Group-Aware Reinforcement Learning for Output Diversity in Large Language Models

Oron Anschel [*], **Alon Shoshan, Adam Botach, Shunit Haviv Hakimi, Asaf Gendler, Emanuel Ben Baruch, Nadav Bhonker, Igor Kviatkovsky, Manoj Aggarwal, Gerard Medioni**

Amazon

{oronans, alonshos, kabotach, havivs, gendlasa, emanbb, nadavb, kviat, manojagg, medioni}@amazon.com

Abstract

Large Language Models (LLMs) often suffer from mode collapse, repeatedly generating the same few completions even when many valid answers exist, limiting their diversity across a wide range of tasks. We introduce **Group-Aware Policy Optimization (GAPO)**, a simple extension of the recent and popular Group Relative Policy Optimization (GRPO) that computes rewards over the group as a whole. GAPO enables learning from the group-level properties such as diversity and coverage. We demonstrate GAPO using a frequency-aware reward function that encourages uniform sampling over valid LLM completions, and show that GAPO-trained models produce valid and more diverse model responses. Beyond this setup, GAPO generalizes to open-ended prompts and improves response diversity without compromising accuracy on standard LLM benchmarks (GSM8K, MATH, HumanEval, MMLU-Pro). Our code will be made publicly available.

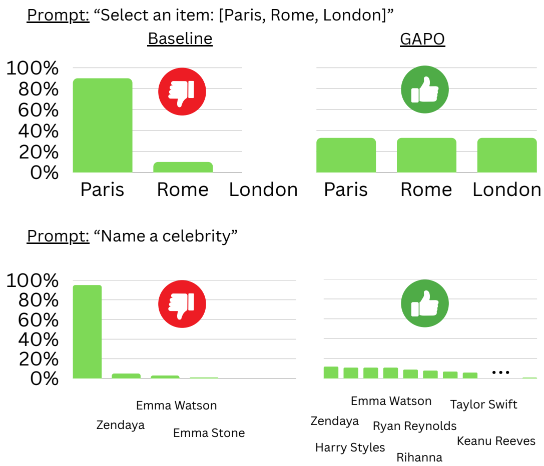


Figure 1: GAPO with a frequency-aware reward promotes output uniformity and diversity. GAPO mitigates over-representation of frequent completions and encourages more balanced generation, both in list selection tasks and open-ended prompts.

1 Introduction

Large Language Models (LLMs), particularly sophisticated instruction-following systems such as ChatGPT, Claude, Gemini, Qwen, and DeepSeek, are experiencing rapidly increasing deployment across a diverse range of real-world applications and use cases (Ouyang et al., 2022; Anil et al., 2023; Yang et al., 2024; Guo et al., 2025). While reinforcement learning from human feedback (RLHF) (Ouyang et al., 2022) improves factuality and alignment, it often reduces output diversity (Kirk et al., 2023). This limitation is especially problematic in creative or open-ended tasks, where multiple distinct completions may be equally valid.

This concerning reduction in output diversity is frequently characterized and analyzed in the literature as mode collapse (O'Mahony et al., 2024), a phenomenon where models exhibit a strong tendency to repeatedly generate the same limited set of responses across multiple interactions. For example, when prompted with the seemingly open-ended request *tell me a joke*", popular models like ChatGPT-4o and Claude Sonnet 3.5 frequently respond with virtually identical outputs such as: *Why don't scientists trust atoms? Because they make up everything!"* (Jentzsch and Kersting, 2023). While these responses remain both well-aligned with human preferences and linguistically fluent, such persistent repetition clearly demonstrates a problematic overconcentration of probability mass on an extremely limited subset of the vast space of possible completions, thereby significantly limiting the model's overall expressiveness and creative potential.

Recent work has identified this behavior as a consequence of the training pipeline. Both supervised fine-tuning (SFT) and RLHF have been shown to push models toward high-probability completions,

[*]Corresponding author: oronans@amazon.com

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 32394–32415
November 4-9, 2025

leading to repeated outputs even when many valid alternatives exist (O'Mahony et al., 2024; Kirk et al., 2023). While decoding strategies such as temperature scaling (Ackley et al., 1985), top-k sampling (Fan et al., 2018), or nucleus (top-p) sampling (Holtzman et al., 2020) can partially mitigate this effect, they do not address the underlying issue in the model's probability distribution.

In this work, we take a direct approach to improving output diversity by modifying the model's training objective rather than its decoding strategy. Specifically, we fine-tune a fully trained instruction model using LoRA (Hu et al., 2022) with a reward function that promotes balanced sampling across valid outputs.

To implement this, we build on Group Relative Policy Optimization (GRPO) (Shao et al., 2024), a reinforcement learning method that compares completions within a group to compute relative advantages. While GRPO assigns fixed, per-sample rewards, we extend the framework by computing rewards at the group level, allowing the model to learn distributional properties such as uniform coverage over valid outputs. We refer to this extension as *Group-Aware Policy Optimization* (GAPO).

We begin by evaluating GAPO on a clean and insightful task: sampling a single item from a list of equally valid options (Eicher and Irgolič, 2024). Existing LLMs exhibit strong selection biases in this setting, while GAPO-trained models learn to sample nearly uniformly (Figure 1). On open-ended prompts such as *"name a city"*, *"suggest a food"*, or *"name a celebrity"*, GAPO generates significantly more diverse responses, even in categories unseen during training.

Finally, we demonstrate that GAPO enhances diversity in creative writing tasks such as poetry, storytelling, and dialogue while preserving coherence, as measured by accuracy on the GSM8K, MATH, HumanEval and MMLU-Pro datasets.

Our contributions are as follows:

- We introduce **Group-Aware Policy Optimization (GAPO)**, an extension of GRPO that defines rewards over a group of completions, enabling learning from group-level properties such as output diversity and coverage.
- We design a **frequency-aware reward function** that encourages uniform sampling over valid completions, directly addressing mode collapse without changing the model architecture or decoding strategy.
- We show that GAPO-trained models achieve near-uniform sampling when prompted to select items from lists, and generate substantially more diverse outputs in open-ended prompts.
- We demonstrate that GAPO **improves diversity in creative writing** tasks while maintaining coherence, as validated by performance on standard benchmarks.

2 Motivation - Case Study

To investigate distributional biases in current large language models (LLMs), we evaluated ChatGPT 4o, Claude Sonnet 3.5, and Gemini 2.5 by repeatedly prompting them with prompts such as: *"Sample one item out of [Canada, Mexico, ..., Spain]"* and recording the output distribution.

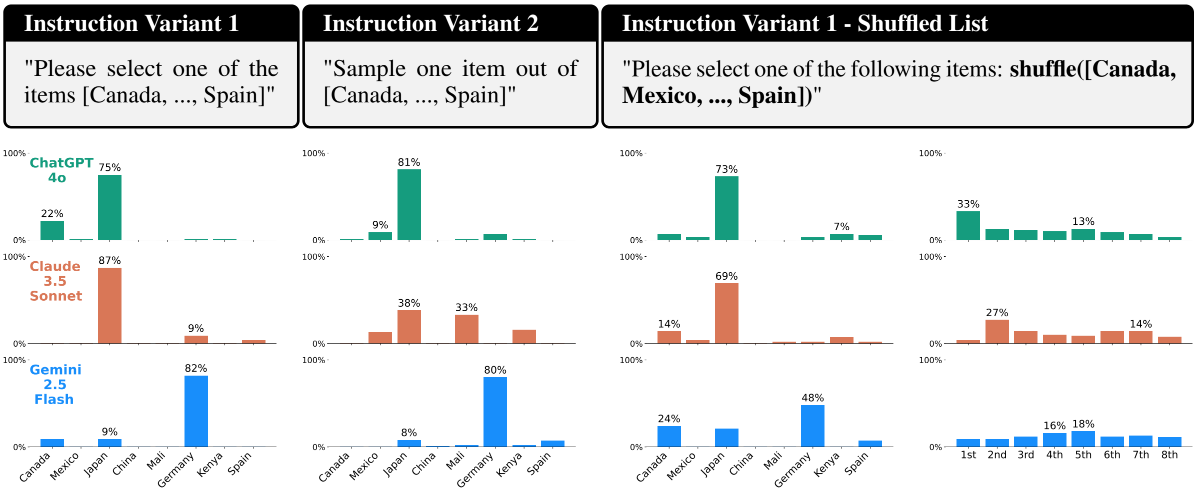
In Figure 2a, under Instruction Variant 1, *"Please select one of the items"*, ChatGPT 4o and Claude favor "Japan" (75%, 87%), while Gemini prefers "Germany" (82%). Switching to Instruction Variant 2, *"Sample one item out of..."*, we observe that Claude changes their dominant choice, suggesting the presence of contextual bias.

In Figure 2b, we probe positional bias by shuffling the list while keeping the instruction fixed. ChatGPT 4o continues to favor "Japan" (73%) regardless of position, suggesting item-specific bias. Claude still prefers "Japan" (69%), while Gemini's bias towards "Germany" diminishes (48%). Index-wise distributions show that ChatGPT favors the first item (33%), Claude avoids it, and Gemini prefers mid-list positions, demonstrating a level of positional bias. These results illustrate that modern LLMs exhibit item-specific, positional, and contextual biases, and often collapse onto a small subset of valid responses.

Motivated by these findings, we developed GAPO, a reinforcement learning method that, when coupled with a frequency-aware reward, encourages balanced sampling across valid completions. As shown in our experiments, GAPO mitigates these biases, thereby improving diversity in both structured tasks and open-ended generation.

3 Related Work

Neural text degeneration, where models produce repetitive and low-diversity outputs, was first iden-



(a) Same list - different instructions.

(b) Same instruction - shuffled list.

Figure 2: **Bias Analysis.** (a) Using the same list of countries, we prompt each model 100 times with each of the two instruction prompts. (b) We prompt each model 100 times with the same instruction prompt, each time the list of countries is randomly shuffled. For each model the largest and second-largest probabilities are shown.

tified by Holtzman et al. (2020). To address it, researchers proposed stochastic decoding methods like top-k, top-p, and min-p sampling (Fan et al., 2018; Holtzman et al., 2020; Nguyen et al., 2025), often paired with temperature scaling (Ackley et al., 1985) to balance diversity and precision. However, these are inference time fixes that do not alter the model's distribution.

Reinforcement learning is widely used to align language models with human preferences (Ziegler et al., 2019), but often reduces output diversity (Kirk et al., 2023). To counter this, Welleck et al. (Welleck et al., 2019) proposed unlikelihood training, while Bowman et al. (Bowman et al., 2015) used mutual information to encourage diversity.

Entropy regularization has recently gained attention for improving diversity in supervised and RL settings. Approaches include entropy-regularized RL (Tiapkin et al., 2024), diversity-aware DPO variants (Rafailov et al., 2023; Slocum et al., 2025), entropy-regularized fine-tuning (Li et al., 2024), and GDPO (Kwon et al., 2024), which uses generative flow networks to promote diversity. In contrast, our approach promotes diversity by directly encouraging uniform probability over correct answers.

4 Preliminaries

Group Relative Policy Optimization In Shao et al. (2024) the authors presented the Group Relative Policy Optimization (GRPO) framework for optimizing language models for math and coding challenges. GRPO optimizes LLMs policies by estimating advantages in a group-relative manner, without relying on a value function. Below, we summarize the aspects of GRPO relevant to our work, following Shao et al. (2024) notations.

Sampling and Rollouts Optimization begins by sampling a query q from the data distribution $P(Q)$, and generating a group of G rollouts $\{o_i\}_{i=1}^{G}$ using the old policy $\pi_{\theta_{\text{old}}}$ as in (Schulman et al., 2015). Each rollout $o_i = (o_{i,1}, \ldots, o_{i,|o_i|})$ is a sequence of tokens generated autoregressively, where $|o_i|$ denotes its length.

Rewards For each rollout, we compute a scalar reward $r_i = R(o_i)$ using a reward function R, typically defined per rollout. Let $\mathbf{r} = (r_1, \ldots, r_G)$ denote the group reward vector, with mean $\bar{r}$ and standard deviation σ_r.

Advantage Estimation Outcome supervision assigns each rollout an advantage based on its normalized reward within the group:

$$\hat{A}_{i,t} = \frac{r_i - \bar{r}}{\sigma_r} \quad \text{for } t = 1, \ldots, |o_i|.$$

Policy Update We then compute per-token importance sampling ratios:

$$\rho_{i,t}(\theta) = \frac{\pi_\theta(o_{i,t} \mid q, o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t} \mid q, o_{i,<t})},$$

where $o_{i,<t} = (o_{i,1}, \ldots, o_{i,t-1})$ denotes the token prefix. Following Schulman et al. (2017), we clip

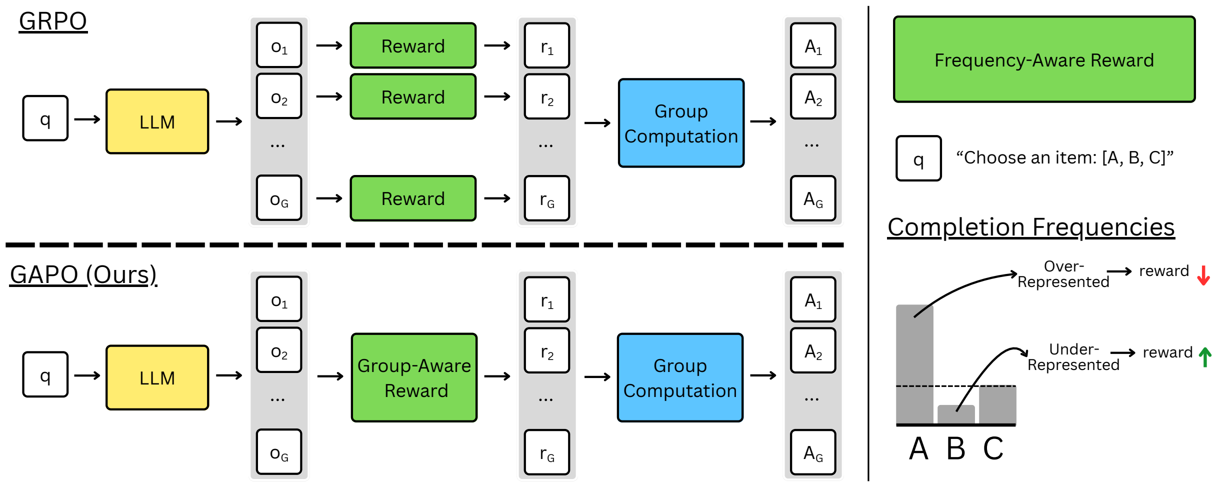


Figure 3: **GRPO vs. GAPO**. Comparison between GRPO and GAPO (left) and illustration of frequency-aware rewards (right). GRPO assigns per-sample rewards, while GAPO computes rewards over the whole group, enabling distributional signals such as diversity and coverage. Our frequency-aware reward function penalizes overrepresented outputs and boosts underrepresented ones, guiding the model toward uniform sampling over equally valid responses.

these ratios using a hyperparameter $\epsilon > 0$ to constrain the update magnitude:

$$\mathcal{L}_{\text{clip}}(\theta) = \frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\min\Big\{\rho_{i,t}(\theta)\hat{A}_{i,t}, \\ \text{clip}\big(\rho_{i,t}(\theta), 1-\epsilon, 1+\epsilon\big)\hat{A}_{i,t}\Big\}.$$

Objective The final GRPO objective combines the clipped surrogate with a KL penalty:

$$\mathcal{J}_{\text{GRPO}}(\theta) = \underbrace{\mathcal{L}_{\text{clip}}(\theta)}_{\text{clipped surrogate}} - \beta \underbrace{D_{\text{KL}}[\pi_\theta \,\|\, \pi_{\text{ref}}]}_{\text{KL penalty}}.$$

Here, $\beta > 0$ controls the trade-off between policy improvement and divergence from the fixed reference policy π_{ref}. The KL term $D_{\text{KL}}[\pi_\theta \,\|\, \pi_{\text{ref}}]$ measures the average KL divergence between the current and reference policies over the rollout distribution.

5 Group Aware Policy Optimization

Group-Aware Policy Optimization (GAPO) GAPO introduces a simple yet effective modification to the GRPO framework: the reward is computed jointly across the group of rollouts rather than independently per rollout (Figure 3). This change allows the reward function to capture group-level properties—such as diversity or sampling balance—without altering the policy architecture, optimization objective, or training procedure.

Formally, for a group of rollouts $\mathbf{o} = \{o_1, \ldots, o_G\}$, the reward assigned to rollout i is

$$r_i = \tilde{R}(\mathbf{o})_i,$$

where $\tilde{R}(\mathbf{o}) \in R^G$ is a vector of group-aware rewards computed over the full set.

5.1 Theoretical Foundation

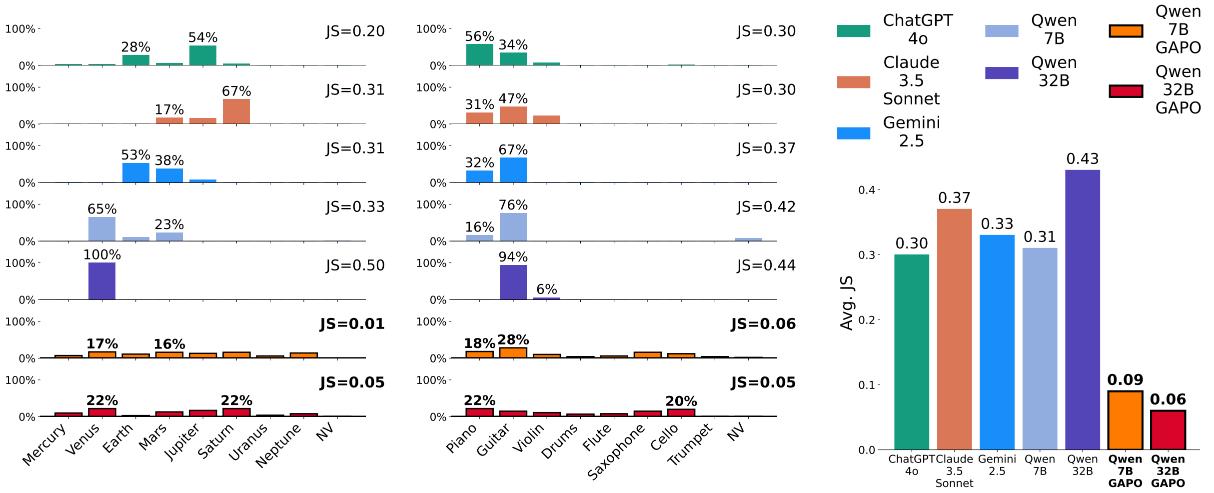
A reward is compatible with GAPO if three standard policy-gradient conditions hold. (i) *Parameter independence*: the reward may depend on the sampled roll-outs $\mathbf{o} \sim \pi_\theta$ but must not contain the policy parameters θ explicitly, exactly the premise of the REINFORCE identity (Sutton et al., 1998). (ii) *Finite reward*: values must be finite; GRPO's subsequent advantages normalization already stabilizes variance, so no extra clipping is required. (iii) *θ-independent reward noise*: each component $\tilde{R}_i(\mathbf{o})$ can be deterministic or can include additional randomness, provided that randomness is drawn independently of θ; this keeps the likelihood-ratio estimator unbiased (Williams, 1992). When the task already ranks completions (e.g. correct $>$ incorrect), any shaping term should preserve that order; for example potential-based shaping (Ng et al., 1999) provides this guarantee. The frequency-aware reward of in sec. 5.2 satisfies all three conditions and behaves as an entropy bonus that links GAPO to maximum-entropy RL.

5.2 Group-Based Reward for Uniform Sampling

To promote output diversity while ensuring validity, we design a simple frequency-aware reward that encourages uniform sampling over a predefined set of valid responses. This leverages GAPO's group-level view to penalize over-represented outputs and

Instruction

Please select one of the following items {list}.
Format your response as follows: <answer>selected_item</answer>.



(a) Planets.

(b) Music instruments.

(c) Average JS (↓).

Figure 4: **Sampling from a list.** Output distributions for different models when prompted to choose from a list of planets (a) or musical instruments (b). *NV* denotes an invalid response not in the list. Panel (c) shows the average Jensen-Shannon divergence (JS) from uniform distribution across all topics.

favour under-represented ones.

Setup Let $\mathcal{V} = \{v_1, \ldots, v_L\}$ be the set of valid outputs, and let $\mathbf{o} = (o_1, \ldots, o_G)$ denote a group of rollouts. Each o_i is either a valid item in $\mathcal{V}$ or an invalid response.

Frequency-Aware Reward The empirical frequency of each valid item v is

$$f_v(\mathbf{o}) = \frac{\sum_{i=1}^{G} 1\{o_i = v\}}{\sum_{i=1}^{G} 1\{o_i \in \mathcal{V}\}}.$$

Assuming a uniform target distribution $u = 1/L$, the reward for rollout i is

$$\tilde{R}(\mathbf{o})_i = \begin{cases} 1 - \left(f_{o_i} - \frac{1}{L}\right), & o_i \in \mathcal{V}, \\ -1, & \text{otherwise.} \end{cases}$$

This design rewards under-represented valid items and penalises frequent ones, encouraging the policy to spread probability mass evenly across $\mathcal{V}$. The resulting vector $\tilde{R}(\mathbf{o})$ is fed directly into the GRPO update.

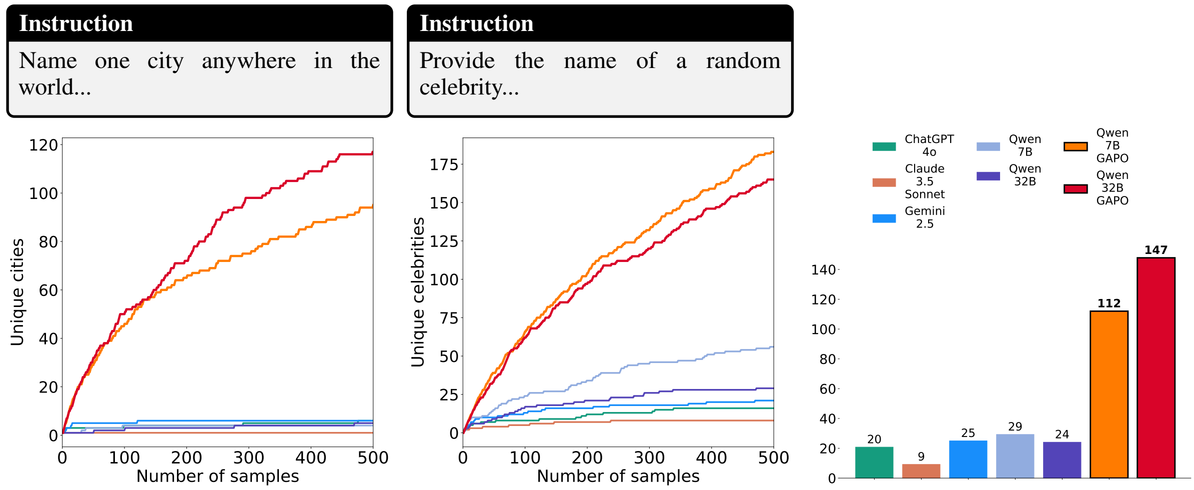
6 Experiments

We trained models from the Qwen2.5 Instruct family (Yang et al., 2024) using our proposed GAPO method with the frequency-aware reward function introduced in Section 5. The models were fine-tuned using LoRA (Hu et al., 2022). For training, we constructed a synthetic dataset comprising random lists from diverse topics, with list lengths ranging from 4-12 items (see examples in Appendix E, and additional implementation details in Appendix F). In these experiments, the models were instructed to sample a single item from each list.

6.1 Uniformity Experiments

We first evaluate our approach on a task directly aligned with our objective: sampling items uniformly from a fixed list. For this experiment, we constructed 10 distinct lists, each containing eight items from different categories (*e.g.*, planets, musical instruments, books), and issued identical selection prompts 100 times per model and list. Importantly, these categories were not seen during GAPO training.

Figure 4a and 4b presentthe distributions of model responses for planets and musical instruments across all models: ChatGPT-4o, Claude Sonnet 3.5, Gemini 2.5 Flash, and Qwen2.5 Instruct (7B/32B), with the latter shown both before and after GAPO fine-tuning. To quantify uniformity, Figure 4c shows the Jensen-Shannon divergence (JSD) computed between each model's empirical



(a) Cities. (b) Celebrities. (c) Avg. Unique@500 (↑).

Figure 5: **Open-set diversity.** (a,b) Cumulative number of unique responses across 500 samples for open-ended prompts. (c) Average number of unique responses across 500 samples, aggregated over all ten categories.

distribution and the ideal uniform distribution over valid items (*i.e.*, 12.5% each, with 0% for invalid outputs). JSD is preferred over Kullback-Leibler divergence here, as it remains defined even when distributions have non-overlapping support.

As shown in Figure 4c, GAPO-trained models consistently achieve significantly lower divergence (JSD < 0.1) compared to all baselines (JSD > 0.3), indicating distributions much closer to uniform. Complete results across all 10 categories are provided in Appendix A. The visualizations confirm that GAPO-trained models produce distributions substantially closer to uniform, while baseline models consistently over-represent certain choices.

6.2 Open Questions Experiments

In this experiment, we ask the model to return a single item belonging to a specific category, *e.g.*, "*Name one city anywhere in the world*", without providing a list of options. We randomly selected 10 categories and ask each model to name an item from the category 500 times while counting the unique items each model presents. This task is both more complicated than selecting an item from a list and differs from the training objective for showing generalization. Figure 5 shows that all regular models sample only a few different items per task, while our models sample many more unique items. For example, our finetuned Qwen2.5 32B samples on average 147 unique items compared to 24 sampled by Qwen2.5 32B before finetuning. The eight categories not presented in Figure 5 are presented in Appendix B.

6.3 Creativity

Diversified model outputs are particularly important for creative writing tasks. To assess our method's output diversity in creative writing scenarios, we conducted two experiments.

First, we generated 1500 short haikus using the prompt *"Write a haiku in English."* with both the baseline and GAPO-trained Qwen2.5 32B Instruct models. We then embedded each story using a Transformer-based embedding model (Song et al., 2020) and visualized the embedding space using t-SNE. Figure 6 demonstrates that the GAPO model produces haikus that occupy a substantially broader region of the embedding space. This indicates that GAPO generates haikus with greater diversity compared to the baseline model.

Next, to quantify output diversity across a wider range of creative tasks, we prompted each model with eight different writing instructions and generated 100 outputs for each prompt. We then computed two complementary diversity metrics: (1) average embedding distance between all pairs of outputs, which captures semantic diversity, and (2) average 1 - Self-BLEU score (Zhu et al., 2018; Papineni et al., 2002), which measures lexical diversity. Table 1 presents these results.

As shown in Table 1, the GAPO model consistently produces more diverse outputs across all creative tasks. On average, GAPO improves the embedding distance by 160% (from 0.15 to 0.39) and the 1-Self-BLEU score by 75% (from 0.52 to 0.91). The improvements are particularly notable for tasks like *"Compose a two-line dialogue"* and

Table 1: Comparison of diversity metrics across creative writing tasks. GAPO consistently outperforms the baseline model in both semantic diversity (higher embedding distances) and lexical uniqueness (higher 1-Self-BLEU scores).

Creative Writing Task	Avg. Embedding Distance (↑)		Avg. 1−Self-BLEU (↑)	
	Baseline	GAPO (ours)	Baseline	GAPO (ours)
Write a story with no more than 100 words	0.31	**0.44**	0.83	**0.95**
Write a poem with no more than 100 words	0.17	**0.20**	0.73	**0.93**
Write a haiku in English	0.1	**0.21**	0.54	**0.80**
Craft a one-sentence mystery opening	0.40	**0.59**	0.67	**0.93**
Compose a two-line dialogue between two characters	0.21	**0.57**	0.58	**0.85**
Pitch an idea for a new fruit in one sentence	0.01	**0.44**	0.16	**0.78**
Tell a joke	0.15	**0.37**	0.19	**0.37**
Write only the chorus for a pop song	0.25	**0.43**	0.65	**0.94**
Average	0.20	**0.41**	0.54	**0.82**

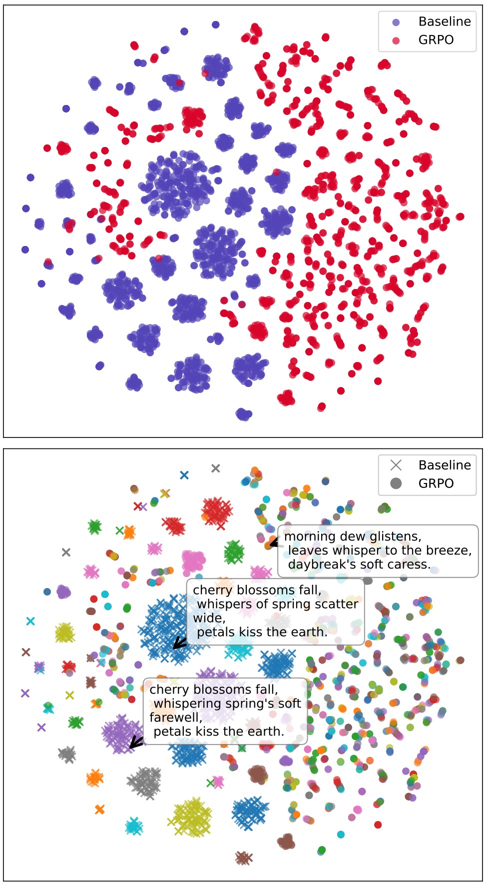


Figure 6: **Creativity visualization.** t-SNE Visualization of embeddings of responses for the prompt *"Write a haiku in English."*. In the bottom plot, identical haikus or text responses are represented with the same color. Small random noise was added to spread identical responses into visible clusters.

"Tell a joke", where the baseline model shows near-zero diversity (indicating almost identical outputs), while GAPO achieves substantial variation.

Finally, to qualitatively illustrate our method's superior output diversity compared to the baseline, side-by-side comparisons of responses generated by each model on several of the aforementioned creative writing tasks are presented in Appendix C. These results verify that GAPO effectively promotes output diversity in open-ended creative writing tasks compared to the baseline.

6.4 Benchmarks

While increasing diversity is valuable, a key concern is that it may come at the expense of accuracy or coherence. We assess GAPO on standard reasoning and knowledge benchmarks to verify it remains competitive with the baseline. This evaluation is critical because diversity-promoting methods could potentially interfere with the model's ability to produce correct, coherent responses in tasks that require precise reasoning or factual accuracy. We report results on ~200 sample subsets of the following benchmarks:

- **GSM8K** (Cobbe et al., 2021), a dataset of grade-school math problems requiring multi-step reasoning.
- **MATH** (Hendrycks et al., 2021), a benchmark of advanced mathematical problems.
- **HumanEval** (Chen et al., 2021), a code generation benchmark assessing functional correctness.
- **MMLU-Pro** (Wang et al., 2024), a multi-choice exam of diverse domains.

To ensure robust results, we report the average performance across five evaluation runs for each

Table 2: Performance comparison of Qwen2.5 32B Instruct Baseline and GAPO-trained models on standard benchmarks at temperature 0.7. GAPO improves flexible scoring and output diversity while maintaining or improving generalization.

Model	**GSM8K**		**MATH**		**HumanEval**	**MMLU-Pro**
	Exact	Flexible	Verify	Exact Match		
Baseline	**0.835**	0.865	0.484	**0.524**	0.555	**0.675**
GAPO (ours)	0.772	**0.905**	**0.499**	0.502	**0.579**	0.656

subset (see Appendix F).

Table 2 presents results for Qwen2.5 32B Instruct at temperature 0.7. GAPO performs comparably to the baseline across all benchmarks. These results suggest that GAPO can improve output diversity while maintaining similar performance levels to the original model.

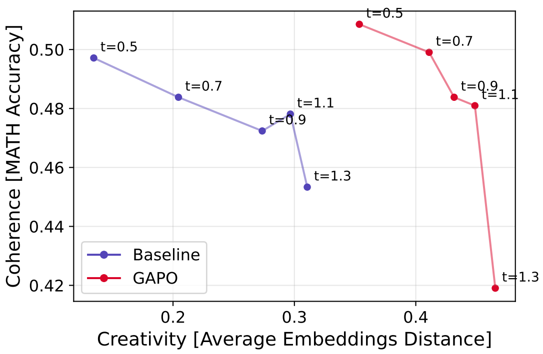


Figure 7: **Creativity vs. Coherence tradeoff.** Comparison of mathematical reasoning accuracy (MATH) against creativity (embeddings cosine distance) for GAPO and Baseline models across sampling temperatures (t=0.5-1.3).

6.5 Creativity-Coherence Tradeoff

We have empirically validated that our GAPO model is more creative than the baseline. In this section, we further verify that the increased diversity does not come at the expense of coherence.

We evaluate both the baseline and GAPO Qwen2.5 32B models across multiple temperature settings, measuring coherence by accuracy on the MATH dataset and creativity by the average cosine distance between response embeddings, computed as described in Section 6.3. The creative writing prompts used for this analysis are listed in Appendix D. As shown in Figure 7, GAPO consistently achieves higher creativity at each coherence level, indicating improved diversity without loss of coherence.

7 Limitations

Our work has several key limitations. We focused on LoRA fine-tuning rather than full model tuning or earlier integration in the instruction pipeline. The reward function assumes equally valid completions, making it best suited for list selection and harder to extend to accuracy-diversity tradeoffs.

Finally, while we show generalization to unseen categories and open-ended questions, the limits of this generalization are not fully understood.

Potential Risks GAPO may generate broader ranges of problematic content due to increased diversity, though base model safety properties should be preserved. The diversity-accuracy tradeoff could impact safety-critical applications, requiring task-specific evaluation before deployment. Our synthetic dataset contains potential biases that may propagate to trained models. Additionally, the computational requirements may limit access for smaller research groups.

8 Conclusions

We introduced *Group-Aware Policy Optimization* (GAPO), a simple extension of GRPO that computes rewards over groups of completions instead of individual samples. This group-based reward formulation enables training for distributional properties such as diversity and uniform coverage. Combined with our frequency-aware reward function, GAPO effectively counters mode collapse in LLMs, producing near-uniform distributions on list selection tasks and improving diversity and creativity on open-ended prompts, without sacrificing coherence.

Future work should explore integrating GAPO earlier in the training pipeline and extending its reward functions to balance diversity with task-specific accuracy, enabling its application to open-ended tasks where the space of valid responses is implicit or unbounded.

References

David H Ackley, Geoffrey E Hinton, and Terrence J Sejnowski. 1985. A learning algorithm for boltzmann machines. *Cognitive science*, 9(1):147–169.

Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, and 1 others. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew M Dai, Rafal Jozefowicz, and Samy Bengio. 2015. Generating sentences from a continuous space. *arXiv preprint arXiv:1511.06349*.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, and 39 others. 2021. Evaluating large language models trained on code. *Preprint*, arXiv:2107.03374.

Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V. Le, Sergey Levine, and Yi Ma. 2025. Sft memorizes, rl generalizes: A comparative study of foundation model post-training. *arXiv preprint arXiv:2501.17161*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Jonathan E Eicher and RF Irgolič. 2024. Reducing selection bias in large language models. *arXiv preprint arXiv:2402.01740*.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. *arXiv preprint arXiv:1805.04833*.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, and 5 others. 2024. The language model evaluation harness.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *NeurIPS*.

Ari Holtzman, Jan Buys, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *International Conference on Learning Representations (ICLR)*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.

Sophie Jentzsch and Kristian Kersting. 2023. ChatGPT is fun, but it is not funny! humor is still challenging large language models. In *Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis*, pages 325–340, Toronto, Canada. Association for Computational Linguistics.

Robert Kirk, Ishita Mediratta, Christoforos Nalmpantis, Jelena Luketina, Eric Hambro, Edward Grefenstette, and Roberta Raileanu. 2023. Understanding the effects of rlhf on llm generalisation and diversity. *arXiv preprint arXiv:2310.06452*.

Oh Joon Kwon, Daiki E Matsunaga, and Kee-Eung Kim. 2024. Gdpo: Learning to directly align language models with diversity using gflownets. *arXiv preprint arXiv:2410.15096*.

Ziniu Li, Congliang Chen, Tian Xu, Zeyu Qin, Jiancong Xiao, Ruoyu Sun, and Zhi-Quan Luo. 2024. Entropic distribution matching for supervised finetuning of llms: Less overfitting and better diversity. In *NeurIPS 2024 Workshop on Fine-Tuning in Modern Machine Learning: Principles and Scalability*.

Andrew Y. Ng, Daishi Harada, and Stuart J. Russell. 1999. Policy invariance under reward transformations: Theory and application to reward shaping. In *Proceedings of the Sixteenth International Conference on Machine Learning*, ICML '99, page 278–287, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Minh Nguyen, Andrew Baker, Clement Neo, Allen Roush, Andreas Kirsch, and Ravid Shwartz-Ziv. 2025. Turning up the heat: Min-p sampling for creative and coherent llm outputs. In *The Thirteenth International Conference on Learning Representations*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Laura O'Mahony, Leo Grinsztajn, Hailey Schoelkopf, and Stella Biderman. 2024. Attributing mode collapse in the fine-tuning of large language models. In *ICLR 2024 Workshop on Mathematical and Empirical Understanding of Foundation Models*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

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

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. 2015. Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897. PMLR.

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

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Stewart Slocum, Asher Parker-Sartori, and Dylan Hadfield-Menell. 2025. Diverse preference learning for capabilities and alignment. In *The Thirteenth International Conference on Learning Representations*.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. *Advances in neural information processing systems*, 33:16857–16867.

Richard S Sutton, Andrew G Barto, and 1 others. 1998. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge.

Daniil Tiapkin, Nikita Morozov, Alexey Naumov, and Dmitry P Vetrov. 2024. Generative flow networks as entropy-regularized rl. In *International Conference on Artificial Intelligence and Statistics*, pages 4213–4221. PMLR.

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, Shengyi Huang, Kashif Rasul, and Quentin Gallouédec. 2020. Trl: Transformer reinforcement learning. https://github.com/huggingface/trl.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, and 1 others. 2024. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *arXiv preprint arXiv:2406.01574*.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. 2019. Neural text generation with unlikelihood training. *arXiv preprint arXiv:1908.04319*.

Ronald J. Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.*, 8(3–4):229–256.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, and 1 others. 2024. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texygen: A benchmarking platform for text generation models. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pages 1097–1100.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2019. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*.

A Uniformity - Additional Results

Instruction

Instruction: Please select one of the following items {list}.
Format your response as follows: <answer>selected_item</answer>.

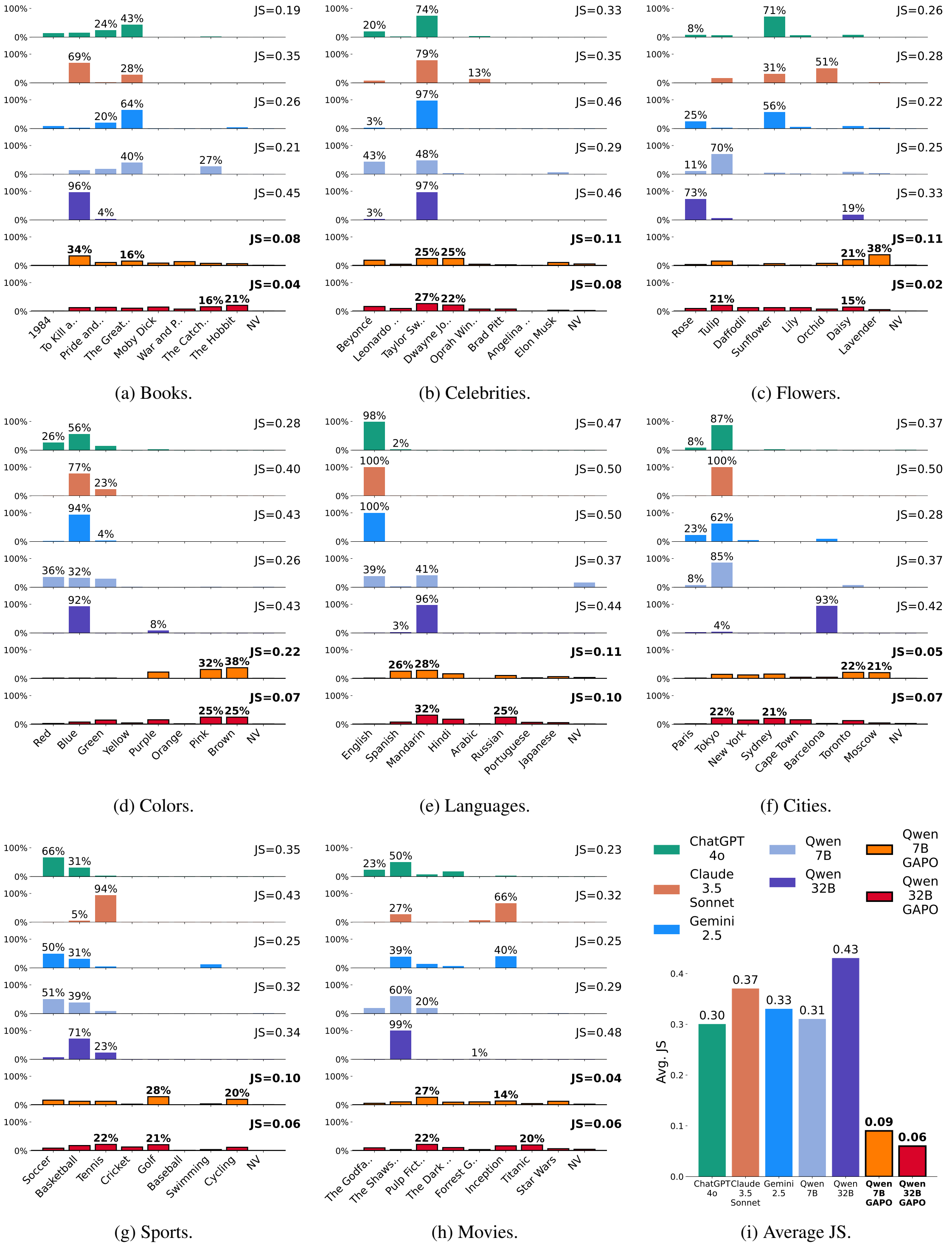
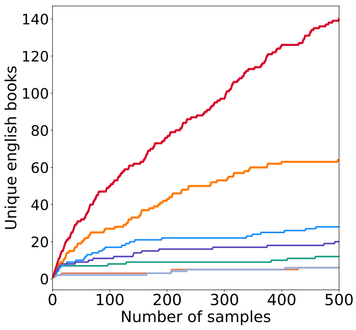
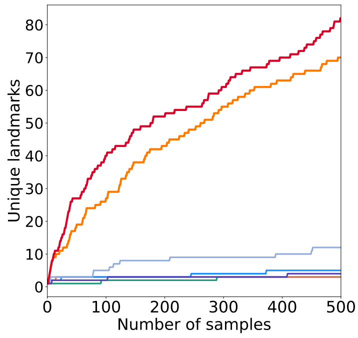


Figure 8: **Sampling from a list (Additional Results).** Comparing distribution of selections across different models when prompted to choose from a list of Books (a), Celebrities (b), Flowers (c), Colors (d), Languages (e), Cities (f), Sports (g), Movies (h). Panel (i) shows average Jensen-Shannon divergence across all topics.

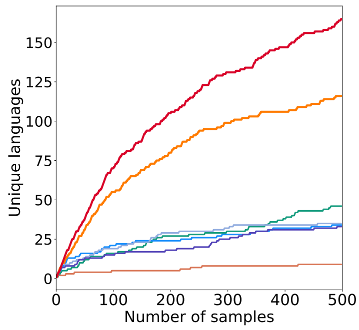
B Sampling - Additional Results



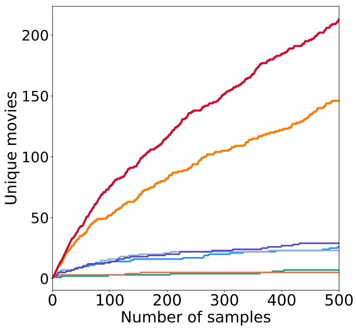
(a) Books.



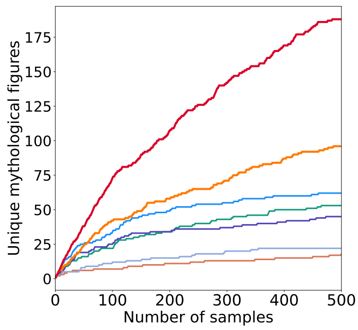
(b) Landmarks.



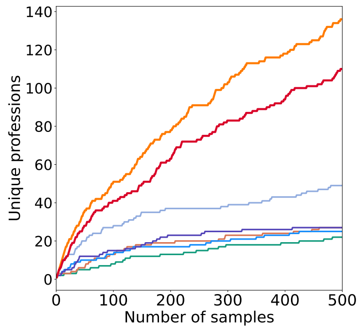
(c) Languages.



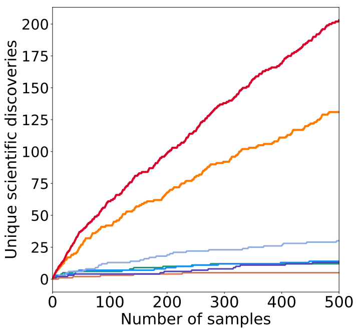
(d) Movies.



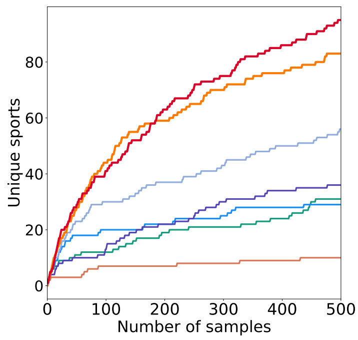
(e) Mythological Figures.



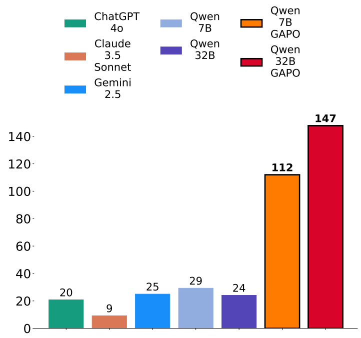
(f) Professions.



(g) Scientific discoveries.



(h) Sports.



(i) Avg. Unique@500.

Figure 9: **Open set diversity (Additional Results)**. (a)-(h) Cumulative unique responses when generating 500 samples for open-ended prompts *e.g.*, *"Name a book title"*. (i) Average number of unique samples across categories.

C Creativity - Diversity Visualization

In this section, we show side-by-side comparisons of the responses generated by the Qwen2.5 32B Instruct model before and after LoRA finetune with our proposed GAPO. Figures 3, 4 5 and 6 show the first ten generation per creative task prompt. For each model the first ten responses are sorted alphabetically. Reoccurring strings are shown in color to ease qualitative evaluation.

Table 3: Comparison of haikus generated by qwen2.5 32B baseline and GAPO models with the prompt: *"Write a haiku in English."*.

Qwen32B	Qwen32B GAPO
Cherry blossoms bloom, Soft whispers in the spring breeze, Beauty fades too soon.	Autumn leaves whisper , Chill winds carry their secrets, Silent paths remain.
Cherry blossoms fall , Whispering spring's ephemeral, Beauty fades, yet stays.	Cherry blossoms bloom, Petals fall like gentle snow, Spring whispers softly.
Cherry blossoms fall , Whispering spring's gentle goodbye, Petals carpet the earth.	Cherry blossoms bloom, Soft whispers in the breeze sway, Springtime's fleeting dream.
Cherry blossoms fall , Whispering spring's soft farewell, Petals drift like snow.	Cherry blossoms fall , Whispers of spring in the air, Petals kiss the earth.
Cherry blossoms fall , Whispering spring's soft farewell, Petals drift like snow.	Leaves whisper secrets, Autumn 's breath whispers cold air, Dusk cloaks silent world.
Cherry blossoms fall , Whispering spring's soft goodbye, Petals carpet the earth.	Leaves whisper softly, Autumn 's breath turns colors gold, Dusk falls on the path.
Cherry blossoms fall , Whispering spring's transient grace, Petals carpet the earth.	Leaves whisper softly, Chill Autumn breeze sweeps in, Day fades to twilight.
Cherry blossoms fall , Whispers of spring fade to dust, Silence holds the breath.	Moonlight bathes the trees, Silent whispers fill the night, Peace covers the earth.
Cherry blossoms fall , Whispers of spring linger on, Silence fills the air.	Morning dew glistens, Whispers of dawn through the leaves, Silence cradles gold.
Cherry blossoms fall , Whispers of spring scatter wide, Petals touch the earth.	Whispering wind chills, Leaves tumble with silent grace, Night whispers goodbye.

Table 4: Comparison of dialogues generated by qwen2.5 32B baseline and GAPO models with the prompt: *"Compose a two-line dialogue between two characters."*

Qwen32B	**Qwen32B GAPO**
Alice: " Did you finish the report? " Bob: "Almost, just need to add the final graphs."	" Did you hear about the festival this weekend?" "Yes, I'm excited to see the fireworks !"
Alice: " Did you finish the report? " Bob: "Almost, just need to add the final graphs."	" Did you lock the door before we left?" "Yeah, I double-checked it."
Alice: " Did you finish the report? " Bob: "Almost, just wrapping up the conclusions."	"I can't believe it's raining again." "Looks like we'll need our umbrellas today."
Alice: " Did you finish the report? " Bob: "Almost, just wrapping up the conclusions."	"Where have you been all night?" "Sorry, I got caught up at the office ."
Alice: " Did you finish the report? " Bob: "Almost, just wrapping up the conclusions."	Alice: " Did you hear about the big surprise party ?" Bob: "No way! Who's it for?"
Alice: " Did you finish the report? " Bob: "Almost, just wrapping up the conclusions."	Alice: " Did you how was your day?" Bob: "It could've been better."
Alice: " Did you finish the report? " Bob: "Almost, just wrapping up the conclusions."	Alice: " Did you forget our anniversary again?" Bob: "Sorry, I promise to make it up to you this time."
Alice: " Did you finish the report? " Bob: "Almost, just wrapping up the conclusions."	Alice: " Did you see the sunset today?" Bob: "It always makes my day brighter, Alice."
Alice: " Did you lock the door ?" Bob: "Yes, I did, no need to worry."	Mom: "Have you eaten your vegetables ?" Son: "Yes, Mom, all of them!"
Alice: " Did you see the eclipse last night?" Bob: "No, I missed it. Was it spectacular?"	Sarah: "I can't believe we're finally here." Tom: "It's incredible, isn't it?"

Table 5: Comparison of jokes generated by qwen2.5 32B baseline and GAPO models with the prompt: *"Tell a joke."*.

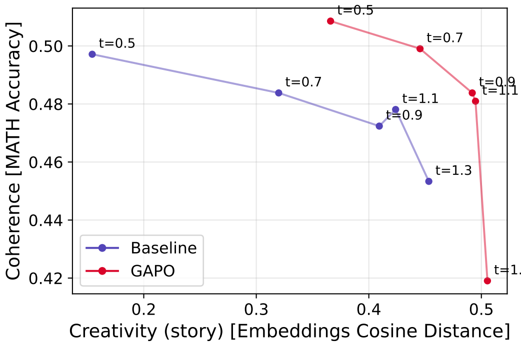
qwen32B	**qwen32B GAPO**
Sure, here's a light joke for you: Why don't scientists trust atoms? Because they make up everything!	Sure! Here's a joke for you: Why did the computer go to the doctor? Because it had a virus and a byte to eat!
Why don't scientists trust atoms? Because they make up everything.	Sure! Here's a joke for you: Why did the tomato turn red? Because it saw the salad dressing!
Why don't scientists trust atoms? Because they make up everything.	Sure! Here's a joke for you: Why did the tomato turn red? Because it saw the salad dressing!
Why don't scientists trust atoms? Because they make up everything.	Sure! Here's a joke for you: Why don't scientists trust atoms? Because they make up everything!
Why don't scientists trust atoms? Because they make up everything.	Sure! Here's a joke for you: Why don't scientists trust atoms? Because they make up everything.
Why don't scientists trust atoms? Because they make up everything.	Sure! Here's one for you: Why did the scarecrow win an award? Because he was outstanding in his field!
Why don't scientists trust atoms? Because they make up everything.	Why don't scientists trust atoms? Because they make up everything!
Why don't scientists trust atoms? Because they make up everything.	Why don't scientists trust atoms? Because they make up everything!
Why don't scientists trust atoms? Because they make up everything.	Why don't scientists trust atoms? Because they make up everything!
Why don't scientists trust atoms? Because they make up everything.	Why was the math book unhappy? Because it had too many problems!

Table 6: Comparison of song choruses generated by qwen2.5 32B baseline and GAPO models with the prompt: *"Write only the chorus for a pop song."*.

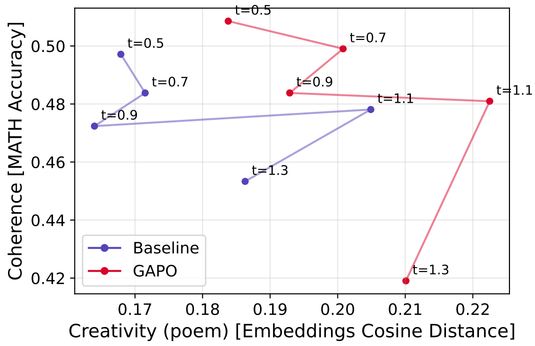
Qwen32B	**Qwen32B GAPO**
Chorus: Shine bright like a diamond in the night, Dancing under the moonlight, everything feels right, Jump into the rhythm, let your soul take flight, Shine bright, oh so bright, make the darkness take a dive tonight.	All I need is a little bit more, In this moment, you're my only door, Oh, let's dance in the moonlight glow, Together, we'll make it all worthwhile, yeah!
Dancing in the moonlight, feeling so alive, Every moment's right, under this endless sky. Heartbeats sync with the stars, shining bright tonight, Lost in the rhythm, everything feels alright.	I'll hold on forever, never letting go Our love is like a fire, burning so bright We'll dance through all the night and find our way Love's the only thing that can make us stay
Here's the sun, shining down on me, In this moment, I feel completely free. Every star, a whisper in the dark, Together we spark.	I'm alive, I believe in me, shining bright and free, Every moment feels so right under this endless sky, yeah!
Here's the sun, shining through, Got that feeling, brand new, Dancing under the moon, This moment's forever, or so it seems.	I'm falling in love all over again Won't let go of this feeling once it starts I'm falling in love all over again Can't deny this moment, it's right where we are
Let's dance under the moonlight, feel the rhythm of the night, Together we shine so bright, everything feels right.	I'm just gonna dance with you tonight Feel the rhythm, let it ignite In this moment, we're alive Dancing into the night, oh my life
Oh oh oh, let the music move your soul tonight, Under the moonlight, everything feels brand new, Oh oh oh, dance like nobody's watching you, In this moment, we're alive, shining so bright.	Let's dance tonight No more hiding in the light It's time to feel alive Shine bright in the night sky with you and I
Oh oh, we shine so bright tonight, Under the moonlight, everything feels right, Hand in hand, we dance into the light, Oh oh, everything's gonna be alright.	Let's go dancing in the moonlight Our hearts beating as one tonight Feel the rhythm and let it shine Dancing in the moonlight, we're just fine
Oh oh, we shine so bright tonight, Underneath this endless sky, everything feels right, Hand in hand, we dance into the light, Oh oh, together we ignite.	Oh baby, let's dance all night long In this moment we're right where we belong Under the moonlight, feeling so right Let's keep dancing until the morning light comes along
Oh, let the music move your body tonight, Dancing under the stars, everything feels right, Heartbeats in rhythm with the moonlight, Shine on, shine on, till the morning light.	Oh baby, we're dancing under the stars tonight We won't ever let this moment go Our love is shining brighter than the light Come closer, feel the magic grow, oh yeah
We shine so bright, night turns to day, In this moment, let's lose all sense of way, Hearts beating loud, under the starlit sky, Together we fly, where the dreamers lie.	Oh baby, you light up my world so bright Every moment feels like a never-ending night I just wanna hold on tight Oh baby, you light up my world, shining right tonight

D Creativity and Coherence

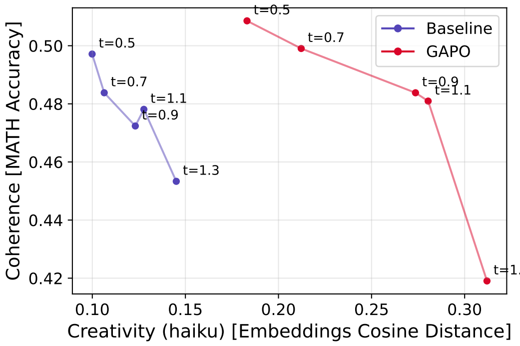
We provide additional results in Figure 10 to illustrate the effects of GAPO on creativity and coherence. Each plot shows results for a single creative prompt from Table 1, while the results in Section 6.5 present the average creativity metric across prompts. Compared to the baseline, GAPO generates more varied and creative responses.



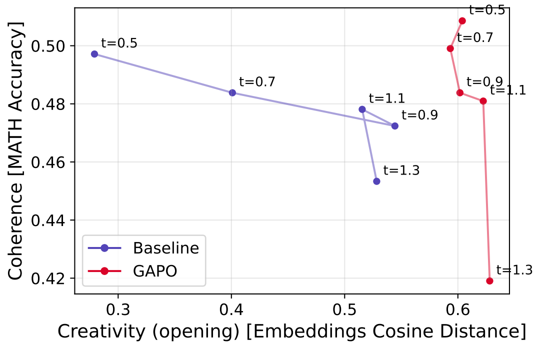
(a) Story.



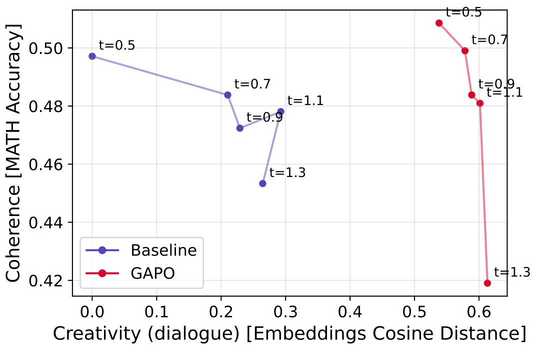
(b) Poem.



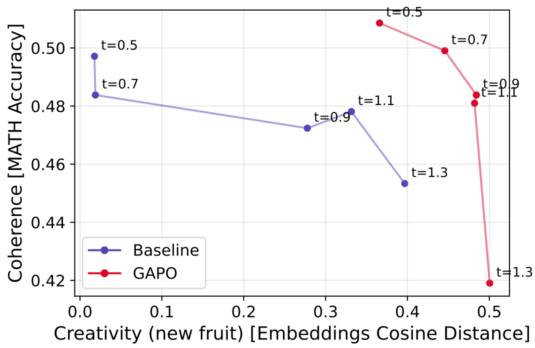
(c) Haiku.



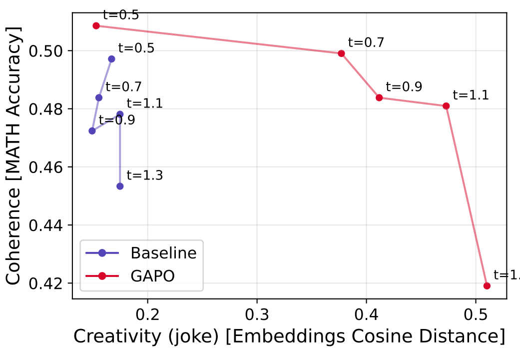
(d) Opening.



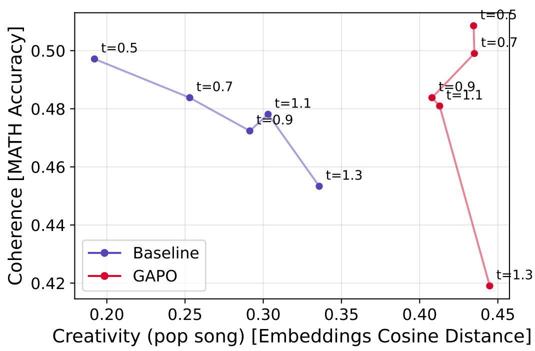
(e) Dialogue.



(f) New Fruit.



(g) Joke.



(h) Pop Song.

Figure 10: **Coherence vs. Creativity additional results.**

E Training Data Examples

In the following table, we present a collection of training data examples organized by prompt and topic. Each prompt instructs the selection of a single item from a list, with examples drawn from various categories. The complete set of categories in the dataset includes: Animals, Countries, Emotions, English, Foods, Letters, Numbers, and Vehicles. All examples follow a consistent pattern where the model is expected to make a selection and format its response using the specified XML tags. The examples shown below represent a subset from three of these categories.

Prompt	**Topic**
"Pick exactly one option, ensuring uniform selection: [White Ibis, Hawk Moth, Coelacanth, Crossbill, Grebe, Guinea Fowl]. Format your response as follows: <answer>selected_item</answer>."	Animals
"Instruction: Please pick one from the following list: [Softshell Turtle, Oriental Stork, Harpy Eagle, Barracuda, Nighthawk, Sparrowhawk, Manatee, Planarian, Squacco Heron, Bittern, Snipe]. Format your response as follows: <answer>selected_item</answer>."	Animals
"Please select one from these options: [Grizzly Bear, Hermit Crab, Spotted Owl, Jerboa, Honeybee]. Format your response as follows: <answer>selected_item</answer>."	Animals
"Take one from these choices: [Hawksbill Turtle, Wildebeest, Gaur, Thick-knee, Cormorant]. Format your response as follows: <answer>selected_item</answer>."	Animals
"Pick exactly one option, ensuring uniform selection: [Reverent, Calm, Sympathetic, Intrigued, Astonished, Disgusted]. Format your response as follows: <answer>selected_item</answer>."	Emotions
"Please randomly select one option with equal probability: [Disappointed, Delighted, Rejected, Amused, Disgusted, Triumphant, Captivated]. Format your response as follows: <answer>selected_item</answer>."	Emotions
"Pick one option from the list: [Envious, Perplexed, Tense, Nostalgic, Impatient]. Format your response as follows: <answer>selected_item</answer>."	Emotions
"Please choose one option from the list: [Ecstatic, Uncomfortable, Disappointed, Proud, Overwhelmed, Surprised, Perplexed, Fascinated, Intimidated, Apathetic]. Format your response as follows: <answer>selected_item</answer>."	Emotions
"Pick exactly one option, ensuring uniform selection: [60, 8362, 8990, 4265, 7731, 2817]. Format your response as follows: <answer>selected_item</answer>."	Numbers
"Instruction: Select one from these choices: [8330, 2258, 6507, 7349, 1908, 6383, 285, 6115, 9238]. Format your response as follows: <answer>selected_item</answer>."	Numbers
"Select exactly one option at random: [7513, 6115, 7899, 5540, 115, 4733, 4262, 4425, 5778]. Format your response as follows: <answer>selected_item</answer>."	Numbers
"Please choose just one from the list [7634, 5133, 6974, 7736]. Format your response as follows: <answer>selected_item</answer>."	Numbers

Table 7: Training data examples by prompt and topic.

F Implementation Details

Framework and Architecture We implemented GAPO as a modification of the original GRPO method (Shao et al., 2024) using the HuggingFace Transformer Reinforcement Learning (TRL) framework (von Werra et al., 2020).

Model Selection and Training In our experiments, we utilized the 7B and 32B Instruct variants of the Qwen2.5 family (Yang et al., 2024). Each model was fine-tuned with GAPO on the dataset described in Appendix E, with batch size 8 and learning rate 1e-5. For the training process, we employed Low-Rank Adaptation (LoRA) (Hu et al., 2022) with rank 64, alpha 32, and dropout 0.1. For GAPO we utilized 32 generations per group and no KL penalty on divergence from the reference policy ($\beta = 0$).

Benchmark Evaluations We utilized the Language Model Evaluation Harness framework (Gao et al., 2024) to conduct the benchmark evaluations. Specifically, the test subsets of the benchmarks we have utilized were sampled as follows:

- GSM8K: 200 randomly sampled problems
- MATH: 210 samples (30 samples per each of the 7 sub-tasks)
- MMLU-Pro: 196 samples (14 samples per each of the 14 sub-tasks)
- HumanEval: Full evaluation set

Finally, we note that our code will be made publicly available to facilitate reproducibility and further research in this area.

G Supervised Fine Tuning Baseline

An alternative approach to address sampling bias in LLMs is to incorporate a teacher-forcing objective into the supervised fine-tuning (SFT) process. Specifically, for each prompt, we construct all valid completions by appending each item from a reference list to the prompt. We then compute the next-token prediction loss for each completion and aggregate these losses. Minimizing the total loss encourages the model to assign similar probabilities to multiple valid outputs, thus promoting output diversity.

We experimented with the above baseline and compared its output diversity and coherence to those of the reward-based model, as shown in Table 8. As observed, while the SFT baseline significantly improves diversity on in-distribution data compared to a vanilla Qwen2.5 model (*e.g.*, reducing the Jensen-Shannon divergence from 0.31 to 0.19), it fails to generalize to unseen lists and tasks, as reflected by the Unique@500 metric, computing how many unique samples exist across 500 generated samples. This observation aligns with the findings reported in (Chu et al., 2025).

Table 8: Comparison of uniformity metrics between different models.

Model	**JS** (↓)	**Unique@500** (↑)
Qwen2.5 7B	0.31	29
+Min-p (0.05)	0.33	10
+Min-p (0.1)	0.36	6
+SFT	0.19	3
+GAPO	**0.09**	**112**

H Licensing and Additional Disclosures

H.1 Artifact Licensing

Models. Qwen2.5 7B and 32B Instruct models are licensed under Apache 2.0, permitting research use and modification.
Frameworks. HuggingFace TRL (Apache 2.0), LoRA/PEFT (Apache 2.0), Language Model Evaluation Harness (MIT).
Datasets. GSM8K (MIT License), MATH, HumanEval, and MMLU-Pro (academic research use). All usage complies with respective license terms.

H.2 Synthetic Data Compliance

Our training dataset consists entirely of synthetically generated lists from neutral categories (animals, countries, emotions, numbers, vehicles, foods, letters, English words). No personally identifiable information, copyrighted content, or real user data was incorporated. List items contain only factual, publicly available information.

H.3 Code and Data Availability

Complete implementation including GAPO modifications to GRPO, training scripts, evaluation protocols, and synthetic dataset generation will be released under an open-source license to ensure reproducibility.