GROUP REWARD-MASS OPTIMIZATION FOR GENERATIVE RECOMMENDATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Generative recommendation formulates personalized recommendation as a conditional sequence generation task, allowing models to directly generate items aligned with user preferences. Because a user's intent can correspond to multiple valid items, sparse-reward policy optimization can concentrate generation probability on a narrow set of outputs, reducing recommendation diversity and coverage. To address this issue, we propose Group Reward-Mass Optimization (GRMO), a framework that formulates policy learning as direct probability mass matching within candidate groups. GRMO leverages recommendation rewards to construct a target probability mass distribution over sampled candidates. By performing intra-group normalization onboth the target and current policy distributions, GRMO minimizes their KL divergence, directly optimizing the relative probability allocation. Experiments on Reddit-v2 show that GRMO improves ranking performance over GRPO while producing more stable, query-conditioned recommendations and preserving access to high-quality alternatives.

1 INTRODUCTION

Conversational recommender systems allow users to express and refine their preferences through natural-language interaction (Li et al., 2018; Chen et al., 2019; Zhou et al., 2020; Wang et al., 2022). Large language models (LLMs) provide a flexible foundation for this task supporting the interpretation of user requests, reasoning over dialogue context, and generation of ranked recommendation lists within a unified text interface.

Supervised fine-tuning (SFT) adapts LLMs to recommendation data and output formats. However, its likelihood objective focuses on reproducing demonstration responses and does not directly optimize ranking metrics. In addition, a single user intent can admit multiple relevant items and valid recommendation lists. Effective recommendation therefore requires attention to both ranking quality and the allocation of generation probability across useful alternatives.

Reinforcement learning (RL) provides a natural way for aligning generated recommendations with task-specific feedback. Recent industrial systems have incorporated reinforcement learning into recommendation model post-training (Deng et al., 2025; Team et al., 2026; Jiang et al., 2026). Group Relative Policy Optimization (GRPO) (Shao et al., 2024) updates policies using relative advantages computed from responses sampled for the same context. Rank-GRPO (Zhu et al., 2026b) adapts this approach to recommendation by treating each recommendation position as an action unit and assigning credit according to the relevance of the generated item. Under sparse feedback, probability may concentrate on a narrow set of outputs, potentially limiting access to alternative high-quality recommendations.

FlowRL (Zhu et al., 2026a) applies flow-based learning (Bengio et al., 2021) to policy optimization, using a trajectory-balance objective to learn a reward-dependent distribution over trajectories. This perspective makes the allocation of generation probability across alternative trajectories an explicit learning objective. It is particularly relevant to recommendation, where a single user intent can admit multiple relevant items and valid recommendation lists. Motivated by this connection, we bring reward-dependent distribution matching to recommendation RL, aiming to improve ranking quality while preserving access to distinct high-quality recommendations.

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: Conceptual illustration of reward-mass matching. Given multiple useful candidates for the same user intent, standard methods often concentrate probability on a single output. Reward-mass matching, in contrast, aligns groupwise probability mass with a reward-dependent target, thereby maintaining support across multiple high-quality alternatives. Stars denote candidate rewards, and bars denote policy probabilities.

We propose **Group Reward-Mass Optimization (GRMO)** to generate recommendations. GRMO constructs reward-dependent target masses within each sampled group and matches the policy's normalized probability masses to these targets. By combining group distribution matching with position-wise recommendation feedback, GRMO guides probability allocation across sampled candidates, aiming to preserve access to multiple high-quality recommendation paths. We formulate a tractable groupwise reverse-KL objective with behavior-policy correction, retaining GRPO's sampled-group training structure. Experiments show improvements in ranking quality and quality-adjusted coverage of high-quality recommendation paths. Figure 1 provides a schematic comparison between concentrated probability allocation and reward-mass matching.

2 Related Work

2.1 Feedback Optimization for Recommendation

Preference alignment and reinforcement learning provide complementary ways to incorporate task feedback into recommendation models. LPO4Rec studies listwise preference alignment for tail-item recommendation(Li et al., 2025), while RankGR applies listwise direct preference optimization to generative retrieval (Fu et al., 2026). Their objectives use preference relations among candidate items. For conversational recommendation, Rank-GRPO extends GRPO(Shao et al., 2024) by treating each recommendation position as an action unit, introducing position-wise credit assignment and importance ratios based on geometric mean token probabilities (Zhu et al., 2026b). This formulation aligns the update granularity with individual recommended items.

2.2 Reward Distribution Learning and Flow-Based Training

Constructing a reward-dependent target and fitting a policy to it has precedents in reward-augmented maximum likelihood (Norouzi et al., 2016) and maximum a posteriori policy optimization (Abdolmaleki et al., 2018). Language-model distribution matching (Korbak et al., 2022) and DPO further connect KL-regularized reward maximization with a reward-tilted reference distribution (Rafailov et al., 2023). In recommendation, GFN4Rec aligns list-generation probabilities with rewards (Liu et al., 2023), while Flower uses flows on an item-prefix tree to derive token-level process supervision (Gao et al., 2025b). These approaches establish distribution learning at different granularities, from complete recommendation lists to item tokens. FlowRL applies reward-distribution matching to LLM reasoning through a trajectory-balance objective (Zhu et al., 2026a).

2.3 GROUPWISE TARGET MATCHING

Explicit matching on sampled candidate groups is closely related to several language-model alignment methods. BRAIn constructs a forward KL objective from self-normalized importance weights for both the target and the current policy (Pandey et al., 2024). LPO also normalizes current-to-behavior probability ratios within a response group and studies both forward and reverse KL projections (Qu et al., 2026). However, these studies do not examine whether groupwise mass matching improves ranking quality while preserving access to distinct high-quality alternatives in generative recommendation. To address this issue, we propose Group Reward-Mass Optimization (GRMO), which uses position-wise recommendation feedback for groupwise reverse-KL matching. We evaluate its effects on ranking quality and access to multiple high-quality recommendation alternatives.

3 METHODOLOGY

3.1 PROBLEM FORMULATION

We formulate the conversational recommendation task as an end-to-end conditional sequence generation problem. Let $x = (u_1, u_2, \ldots, u_T)$ denote the conversation history up to the current turn, comprising multi-turn dialogue utterances between the user and the system. Given the platform item catalog $\mathcal{C}$, the objective of the Conversational Recommender System is to generate a target text response $y = (t_1, t_2, \ldots, t_N)$ containing a ranked list of recommended items.

Formally, the generative model P_θ parameterized by θ models the conditional probability of the sequence y given the context x as:

$$P_\theta(y \mid x) = \prod_{i=1}^{N} P_\theta(t_i \mid x, t_{<i}), \tag{1}$$

where t_i represents the i-th token in the generated response, and $t_{<i}$ denotes the sequence of tokens preceding index i. The ultimate goal is to optimize θ such that the item titles embedded within the generated output y maximize target recommendation metrics (e.g., Recall@K, NDCG@K) with respect to the user's ground-truth target items $\mathcal{I}^*(x) \subset \mathcal{C}$.

3.2 SUPERVISED FINE-TUNING

Supervised Fine-Tuning (SFT) serves as the first stage of our training pipeline. The primary objective of SFT is to adapt the base Large Language Model (LLM) to the task interface, establishing catalog grounding, adherence to specified output formats, and basic ranking capabilities prior to reinforcement learning alignment.

Let $\mathcal{D}_{\text{SFT}} = \{(x^{(n)}, y^{*(n)})\}_{n=1}^{N_{\text{SFT}}}$ denote the dataset of demonstration pairs, where $x^{(n)}$ represents the conversational context and $y^{*(n)}$ is the ground-truth response containing a catalog-compliant ranked item list. We optimize the model parameters θ using the standard cross-entropy loss over the target response tokens via teacher forcing:

$$\mathcal{L}_{\text{SFT}}(\theta) = -\frac{1}{N_{\text{SFT}}} \sum_{n=1}^{N_{\text{SFT}}} \sum_{t=1}^{|y^{*(n)}|} \log \pi_\theta \left(y_t^{*(n)} \mid x^{(n)}, y_{<t}^{*(n)} \right). \tag{2}$$

This cold-start phase ensures that the model learns the candidate item domain and output syntax, providing a robust initialization for the subsequent RL-based ranking optimization.

3.3 GRMO: GROUP REWARD-MASS OPTIMIZATION

In this section, we will present **GRMO** (Group Reward-Mass Optimization), a reinforcement learning algorithm that frames policy alignment as distributional probability-mass matching. We first present the mathematical formulation of GRMO, followed by our reward design and total training objectives.

Let $\mathcal{X}$ denote the space of conversational contexts and $\mathcal{Y}$ denote the space of generated response sequences. For an input context $x \in \mathcal{X}$, an autoregressive language model parameterizes a distribution over sequences $y = (y_1, y_2, \ldots, y_T) \in \mathcal{Y}$ as:

$$\pi_\theta(y \mid x) = \prod_{t=1}^{T} \pi_\theta(y_t \mid x, y_{<t}). \tag{3}$$

Given a frozen reference policy $\pi_{\text{ref}}(\cdot \mid x)$ and a task reward function $R(x, y)$, the standard KL-regularized reinforcement learning objective is formulated as:

$$J_\beta(\pi; x) = \mathbb{E}_{y \sim \pi(\cdot|x)}[\beta R(x, y)] - \mathbb{D}_{\text{KL}}\left(\pi(\cdot \mid x) \parallel \pi_{\text{ref}}(\cdot \mid x)\right), \tag{4}$$

where $\beta > 0$ is a temperature hyperparameter that balances reward maximization against the KL divergence constraint.

By writing out the explicit summation over the discrete sequence space $\mathcal{Y}$, the objective can be expanded as:

$$J_\beta(\pi; x) = \sum_{y \in \mathcal{Y}} \pi(y \mid x) \left[\beta R(x, y) + \log \pi_{\text{ref}}(y \mid x) - \log \pi(y \mid x)\right]. \tag{5}$$

We define $\tilde{q}^*_\beta(y \mid x) = \pi_{\text{ref}}(y \mid x) \exp(\beta R(x, y))$ as the unnormalized reward-tilted reference measure. Its corresponding normalized probability mass function $q^*_\beta(y \mid x)$, which serves as the optimal target distribution, is

$$q^*_\beta(y \mid x) = \frac{\pi_{\text{ref}}(y \mid x) \exp(\beta R(x, y))}{Z_\beta(x)}, \quad \text{where} \quad Z_\beta(x) = \sum_{y' \in \mathcal{Y}} \pi_{\text{ref}}(y' \mid x) \exp(\beta R(x, y')). \tag{6}$$

Here, $Z_\beta(x)$ is the sequence-level partition function, and $\beta R(x, y) + \log \pi_{\text{ref}}(y \mid x) = \log q^*_\beta(y \mid x) + \log Z_\beta(x)$. Substituting this identity into Equation 5 gives

$$\begin{aligned} J_\beta(\pi; x) &= \sum_{y \in \mathcal{Y}} \pi(y \mid x) \left[\log q^*_\beta(y \mid x) + \log Z_\beta(x) - \log \pi(y \mid x)\right] \\ &= \log Z_\beta(x) \sum_{y \in \mathcal{Y}} \pi(y \mid x) - \sum_{y \in \mathcal{Y}} \pi(y \mid x) \log \frac{\pi(y \mid x)}{q^*_\beta(y \mid x)}. \end{aligned} \tag{7}$$

Since $\sum_{y \in \mathcal{Y}} \pi(y \mid x) = 1$, we obtain the exact functional identity:

$$J_\beta(\pi; x) = \log Z_\beta(x) - \mathbb{D}_{\text{KL}}\left(\pi(\cdot \mid x) \parallel q^*_\beta(\cdot \mid x)\right). \tag{8}$$

Because the partition function $Z_\beta(x)$ is independent of the policy parameters, optimizing the KL-regularized reward is equivalent to minimizing the reverse KL divergence between the current policy π_θ and the target distribution q^*_β:

$$\arg\max_\pi J_\beta(\pi; x) = \arg\min_\pi \mathbb{D}_{\text{KL}}\left(\pi(\cdot \mid x) \parallel q^*_\beta(\cdot \mid x)\right). \tag{9}$$

Groupwise Categorical Parameterization. Computing the global partition function $Z_\beta(x)$ over the high-dimensional output space $\mathcal{Y}$ is intractable. For each context x, we independently sample a sibling group $\mathcal{S}_x = \{y_1, \ldots, y_G\}$ from the behavior policy, i.e., $y_i \sim \pi_{\text{old}}(\cdot \mid x)$. The global reward-tilted distribution is $q^\star_\beta(y \mid x) = \pi_{\text{ref}}(y \mid x) \exp(\beta R(x, y)) / Z_\beta(x)$, where $Z_\beta(x) = \sum_{y \in \mathcal{Y}} \pi_{\text{ref}}(y \mid x) \exp(\beta R(x, y))$. Although $Z_\beta(x)$ cannot be evaluated directly, it cancels when the distribution is self-normalized over the sampled sibling group. To correct for the sampling distribution induced by π_{old}, we define categorical current-policy and target masses over $\mathcal{S}_x$ as

$$\begin{aligned} p_i &= \frac{\pi_\theta(y_i \mid x) / \pi_{\text{old}}(y_i \mid x)}{\sum_{j=1}^{G} \pi_\theta(y_j \mid x) / \pi_{\text{old}}(y_j \mid x)} \\ &= \text{Softmax}_i\left(\log \pi_\theta(y_i \mid x) - \log \pi_{\text{old}}(y_i \mid x)\right), \end{aligned} \tag{10}$$

$$\begin{aligned} q_i &= \frac{\pi_{\text{ref}}(y_i \mid x) \exp(\beta R(x, y_i)) / \pi_{\text{old}}(y_i \mid x)}{\sum_{j=1}^{G} \pi_{\text{ref}}(y_j \mid x) \exp(\beta R(x, y_j)) / \pi_{\text{old}}(y_j \mid x)} \\ &= \text{Softmax}_i\left(\beta R(x, y_i) + \log \pi_{\text{ref}}(y_i \mid x) - \log \pi_{\text{old}}(y_i \mid x)\right). \end{aligned} \tag{11}$$

Writing $p = [p_1, \ldots, p_G]^\top$ and $q = [q_1, \ldots, q_G]^\top$, GRMO minimizes the reverse KL divergence

$$\mathcal{L}_{\mathrm{GRMO}}(\theta) = D_{\mathrm{KL}}(p \,\|\, q) = \sum_{i=1}^{G} p_i \log \frac{p_i}{q_i}. \tag{12}$$

Self-normalization removes $Z_\beta(x)$, leaving a categorical KL over the sampled sibling group.

Gradient Analysis. During one update, the sampled candidates, π_{old}, π_{ref}, and target masses q_i are fixed; only π_θ is optimized. Let $\Delta_k = \log(p_k/q_k)$. Using $\partial p_i/\partial s_k = p_i(\mathbb{I}[i = k] - p_k)$, the reverse-KL logit gradient simplifies to

$$\frac{\partial \mathcal{L}_{\mathrm{GRMO}}}{\partial s_k} = p_k \left(\log p_k - \log q_k - \mathcal{L}_{\mathrm{GRMO}}\right). \tag{13}$$

Since π_{old} is stop-gradient within the update,

$$\nabla_\theta \mathcal{L}_{\mathrm{GRMO}} = \sum_{k=1}^{G} p_k \left(\log p_k - \log q_k - \mathcal{L}_{\mathrm{GRMO}}\right) \nabla_\theta \log \pi_\theta(y_k \mid x). \tag{14}$$

The coefficients sum to zero, so the update depends only on relative allocation within the sampled group and the unique interior stationary allocation is $p = q$. Under gradient descent, $\Delta_k > \mathcal{L}_{\mathrm{GRMO}}$ gives a suppressive logit-gradient contribution, whereas $\Delta_k < \mathcal{L}_{\mathrm{GRMO}}$ gives a promotive contribution. Because logits share both softmax normalization and neural-network parameters, these signs do not imply monotonic changes in individual probabilities after a finite update. Unlike GRPO's centered-advantage weighting, GRMO directly penalizes discrepancies between current and target groupwise masses; its effect on high-quality path access is evaluated empirically in Section 4.2.

3.4 Reward Design and Training Objectives

Position-Preserving Item Credit. GRMO evaluates every reward $r_{i,k}$ at the position where the corresponding item is emitted. Invalid catalog entries, items already observed in the conversation history, and unmatched recommendations receive zero credit at their original ranks; credit is never reassigned across positions. Let $\mathcal{T}_{i,k}$ denote the token span of the k-th item in sibling completion y_i. For ground-truth annotation m, the raw item credit is

$$c_{i,k}^{(m)} = \mathbf{1}_{\mathcal{C}}(a_{i,k}, \widehat{v}_{i,k})\, \mathbf{1}_{\mathrm{unseen}}(a_{i,k})\, h_{i,k}^{(m)}, \tag{15}$$

where $\mathbf{1}_{\mathcal{C}}$ checks catalog-valid title/year pairs under the native tolerance, $\mathbf{1}_{\mathrm{unseen}}$ excludes previously mentioned items, and $h_{i,k}^{(m)} \in \{0, 1\}$ applies single-use greedy matching against annotation m.

Unlike sampling one annotation, we analytically average over all V_x unique annotations with empirical multiplicities n_m:

$$r_{i,k} = \mathbb{E}_{m|x}\left[c_{i,k}^{(m)}\right] = \frac{1}{N_x} \sum_{m=1}^{V_x} n_m c_{i,k}^{(m)}, \qquad N_x = \sum_{m=1}^{V_x} n_m. \tag{16}$$

This Rao–Blackwellized reward removes annotation-sampling variance while preserving the original item position. A scalar sequence-level NDCG reward can be obtained from the DCG-weighted sum of $\{r_{i,k}\}_{k=1}^{K}$ for evaluation and for the motivating sequence-level formulation.

Position-Local Policy and Target Masses. For each sibling i and rank k, we summarize the owned item tokens by their mean log-ratios to the behavior policy:

$$s_{i,k}^{\theta} = \frac{1}{|\mathcal{T}_{i,k}|} \sum_{t \in \mathcal{T}_{i,k}} \left[\log \pi_\theta(y_{i,t} \mid x, y_{i,<t}) - \log \pi_{\mathrm{old}}(y_{i,t} \mid x, y_{i,<t})\right],$$

$$s_{i,k}^{\mathrm{ref}} = \frac{1}{|\mathcal{T}_{i,k}|} \sum_{t \in \mathcal{T}_{i,k}} \left[\log \pi_{\mathrm{ref}}(y_{i,t} \mid x, y_{i,<t}) - \log \pi_{\mathrm{old}}(y_{i,t} \mid x, y_{i,<t})\right]. \tag{17}$$

Thus, we use item-level token means but no global sequence-length normalization.

For sibling group g, let $\mathcal{I}_{g,k}$ contain the siblings with a valid item-token span at rank k. We center the position rewards within the supported siblings and divide by the detached batch-RMS scale s_{B}:

$$\widetilde{r}_{g,i,k} = \frac{r_{g,i,k} - |\mathcal{I}_{g,k}|^{-1} \sum_{j \in \mathcal{I}_{g,k}} r_{g,j,k}}{s_{\mathrm{B}}}. \tag{18}$$

The current-policy mass, frozen-reference base mass, and reward-tilted target are

$$\begin{aligned} p_{g,i,k} &= \mathrm{Softmax}_{i \in \mathcal{I}_{g,k}} \left(s^{\theta}_{g,i,k} \right), \\ q^{0}_{g,i,k} &= \mathrm{Softmax}_{i \in \mathcal{I}_{g,k}} \left(s^{\mathrm{ref}}_{g,i,k} \right), \\ q_{g,i,k} &= \mathrm{Softmax}_{i \in \mathcal{I}_{g,k}} \left(s^{\mathrm{ref}}_{g,i,k} + \alpha_{g,k} \beta \widetilde{r}_{g,i,k} \right). \end{aligned} \tag{19}$$

The largest feasible $\alpha_{g,k} \in [0, 1]$ is selected so that $D_{\mathrm{KL}}(q_{g,\cdot,k} \| q^{0}_{g,\cdot,k}) \leq \epsilon_{\mathrm{target}}$. This target-KL budget regularizes the reward tilt itself; it is not an additional PPO KL loss.

Each sibling item is conditioned on its own generated prefix. Therefore, Equation 19 defines a position-local groupwise surrogate over sampled item spans. This task-aligned instantiation preserves recommendation-position credit while retaining GRMO's probability-mass matching principle.

Final Training Objective. Let $C_{g,k}$ be the number of valid item-owned tokens at rank k, ω_k the configured position weight, and N_{tok} the total valid-token count. The mass-matching objective is

$$\mathcal{L}_{\mathrm{mass}} = \frac{1}{N_{\mathrm{tok}}} \sum_{g} \sum_{k=1}^{K} C_{g,k} \omega_k D_{\mathrm{KL}} \left(p_{g,\cdot,k} \parallel q_{g,\cdot,k} \right). \tag{20}$$

The selected configurations optimize

$$\mathcal{L}_{\mathrm{train}} = \mathcal{L}_{\mathrm{mass}} + \mathcal{L}_{\mathrm{length}}, \tag{21}$$

where $\mathcal{L}_{\mathrm{length}}$ contains end-of-list, overflow, and premature-stop shaping. Ratio clipping is applied only to $\mathcal{L}_{\mathrm{length}}$. The actor-KL, KL-in-reward, entropy, and weight-decay coefficients are zero, and the gradient norm is clipped at 1.0.

Further details on sequence-level evaluation, masks, position weights, and reward aggregation are provided in Appendix B.

4 EXPERIMENTS

In this section, We evaluate GRMO on REDDIT-V2 under two backbone scales for conversational recommendation.

4.1 EXPERIMENTAL SETUP

Dataset and Experimental Setup. We conduct experiments on REDDIT-V2, a large-scale benchmark for conversational movie recommendation. Following Rank-GRPO (Zhu et al., 2026b), we use the same data splits and evaluation protocol, with 383,013, 9,421, and 10,972 dialogues for training, validation, and testing.

Implementation Details. We instantiate our framework on two compact open-source model families: **Qwen2.5-0.5B-Instruct** and **Llama-3.2-3B-Instruct**. Both backbones are initialized from SFT checkpoints trained under identical data conditions for fair comparison.

During the RL alignment phase, models are trained on 8×80GB A100 GPUs. Rollout generations are sampled with a nucleus temperature of 1.0, producing candidate groups of size $G = 8$ for Qwen2.5-0.5B and Llama-3.2-3B. Evaluation is conducted under deterministic greedy decoding (temperature 0.0) with cutoffs at $K \in \{5, 10, 15, 20\}$, measuring Recall@K ($R@K$) and NDCG@K ($N@K$).

Table 1: Comparison of GRMO with baselines on the REDDIT-V2 dataset. Here, $R@k$ and $N@k$ stand for Recall@k and NDCG@k, respectively.

Method	R@5	R@10	R@15	R@20	N@5	N@10	N@15	N@20
Traditional and Transformer-based CRS								
Redial (Li et al., 2018)	0.0103	0.0228	0.0274	0.0322	0.0073	0.0113	0.0128	0.0143
KBRD (Chen et al., 2019)	0.0444	0.0813	0.1058	0.1223	0.0305	0.0418	0.0490	0.0545
KGSF (Zhou et al., 2020)	0.0579	0.0921	0.1246	0.1433	0.0405	0.0503	0.0599	0.0662
UniCRS (Wang et al., 2022)	0.0722	0.1053	0.1344	0.1494	0.0548	0.0640	0.0726	0.0778
NBCRS (Xie et al., 2024)	0.0801	0.1290	0.1655	0.2019	0.0661	0.0833	0.0954	0.1048
LLM Prompting-Based Methods								
GPT-4o-mini (zero-shot)	0.0949	0.1348	0.1600	0.1687	0.0747	0.0877	0.0950	0.0973
GPT-4o (zero-shot) (He et al., 2023)	0.1106	0.1625	0.1992	0.2147	0.0861	0.1028	0.1136	0.1197
CRAG (GPT-4o) (Zhu et al., 2025)	0.1146	0.1715	0.2030	0.2212	0.0885	0.1065	0.1164	0.1227
LLM Post-Training-Based Methods								
Qwen2.5-0.5B (zero-shot)	0.0021	0.0021	0.0021	0.0021	0.0023	0.0022	0.0021	0.0021
+ SFT	0.0642	0.1027	0.1212	0.1308	0.0502	0.0625	0.0678	0.0704
+ SFT (DPO) + DPO (Rafailov et al., 2023)	0.0716	0.1058	0.1300	0.1473	0.0570	0.0681	0.0752	0.0797
+ SFT (DPO) + S-DPO (Chen et al., 2024)	0.0875	0.1240	0.1488	0.1699	0.0657	0.0772	0.0845	0.0888
+ SFT (DPO) + SPRec(Gao et al., 2025a)	0.0841	0.1256	0.1524	0.1686	0.0657	0.0775	0.0855	0.0898
+ SFT + Vanilla GRPO	0.0834	0.1298	0.1617	0.1803	0.0651	0.0801	0.0895	0.0945
+ SFT + Rank-GRPO (log)	0.0892	0.1353	0.1720	0.1946	0.0701	0.0849	0.0957	0.1017
+ SFT + Rank-GRPO ($\exp\infty$)	<u>0.0946</u>	<u>0.1468</u>	<u>0.1813</u>	<u>0.2047</u>	<u>0.0744</u>	<u>0.0915</u>	<u>0.1016</u>	<u>0.1079</u>
+ SFT + **GRMO (Ours)**	**0.0948**	**0.1472**	**0.1817**	**0.2071**	**0.0746**	**0.0918**	**0.1019**	**0.1086**
Llama-3.2-3B (zero-shot)	0.0574	0.0805	0.0912	0.0961	0.0463	0.0535	0.0566	0.0580
+ SFT	0.0788	0.1191	0.1410	0.1541	0.0615	0.0744	0.0807	0.0842
+ SFT (DPO) + DPO (Rafailov et al., 2023)	0.0894	0.1279	0.1491	0.1697	0.0708	0.0808	0.0915	0.0896
+ SFT (DPO) + S-DPO (Chen et al., 2024)	0.1072	0.1419	0.1734	0.1943	0.0807	0.0933	0.1019	0.1079
+ SFT (DPO) + SPRec (Gao et al., 2025a)	0.1053	0.1491	0.1809	0.1965	0.0828	0.0956	0.1043	0.1109
+ SFT + Vanilla GRPO	0.1034	0.1568	0.1946	0.2186	0.0824	0.0996	0.1106	0.1170
+ SFT + Rank-GRPO (log)	0.1103	0.1638	0.2033	0.2302	0.0862	0.1035	0.1169	0.1239
+ SFT + Rank-GRPO ($\exp\infty$)	<u>0.1178</u>	<u>0.1756</u>	<u>0.2150</u>	<u>0.2368</u>	<u>0.0919</u>	<u>0.1107</u>	<u>0.1223</u>	<u>0.1283</u>
+ SFT + **GRMO (Ours)**	**0.1183**	**0.1761**	**0.2157**	**0.2377**	**0.0923**	**0.1110**	**0.1225**	**0.1289**

Baselines. The conversational recommendation baselines include ReDial (Li et al., 2018), which models dialogue and user preferences; knowledge-enhanced methods KBRD (Chen et al., 2019), KGSF (Zhou et al., 2020), and UniCRS (Wang et al., 2022); and the neighborhood-based method NBCRS (Xie et al., 2024).We also consider GPT-4o-mini and GPT-4o in the zero-shot recommendation setting (He et al., 2023), together with CRAG (Zhu et al., 2025), which augments LLM recommendations with collaborative retrieval and reflection. We also include DPO (Rafailov et al., 2023), S-DPO (Chen et al., 2024), SPRec (Gao et al., 2025a),GRPO and Rank-GRPO (Zhu et al., 2026b) as preference-optimization baselines.

4.2 Main Results and Empirical Analysis

Table 1 presents the quantitative benchmarking results across all evaluated paradigms on REDDIT-V2. GRMO consistently improves upon the strongest same-backbone policy-optimization baseline across both Qwen2.5-0.5B and Llama-3.2-3B. Specifically, compared to Rank-GRPO, GRMO achieves consistent gains across all metrics, with Recall@20 improving from 0.2047 to 0.2071 ($+1.17\%$) and NDCG@20 from 0.1079 to 0.1086 ($+0.65\%$) on Qwen2.5-0.5B, and further scaling up to 0.2377 ($+0.38\%$) and 0.1289 ($+0.47\%$) on Llama-3.2-3B. When compared against Vanilla GRPO, the performance improvements are even more pronounced. These results show consistent ranking gains across model scales and evaluation cutoffs.

We also study whether the improvement of GRMO is accompanied by meaningful diversity rather than uncontrolled sampling noise. We compare GRPO and GRMO using the same 512 fixed evaluation query contexts and draw 32 stochastic recommendation lists per query and policy at temperature 1.0, yielding $65{,}536$ trajectories across the two backbones. Each generated list is represented as a rank-discounted distribution over valid unseen movies. We then use the Hellinger distance to mea-

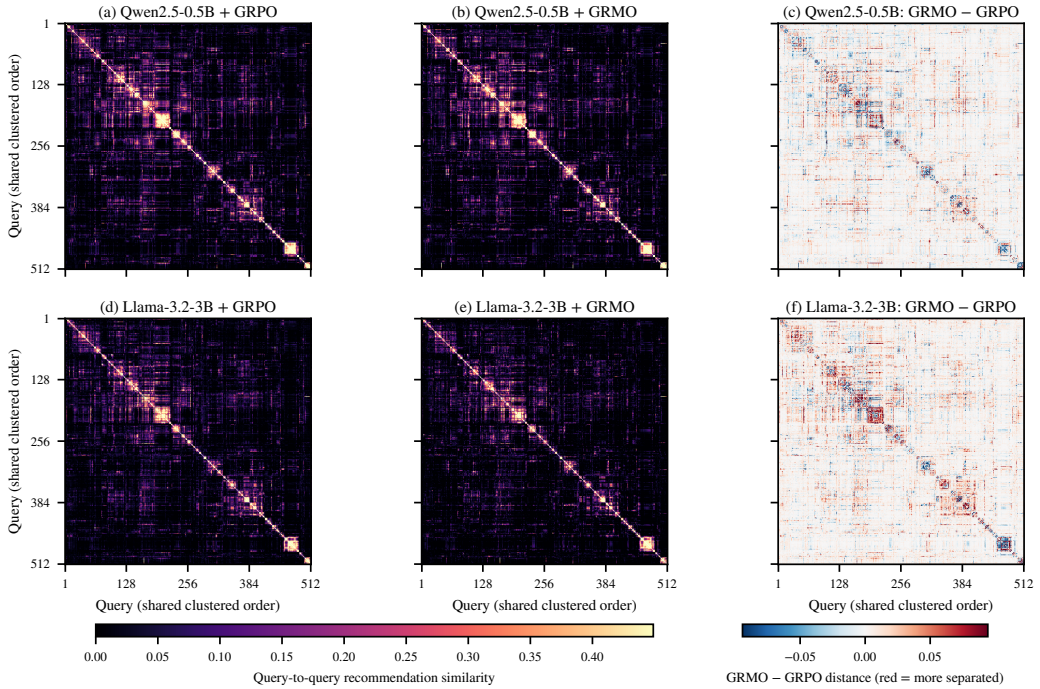

Figure 2: Query-conditioned recommendation diversity. Rows show **Qwen2.5-0.5B** and **Llama-3.2-3B**. The first two columns visualize query-to-query recommendation similarity under the same ordering and color scale. In the third column, red indicates that GRMO separates a query pair more than GRPO, while blue indicates greater similarity.

Table 2: Query-conditioned diversity comparison.

Backbone	Policy	Consistency	Inter-query	Separation	HQ rate	HQ coverage
Qwen2.5-0.5B	GRPO	0.4001	0.9752	0.3753	31.78%	0.9480
Qwen2.5-0.5B	**GRMO**	**0.4421**	**0.9763**	**0.4184**	**33.57%**	**1.0078**
Llama-3.2-3B	GRPO	0.3983	0.9797	0.3780	36.11%	0.8183
Llama-3.2-3B	**GRMO**	**0.4799**	**0.9833**	**0.4632**	**42.18%**	**0.9346**

sure both diversity among repeated samples for the same query and diversity between the mean recommendation distributions of different queries.

GRMO improves query-conditioned diversity. Figure 2 and Table 2 reveal the same pattern at both model scales. GRMO reduces within-query stochastic variation, increasing consistency by 10.5% for Qwen2.5-0.5B and 20.5% for Llama-3.2-3B. At the same time, inter-query diversity increases by 0.00111 for Qwen2.5-0.5B and 0.00367 for Llama-3.2-3B, with paired-query bootstrap intervals that exclude zero. Together, lower within-query variation and higher inter-query diversity indicate more stable recommendations for the same intent and clearer differences across user intents.

GRMO shifts probability mass toward high-quality paths. Table 2 demonstrates the shifts in high-quality generation behavior: the high-quality trajectory rate increases from 31.78% to 33.57% for Qwen2.5-0.5B and from 36.11% to 42.18% for Llama-3.2-3B, while the corresponding probability-weighted high-quality path coverage rises from 0.9480 to 1.0078 and from 0.8183 to 0.9346, respectively. HQCoverage combines the probability of sampling a high-quality trajectory with its effective path support. Together with the consistency and inter-query results, the higher HQCoverage characterizes GRMO's diversity as query-conditioned and quality-aware. Formal definitions are provided in Appendix A.

The quality advantage is robust to sampling temperature. Figure 3(a) isolates temperature while fixing the policies, query set, sample count, top-p, and sampling seed. GRMO improves

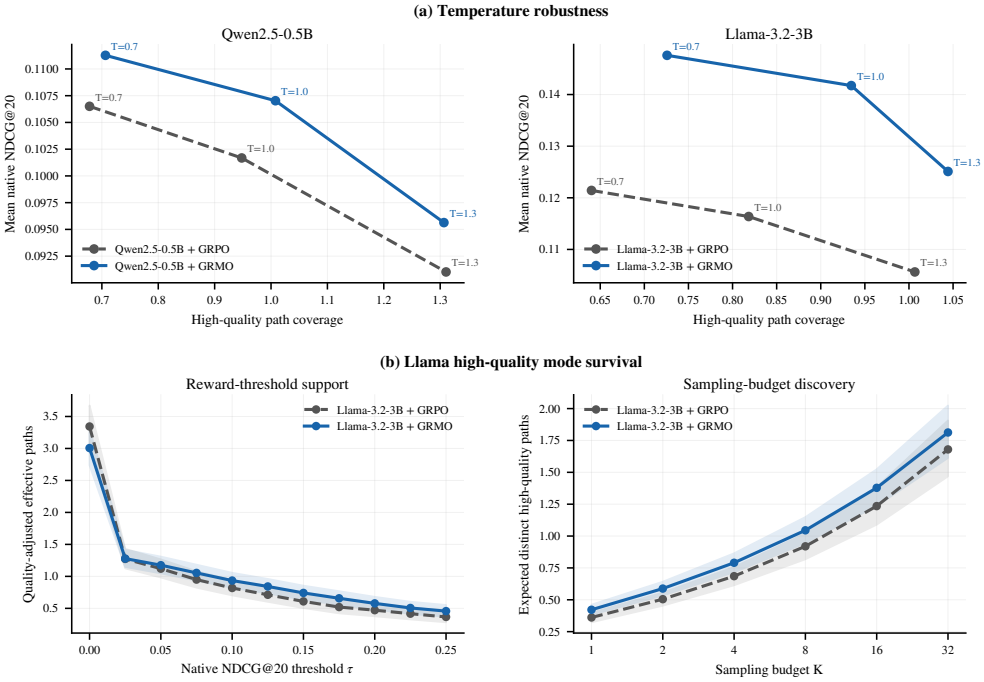

Figure 3: Robustness analysis of GRMO. (a) Quality robustness across sampling temperatures $T \in \{0.7, 1.0, 1.3\}$ (evaluated on 512 queries with 32 trajectories each). (b) Accessible high-quality path support on Llama-3.2-3B: expected number of distinct high-quality paths across reward thresholds and sampling budgets K. Shaded areas denote 95% confidence intervals.

average NDCG@20 across all six backbone-temperature configurations. For Llama-3.2-3B, when the temperature T is set to 0.7, 1.0, and 1.3, the corresponding paired gains are 0.02617, 0.02536, and 0.01946, respectively. The GRMO–GRPO performance gap remains positive at all three temperatures.

Improved access to high-quality paths. Figure 3(b) separates unfiltered path support from quality-weighted path access on Llama-3.2-3B. At threshold zero, GRMO contracts low-value branches. After quality filtering, the probability-weighted support is higher throughout thresholds 0.10–0.25, and a fixed unconditional sampling budget discovers more distinct high-quality paths for $K = 1$–16. These metrics measure accessible high-quality support under the sampling distribution, showing that GRMO makes high-quality alternatives easier to reach under a fixed sampling budget.

5 Conclusion

We presented Group Reward-Mass Optimization (GRMO), a probability-mass matching framework for generative recommendation. By directly optimizing the allocation of probability mass among sampled candidates, GRMO improves recommendation quality while retaining multiple high-quality alternatives. Experiments on Reddit-v2 with two model backbones show consistent gains in ranking performance, together with a favorable balance between recommendation quality and meaningful diversity. GRMO also remains robust across different decoding temperatures and sampling budgets. Future work will explore extending GRMO beyond recommendation to general generation and reasoning tasks with multiple valid outputs.

AI use statement

We used generative AI tools to assist with language polishing and improve the clarity and readability of the manuscript. The authors have reviewed the AI-assisted revisions and take full responsibility for the final content of this work.

REFERENCES

Abbas Abdolmaleki, Jost Tobias Springenberg, Yuval Tassa, Remi Munos, Nicolas Heess, and Martin Riedmiller. Maximum a posteriori policy optimisation. *arXiv preprint arXiv:1806.06920*, 2018.

Emmanuel Bengio, Moksh Jain, Maksym Korablyov, Doina Precup, and Yoshua Bengio. Flow network based generative models for non-iterative diverse candidate generation, 2021. URL https://arxiv.org/abs/2106.04399.

Qibin Chen, Junyang Lin, Yichang Zhang, Ming Ding, Yukuo Cen, Hongxia Yang, and Jie Tang. Towards knowledge-based recommender dialog system. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 1803–1813, 2019.

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.

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, 2025. URL https://arxiv.org/abs/2502.18965.

Kairui Fu, Changfa Wu, Kun Yuan, Binbin Cao, Dunxian Huang, Yuliang Yan, Junjun Zheng, Jianning Zhang, Silu Zhou, Jian Wu, et al. Rankgr: Rank-enhanced generative retrieval with listwise direct preference optimization in recommendation. *arXiv preprint arXiv:2602.08575*, 2026.

Chongming Gao, Ruijun Chen, Shuai Yuan, Kexin Huang, Yuanqing Yu, and Xiangnan He. Sprec: Self-play to debias llm-based recommendation. In *Proceedings of the ACM on Web Conference 2025*, pp. 5075–5084, 2025a.

Chongming Gao, Mengyao Gao, Chenxiao Fan, Shuai Yuan, Wentao Shi, and Xiangnan He. Process-supervised llm recommenders via flow-guided tuning. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1934–1943, 2025b.

Zhankui He, Zhouhang Xie, Rahul Jha, Harald Steck, Dawen Liang, Yesu Feng, Bodhisattwa Prasad Majumder, Nathan Kallus, and Julian McAuley. Large language models as zero-shot conversational recommenders. In *Proceedings of the 32nd ACM international conference on information and knowledge management*, pp. 720–730, 2023.

Jie Jiang, Hongbo Tang, Wenjie Wu, Yangru Huang, Zhenmao Li, Qian Li, Changping Wang, Jun Zhang, and Huan Yu. S-grec: Personalized semantic-aware generative recommendation with asymmetric advantage. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 7465–7476, 2026.

Tomasz Korbak, Hady Elsahar, Germán Kruszewski, and Marc Dymetman. On reinforcement learning and distribution matching for fine-tuning language models with no catastrophic forgetting. *Advances in Neural Information Processing Systems*, 35:16203–16220, 2022.

Raymond Li, Samira Ebrahimi Kahou, Hannes Schulz, Vincent Michalski, Laurent Charlin, and Chris Pal. Towards deep conversational recommendations. *Advances in neural information processing systems*, 31, 2018.

Zihao Li, Chao Yang, Tong Zhang, Yakun Chen, Xianzhi Wang, Guandong Xu, and Daoyi Dong. Listwise preference alignment optimization for tail item recommendation. *IEEE Transactions on Computational Social Systems*, 2025.

Shuchang Liu, Qingpeng Cai, Zhankui He, Bowen Sun, Julian McAuley, Dong Zheng, Peng Jiang, and Kun Gai. Generative flow network for listwise recommendation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1524–1534, 2023.

Mohammad Norouzi, Samy Bengio, Navdeep Jaitly, Mike Schuster, Yonghui Wu, Dale Schuurmans, et al. Reward augmented maximum likelihood for neural structured prediction. *Advances In Neural Information Processing Systems*, 29, 2016.

Gaurav Pandey, Yatin Nandwani, Tahira Naseem, Mayank Mishra, Guangxuan Xu, Dinesh Raghu, Sachindra Joshi, Asim Munawar, and Ramón Fernandez Astudillo. Brain: Bayesian reward-conditioned amortized inference for natural language generation from feedback. *arXiv preprint arXiv:2402.02479*, 2024.

Yun Qu, Qi Wang, Yixiu Mao, Heming Zou, Yuhang Jiang, Yingyue Li, Wutong Xu, Lizhou Cai, Weijie Liu, Clive Bai, et al. Listwise policy optimization: Group-based rlvr as target-projection on the llm response simplex. *arXiv preprint arXiv:2605.06139*, 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.

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.

OneRec Team, Biao Yang, Boyang Ding, Chenglong Chu, Dunju Zang, Fei Pan, Han Li, Hao Jiang, Honghui Bao, Huanjie Wang, et al. Onereason technical report, 2026. URL https://arxiv.org/abs/2606.06260.

Xiaolei Wang, Kun Zhou, Ji-Rong Wen, and Wayne Xin Zhao. Towards unified conversational recommender systems via knowledge-enhanced prompt learning. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, pp. 1929–1937, 2022.

Zhouhang Xie, Junda Wu, Hyunsik Jeon, Zhankui He, Harald Steck, Rahul Jha, Dawen Liang, Nathan Kallus, and Julian McAuley. Neighborhood-based collaborative filtering for conversational recommendation. In *Proceedings of the 18th ACM Conference on Recommender Systems*, pp. 1045–1050, 2024.

Kun Zhou, Wayne Xin Zhao, Shuqing Bian, Yuanhang Zhou, Ji-Rong Wen, and Jingsong Yu. Improving conversational recommender systems via knowledge graph based semantic fusion. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 1006–1014, 2020.

Xuekai Zhu, Daixuan Cheng, Dinghuai Zhang, Henry Li, Kaiyan Zhang, Che Jiang, Youbang Sun, Ermo Hua, Yuxin Zuo, Xingtai Lv, et al. Flowrl: Matching reward distributions for llm reasoning. In *International Conference on Learning Representations*, 2026a.

Yaochen Zhu, Chao Wan, Harald Steck, Dawen Liang, Yesu Feng, Nathan Kallus, and Jundong Li. Collaborative retrieval for large language model-based conversational recommender systems. In *Proceedings of the ACM on Web Conference 2025*, pp. 3323–3334, 2025.

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 Diversity Metric Definitions

Let Q denote the number of queries, S the number of sampled trajectories per query, and $K = 20$ the recommendation cutoff. For query q and trajectory s, we represent the generated recommendation list as a rank-discounted distribution over valid unseen items:

$$p_{q,s}(i) = \frac{\sum_{r \in \mathcal{R}_{q,s}} w_r \, \mathbb{I}[x_{q,s,r} = i]}{\sum_{r \in \mathcal{R}_{q,s}} w_r}, \qquad w_r = \frac{1}{\log_2(r+1)}, \tag{22}$$

where $\mathcal{R}_{q,s}$ contains the valid emitted ranks and $x_{q,s,r}$ is the item generated at rank r.

We measure the distance between two recommendation distributions using the Hellinger distance:

$$H(p, p') = \sqrt{1 - \sum_i \sqrt{p(i)p'(i)}}. \tag{23}$$

Within-query diversity and consistency. Within-query diversity measures stochastic variation among repeated samples for the same query:

$$D_{\text{within}} = \frac{1}{Q} \sum_{q=1}^{Q} \frac{2}{S(S-1)} \sum_{1 \le s < t \le S} H(p_{q,s}, p_{q,t}). \tag{24}$$

We report consistency as

$$\text{Consistency} = 1 - D_{\text{within}}. \tag{25}$$

Inter-query diversity and separation. The mean recommendation distribution for query q is

$$\bar{p}_q = \frac{1}{S} \sum_{s=1}^{S} p_{q,s}. \tag{26}$$

Inter-query diversity is then

$$D_{\text{inter}} = \frac{2}{Q(Q-1)} \sum_{1 \le q < q' \le Q} H(\bar{p}_q, \bar{p}_{q'}). \tag{27}$$

The query-separation margin is

$$\text{Separation} = D_{\text{inter}} - D_{\text{within}}. \tag{28}$$

A larger value indicates that differences between user queries dominate uncontrolled sampling variation within the same query.

High-quality trajectory rate. Let $R_{q,s}$ denote the native NDCG@20 reward and let $\tau = 0.10$. Define

$$z_{q,s} = \mathbb{I}[R_{q,s} \ge \tau], \qquad \rho_q = \frac{1}{S} \sum_{s=1}^{S} z_{q,s}. \tag{29}$$

The reported high-quality trajectory rate is

$$\text{HQRate} = \frac{1}{Q} \sum_{q=1}^{Q} \rho_q. \tag{30}$$

Quality-adjusted high-quality path coverage. Let $c_{q,s}$ denote the reward-independent path cluster assigned to trajectory s. Among the high-quality trajectories for query q, the probability mass of cluster c is

$$\pi_{q,c}^{\text{HQ}} = \frac{\sum_{s=1}^{S} z_{q,s} \, \mathbb{I}[c_{q,s} = c]}{\sum_{s=1}^{S} z_{q,s}}. \tag{31}$$

Its effective number of high-quality paths is the exponential Shannon entropy:

$$N_{\mathrm{eff},q}^{\mathrm{HQ}} = \exp\left(-\sum_{c} \pi_{q,c}^{\mathrm{HQ}} \log \pi_{q,c}^{\mathrm{HQ}}\right). \tag{32}$$

Finally, quality-adjusted high-quality path coverage is

$$\mathrm{HQCoverage} = \frac{1}{Q}\sum_{q=1}^{Q} \rho_q \, N_{\mathrm{eff},q}^{\mathrm{HQ}}. \tag{33}$$

We set the contribution to zero when a query has no trajectory satisfying the quality threshold. HQCoverage is an effective path-count metric and is therefore not restricted to the interval $[0, 1]$. Because it multiplies the high-quality trajectory rate by effective path support, an increase may be caused by a higher probability of sampling high-quality trajectories, a larger conditional path support, or both. For diagnostic decomposition, we additionally define the high-quality-probability-weighted effective path count

$$\widetilde{N}_{\mathrm{eff}}^{\mathrm{HQ}} = \frac{\mathrm{HQCoverage}}{\mathrm{HQRate}} = \frac{\sum_q \rho_q N_{\mathrm{eff},q}^{\mathrm{HQ}}}{\sum_q \rho_q}, \tag{34}$$

whenever $\mathrm{HQRate} > 0$. This quantity remains probability weighted and is not the unweighted per-query mean conditional mode count.

B From Position-Level Rewards to GRMO Targets

Let $y_i = (a_{i,1}, \ldots, a_{i,K})$ denote the recommendation list generated by the i-th sibling, and let $r_{i,k}$ denote the reward assigned to the item emitted at position k. Invalid, previously observed, or unmatched items receive $r_{i,k} = 0$ at their original positions.

Sequence-Level Evaluation. For reporting a scalar list-quality score, position credits are aggregated using the native DCG normalization:

$$R(x, y_i) = \frac{\sum_{k=1}^{K} d_k r_{i,k}}{\mathrm{IDCG}_x@K}, \qquad d_k = \frac{1}{\log_2(k+1)}. \tag{35}$$

This scalar is used for evaluation and motivates the sequence-level derivation in Section 3.3. The implemented rank-local objective does not consume $R(x, y_i)$; it retains the complete vector $(r_{i,1}, \ldots, r_{i,K})$.

Support and Token Scoring. An item participates in the rank-k mass-matching term only if the parser assigns it a nonempty owned-token span $\mathcal{T}_{i,k}$. Missing ranks have no target support. A parsed but invalid, seen, or unmatched item remains on the sampled support with reward zero. Under the selected sample-occurrence measure, repeated recommendations remain distinct sampled occurrences rather than being merged into one candidate.

Both current and reference scores are means of token log-ratios over $\mathcal{T}_{i,k}$, as defined in Equation 17. Hence, item length does not multiply its candidate logit. No additional normalization by total response length is applied.

Reward Normalization and Target Calibration. For every complete sibling group g and active rank k, rewards are centered over the supported siblings. A detached scale s_{B} is computed once from the complete rollout batch so that microbatching and data-parallel partitioning do not change the reward scale. The normalized credit is

$$\widetilde{r}_{g,i,k} = \frac{r_{g,i,k} - |\mathcal{I}_{g,k}|^{-1} \sum_{j \in \mathcal{I}_{g,k}} r_{g,j,k}}{s_{\mathrm{B}}}. \tag{36}$$

Starting from $q^0_{g,\cdot,k} = \mathrm{Softmax}_i(s^{\mathrm{ref}}_{g,i,k})$, GRMO follows the exponential reward-tilting path

$$q_{g,i,k}(\alpha) = \mathrm{Softmax}_{i \in \mathcal{I}_{g,k}}\left(s^{\mathrm{ref}}_{g,i,k} + \alpha\beta\widetilde{r}_{g,i,k}\right). \tag{37}$$

Because $D_{\mathrm{KL}}(q(\alpha)\|q^0)$ is monotone along this exponential path, a fixed bisection selects the largest $\alpha \in [0, 1]$ satisfying the configured target budget $\epsilon_{\mathrm{target}} = 0.25$. Flat-reward ranks therefore retain their reference base mass, while informative ranks receive the strongest feasible reward tilt.

Position Weights and Complete Loss. The outer weight in Equation 20 is configuration-specific. Qwen2.5-0.5B uses normalized DCG weights $\omega_k \propto 1/\log_2(k+1)$. The selected Llama-3.2-3B configuration embeds position sensitivity in its native local NDCG-marginal credit and uses $\omega_k = 1$, avoiding duplicate discounting. In both cases, $C_{g,k}$ counts owned item tokens and the final denominator is the total number of valid response tokens:

$$\mathcal{L}_{\text{mass}} = \frac{1}{N_{\text{tok}}} \sum_{g} \sum_{k=1}^{K} C_{g,k} \omega_k D_{\text{KL}}(p_{g,\cdot,k} \parallel q_{g,\cdot,k}) \,. \tag{38}$$

The selected runs add the existing length/EOS shaping term, giving $\mathcal{L}_{\text{train}} = \mathcal{L}_{\text{mass}} + \mathcal{L}_{\text{length}}$. Format-mass and unowned-token KL coefficients are zero, as are the generic actor KL loss, KL-in-reward term, and entropy coefficient.

The loss compares sampled trajectories at a common recommendation rank; their autoregressive prefixes may differ, so the resulting objective is a rank-local surrogate.