLE	144 ± 0	0	68	**7.790**

Table 4: Component and reward-allocation ablations on Office ($G = 12$). Reward Mean excludes exploration bonuses.

Variant	U_Q	Reward Mean	HR@10 (%)
Base: no roles, no bonus	2.15	0.751	7.35 ± 0.05
Roles Only	3.42	0.762	7.52 ± 0.06
Gated Bonus Only	3.85	0.768	7.61 ± 0.05
Roles + Ungated Bonus	**4.78**	0.710	7.45 ± 0.07
Random Gating (72%)	4.18	0.742	7.510
Ungated, $\beta = 0.0035$	4.30	0.735	7.485
Matched Random Gate	4.45	0.738	7.565
Full Q-ROLE	4.52	**0.778**	$\mathbf{7.79} \pm 0.04$

Full's empirical mean bonus. Full retains higher Reward Mean and HR@10 than all three controls, supporting quality-dependent reward allocation beyond the mean incentive magnitude.

Training-time Explorer identities. We train models with shared or distinct Explorer identities, retaining a separate Anchor and identical gating, sampling, and training budgets. Table 5 shows that distinct identities reduce repetition from 32.4% to 11.2% and improve Pool Hit and HR@10, supporting their benefit for complementary exploration in this setting.

Inference-time role interventions. We freeze the Full checkpoint and change only slot role conditions, retaining slot-specific decoding rules; the repeated-ID condition averages across trained Explorer IDs. Table 6(a) shows higher Pool Hit and HR@10 with distinct IDs. The Pool Hit–HR@10 gap grows from 2.128 to 2.645 percentage points: the gain is consistent with reaching more targets, without establishing a smaller selection gap.

Sensitivity to quality tolerance. Table 6(b) varies the training tolerance at fixed $\beta = 0.005$ and $G = 12$, with separate training runs and a fixed evaluation-quality reference. Pass denotes the fraction of Explorers eligible for bonuses. The tested sweep favors intermediate tolerances; increasing δ to 0.05 raises acceptance to 98% but lowers HR@10 to 7.430%.

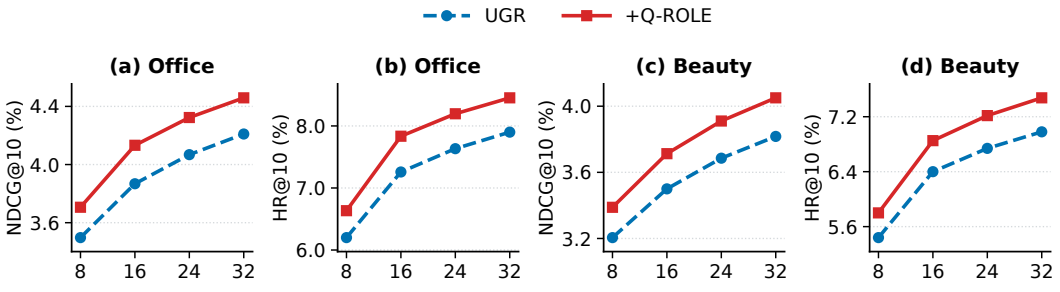


Figure 4: Candidate-budget scaling on Office Products and Beauty; horizontal axes show slots.

Table 5: Training-time Explorer identities on Office ($G = 12$). Metrics except U_Q are percentages.

Configuration	U_Q	Repetition	Pool Hit	HR@10	NDCG@10
Shared Explorer ID	3.85	32.4	9.950	7.625	4.020
Distinct Explorer IDs	**4.52**	**11.2**	**10.435**	**7.790**	**4.110**

Table 6: Additional mechanism analyses on Office ($G = 12$): (a) role interventions with a frozen checkpoint; (b) separately trained gate settings. All metrics except δ are percentages.

(a) Inference-time roles

Role IDs	Repetition	Pool Hit	HR@10
Anchor only	71.2	9.105	7.288
Repeated Explorer	46.8	9.680	7.552
Distinct Explorers	**11.2**	**10.435**	**7.790**

(b) Quality-gate tolerance

δ	Pass	HR@10	NDCG@10
0.000	35	7.549	4.041
0.005	58	7.766	4.100
0.010	72	**7.790**	**4.110**
0.050	98	7.430	3.930

5.5 Candidate-Budget Scaling

Figure 4 compares HR@10 and NDCG@10 over 8–32 candidate slots. Q-ROLE performs better across the displayed budgets; its 16-slot HR@10 nearly matches the 32-slot host on both datasets. Because duplicates are not refilled, eight slots permit at most eight unique items. These curves measure candidate-slot efficiency.

Table 7: Online A/B results: relative changes (%) against the production baseline.

ADV	Model-ADV	Ad Impressions	Avg. Advertiser Bid
+1.35	**+1.68**	+0.18	+0.29

5.6 Online Evaluation

We report a six-day online A/B evaluation allocating 4% of request traffic to Q-ROLE against the production generative recommender. ADV measures realized advertising value, while Model-ADV measures model-evaluated advertising value. Table 7 reports positive relative changes of 1.35% and 1.68% in these metrics, accompanied by smaller changes in impressions and advertiser bids. These are aggregate business-outcome point estimates.

6 Conclusion

We propose Q-ROLE to rethink exploration in generative recommendation, shifting the focus from merely generating *more* candidates to constructing *better* candidate sets. By coupling learnable Anchor/Explorer roles with quality-gated coverage rewards, Q-ROLE effectively mitigates both candidate homogenization and quality degradation. Offline benchmarks and industrial online deployments validate its robust performance under fixed candidate budgets. Ultimately, we hope this work inspires the community to evaluate generative exploration beyond simple candidate counts or maximum rewards, prioritizing genuine set complementarity and practical task utility.

AI Use Statement

Artificial intelligence tools were utilized solely for the purpose of language polishing, grammatical correction, and enhancing the overall readability and stylistic fluency of the manuscript. No AI tools were involved in the generation of core research ideas, mathematical derivation, experimental design or empirical analysis. The conception, execution, validation, and intellectual content of this research were independently conducted and verified by the human authors.

REFERENCES

Shihao Cai, Chongming Gao, Haoyan Liu, Wentao Shi, Jianshan Sun, Ruiming Tang, and Fuli Feng. MGFRec: Towards reinforced reasoning recommendation with multiple groundings and feedback. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.1*, pp. 49–58. Association for Computing Machinery, 2026. doi: 10.1145/3770854.3780206.

Mehul Damani, Isha Puri, Stewart Slocum, Idan Shenfeld, Leshem Choshen, Yoon Kim, and Jacob Andreas. Beyond binary rewards: Training LMs to reason about their uncertainty. In *The Fourteenth International Conference on Learning Representations*, 2026.

Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is all you need: Learning skills without a reward function. In *International Conference on Learning Representations*, 2019.

Chenxiao Fan, Chongming Gao, Wentao Shi, Yaxin Gong, Zihao Zhao, and Fuli Feng. Fine-grained list-wise alignment for generative medication recommendation. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Chenxiao Fan, Chongming Gao, Yaxin Gong, Haoyan Liu, Fuli Feng, and Xiangnan He. Uncertainty-aware generative recommendation. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*. Association for Computing Machinery, 2026. doi: 10.1145/3770855.3817975.

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

Kaifeng Guo, Yiming Yang, Jingtong Gao, Guolei Zeng, Fukang Yang, Yukang Liang, Peng Jiang, Qingpeng Cai, and Xiangyu Zhao. Hierarchical residual policy optimization for generative recommendations. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*. Association for Computing Machinery, 2026. doi: 10.1145/3770855.3818206.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*, 2019.

Yupeng Hou, Jianmo Ni, Zhankui He, Noveen Sachdeva, Wang-Cheng Kang, Ed H. Chi, Julian McAuley, and Derek Zhiyuan Cheng. ActionPiece: Contextually tokenizing action sequences for generative recommendation. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 24004–24024. PMLR, 2025.

Jie Jiang, Yangru Huang, Zeyu Wang, Changping Wang, Yuling Xiong, Jun Zhang, and Huan Yu. Spend search where it pays: Value-guided structured sampling and optimization for generative recommendation. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 2098–2109, 2026a.

Jie Jiang, Yangru Huang, Zeyu Wang, Changping Wang, Yuling Xiong, Jun Zhang, and Huan Yu. Spend search where it pays: Value-guided structured sampling and optimization for generative recommendation. *arXiv preprint arXiv:2602.10699*, 2026b.

Xinyu Lin, Chaoqun Yang, Wenjie Wang, Yongqi Li, Cunxiao Du, Fuli Feng, See-Kiong Ng, and Tat-Seng Chua. Efficient inference for large language model-based generative recommendation. In *The Thirteenth International Conference on Learning Representations*, 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. Association for Computing Machinery, 2025.

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, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Chaotian Song, Jingyao Zhang, Chenghao Chen, Zisen Sang, Dehai Zhao, Guodong Cao, Boxi Wu, Deng Cai, and Jia Jia. DeGRe: Dense-supervised generative reranking for recommendation. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*. Association for Computing Machinery, 2026. doi: 10.1145/3770855.3818363.

Ashwin Vijayakumar, Michael Cogswell, Ramprasaath Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. Diverse beam search for improved description of complex scenes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Yejing Wang, Shengyu Zhou, Jinyu Lu, Qidong Liu, Xinhang Li, Wenlin Zhang, Feng Li, Pengjie Wang, Chuan Yu, Jian Xu, Bo Zheng, and Xiangyu Zhao. GFlowGR: Fine-tuning generative recommendation frameworks with generative flow networks. In *Proceedings of the 49th International ACM SIGIR Conference on Research and Development in Information Retrieval*. Association for Computing Machinery, 2026. doi: 10.1145/3805712.3809663.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822, 2023.

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*. Association for Computing Machinery, 2026. doi: 10.1145/3770855.3818052.

Binglei Zhao, Xuanhua Yang, Xiwei Zhao, and Sulong Xu. DEGR: Dual exploration-driven generative re-ranking for adaptive cross-request context bridging. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*. Association for Computing Machinery, 2026. doi: 10.1145/3770855.3818362.

A Data and Task Definition

This section specifies the datasets, evaluation task, candidate accounting, and quality thresholds used in the main experiments.

Table 8: Public-benchmark dataset statistics and evaluation-pool sizes.

Dataset	Users	Items	Interactions	Negatives/target
Office Products	101,202	27,102	732,323	999
Beauty	352,341	57,281	2,123,492	999

Chronological splits and evaluation pools. For each user, the final interaction is held out for testing, the second-to-last interaction is used for validation, and earlier interactions form the training sequence. Each evaluation request contains one held-out target and 999 negative items. Generation is restricted to this pool. Checkpoints and decoding settings are selected using validation data; the final inference scorer does not access the held-out target label.

Output representation and candidate budgets. For item generation, a candidate is an item represented by a Semantic ID (SID); for slate generation, it is a recommendation sequence. The candidate budget G counts generated slots, including duplicate outputs. Duplicates are not replaced by additional generations. The main UGR comparison uses 16 slots, the Office mechanism diagnostics use $G = 12$, and the budget analysis considers 8, 16, 24, and 32 slots. Candidate count, prefix depth, and the number of learned role identities describe different aspects of the configuration.

Recommendation metrics. Let r_x be the rank of the first occurrence of the held-out target in the final recommendation list for request x, with $r_x = \infty$ when the target is absent. With one relevant target per request, the reported percentage metrics are

$$\mathrm{HR@}k = \frac{100}{N} \sum_{x=1}^{N} \mathbf{1}[r_x \le k], \tag{10}$$

$$\mathrm{NDCG@}k = \frac{100}{N} \sum_{x=1}^{N} \frac{\mathbf{1}[r_x \le k]}{\log_2(r_x + 1)}, \tag{11}$$

where an absent target contributes zero. Pool Hit is the percentage of requests for which the generated candidate pool contains the target, before final ranking. It measures target reachability and is distinct from HR@10 after selection.

Fixed evaluation threshold. Write $R_i = R(x, y^{(i)})$ for the base task reward, without the exploration bonus. For comparisons of qualified coverage, we use

$$R_i \ge R_{\mathrm{ref}}(x) - \delta_{\mathrm{eval}}, \qquad \delta_{\mathrm{eval}} = 10^{-4}, \tag{12}$$

where $R_{\mathrm{ref}}(x)$ is the beam-0 reward of the frozen base-model reference. The reference model and tolerance are shared across the methods in a comparison. This evaluation reference is separate from the Anchor generated by the current policy during training.

Prefix coverage and repetition. Following the main text, let $p_4(y)$ denote the first four positions of the generated output. Positions refer to SID tokens for item generation and item positions for industrial slate generation. Prepended role tokens are conditioning inputs and are excluded from the output-prefix statistics. Qualified coverage for request x is

$$U_Q(x) = \left| \left\{ p_4(y^{(i)}) : R_i \ge R_{\mathrm{ref}}(x) - \delta_{\mathrm{eval}} \right\} \right|. \tag{13}$$

Thus, a prefix is counted once if at least one candidate with that prefix qualifies. The reported U_Q averages this count over requests. Prefix depth four is a diagnostic choice, not a specification of the complete output length.

For the $G-1$ Explorer slots, let $U_E(x)$ be their number of distinct prefixes. Explorer repetition is the request-averaged percentage $100\,[1 - U_E(x)/(G-1)]$: every duplicate slot beyond one representative of a prefix contributes to repetition. Reward Mean and Reward Max are computed within each candidate group from base rewards and then averaged over requests. The industrial pilot separately specifies when its fixed eCPM candidate is included in a reward statistic; it is not silently included in the model-only prefix counts.

B Implementation and Baseline Adaptation

Role conditioning. Learnable role embeddings are prepended to the decoder input and optimized jointly with the shared generator. The default configuration distinguishes an Anchor identity from Explorer identities. The distinct-ID configuration conditions Explorer slots on different learned identities; the shared-ID ablation uses one Explorer identity across those slots. Current-policy, old-policy, and, when applicable, reference-policy probability evaluations retain the role conditioning used to generate each candidate. Increasing the inference budget allocates more generation opportunities to Explorers while retaining the learned role conditions and Anchor decoding rule.

Reward and policy update. For a length-L output, the coverage score compares each Explorer with the Anchor at every prefix depth:

$$C_i = \frac{1}{L} \sum_{\ell=1}^{L} \mathbf{1}\left[y_{1:\ell}^{(i)} \neq y_{1:\ell}^{(0)} \right]. \tag{14}$$

The training gate and shaped rewards are

$$g_i = \mathbf{1}[R_i \geq R_0 - \delta], \quad i > 0, \tag{15}$$

$$\widetilde{R}_0 = R_0, \qquad \widetilde{R}_i = R_i + \beta g_i C_i, \quad i > 0. \tag{16}$$

The Anchor reward R_0 depends on the trajectory generated for the current training context. An Explorer that fails the gate retains its base reward. The gate controls the additional reward; it does not remove the candidate. A passing Explorer can still have base reward below the Anchor by up to δ, so passing is an Anchor-relative condition rather than an absolute guarantee of task quality.

In the sequence-level group-relative optimization example, shaped rewards are normalized within the candidate group to produce an advantage that is applied through token probability ratios. This shaped-reward definition is separate from token-level credit assignment. Existing host-specific optimization, credit assignment, auxiliary losses, and regularization are retained. Rewards, gates, and advantages are detached during policy updates.

Table 9: Reward settings and decoding configurations. Sampling settings are distinguished from role-conditioned beam search.

Category	Parameter	Setting
Reward	Coverage scale β	0.005
	Training tolerance δ	0.01
	Evaluation tolerance δ_{eval}	10^{-4}; fixed reference
Optimization	Policy update	Host optimizer with consistent role conditioning
Beam search	Role handling	Retain each hypothesis's role through expansion and pruning
Sampling	Anchor	Greedy decoding
	Explorer temperature T_i	1.5
	Explorer min-p threshold α	0.05

Decoding and final selection. Role-conditioned beam search and industrial sampling are separate decoding configurations. Beam-search hypotheses retain their role identities during expansion and pruning. In the industrial sampling configuration, the Anchor is decoded greedily and Explorers

are sampled independently under their assigned identities. Temperature and min-p control sampling stochasticity; they do not imply that all beam-search comparisons use stochastic sampling. Final selection uses the task evaluator without an exploration bonus. Retained industrial eCPM candidates are accounted for separately from model-generated slots.

Host adaptation and comparison scope. UGR, HRPO, GFlowGR, GoalRank, and DEGR are evaluated under the shared task described in Appendix A. Each host–Q-ROLE pair shares initialization, data, evaluator, and training budget. Slate hosts use adapted supervision and evaluators. These comparisons evaluate the incremental effect of Q-ROLE under that formulation, rather than claiming a reproduction of every host's original benchmark protocol.

The additional Office HRPO adaptation retains cohort-based supervision, prefix utilities, and token-level credit-to-go. Its HR@10 changes from 6.720% to 7.215% (7.4% relative), supporting compatibility with these optimization mechanisms. This adaptation does not reproduce the original KuaiRand/Kuai task. Beam Search in Table 2 is the reference for candidate construction; Base in Table 4 is the reference for removing role conditioning and the coverage bonus.

C Complete Results and Statistical Reporting

Table 10 provides the three-seed summary for the main UGR comparison on Office Products at $G = 16$. HR@10 and NDCG@10 are percentages and are reported as mean $\pm$ standard deviation across three random seeds. The final column reports the paired HR@10 improvement and its 95% confidence interval, both in percentage points (pp).

Table 10: Three-seed Office results ($G = 16$). Metric columns show mean $\pm$ SD; the paired gain is in percentage points with a 95% CI.

Model	HR@10 (%)	NDCG@10 (%)	Paired HR gain [95% CI]
UGR	7.258 ± 0.035	3.868 ± 0.021	—
UGR + Q-ROLE	$\mathbf{7.834} \pm 0.041$	$\mathbf{4.132} \pm 0.028$	$+0.576$ [0.528, 0.624]

For paired seed-level results h_s^{Q} and h_s^{host}, the gain is the mean of $d_s = h_s^{\mathrm{Q}} - h_s^{\mathrm{host}}$. The interval in Table 10 concerns this paired mean difference; it is not the standard deviation of either method. Its center, $7.834 - 7.258 = 0.576$ pp, is distinct from a relative percentage gain. This three-seed reporting convention applies to the Office $G = 16$ comparison in this table and is not a definition of the uncertainty notation in every other experiment. Online results are reported separately as aggregate relative-change point estimates.

D Role and Reward Mechanism Analysis

This section distinguishes training-time identity configuration, inference-time role interventions, and reward-allocation controls. These comparisons answer different questions and are not interchangeable.

Training-time identities. Table 11 expands the training-configuration comparison in Table 5. Both variants retain a separate Anchor identity and use identical gating, sampling, and training budgets. The shared-ID variant conditions every Explorer slot on the same learned Explorer embedding, whereas the distinct-ID variant uses separate identities.

Table 11: Training-time Explorer identities on Office ($G = 12$). All metrics except U_Q are percentages.

Configuration	U_Q	Repetition	Pool Hit	HR@10	NDCG@10
Shared Explorer ID	3.85	32.4	9.950	7.625	4.020
Distinct Explorer IDs	**4.52**	**11.2**	**10.435**	**7.790**	**4.110**

Distinct identities reduce repetition by 21.2 percentage points and improve Pool Hit and HR@10. This supports the benefit of multiple persistent identities in this setting. A shared identity remains a functioning alternative, so the comparison does not establish that distinct identities are necessary for every form of exploration.

Inference-time interventions. Table 6(a) instead freezes the Full checkpoint and changes only the role conditions supplied to the slots, retaining their decoding rules. Repeated-ID results are averaged across trained Explorer identities. Anchor-only conditioning in that table is an inference intervention on the trained model, not an independently trained no-role baseline. In particular, a model trained with a shared Explorer identity is different from repeatedly supplying one identity to a model trained with distinct identities.

Distinct inference identities increase Pool Hit from 9.680% to 10.435% and HR@10 from 7.552% to 7.790% relative to repeated identities. The Pool Hit–HR@10 gap increases from 2.128 to 2.645 pp. These results are consistent with reaching more targets and converting some additional targets into final hits; they do not show a smaller selection gap.

Reward allocation and average bonus magnitude. Table 12 collects the random-allocation controls from Table 4. Full uses $\beta = 0.005$ and $\delta = 0.01$. Random Gating activates bonuses independently of task quality at a target rate of 72%; the low-β variant activates its bonus for every Explorer. Neither control necessarily matches the mean realized bonus because $\mathbb{E}[\beta g_i C_i]$ depends on the joint distribution of qualification and coverage. The Matched Random Gate dynamically scales randomly allocated bonuses to match Full's empirical mean exploration bonus.

Table 12: Reward-allocation controls on Office ($G = 12$). Reward Mean excludes bonuses; HR@10 is a percentage.

Gating strategy	U_Q	Reward Mean	HR@10 (%)
Random Gating (72%)	4.18	0.742	7.510
Ungated, $\beta = 0.0035$	4.30	0.735	7.485
Matched Random Gate	4.45	0.738	7.565
Quality Gate (Q-ROLE)	**4.52**	**0.778**	**7.790**

After matching the mean bonus, Full retains higher Reward Mean (0.778 versus 0.738) and HR@10 (7.790% versus 7.565%) at similar qualified coverage. This supports quality-dependent reward allocation beyond the mean incentive magnitude. Matching a mean does not equate every aspect of the reward distribution or establish that all performance differences have a single cause.

Training tolerance versus evaluation qualification. Table 6(b) varies the training-time δ at fixed $\beta = 0.005$ and $G = 12$, with a separate training variant for each setting. The evaluation reference and $\delta_{\text{eval}} = 10^{-4}$ remain fixed. Gate pass rate counts training-bonus eligibility, whereas U_Q counts prefixes passing the fixed evaluation criterion. Consequently, training-gate acceptance and evaluation-qualified coverage need not move together. In the tested sweep, $\delta = 0.005$ performs close to 0.01, while 0.05 raises acceptance to 98% but lowers HR@10 to 7.430%.

E Computation Overhead and Budget Scaling

Table 13 decomposes the latency measurements associated with Table 3. Measurements are reported in milliseconds per request on one NVIDIA A100 at batch size 64. Generation, auxiliary prefix-value queries, and final task evaluation are listed separately. Total latency includes all three stages; excluding the final selector from the *value-call count* does not exclude its time from the total latency.

The totals satisfy $42 + 0 + 26 = 68$ and $57 + 62 + 26 = 145$. In this configuration, Q-ROLE and standard-beam V-STAR have the same measured generation and total latency. VED improves the V-STAR HR@10 point estimate from 7.450% to 7.680% with additional decoding and value evaluation. Q-ROLE attains 7.790% HR@10 at the measured 68 ms total latency. These comparisons characterize the tested configurations rather than a universal minimum inference cost.

Table 13: Inference latency breakdown on Office ($G = 12$), in ms/request at batch size 64 on one A100.

Method	Generation	Value queries	Task evaluator	Total
V-STAR (Standard Beam)	42	0	26	68
V-STAR (+ VED)	57	62	26	145
Q-ROLE	42	0	26	68

Table 14: Reported online configuration and metric conventions. Detailed outcome-based definitions appear in Section G.

Parameter	Description
Evaluation window	Six days.
Q-ROLE traffic	4% of live request traffic.
Request scale	Millions of requests per day per bucket.
ADV	Posterior advertising value, using attributed outcomes and target values; the tCPA example is $\sum_i v_i t_i$.
Model-ADV	Attributed outcomes weighted by the corresponding bidding expressions; the tCPA example is $\sum_i v_i t_i r_i$.

Candidate-slot efficiency. Figure 4 reports HR@10 and NDCG@10 at budgets of 8, 16, 24, and 32 generated slots. All slots count toward the budget, including duplicates; eight slots therefore permit at most eight distinct items. Q-ROLE's 16-slot HR@10 nearly matches the 32-slot host on both datasets. This is a comparison of candidate-slot efficiency. It does not imply that runtime scales linearly with the number of slots or that doubling the slot budget doubles latency. The reported decoding-token counts measure a different quantity from either the number of candidates or the prefix depth used for coverage analysis.

F Industrial Experiment Setup

The online evaluation in Section 5.6 was conducted in an industrial generative reranking system over six days, with 4% of request traffic allocated to Q-ROLE against the production baseline. Table 14 summarizes the reported configuration. The relative changes in Table 7 are aggregate point estimates.

G Industrial Advertising-Value Metrics

We distinguish the predicted score used for ranking from the outcome-based values used for evaluation. Both Model-ADV and Posterior ADV use realized, attributed outcomes. Model-ADV retains the strategy-specific bidding coefficients, whereas Posterior ADV uses the corresponding target values. We abbreviate Posterior ADV as ADV in the online results.

Units and ranking score. Let i index served ad impressions in the evaluation set $\mathcal{I}$. Let c_i denote the attributed click indicator, v_i the attributed conversion indicator, and w_i the attributed realized gross merchandise volume (GMV). Write $p_i = pCTR_i$ and $q_i = pCVR_i$, where q_i is the predicted conversion rate conditional on a click. Let b_i denote the effective per-click bid. The expected value per thousand impressions is

$$\mathrm{eCPM}_i = 1000\, p_i b_i. \tag{17}$$

The common factor 1000 can be omitted when describing ranking order. Thus, the bid expressions in Table 15 must be multiplied by p_i to obtain expected value per impression.

For tCPA bidding, $t_i = tCPA_i$ has units of currency per conversion and $r_i = bid_ratio_i$ is dimensionless. For the pCPA-based rule, $a_i = pCPA_i$ likewise denotes a conversion-price parameter in currency per conversion. For ROI objectives, $\rho_i = tROI_i$ is dimensionless and $\widehat{w}_i = pGMV_i$ is predicted GMV *per conversion*. With this definition, $q_i \widehat{w}_i$ is predicted GMV per click. CPC bids d_i

Table 15: Linear, unclipped bidding expressions and outcome-based Model-ADV contributions. Each m_i has monetary units. The normalized base-bid row is the algebraic form obtained under the same outcome substitution rule.

Strategy	Per-click bid b_i	Model-ADV contribution m_i
Target CPA	$q_i t_i r_i$	$v_i t_i r_i$
Target ROI	$q_i \widehat{w}_i r_i / \rho_i$	$w_i r_i / \rho_i$
Whole-Site ROI	$q_i \widehat{w}_i r_i / \rho_i$	$w_i r_i / \rho_i$
pCPA-based	$q_i a_i r_i$	$v_i a_i r_i$
CPC	d_i	$c_i d_i$
Normalized base bid	$q_i d_i^0 / q_i^0$	$v_i d_i^0 / q_i^0$

and base bids d_i^0 have units of currency per click; $q_i^0 > 0$ denotes the reference conversion rate used by the normalized base-bid rule.

Outcome-based evaluation. For the linear tCPA rule, the per-click bid and its two evaluation contributions are

$$b_i = q_i t_i r_i, \tag{18}$$

$$m_i = v_i t_i r_i, \tag{19}$$

$$u_i^{\mathrm{ADV}} = v_i t_i. \tag{20}$$

Consequently, the tCPA totals are

$$\text{Model-ADV} = \sum_{i \in \mathcal{I}} v_i t_i r_i, \tag{21}$$

$$\text{ADV} = \sum_{i \in \mathcal{I}} v_i t_i. \tag{22}$$

The observed conversion outcome replaces the predicted click–conversion outcome in the ranking expression. The bidding adjustment associated with each impression remains a weight in Model-ADV. These quantities have monetary units and represent outcome-weighted values; they need not equal auction payments.

Table 15 gives the corresponding Model-ADV contribution m_i for each listed linear bidding rule. Its aggregate form is $\text{Model-ADV} = \sum_{i \in \mathcal{I}} m_i$. The two ROI rows share the same algebra, with predicted and realized GMV evaluated under the attribution scope of the respective objective: ad-attributed value for Target ROI and whole-site-attributed value for Whole-Site ROI.

For the normalized base-bid rule, substituting the realized conversion outcome for the predicted click–conversion outcome gives $m_i = v_i d_i^0 / q_i^0$. This row specifies the algebraic extension of the metric convention; it is not a separate online validation result. For the tCPA expressions above, the bidding-weight function is explicitly $f(t_i, r_i) = t_i r_i$. The displayed formulas describe linear, unclipped rules. Caps or additional nonlinear constraints require their own explicit transformations and are outside these displayed formulas.

Aggregation and interpretation. The formulas define per-impression contributions and their totals over $\mathcal{I}$. Comparisons between traffic groups require a common exposure-normalization basis; the sum notation does not assume equal group sizes or specify a particular normalization denominator. The online lifts in Table 7 retain their interpretation as reported aggregate point estimates. Using the bidding coefficients associated with served impressions defines an evaluation convention. The weighting rule alone does not isolate a causal effect of the generator or remove interactions with bidding and allocation; causal interpretation depends on the experiment design.

H Illustrative Scenarios and Potential Failure Modes

This section uses conceptual examples to explain Q-ROLE's reward mechanism and its limitations. The examples are hypothetical rather than logged evaluation episodes. In Table 16, each three-product list denotes an illustrative recommendation slate; its products are not the three tokens of an

item's Semantic ID (SID). Gate conditions describe training-time bonus eligibility. At inference, the task evaluator scores candidates using task utility without the exploration bonus. Failing the training gate removes the bonus, not the candidate.

Useful alternatives. An Explorer receives a coverage bonus when it diverges from the Anchor and its base reward satisfies $R_i \geq R_0 - \delta$. Earlier prefix divergence receives more coverage credit. This incentive can reinforce alternatives with acceptable Anchor-relative reward, making useful candidates available for subsequent selection. Prefix divergence alone, however, does not establish a different semantic intent or guarantee a correct recommendation. Moreover, the threshold is relative to the Anchor and allows a tolerance; it is not an absolute quality guarantee.

Target unreachability within a decoding pass. If a target prefix is excluded from every active decoding path by beam pruning or sampling filters, the target cannot be recovered within that decoding pass. The coverage bonus is assigned after candidate generation and cannot restore an already discarded path. This limitation applies to the current pass rather than to all future generation: subsequent policy updates may change the probability of generating the target.

Errors in the reward proxy. When base reward is an imperfect proxy for recommendation utility, an unsuitable Explorer may satisfy the gate and receive a coverage bonus. If the inference evaluator shares this error, it may also rank that candidate highly. Quality gating therefore inherits limitations of the base reward signal; eligibility for a bonus does not establish that a candidate matches the user's preference. This is a possible mechanism of failure, rather than a measured attribution of errors in the reported experiments.

Concentrated role-conditioned distributions. If the Explorer-conditioned distributions concentrate on the same outputs and pruning or filtering removes alternatives, candidates may repeat the Anchor despite having distinct role identities. The coverage score is zero when an Explorer's complete trajectory equals the Anchor's; a late divergence still receives a positive, smaller coverage score. Concentration of the Anchor's distribution alone does not establish concentration of every Explorer distribution, since they use different role conditions. Repeated outputs also do not imply that the decoding procedure is equivalent to standard beam search.

Anchor-relative coverage and Explorer overlap. The coverage score compares each Explorer with the Anchor rather than directly penalizing overlap among Explorers. Multiple Explorers can therefore receive the same bonus for an identical alternative trajectory that diverges from the Anchor. Complementarity among Explorer outputs is an empirical property of the learned roles, not a pairwise-diversity guarantee imposed by the coverage score.

Table 16: Conceptual slate scenarios illustrating the reward mechanism and potential limitations. Product lists are hypothetical slates, not SID token sequences or measured case records. Here E_i denotes Explorer i; qualifying means $R_i \geq R_0 - \delta$ during training.

Scenario	Illustrative trajectories and mechanism
Success: useful alternatives	*Anchor:* [Smartphone, Case, Film]. E_1: [Smartwatch, Band, Charger]; E_2: [Tablet, Stylus, Cover]; E_3: [Snack, Drink, Candy]. Suppose E_1 and E_2 qualify, while E_3 does not. The first two receive coverage bonuses; E_3 retains only its base reward. A useful alternative to the Anchor's slate can then be available for task-based selection.
Failure: target unreachability	*Anchor:* [Running Shoes, Socks, Hat]. E_1: [Running Shoes, Shorts, Tee]; E_2: [Basketball, Hoop, Net]; E_3: [Dumbbell, Mat, Towel]. Suppose all paths to a desired [Dress, Heels, Bag] slate have been pruned. Even if every remaining Explorer qualifies for a bonus, rewarding these candidates cannot restore the omitted slate within the same decoding pass.
Failure: reward miscalibration	*Anchor:* [Laptop, Mouse, Keyboard]. E_1: [Gaming Chair, Desk, Lamp]; E_2: [Clickbait Item, Novelty, Toy]; E_3: [Monitor, Cable, Hub]. Suppose all Explorers qualify, but the base reward overvalues E_2. The gate permits its bonus despite poor underlying utility. An inference evaluator with a similar error may also favor E_2; the training bonus itself is not used for inference selection.
Failure: concentrated outputs	Suppose role-conditioned policies concentrate on the same surviving choices, yielding $y^{(1)} = \cdots = y^{(G-1)} = y^{(0)}$. Every Explorer then has coverage score $C_i = 0$. If an Explorer instead branches late, its coverage score is positive but smaller than for an earlier branch. Distinct identities alone do not rule out these patterns.

Connecting the scenarios to the reported diagnostics. These scenarios motivate reading the candidate statistics jointly. Repetition describes redundant prefix allocation, while U_Q combines prefix uniqueness with the fixed evaluation-quality threshold. Neither quantity records whether a particular held-out target is reachable; Pool Hit supplies that information. HR@10 records the percentage of all evaluated requests whose target appears within the final ten positions. Thus, an increase in U_Q accompanied by lower Reward Mean can indicate more qualified prefixes alongside weaker average candidate quality, as in the ungated-bonus comparison. The aggregate statistics do not identify which individual requests exhibit the hypothetical failures in Table 16. The analysis below formalizes the connection between the reward definition and these diagnostics.

I Analytical Properties and Diagnostic Interpretation

The following identities clarify the reward mechanism and the relationship between reachability and final accuracy. They follow from the definitions and complement the empirical comparisons.

First divergence determines coverage. For equal-length outputs, let τ_i be the first position at which Explorer i differs from the Anchor, with $\tau_i = L + 1$ when their complete trajectories are identical. Prefixes before τ_i match, and all prefixes from τ_i onward differ, even if subsequent tokens agree. Therefore, Eq. equation 14 becomes

$$C_i = \frac{L - \tau_i + 1}{L}. \tag{23}$$

Moving the first divergence one position earlier increases coverage by $1/L$, and increases the bonus by β/L if the Explorer remains eligible. The score does not count subsequent token mismatches or distinguish Explorers following the same alternative. Its interpretation is the location of branching from the Anchor, rather than semantic distance or pairwise diversity throughout the group.

Bounded shaping and group-relative advantages. Write $B_0 = 0$ and $B_i = \beta g_i C_i$ for Explorer bonuses. Since $0 \le B_i \le \beta$ for $\beta \ge 0$, any two candidates satisfy

$$\widetilde{R}_i - \widetilde{R}_j = (R_i - R_j) + (B_i - B_j), \qquad |B_i - B_j| \le \beta. \tag{24}$$

Shaping cannot reverse a base-reward ordering whose gap is strictly greater than β. Smaller gaps may be reordered by qualified divergence. Passing the gate need not place an Explorer above the Anchor after shaping: the gate allows a base-reward shortfall of δ, while the bonus is at most β. In the sequence-level group-relative formulation, a passing Explorer can also remain below the group's shaped-reward mean and have negative advantage. Conversely, a candidate receiving no bonus can have positive advantage if its base reward exceeds that mean. Gate acceptance, bonus receipt, and advantage sign are therefore distinct. This bound concerns rewards within a given group; it does not bound parameter updates or guarantee monotonic improvement of task performance.

Decomposing misses into reachability and selection. Consider the public evaluation in which final recommendations are selected from the measured candidate pool. Let P denote target presence in that pool and H its presence in the final top ten. Then $H \subseteq P$. For $p = \Pr(P) > 0$, define $h = \Pr(H)$ and $c = \Pr(H \mid P)$, giving

$$h = pc, \qquad 1 - h = (1 - p) + p(1 - c). \tag{25}$$

Here Pool Hit and HR@10 equal $100p$ and $100h$. Misses partition into targets absent from the pool and targets present but outside the final top ten. A larger Pool Hit can improve HR@10 even when the Pool Hit–HR@10 gap grows, as observed in the frozen-checkpoint role intervention. Different generators also make different targets reachable, so comparing their conditional rates c does not isolate the evaluator's intrinsic selection quality. Reporting reachability and final accuracy together separates these observable stages of recommendation while retaining this dependence on pool composition.