Does iGRPO Need a Good Draft? Best-vs-Worst Self-Conditioning Ablation for RLVR Math

FARS
Analemma
fars@analemma.ai

Abstract

Iterative Group Relative Policy Optimization (iGRPO) improves mathematical reasoning in large language models by conditioning refinement training on self-generated drafts selected by reward. However, it remains unclear whether iGRPO's benefit stems from conditioning on high-quality drafts or from the two-stage structure itself. We design a controlled ablation study comparing three conditions: GRPO baseline, iGRPO with best-of-N draft selection, and iGRPO with worst-of-formatted draft selection (intentionally selecting low-quality but well-formatted drafts). Surprisingly, worst-of-formatted selection not only recovers but *exceeds* best-of-N performance, achieving 64.37% vs 61.94% macro-average accuracy across six math benchmarks. The recovery ratio of 1.34 (95% CI: [1.21, 1.47]) on MATH500 demonstrates that draft quality is not necessary for iGRPO's benefit. Analysis reveals that worst-of-formatted selection produces 50% more gradient-active training groups, potentially explaining its superior performance. These findings suggest that iGRPO can be simplified by removing reward-based draft selection.

WARNING: This paper was generated by an automated research system. The code is publicly available.[1]

1 Introduction

Reinforcement learning has emerged as a powerful paradigm for improving mathematical reasoning in large language models. DeepSeek-R1 (DeepSeek-AI et al., 2025) demonstrated that RL can incentivize sophisticated reasoning behaviors, while Group Relative Policy Optimization (GRPO) (Shao et al., 2024) introduced an efficient approach that eliminates the need for a separate value function by using group-relative advantage estimation. Building on these advances, iGRPO (Hatamizadeh et al., 2026) extends GRPO with a two-stage draft-conditioned training procedure: the model first generates multiple draft solutions, selects the best draft based on reward, and then trains on refinements conditioned on this selected draft. This approach achieves strong gains on math reasoning benchmarks under matched compute budgets.

However, a fundamental question remains unanswered: *does iGRPO's benefit come from conditioning on high-quality drafts, or from the two-stage structure itself?* The original iGRPO method selects the highest-reward draft for conditioning, but it is unclear whether this quality-based selection is necessary. If the benefit primarily comes from exposing the model to an in-context "attempt" (regardless of correctness), then even a low-quality draft should provide similar gains. Conversely, if draft quality is essential, conditioning on incorrect drafts should substantially reduce performance.

To answer this question, we design a controlled ablation study with three conditions: (A) GRPO baseline without draft conditioning, (B) iGRPO with best-of-N draft selection (the original method), and (C) iGRPO with worst-of-formatted draft selection (our ablation variant that intentionally selects low-quality but well-formatted drafts). All conditions use matched compute, enabling a clean comparison of draft selection strategies.

[1] `https://gitlab.com/fars-a/igrpo-best-draft-ablation`

Our results reveal a surprising finding: conditioning on worst-of-formatted drafts not only recovers but *exceeds* the performance of best-of-N selection. iGRPO-worst achieves 64.37% macro-average accuracy compared to iGRPO-best's 61.94% (+2.43 pp), with a recovery ratio of 1.34 (95% CI: [1.21, 1.47]) on MATH500. Analysis reveals that worst-of-formatted selection produces 50% more gradient-active training groups, potentially explaining its superior performance.

Our contributions are threefold: (1) we design a clean ablation study that isolates the role of draft quality in iGRPO by comparing best-of-N vs worst-of-formatted draft selection under matched compute; (2) we demonstrate that draft quality is *not* necessary for iGRPO's benefit, with worst-of-formatted selection achieving a recovery ratio of 1.34 over best-of-N; and (3) we provide mechanistic analysis showing that worst-of-formatted selection increases gradient-active training groups by 50%, offering a potential explanation for its superior performance.

2 Method

2.1 Background: GRPO and iGRPO

Group Relative Policy Optimization (GRPO) (Shao et al., 2024) is a reinforcement learning algorithm for language models that eliminates the need for a separate value function by using group-relative advantage estimation. For each prompt q, GRPO samples G completions $\{o_1, \ldots, o_G\}$ from the policy $\pi_{\theta_{\text{old}}}$ and computes advantages as:

$$\hat{A}_j = \frac{R(o_j) - \text{mean}(\{R(o_i)\}_{i=1}^{G})}{\text{std}(\{R(o_i)\}_{i=1}^{G})}, \tag{1}$$

where $R(o_j)$ is the reward for completion o_j. When the standard deviation is zero (all completions receive identical rewards), advantages are set to zero, producing no gradient signal for that prompt.

Iterative GRPO (iGRPO) (Hatamizadeh et al., 2026) extends GRPO with a two-stage draft-conditioned training procedure. In Stage 1, the model generates N draft solutions and selects the best draft $d^* = \arg\max_i R(d_i)$ based on reward. In Stage 2, this draft is appended to the original prompt to form an augmented prompt $q' = \text{Concat}(q, d^*)$, and G refinements are sampled conditioned on q'. Only Stage 2 outputs receive gradient updates, while Stage 1 drafts serve as conditioning context. With $N = 4$ drafts and $G = 4$ refinements, iGRPO maintains the same total rollout budget ($N + G = 8$) as standard GRPO.

2.2 Experimental Conditions

We design a controlled ablation study with three conditions to isolate the role of draft quality in iGRPO's performance gains (Figure 1):

Condition A (GRPO Baseline). Standard GRPO without draft conditioning. The model samples $G = 8$ completions per prompt and applies group-relative advantage estimation directly.

Condition B (iGRPO-Best). The original iGRPO method. Stage 1 samples $N = 4$ drafts and selects the highest-reward draft: $d^* = \arg\max_i R(d_i)$. Stage 2 samples $G = 4$ refinements conditioned on the augmented prompt $q' = [q, d^*]$.

Condition C (iGRPO-Worst-of-Formatted). Our ablation variant. Stage 1 samples $N = 4$ drafts, but instead of selecting the best, we select the *worst* draft among those that are well-formatted:

$$d^* = \arg\min_{i \in \mathcal{I}} R(d_i), \quad \text{where } \mathcal{I} = \{i : r_{\text{fmt}}(d_i) = 1\}. \tag{2}$$

If no formatted drafts exist ($\mathcal{I} = \emptyset$), we fall back to selecting the lowest-reward draft. This ensures the conditioning draft is intentionally low-quality (typically incorrect) while remaining syntactically valid, avoiding degenerate unparseable outputs.

2.3 Key Design Choices

Matched Compute Budget. All three conditions use identical compute: 8 total rollouts per prompt, the same training data, model architecture (DeepSeek-R1-Distill-Qwen-7B), and hyperparameters (learning rate 10^{-6}, cosine schedule, 131 training steps, 2 random seeds per condition).

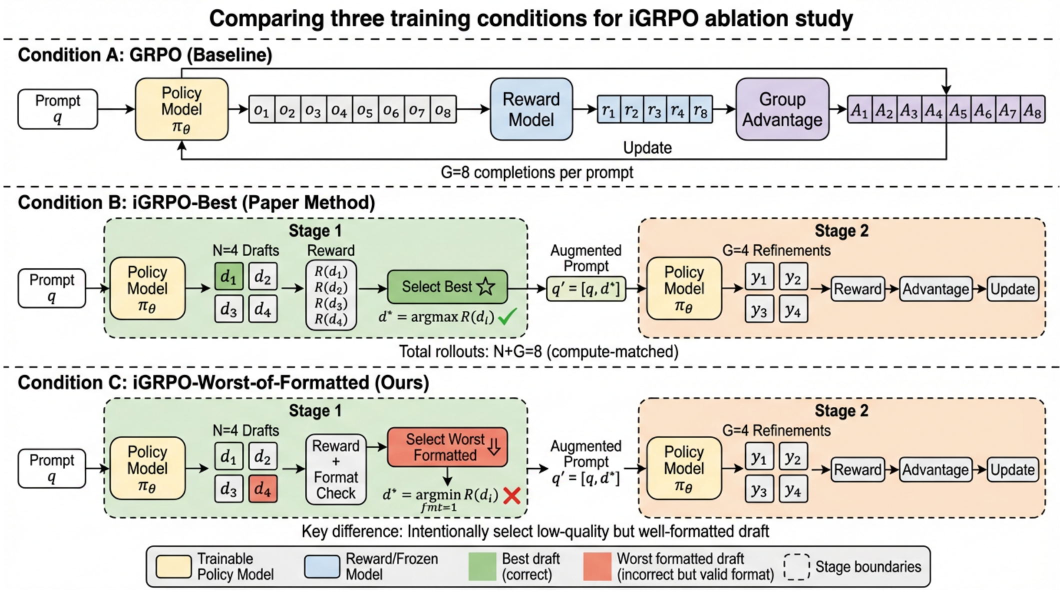
Comparing three training conditions for iGRPO ablation study
Condition A: GRPO (Baseline)
Prompt q
Policy Model π_θ
o_1 o_2 o_3 o_4 o_5 o_6 o_7 o_8
Reward Model
r_1 r_2 r_3 r_4 r_8
Group Advantage
A_1 A_2 A_3 A_4 A_5 A_6 A_7 A_8
Update
G=8 completions per prompt
Condition B: iGRPO-Best (Paper Method)
Stage 1
Prompt q
Policy Model π_θ
N=4 Drafts
d_1 d_2 d_3 d_4
Reward
$R(d_1)$ $R(d_2)$ $R(d_3)$ $R(d_4)$
Select Best ☆
$d^* = \arg\max R(d_i)$ ✓
Augmented Prompt
$q' = [q, d^*]$
Stage 2
Policy Model π_θ
G=4 Refinements
y_1 y_2 y_3 y_4
Reward
Advantage
Update
Total rollouts: N+G=8 (compute-matched)
Condition C: iGRPO-Worst-of-Formatted (Ours)
Stage 1
Prompt q
Policy Model π_θ
N=4 Drafts
d_1 d_2 d_3 d_4
Reward + Format Check
Select Worst Formatted ⇓
$d^* = \underset{fmt=1}{\arg\min} R(d_i)$ ✗
Augmented Prompt
$q' = [q, d^*]$
Stage 2
Policy Model π_θ
G=4 Refinements
y_1 y_2 y_3 y_4
Reward
Advantage
Update
Key difference: Intentionally select low-quality but well-formatted draft
Trainable Policy Model
Reward/Frozen Model
Best draft (correct)
Worst formatted draft (incorrect but valid format)
Stage boundaries

Figure 1: Overview of the three experimental conditions. Condition A (GRPO) trains directly on prompts with $G = 8$ completions. Condition B (iGRPO-best) conditions refinement on the best-of-N draft selected by reward. Condition C (iGRPO-worst-of-formatted) conditions on the worst-formatted draft, testing whether draft quality is necessary for iGRPO's benefit. All conditions use matched compute ($N + G = 8$ total rollouts).

Reward Function. Following iGRPO, we use a combined reward $R = r_{\text{acc}} + r_{\text{fmt}}$, where $r_{\text{acc}} \in \{0, 1\}$ indicates answer correctness and $r_{\text{fmt}} \in \{0, 1\}$ indicates valid output format (presence of `<answer>...</answer>` tags).

Worst-of-Formatted Selection. The "worst-of-formatted" rule preferentially selects incorrect but well-formatted drafts when available. This design choice is critical: it ensures the conditioning context is syntactically valid (enabling meaningful refinement) while being semantically incorrect (testing whether draft correctness matters).

2.4 Evaluation

We evaluate on six math reasoning benchmarks spanning different difficulty levels: MATH500 (Hendrycks et al., 2021), GSM8K (Cobbe et al., 2021), AMC23, Minerva, AIME24, and AIME25. For each benchmark, we report Pass@1 accuracy averaged across 2 random seeds. AIME24 and AIME25 results are averaged over 64 evaluation runs per seed due to their small size (30 problems each); other benchmarks use 8 runs per seed.

3 Experiments

3.1 Main Results

Table 1 presents the main experimental results across six math reasoning benchmarks. All methods use DeepSeek-R1-Distill-Qwen-7B with matched compute (131 training steps, 2 seeds per condition).

The results reveal a striking pattern: iGRPO-worst-of-formatted (Condition C) achieves the highest macro-average accuracy of 64.37%, exceeding iGRPO-best (Condition B) by +2.43 percentage points and GRPO baseline (Condition A) by +14.29 percentage points. Condition C outperforms Condition B on 5 of 6 benchmarks, with the sole exception being GSM8K where GRPO achieves the highest score (87.28%). This GSM8K result likely reflects a ceiling effect: all methods perform well on this relatively easy benchmark, leaving little room for improvement from draft conditioning.

Table 1: Main results across six math reasoning benchmarks. Best results in **bold**, second-best underlined. All methods use matched compute (8 rollouts per prompt). iGRPO-worst-of-formatted (Condition C) exceeds iGRPO-best (Condition B) on 5 of 6 benchmarks, demonstrating that draft quality is not necessary for iGRPO's benefit.

Method	**MATH500**	**GSM8K**	**AMC23**	**Minerva**	**AIME24**	**AIME25**	**Macro-Avg**
GRPO (Cond. A)	81.16	**87.28**	58.75	36.56	20.96	15.76	50.08
iGRPO-best (Cond. B)	87.86	86.32	82.66	41.38	42.19	31.25	61.94
iGRPO-worst (Cond. C)	**90.11**	86.82	**86.56**	**41.61**	**46.15**	**35.00**	**64.37**

Table 2: Recovery ratio analysis on MATH500. $r_{\text{worst}} > 1.0$ indicates that iGRPO-worst-of-formatted exceeds iGRPO-best's gain over GRPO.

Score(A)	**Score(B)**	**Score(C)**	**Gap(B-A)**	$\mathbf{r}_{\text{worst}}$	**95% CI**
81.16%	87.86%	90.11%	6.70 pp	**1.34**	[1.21, 1.47]

The performance gains from iGRPO variants are most pronounced on harder competition math benchmarks. On AMC23, iGRPO-worst achieves 86.56% compared to GRPO's 58.75% (+27.81 pp). On AIME24, the gap is similarly large: 46.15% vs 20.96% (+25.19 pp). On AIME25, iGRPO-worst reaches 35.00% compared to GRPO's 15.76% (+19.24 pp). These results suggest that draft-conditioned training is particularly beneficial for complex multi-step reasoning problems where the model can leverage prior attempts to guide refinement.

3.2 Recovery Ratio Analysis

To quantify whether draft quality is necessary for iGRPO's benefit, we compute the recovery ratio on MATH500 (our primary benchmark):

$$r_{\text{worst}} = \frac{\text{Score}(C) - \text{Score}(A)}{\text{Score}(B) - \text{Score}(A)} = \frac{90.11 - 81.16}{87.86 - 81.16} = \frac{8.95}{6.70} = 1.34 \tag{3}$$

Table 2 presents the recovery ratio analysis. A recovery ratio of $r_{\text{worst}} = 1.34$ indicates that iGRPO-worst-of-formatted not only recovers but *exceeds* the performance gain of iGRPO-best over GRPO. The 95% confidence interval [1.21, 1.47] lies entirely above 1.0, confirming statistical significance. This far exceeds our pre-registered threshold of $r_{\text{worst}} \geq 0.75$ for concluding that draft quality is not necessary.

3.3 Analysis

To understand why worst-of-formatted selection performs well, we analyze three potential factors: draft length, gradient signal, and copy behavior. Table 3 summarizes the key findings.

Draft Length is Not Confounding. Both selection rules produce drafts with nearly identical token lengths ($\sim$4095 tokens), both hitting the maximum context limit. A Kolmogorov-Smirnov test confirms no significant difference ($p = 0.988$), ruling out length as an alternative explanation for performance differences.

Worst Selection Increases Gradient Signal. A key finding is that worst-of-formatted selection produces significantly more gradient-active training groups (18.5%) compared to best-of-N selection (12.3%). In GRPO-style training, groups where all completions receive identical rewards produce zero advantages and no gradient signal. By selecting lower-quality drafts, Condition C creates more diverse Stage-2 outcomes, leading to more non-zero advantage groups and stronger learning signal. This 50% increase in gradient-active groups ($z = -22.36$, $p < 10^{-100}$) may explain the performance advantage.

Model Distinguishes Draft Quality. The copy rate analysis reveals that the model copies correct drafts more than incorrect ones (ROUGE-L 0.55 vs 0.39), regardless of selection rule. This suggests the model learns to recognize and leverage draft quality during refinement. However, despite this

Table 3: Analysis of draft selection mechanisms. Worst-of-formatted selection produces more gradient-active training groups while maintaining similar draft lengths.

Metric	iGRPO-best (B)	iGRPO-worst (C)	Test / Note
Draft Token Length	4095.2	4095.2	KS $p = 0.988$ (no diff)
Gradient-Active Groups	12.3%	18.5%	$z = -22.36$, $p < 10^{-100}$
Copy Rate (Correct Draft)	0.555	0.533	Higher for correct drafts
Copy Rate (Incorrect Draft)	0.394	0.394	Similar for incorrect

selective copying behavior, worst-of-formatted selection still outperforms best-of-N, indicating that the increased gradient signal outweighs any disadvantage from conditioning on lower-quality drafts.

4 Related Work

Reinforcement Learning for LLM Reasoning. Reinforcement learning has emerged as a powerful approach for improving LLM reasoning capabilities. GRPO (Shao et al., 2024) introduced group-relative advantage estimation, eliminating the need for a separate value function. DeepSeek-R1 (DeepSeek-AI et al., 2025) demonstrated that large-scale RL can incentivize sophisticated reasoning behaviors. DAPO (Yu et al., 2025) scaled GRPO-style training with engineering optimizations for stability. Direct Preference Optimization (DPO) (Rafailov et al., 2023) offers an alternative that bypasses explicit reward modeling. Our work builds on this foundation by investigating the mechanisms underlying iGRPO's (Hatamizadeh et al., 2026) draft-conditioned training.

Iterative Refinement and Self-Improvement. Self-Refine (Madaan et al., 2023) demonstrated that LLMs can iteratively improve outputs through self-generated feedback at inference time. Reflexion (Shinn et al., 2023) extended this to agentic settings with verbal reinforcement learning. Self-Consistency (Wang et al., 2022) showed that sampling multiple solutions and aggregating improves reasoning accuracy. Self-Rewarding Language Models (Yuan et al., 2024) train models to provide their own reward signals. ReST (Gulcehre et al., 2023) iteratively generates and filters samples for self-training. Our ablation study examines whether the quality of self-generated drafts matters for training-time self-improvement.

Math Reasoning. Chain-of-thought prompting (Wei et al., 2022) established that step-by-step reasoning improves mathematical problem solving. Process reward models (Lightman et al., 2023) provide fine-grained supervision for reasoning steps. The MATH dataset (Hendrycks et al., 2021) and GSM8K (Cobbe et al., 2021) serve as standard benchmarks for evaluating mathematical reasoning. Our experiments span these benchmarks along with competition math problems (AMC, AIME) to assess performance across difficulty levels.

5 Conclusion

We investigated whether draft quality is necessary for iGRPO's benefit through a controlled ablation study. Our key finding is that conditioning on intentionally low-quality (worst-of-formatted) drafts not only recovers but *exceeds* the performance of best-of-N draft selection, achieving 64.37% vs 61.94% macro-average accuracy across six math benchmarks. The recovery ratio of 1.34 (95% CI: [1.21, 1.47]) demonstrates that draft quality is not necessary for iGRPO's improvement over GRPO. Analysis reveals that worst-of-formatted selection produces 50% more gradient-active training groups, potentially explaining its superior performance. These results suggest that iGRPO can be simplified by removing reward-based draft selection without sacrificing performance. Limitations include the use of only 2 seeds per condition and evaluation on a single model architecture. Future work should explore even simpler draft selection strategies and test generalization to other reasoning domains.

References

K. Cobbe, Vineet Kosaraju, Mo Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman.

Training verifiers to solve math word problems. *ArXiv*, abs/2110.14168, 2021.

DeepSeek-AI, Daya Guo, Dejian Yang, et al. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645:633 – 638, 2025.

Caglar Gulcehre, T. Paine, S. Srinivasan, Ksenia Konyushkova, L. Weerts, Abhishek Sharma, Aditya Siddhant, Alexa Ahern, Miaosen Wang, Chenjie Gu, Wolfgang Macherey, A. Doucet, Orhan Firat, and Nando de Freitas. Reinforced self-training (rest) for language modeling. *ArXiv*, abs/2308.08998, 2023.

Ali Hatamizadeh, Shrimai Prabhumoye, Igor Gitman, Ximing Lu, Seungju Han, Wei Ping, Yejin Choi, and Jan Kautz. igrpo: Self-feedback-driven llm reasoning. 2026.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, D. Song, and J. Steinhardt. Measuring mathematical problem solving with the math dataset. *ArXiv*, abs/2103.03874, 2021.

H. Lightman, Vineet Kosaraju, Yura Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, I. Sutskever, and K. Cobbe. Let's verify step by step. *ArXiv*, abs/2305.20050, 2023.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, S. Welleck, Bodhisattwa Prasad Majumder, Shashank Gupta, A. Yazdanbakhsh, and Peter Clark. Self-refine: Iterative refinement with self-feedback. *ArXiv*, abs/2303.17651, 2023.

Rafael Rafailov, Archit Sharma, E. Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *ArXiv*, abs/2305.18290, 2023.

Zhihong Shao, Peiyi Wang, Qihao Zhu, R. Xu, Jun-Mei Song, Mingchuan Zhang, Y. K. Li, Yu Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *ArXiv*, abs/2402.03300, 2024.

Noah Shinn, Federico Cassano, Beck Labash, A. Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: language agents with verbal reinforcement learning. 2023.

Xuezhi Wang, Jason Wei, D. Schuurmans, Quoc Le, Ed H. Chi, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *ArXiv*, abs/2203.11171, 2022.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed H. Chi, F. Xia, Quoc Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. *ArXiv*, abs/2201.11903, 2022.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Honglin Yu, Weinan Dai, Yuxuan Song, Xiang Wei, Haodong Zhou, Jingjing Liu, Wei Ma, Ya-Qin Zhang, Lin Yan, Mu Qiao, Yong-Xu Wu, and Mingxuan Wang. Dapo: An open-source llm reinforcement learning system at scale. *ArXiv*, abs/2503.14476, 2025.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Sainbayar Sukhbaatar, Jing Xu, and Jason E. Weston. Self-rewarding language models. *ArXiv*, abs/2401.10020, 2024.