SR-GRPO: Stable Rank as an Intrinsic Geometric Reward for Large Language Model Alignment

Yixuan Tang
The Hong Kong University of Science and Technology
ytangch@connect.ust.hk

Yi Yang
The Hong Kong University of Science and Technology
imyiyang@ust.hk

Abstract

Aligning Large Language Models (LLMs) with human preferences typically relies on external supervision, which faces critical limitations: human annotations are scarce and subjective, reward models are vulnerable to reward hacking, and self-evaluation methods suffer from prompt sensitivity and biases. In this work, we propose **stable rank**, an intrinsic, annotation-free quality signal derived from model representations. Stable rank measures the effective dimensionality of hidden states by computing the ratio of total variance to dominant-direction variance, capturing quality through how information distributes across representation dimensions. Empirically, stable rank achieves 84.04% accuracy on RewardBench and improves task accuracy by an average of 14.1% relative to greedy decoding via Best-of-N sampling. Leveraging this insight, we introduce **Stable Rank Group Relative Policy Optimization (SR-GRPO)**, which uses stable rank as a reward signal for reinforcement learning. Without external supervision, SR-GRPO improves Qwen2.5-1.5B-Instruct by 10% on STEM and 19% on mathematical reasoning, outperforming both learned reward models and self-evaluation baselines. Our findings demonstrate that quality signals can be extracted from internal model geometry, offering a path toward scalable alignment without external supervision.

1 Introduction

The alignment of Large Language Models (LLMs) with human preferences typically relies on Reinforcement Learning from Human Feedback (RLHF) [33, 4]. Despite its success, this paradigm depends heavily on external supervision. Human judgments are subjective and context-dependent, making it difficult to train robust reward models [6, 42], and learned proxies are susceptible to reward hacking [5]. The sparsity of annotations further limits the model's ability to fine-grained behaviors [44].

These limitations have motivated various attempts to reduce annotation dependence. Direct Preference Optimization [34] eliminates explicit reward modeling but still requires preference datasets. Verifiable rewards [9, 24] leverage ground-truth outcomes but only apply where automatic verification is feasible. Self-evaluation methods [49, 26, 13] suffer from prompt sensitivity and systematic biases [53, 43]. Crucially, all these approaches still evaluate quality through external signals while overlooking information embedded in the model's own representations.

Yet the validity of a generation should be reflected in its underlying computation. When a model outputs "The capital of France is Paris," its hidden states encode activated factual knowledge [20]; when it hallucinates, different activation patterns emerge [7]. This raises a natural question: *Can*

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

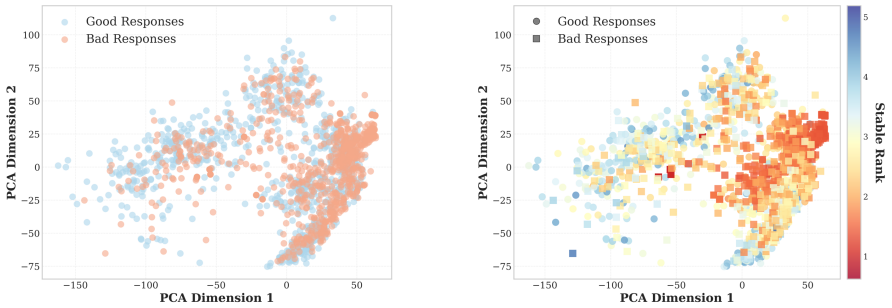

Figure 1: Hidden states visualization for responses in RewardBench [25] based on Qwen3-8B [46]. Left: PCA projection shows good (blue) and bad (red) responses with spatial overlap. Right: Stable rank coloring uncovers clear quality separation: good responses exhibit higher ranks than bad responses. Stable rank can distinguish response quality.

we measure text quality directly from internal signals, enabling LLM alignment without external supervision?

In this work, we show that a simple geometric property of LLM hidden states, their effective dimensionality, provides a reliable signal for LLM response quality. We propose to use **stable rank** [38], a matrix-theoretic measure as an unsupervised proxy for generation quality. Stable rank quantifies how many independent semantic directions a response occupies by measuring the ratio of total variance to dominant-direction variance in the hidden-state representation. It balances representational richness with coherence: a high stable rank indicates that information spreads across many dimensions rather than concentrating in a few. The theoretical motivation comes from two lines of work. First, the softmax bottleneck analysis shows that accurate language modeling requires navigating a high-dimensional semantic manifold, as the contextual probability distribution of natural language is inherently high-rank [47, 16]. Second, literature also shows that when representations collapse into a narrow cone, expressiveness is severely limited and generation quality degrades [12]. We therefore hypothesize that this principle extends to individual LLM responses: high-quality generations should maintain higher effective dimensionality, while low-quality outputs exhibit rank collapse.

We validate this hypothesis in two settings. First, as a zero-shot reward proxy, stable rank achieves 84.04% accuracy on RewardBench [25] with Qwen3-8B [46], matching LLM-as-Judge baselines without any training. Figure 1 illustrates this geometrically: while PCA projections of LLM representations show overlap between good and bad responses, stable rank cleanly separates them. Second, stable rank-guided Best-of-N selection consistently outperforms greedy decoding across four model families on STEM and mathematics benchmarks, with average gains of 11.3% at $N = 16$.

Building on these findings, we introduce **Stable Rank Group Relative Policy Optimization (SR-GRPO)**, which uses stable rank as an intrinsic reward signal for reinforcement learning. By replacing external supervision with this geometric signal derived directly from the reference model, we enable fully annotation-free alignment. On Qwen2.5-1.5B-Instruct [45], SR-GRPO outperforms GRPO with the learned reward model baseline by 9.9% on STEM tasks and 18.7% on mathematical reasoning tasks, and surpasses self-reward baselines by 8 to 9%. Consistent gains on DeepSeek-R1-Distill-Qwen-1.5B [9] confirm robustness across model families.

To understand why stable rank succeeds as a quality signal, we analyze its correlations with interpretable text metrics. We find that stable rank captures three quality dimensions: semantic coherence, where sentences build on each other while staying relevant to the prompt; information density over verbosity, favoring concise, non-repetitive text; and sensitivity to reasoning words like “however" and “because" at key turning points.

Our contributions are summarized as follows:

- We propose SR-GRPO, which uses stable rank as a dense reward signal in reinforcement learning, eliminating dependency on preference datasets or external verifiers.
- We demonstrate that SR-GRPO improves reasoning across multiple benchmarks, thus establishing intrinsic geometric signals as a viable basis for LLM alignment.

- Beyond reinforcement learning, we also show that stable rank offers a reliable zero-shot reward proxy for evaluating LLM responses, and integrates naturally into best-of-N decoding.

2 Stable Rank as an Intrinsic Quality Metric

We propose stable rank [38] as an unsupervised quality signal derived from LLM representations. This section defines the metric, explains its geometric motivation, and validates its effectiveness as a reward proxy.

2.1 Definition

For a sequence of T tokens, we extract the hidden state activation matrix $\mathbf{H} \in \mathbb{R}^{T \times d}$ from the last layer of an LLM. Let $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{\min(T,d)} \geq 0$ denote the singular values of $\mathbf{H}$. The stable rank is defined as:

$$\text{SR}(\mathbf{H}) = \frac{\|\mathbf{H}\|_F^2}{\|\mathbf{H}\|_2^2} = \frac{\sum_i \sigma_i^2}{\sigma_1^2}. \tag{1}$$

Stable rank measures the effective dimensionality of $\mathbf{H}$: if a single singular value dominates, $\text{SR}(\mathbf{H}) \approx 1$, indicating representation collapse; when singular values are balanced, stable rank approaches $\text{rank}(\mathbf{H})$, reflecting a rich, high-dimensional representation. Intuitively, low stable rank means token representations cluster along a few dominant directions, while a high stable rank means they spread across the embedding space.

Theoretical motivation. The theoretical motivation comes from two lines of work. The softmax bottleneck [47, 16] implies that expressive language modeling requires high-rank representations, while representation collapse is a known symptom of degraded generation [12]. Stable rank directly measures this effective dimensionality, making it a natural candidate for quality assessment. We validate this connection empirically below and through correlation analysis in Section 5.

Implementation. Given a prompt-response pair formatted with the model's chat template, we perform a forward pass and extract the hidden state matrix $\mathbf{H} \in \mathbb{R}^{T \times d}$ from the final hidden layer, where T is the total number of tokens. We use final-layer activations because they aggregate information across all preceding layers and are directly related to the final text generation. Stable rank is then computed from $\mathbf{H}$ via Equation 1. Cross-layer analysis in Appendix A confirms that the final layer embeddings yield the strongest quality signal.

2.2 Empirical Validation: Stable Rank as Zero-Shot Reward Proxy

We evaluate stable rank as a zero-shot proxy for human preference by testing whether it can predict the preferred response in a pair. This assessment checks whether stable rank can serve as a reliable reward signal without any model-specific training.

Setup. We use RewardBench [25], a benchmark containing 2,985 preference pairs across five categories: Chat, Chat-Hard, Safety, Code, and Math. For each pair, we compute stable rank for both responses and predict the one with the higher stable rank as preferred. We compare against three baseline methods: (1) Pointwise Scoring (Point.) [22]: prompting the LLM to score each response on a 1-5 scale; (2) Pairwise Comparison (Pair.) [22]: prompting the LLM to directly compare two responses; (3) IPO [13]: using implicit preference scores derived from Yes/No token probabilities. We evaluate five models: Qwen2.5-1.5B-Instruct [45], Qwen3-0.6B [46], Qwen3-8B [46], Llama-3.1-8B-Instruct [10], and Phi-3.5-mini-Instruct [1]. For stable rank, we format inputs as prompt-response pairs using each model's native chat template, extract final-layer hidden states for the response tokens only, and compute stable rank via Eq. 1. Prompts and implementation details are in Appendix B.

Results. Table 1 shows stable rank consistently outperforms baselines across model scales. On Qwen3-8B, it achieves 84.04% accuracy, surpassing Pointwise (83.70%) and IPO (78.02%). The advantage is more pronounced on smaller models: stable rank achieves 75.95% on Qwen2.5-1.5B-Instruct, outperforming the best baseline (IPO, 65.85%) by 10.1 percentage points. Generative baselines exhibit high variance across scales. Pointwise scoring excels on large models (83.70% on

Qwen3-8B) but collapses on smaller ones (37.15% on Qwen2.5-1.5B), likely because smaller models lack the instruction-following capability to produce calibrated scores. Pairwise comparison shows the opposite pattern, performing better on small models but degrading on Qwen3-8B (71.98%). IPO maintains relative stability (65.57%–78.02%) but still underperforms stable rank across all models. Complete results by category are provided in Appendix D.

The consistent advantage of stable rank, especially on smaller models, suggests that LLM's intrinsic geometric signals are more robust than prompt-based evaluation. Prompt-based methods require LLMs to follow evaluation instructions and produce calibrated scores, which smaller models often struggle with. Stable rank has no such dependency, making it particularly suitable for aligning tiny LLMs.

Table 1: RewardBench overall accuracy (%).

Model	Point.	Pair.	IPO	SR
Qwen3-0.6B	38.76	47.12	65.57	**66.96**
Llama-3.1-8B	56.37	56.65	58.14	**68.36**
Phi-3.5-mini	42.44	62.11	66.27	**70.87**
Qwen2.5-1.5B	37.15	53.46	65.85	**75.95**
Qwen3-8B	83.70	71.98	78.02	**84.04**

2.3 Empirical Validation: Best-of-N Decoding with Stable Rank

We next evaluate stable rank as a test-time quality signal in Best-of-N decoding, where the model samples multiple candidate responses and selects the one with the highest stable rank. This setup tests whether stable rank can guide inference and improve task performance beyond standard greedy decoding.

Setup. For each prompt, we sample $N \in \{1, 4, 8, 16\}$ responses (temperature 0.7, top-p 0.9) and select the one with the highest stable rank. We test on STEM benchmarks (GPQA [36], MMLU-redux [15]) and mathematical reasoning tasks (MATH500 [27], OlympiadBench [18], AMC23 [23]) across four models: Qwen2.5-1.5B-Instruct [45], Phi-3.5-mini-Instruct [1], Llama-3.2-1B-Instruct [10], and DeepSeek-R1-Distill-Qwen-1.5B [9]. We compare against random selection to isolate the effect of the selection criterion from sampling diversity.

Results. Figure 2 shows that stable rank-guided selection consistently improves over greedy decoding across all models. At N=16, Llama-3.2-1B achieves the largest gain (+20.5%), followed by Qwen2.5-1.5B (+17.0%) and DeepSeek-R1 (+10.2%), and Phi-3.5-mini shows a moderate improvement of +8.5%. Stable rank also outperforms random selection. On Llama-3.2-1B at N=16, stable rank (26.5% avg.) exceeds random selection (19.8%) by 33.8%. Notably, random selection often degrades performance below greedy decoding, such as -16.1% on Llama-3.2-1B at N=8. In contrast, stable rank consistently yields positive gains, demonstrating that it identifies high-quality responses rather than merely benefiting from sampling diversity. These results confirm that stable rank captures quality signals that correlate strongly with correctness. Complete results are provided in Appendix E.

3 Stable Rank Group Relative Policy Optimization (SR-GRPO)

Having established an empirical correlation between stable rank and response quality, we next ask whether using stable rank as a reward signal in reinforcement learning can further improve LLM alignment behavior. Because stable rank requires no labeled data and can be computed for any generated response, it provides a simple and scalable dense reward for reinforcement learning.

3.1 Setup

Let π_{ref} denote a reference model and π_ϕ the trainable policy initialized from π_{ref}. Given a prompt x, the policy generates a response $y \sim \pi_\phi(\cdot|x)$. Our goal is to optimize π_ϕ to generate higher-quality responses, using stable rank as a proxy reward.

3.2 Training Procedure

We build on Group Relative Policy Optimization (GRPO) [39], which compares responses within a sampled group to obtain relative quality signals, avoiding the need for a learned value function. This

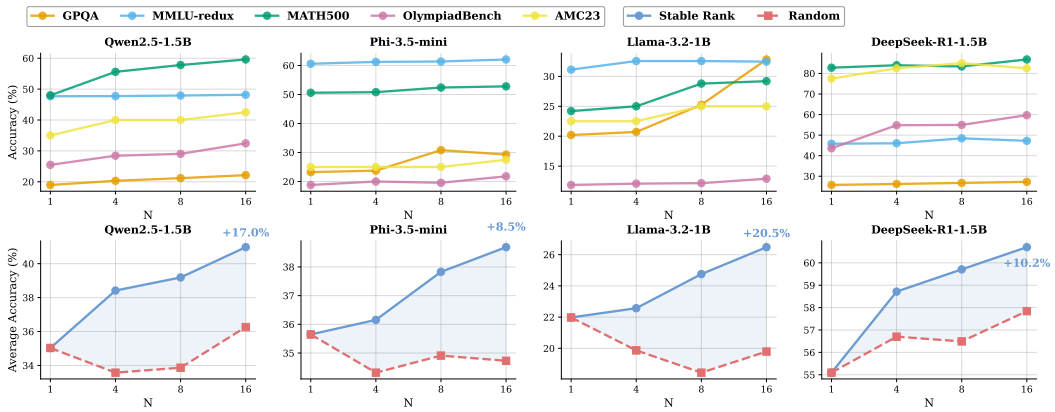

Figure 2: Best-of-N decoding with stable rank selection. **Top**: Performance on individual benchmarks as N increases. **Bottom**: Average accuracy comparing stable rank selection (solid) versus random selection (dashed). Annotations show relative improvement over greedy decoding (N=1). Stable rank consistently outperforms random selection across all models.

avoids the need to train a separate critic and suits our setting where rewards come from a geometric metric.

For each prompt x, we sample K responses from the current policy and compute their stable rank on the frozen reference model π_{ref}. Using a frozen model is critical: it provides a stationary reward signal that the policy cannot manipulate by changing its own internal geometry.

We standardize rewards within each group to obtain scale-invariant learning signals: $A_k = (r_k - \mu)/(\sigma + \epsilon)$, where μ and σ are the group mean and standard deviation. The training objective is:

$$\mathcal{J}(\phi) = \mathbb{E}_x \left[\frac{1}{K} \sum_{k=1}^{K} \rho_k A_k - \beta D_{\text{KL}}(\pi_\phi \| \pi_{\text{ref}}) \right] \tag{2}$$

where $\rho_k = \pi_\phi(y_k|x)/\pi_{\phi_{\text{old}}}(y_k|x)$ is the importance ratio, ϕ_{old} denotes the parameters before the current update, and β controls the KL penalty. The full training procedure is summarized in Appendix F, Algorithm 1.

Computational efficiency. Computing stable rank requires $O(Td)$ operations: $O(Td)$ for the Frobenius norm and $O(Td)$ per power iteration for the spectral norm. This overhead is negligible compared to the transformer forward pass. Stable rank is also robust to input truncation: using only 512 tokens achieves nearly identical accuracy (Appendix J).

4 Alignment Experiments

4.1 Experiment Setup

We evaluate SR-GRPO on two models: Qwen2.5-1.5B-Instruct [45] and DeepSeek-R1-Distill-Qwen-1.5B [9]. All methods are trained on prompts from SmolTalk2 [3], which provides diverse topics and tasks. No preference labels are used during training; methods differ only in how they compute rewards from sampled responses. For SR-GRPO, we train with LoRA [21] and compute stable rank on the base model without LoRA adapters, which also serves as the frozen reference.

Benchmarks. We evaluate on three categories: (1) *STEM tasks*: GPQA [36], a graduate-level science QA benchmark requiring expert knowledge, and MMLU-redux [15], a curated subset of MMLU with corrected labels; (2) *Mathematical reasoning*: MATH500 [27], a subset of competition mathematics problems, AIME25 [51], problems from the 2025 American Invitational Mathematics Examination, OlympiadBench [19], olympiad-level math and physics problems, and AMC23 [23], problems from the 2023 American Mathematics Competition; (3) *General chat*: WildBench [28], which evaluates open-ended conversation quality using GPT-4o mini [32] as a judge. For all benchmarks except

Table 2: Alignment results on Qwen2.5-1.5B-Instruct and DeepSeek-R1-Distill-Qwen-1.5B. SR-GRPO outperforms the reward model (RM) and self-evaluation baselines without external labels. WildBench scores are reported as WB-Elo. Best results in **bold**; ties are all bolded.

		STEM			Math					Chat
Model	**Method**	GPQA	MMLU	Avg.	MATH	AIME	Olymp.	AMC	Avg.	WB-Elo
Qwen2.5-1.5B-Instruct	Base	19.0	**47.7**	33.3	48.0	3.3	25.5	35.0	28.0	1036.2
	+ RM	15.7	47.2	31.4	50.4	3.3	25.6	30.0	27.3	1043.3
	+ Self-Reward	17.7	45.5	31.6	50.6	**13.3**	23.6	32.5	30.0	1041.2
	+ Perplexity	18.7	47.6	33.1	48.4	10.0	25.6	32.5	29.1	1040.9
	+ IPO	17.2	**47.7**	32.4	49.2	3.3	24.0	27.5	26.0	1037.7
	+ SR-GRPO	**21.2**	**47.7**	**34.5**	**52.4**	**13.3**	**26.4**	**37.5**	**32.4**	**1062.4**
DeepSeek-R1-Distill-Qwen-1.5B	Base	25.8	45.8	35.8	82.8	30.0	43.6	**77.5**	58.5	913.5
	+ RM	24.2	46.3	35.3	85.4	**36.7**	56.2	72.5	62.7	918.4
	+ Self-Reward	23.7	45.8	34.8	85.2	**36.7**	57.2	67.5	61.6	919.2
	+ Perplexity	24.2	46.3	35.3	**86.2**	**36.7**	55.0	**77.5**	63.8	917.0
	+ IPO	28.8	45.8	37.3	86.0	26.7	56.7	**77.5**	61.7	922.4
	+ SR-GRPO	**30.3**	**46.6**	**38.4**	**86.2**	**36.7**	**58.5**	**77.5**	**64.7**	**932.5**

WildBench, we report Pass@1 accuracy. WildBench [28] evaluates open-ended conversation quality by comparing model outputs against reference responses using GPT-4o mini as a judge; scores are reported as Elo ratings.

Baselines. We compare SR-GRPO against two categories of methods:

(1) Reward Model (RM): We use Skywork-Reward-V2-Qwen3-1.7B [29], a widely used reward model trained on the Skywork preference dataset with the Bradley-Terry objective. We select a 1.7B model to match the scale of our policy models, ensuring fair comparison with self-evaluation methods that use the policy model itself for reward computation.

(2) Self-Evaluation Methods: These methods derive rewards from the model's own outputs without external labels: *Self-Reward* [49] asks the model to evaluate its own completions with the pointwise scoring prompt (Appendix B) and uses the resulting 1–5 score as the reward. *Perplexity* uses the negative log-likelihood of the completion as the reward, treating lower perplexity as higher fluency. *IPO* [13] casts the model as a binary quality classifier: it prompts the model to answer "Yes" or "No", extracts the logits of both tokens at the first output position, and uses the softmax probability of "Yes" as the reward signal.

All methods use the same GRPO training configurations (K=8 responses per prompt, β=0.01) to ensure fair comparison. Full training details are in Appendix F.

4.2 Results Analysis

Table 2 shows that SR-GRPO consistently outperforms both reward model and self-evaluation baselines across all categories, despite using no external labels.

Reasoning improvements. On Qwen2.5-1.5B-Instruct, SR-GRPO improves average mathematical reasoning accuracy by 4.4 percentage points (28.0% to 32.4%), with particularly strong gains on competition-level problems: AMC improves from 35.0% to 37.5% and MATH from 48.0% to 52.4%. On DeepSeek-R1-Distill-Qwen-1.5B, which already achieves 58.5% on math tasks, SR-GRPO still improves performance to 64.7%, demonstrating effectiveness even on reasoning-specialized models. The gains on OlympiadBench (43.6% to 58.5%) suggest that stable rank rewards the structured reasoning patterns required for complex problem-solving.

Comparison with baselines. The reward model shows minimal or even negative impact on several tasks (e.g., GPQA drops from 19.0% to 15.7% on Qwen2.5-1.5B), suggesting that models trained on general preference data may not transfer well to specialized reasoning domains. Self-evaluation methods show inconsistent patterns: Self-Reward matches SR-GRPO on AIME (13.3%) but underperforms on STEM tasks (31.6% vs 34.5% avg.); Perplexity performs surprisingly well on DeepSeek-R1 but lacks the consistency of SR-GRPO across both models. IPO degrades performance on several benchmarks, possibly because Yes/No probability signals are too coarse for nuanced quality distinctions.

General chat quality. SR-GRPO's advantage extends beyond reasoning to open-ended conversation. On WildBench (WB-Elo), it achieves gains of 26.2 points on Qwen2.5-1.5B and 19.0 points on

Table 3: Correlations between stable rank and interpretable response properties. (a) Semantic coherence and information-density metrics. (b) Discourse-marker categories from paired-difference analysis. Negative correlations indicate lower stable rank for larger metric values. $^{***}p < 0.001$.

(a) Text-quality metrics

Group	Metric	ρ
Semantic	Coherence std.	-0.356^{***}
	QA alignment	$+0.316^{***}$
	Progression	$+0.313^{***}$
	Semantic variance	-0.272^{***}
	Adjacent sim.	$+0.250^{***}$
Density	Sentence count	-0.368^{***}
	Token count	-0.294^{***}
	Unique tokens	-0.286^{***}
	Lexical diversity	$+0.238^{***}$
	Compression ratio	$+0.233^{***}$

(b) Discourse markers

Category	r	ρ	Examples
Contrastive	$+0.187^{***}$	$+0.067^{***}$	however
Causal	$+0.139^{***}$	-0.002	because
Additive	-0.166^{***}	-0.156^{***}	moreover
Conditional	-0.163^{***}	-0.163^{***}	if, when
Enumeration	-0.122^{***}	-0.148^{***}	first, second
Temporal	-0.106^{***}	-0.122^{***}	then, next
Inference	-0.065^{***}	-0.115^{***}	implies, suggests
Total count	-0.149^{***}	-0.204^{***}	*all markers*

DeepSeek-R1, substantially outperforming all baselines. This indicates that stable rank captures qualities valued in general conversation, such as coherence and helpfulness, not just task-specific correctness. The consistent improvements across both reasoning benchmarks and open-ended chat suggest that stable rank provides a general-purpose quality signal rather than optimizing for narrow task metrics.

5 What Stable Rank Captures

To understand why stable rank can serve as an intrinsic reward, we analyze its correlations with interpretable text quality metrics on the RewardBench dataset. We compute 37 metrics spanning semantic coherence, information density, and linguistic structure. For each metric M, we measure sample-level Pearson correlation with stable rank across 5,970 responses, and paired-difference Spearman correlation across 2,985 chosen/rejected pairs. The definitions and correlation tables are provided in Appendix G; here we summarize key findings.

Stable Rank Captures Semantic Coherence Table 3 presents the strongest correlations among semantic coherence metrics. Progression score ($\rho = 0.313$) measures whether each sentence builds on the previous one. QA alignment consistency ($\rho = 0.316$) captures whether relevance to the prompt remains stable throughout the response. Both correlate positively with stable rank. The strongest negative correlation appears with coherence standard deviation ($\rho = -0.356$), where high values indicate erratic transitions between adjacent sentences. These patterns suggest stable rank can capture responses that maintain topical focus while developing ideas smoothly, avoiding the abrupt transitions characteristic of hallucinations or incoherent reasoning.

Stable Rank Distinguishes Density from Verbosity Contrary to common reward hacking, where models maximize length, stable rank correlates negatively with token count ($\rho = -0.294$) and sentence count ($\rho = -0.368$). Instead, stable rank favors information density. Lexical diversity correlates positively ($\rho = 0.238$). This metric measures the ratio of unique tokens to total tokens. Compression ratio also correlates positively ($\rho = 0.233$). It measures the ratio of compressed to original text length. Table 3 presents key correlations. These patterns suggest stable rank penalizes verbose, repetitive text while rewarding concise responses with diverse vocabulary. A length-controlled preference analysis is provided in Appendix C.

Stable Rank Highlights Key Logical Markers In this section, we study the correlation between discourse markers and stable rank. Discourse markers are explicit phrases that signal logical relationships, such as "because" or "if", or structural transitions, such as "however" or "first". We use paired-difference analysis on $N = 2{,}985$ chosen and rejected pairs (Appendix G) to correlate marker frequency, measured per 100 tokens, with stable rank while controlling for prompt context. We report both Pearson and Spearman correlations because the two can diverge when relationships are non-monotonic.

Table 3 shows that most discourse marker categories are negatively correlated with stable rank. In particular, additive ($\rho = -0.156$), conditional ($\rho = -0.163$), and enumeration markers ($\rho = -0.148$) exhibit consistently negative correlations, and the total marker count is also strongly negative ($\rho = -0.204$). Responses that rely heavily on connective-heavy structures (e.g., "moreover", "first", "then") tend to receive lower stable ranks, consistent with Section 5, which shows that stable rank favors compact information over verbose, pattern-like exposition.

Contrastive and causal markers behave differently. Although their Pearson correlations are positive (contrastive $r = 0.187$, causal $r = 0.139$), their Spearman correlations are small, suggesting sensitivity to presence rather than frequency. A small number of well-placed contrastive or causal connectives appears beneficial, while frequent use provides little additional signal. This pattern aligns with their linguistic role: these markers are most informative when highlighting key reasoning transitions rather than serving as structural filler.

5.1 Summary

Together, these correlation patterns align with established findings in text quality research. The strong association with semantic coherence mirrors results from neural coherence models, which show that smooth semantic transitions predict text quality [30, 8], while the negative correlation with length and the preference for information density directly address a known failure mode of RLHF, namely length bias, where reward models favor longer responses regardless of quality [40]. Unlike learned reward models that often exploit this spurious correlation, stable rank's geometric formulation inherently penalizes verbose, low-information content. Although the observed correlations are moderate in magnitude, they are consistent across metrics and statistically significant, indicating that stable rank captures quality signals rather than noise. Overall, these results suggest that stable rank reflects multiple dimensions of text quality, including coherence, information density, and reasoning structure, and captures a meaningful aggregate signal rather than a single superficial feature. Qualitative examples are provided in Appendix L.

6 Ablation Studies

Context length. We examine how sequence truncation affects performance. Accuracy drops from 83.85% at 512 tokens to 62.59% at 128 tokens. Code suffers the largest degradation (87.91% to 24.80%) because truncation removes critical program logic. However, extending beyond 512 tokens yields negligible gains (<0.2 percentage points). This indicates that stable rank captures core semantic structure rather than mechanically rewarding longer sequences. Details are provided in Appendix J.

Prompt format. We test six input templates across three models. Overall accuracy varies by at most 3 percentage points, with no format consistently outperforming others. This robustness simplifies deployment: practitioners can use simple formats without extensive tuning. Full results are shown in Appendix K.

Model scale. We further evaluate SR-GRPO on two 8B models, Qwen3-8B and Llama-3.1-8B-Instruct, to test whether the reward transfers beyond the 1.5B setting. SR-GRPO improves both models and is strongest or tied-strongest across most reported metrics, indicating that stable rank remains useful at larger model scales. Full results are in Appendix I.1.

Different RL algorithm. Stable rank is a reward signal and is not tied to GRPO. We therefore apply the same reward with RLOO and Online DPO. All three optimization algorithms improve over the base model, suggesting that the gains come from the stable-rank reward rather than a GRPO-specific training artifact. Full results are in Appendix I.2.

Alternative intrinsic dimension metrics. We compare stable rank with three alternatives: condition number (ratio of largest to smallest singular value), PCA 95% variance (number of components explaining 95% variance), and effective rank [37] (entropy-based weighting of singular values).On RewardBench, stable rank achieves 84.04% accuracy, outperforming PCA 95% variance (61.91%), effective rank (54.50%), and condition number (36.04%), with larger margins on harder Math and Safety categories. We also compare stable rank against PCA dimension and effective rank when used directly as RL rewards; stable rank gives the strongest overall trade-off across reasoning and chat quality. Full results are in Appendix H and I.6. Additional analyses of pretrained reward

sources, stronger baselines, longer training runs, and variability across random seeds are provided in Appendix I.

7 Related Work

We review three lines of related work: alignment methods using external feedback, approaches that reduce reliance on human labels, and theoretical connections between representation geometry and generation quality.

7.1 Alignment with External Feedback Signals

The standard pipeline for aligning Large Language Models with human preferences is Reinforcement Learning from Human Feedback (RLHF) [33, 4]. RLHF trains a reward model on human preference data and then fine-tunes the policy to maximize this learned signal under a KL constraint to a reference model. This approach underpins LLMs such as ChatGPT and Llama [10], but it inherits well-known issues: preference data are expensive and noisy, and reward models are vulnerable to reward hacking and distribution shift [5, 41]. These limitations motivate supervised preference-based methods such as Direct Preference Optimization (DPO) [34], KTO [11], Bayesian RLHF [42], and MaxMin-RLHF [6], which optimize the policy directly on pairwise comparisons without an explicit value network but still rely on curated preference datasets. Process reward models and generative reward models provide step-level scores and textual critiques for mathematical reasoning and general evaluation [52, 48].All these approaches keep supervision external to the model's internal computation.

7.2 Alignment with Automatic Signals

A complementary line of work reduces reliance on human labels by using automatic supervision or signals derived from the model itself. Verifiable-reward methods such as DeepSeek-R1 [9] and TÜLU 3 [24] replace human feedback with programmatic checkers; these methods work well in domains like mathematics and coding but do not generalize to open-ended tasks. Self-evaluation approaches treat the model, or another LLM, as a judge: Self-Rewarding Language Models [49] and RLAIF [26] generate AI feedback instead of human labels, while IPO [13] and Self-Rewarding PPO [50] derive preference signals from token-level probabilities. Closer to our setting, work on internal activations such as Factoscope [20] and INSIDE [7] shows that hidden states encode factuality and hallucination risk. Our method follows this direction, using a geometric statistic of activations as a reward signal for policy optimization.

7.3 Representation Geometry and Generation Quality

A theoretical foundation for rank-based quality measures comes from the softmax bottleneck analysis. Yang et al. [47] show that the expressiveness of softmax-based models is fundamentally limited by the rank of their hidden representations. Godey et al. [16] extend this to modern LLMs, demonstrating that the contextual probability distribution of natural language is inherently high-rank. Complementary work shows that when embeddings collapse into a narrow cone during training, generation quality degrades [12]. Together, these results establish a link between representational dimensionality and generation quality. Empirical work supports this connection. Garrido et al. [14] demonstrates that the effective rank of learned representations predicts downstream performance without labels. In language models specifically, hidden states encode rich semantic structure: word embeddings capture syntactic and semantic regularities through geometric relationships [31], and intermediate representations encode factuality [20] and hallucination risk [7], confirming that generation quality leaves detectable geometric signatures.

Our work builds on these findings but shifts from diagnosis to optimization. Rather than using rank to analyze pretrained models post-hoc, we employ stable rank as a live reward signal during policy optimization, transforming a geometric insight into a practical alignment method.

8 Conclusion

This work shows that stable rank, a simple intrinsic geometric property of LLM hidden states, provides an effective signal for text quality assessment and model alignment. As a zero-shot reward proxy, stable rank achieves strong performance on RewardBench and consistently improves Best-of-N sampling across reasoning benchmarks. Building on this, we introduce SR-GRPO, which uses stable rank as a reward signal for reinforcement learning without external supervision and yields consistent improvements across STEM tasks and mathematical reasoning, outperforming learned reward models and self-evaluation baselines. Overall, our analysis indicates that stable rank reflects semantic coherence, information density, and key reasoning structures. Together, these results highlight representation geometry as a promising direction for scalable, annotation-efficient alignment.

Limitations

Although stable rank does not require external annotations, it requires access to the scoring model's hidden states, limiting its use in settings where only text outputs are available. Our analysis provides empirical evidence linking stable rank to response quality, while the mechanism underlying this relationship warrants further investigation. Our evaluation also covers a limited set of models and training settings. Extending this evaluation to broader model families and longer training runs would help characterize the generality of the signal and its behavior under sustained optimization.

Acknowledgments

This work was supported by the HKUST Strategic Fund for Futuristic Business Research under grant SBMSFFBR26BM02.

References

[1] Marah I Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat S. Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Caio César Teodoro Mendes, Weizhu Chen, Vishrav Chaudhary, Parul Chopra, Allie Del Giorno, Gustavo de Rosa, Matthew Dixon, Ronen Eldan, Dan Iter, Amit Garg, Abhishek Goswami, Suriya Gunasekar, Emman Haider, Junheng Hao, Russell J. Hewett, Jamie Huynh, Mojan Javaheripi, Xin Jin, Piero Kauffmann, Nikos Karampatziakis, Dongwoo Kim, Mahoud Khademi, Lev Kurilenko, James R. Lee, Yin Tat Lee, Yuanzhi Li, Chen Liang, Weishung Liu, Eric Lin, Zeqi Lin, Piyush Madan, Arindam Mitra, Hardik Modi, Anh Nguyen, Brandon Norick, Barun Patra, Daniel Perez-Becker, Thomas Portet, Reid Pryzant, Heyang Qin, Marko Radmilac, Corby Rosset, Sambudha Roy, Olatunji Ruwase, Olli Saarikivi, Amin Saied, Adil Salim, Michael Santacroce, Shital Shah, Ning Shang, Hiteshi Sharma, Xia Song, Masahiro Tanaka, Xin Wang, Rachel Ward, Guanhua Wang, Philipp A. Witte, Michael Wyatt, Can Xu, Jiahang Xu, Sonali Yadav, Fan Yang, Ziyi Yang, Donghan Yu, Chengruidong Zhang, Cyril Zhang, Jianwen Zhang, Li Lyna Zhang, Yi Zhang, Yue Zhang, Yunan Zhang, and Xiren Zhou. Phi-3 technical report: A highly capable language model locally on your phone. *CoRR*, abs/2404.14219, 2024. doi: 10.48550/ARXIV.2404.14219.

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

[3] Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Guilherme Penedo, Lewis Tunstall, Andrés Marafioti, Hynek Kydlícek, Agustín Piqueres Lajarín, Vaibhav Srivastav, et al. Smollm2: When smol goes big-data-centric training of a small language model. *CoRR*, 2025.

[4] Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath,

Jackson Kernion, Tom Conerly, Sheer El Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom B. Brown, Jack Clark, Sam McCandlish, Chris Olah, Benjamin Mann, and Jared Kaplan. Training a helpful and harmless assistant with reinforcement learning from human feedback. *CoRR*, abs/2204.05862, 2022.

[5] Stephen Casper, Xander Davies, Claudia Shi, Thomas Krendl Gilbert, Jérémy Scheurer, Javier Rando, Rachel Freedman, Tomek Korbak, David Lindner, Pedro Freire, Tony Tong Wang, Samuel Marks, Charbel-Raphael Segerie, Micah Carroll, Andi Peng, Phillip J.K. Christoffersen, Mehul Damani, Stewart Slocum, Usman Anwar, Anand Siththaranjan, Max Nadeau, Eric J Michaud, Jacob Pfau, Dmitrii Krasheninnikov, Xin Chen, Lauro Langosco, Peter Hase, Erdem Biyik, Anca Dragan, David Krueger, Dorsa Sadigh, and Dylan Hadfield-Menell. Open problems and fundamental limitations of reinforcement learning from human feedback. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856.

[6] Souradip Chakraborty, Jiahao Qiu, Hui Yuan, Alec Koppel, Dinesh Manocha, Furong Huang, Amrit Bedi, and Mengdi Wang. MaxMin-RLHF: Alignment with diverse human preferences. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 6116–6135. PMLR, 21–27 Jul 2024.

[7] Chao Chen, Kai Liu, Ze Chen, Yi Gu, Yue Wu, Mingyuan Tao, Zhihang Fu, and Jieping Ye. INSIDE: LLMs' internal states retain the power of hallucination detection. In *The Twelfth International Conference on Learning Representations*, 2024.

[8] Baiyun Cui, Yingming Li, Yaqing Zhang, and Zhongfei Zhang. Text coherence analysis based on deep neural network. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, CIKM '17, page 2027–2030, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450349185. doi: 10.1145/3132847.3133047.

[9] DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *CoRR*, abs/2501.12948, 2025. doi: 10.48550/ARXIV.2501.12948.

[10] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurélien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Rozière, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Grégoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel M. Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, and et al. The llama 3 herd of models. *CoRR*, abs/2407.21783, 2024.

[11] Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. Model alignment as prospect theoretic optimization. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

[12] Jun Gao, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tieyan Liu. Representation degeneration problem in training natural language generation models. In *International Conference on Learning Representations*, 2019.

[13] Shivank Garg, Ayush Singh, Shweta Singh, and Paras Chopra. IPO: your language model is secretly a preference classifier. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pages 19425–19441. Association for Computational Linguistics, 2025.

[14] Quentin Garrido, Randall Balestriero, Laurent Najman, and Yann LeCun. Rankme: assessing the downstream performance of pretrained self-supervised representations by their rank. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

[15] Aryo Pradipta Gema, Joshua Ong Jun Leang, Giwon Hong, Alessio Devoto, Alberto Carlo Maria Mancino, Rohit Saxena, Xuanli He, Yu Zhao, Xiaotang Du, Mohammad Reza Ghasemi Madani, Claire Barale, Robert McHardy, Joshua Harris, Jean Kaddour, Emile Van Krieken, and Pasquale Minervini. Are we done with MMLU? In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5069–5096. Association for Computational Linguistics, April 2025. ISBN 979-8-89176-189-6. doi: 10.18653/v1/2025.naacl-long.262.

[16] Nathan Godey, Éric Villemonte de la Clergerie, and Benoît Sagot. Why do small language models underperform? studying language model saturation via the softmax bottleneck. In *First Conference on Language Modeling*, 2024.

[17] Shangmin Guo, Biao Zhang, Tianlin Liu, Tianqi Liu, Misha Khalman, Felipe Llinares, Alexandre Rame, Thomas Mesnard, Yao Zhao, Bilal Piot, et al. Direct language model alignment from online ai feedback. *arXiv preprint arXiv:2402.04792*, 2024.

[18] Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. OlympiadBench: A challenging benchmark for promoting AGI with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3828–3850. Association for Computational Linguistics, August 2024. doi: 10.18653/v1/2024.acl-long.211.

[19] Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. OlympiadBench: A challenging benchmark for promoting AGI with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3828–3850. Association for Computational Linguistics, August 2024. doi: 10.18653/v1/2024.acl-long.211.

[20] Jinwen He, Yujia Gong, Zijin Lin, Cheng'an Wei, Yue Zhao, and Kai Chen. LLM factoscope: Uncovering LLMs' factual discernment through measuring inner states. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 10218–10230. Association for Computational Linguistics, August 2024.

[21] Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

[22] Seungone Kim, Juyoung Suk, Shayne Longpre, Bill Yuchen Lin, Jamin Shin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. Prometheus 2: An open source language model specialized in evaluating other language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 4334–4353. Association for Computational Linguistics, November 2024. doi: 10.18653/v1/2024.emnlp-main.248.

[23] knoveleng. Amc-23. Hugging Face Dataset, 2023. URL `https://huggingface.co/datasets/knoveleng/AMC-23`.

[24] Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V. Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Chris Wilhelm, Luca Soldaini, Noah A. Smith, Yizhong Wang, Pradeep Dasigi, and Hannaneh Hajishirzi. Tülu 3: Pushing frontiers in open language model post-training. *CoRR*, abs/2411.15124, 2024. doi: 10.48550/ARXIV.2411.15124.

[25] Nathan Lambert, Valentina Pyatkin, Jacob Morrison, Lester James V. Miranda, Bill Yuchen Lin, Khyathi Raghavi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, Noah A. Smith, and Hannaneh Hajishirzi. Rewardbench: Evaluating reward models for language modeling. In *Findings of the Association for Computational Linguistics: NAACL 2025, Albuquerque, New Mexico, USA, April 29 - May 4, 2025*, pages 1755–1797. Association for Computational Linguistics, 2025.

[26] Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Lu, Colton Bishop, Ethan Hall, Victor Carbune, Abhinav Rastogi, and Sushant Prakash. RLAIF vs. RLHF: scaling reinforcement learning from human feedback with AI feedback. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024.

[27] Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. *arXiv preprint arXiv:2305.20050*, 2023.

[28] Bill Yuchen Lin, Yuntian Deng, Khyathi Chandu, Abhilasha Ravichander, Valentina Pyatkin, Nouha Dziri, Ronan Le Bras, and Yejin Choi. Wildbench: Benchmarking LLMs with challenging tasks from real users in the wild. In *The Thirteenth International Conference on Learning Representations*, 2025.

[29] Chris Yuhao Liu, Liang Zeng, Yuzhen Xiao, Jujie He, Jiacai Liu, Chaojie Wang, Rui Yan, Wei Shen, Fuxiang Zhang, Jiacheng Xu, Yang Liu, and Yahui Zhou. Skywork-reward-v2: Scaling preference data curation via human-ai synergy. *arXiv preprint arXiv:2507.01352*, 2025.

[30] Mohsen Mesgar and Michael Strube. A neural local coherence model for text quality assessment. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4328–4339, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1464.

[31] Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751, Atlanta, Georgia, June 2013. Association for Computational Linguistics.

[32] OpenAI. Gpt-4o mini: A more efficient multimodal model, 2025. URL `https://openai.com/blog/gpt-4o-mini`.

[33] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022.

[34] 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. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

[35] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019.

[36] David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*, 2024.

[37] Olivier Roy and Martin Vetterli. The effective rank: A measure of effective dimensionality. In *2007 15th European signal processing conference*, pages 606–610. IEEE, 2007.

[38] Mark Rudelson and Roman Vershynin. Sampling from large matrices: An approach through geometric functional analysis. *J. ACM*, 54(4):21–es, July 2007. ISSN 0004-5411. doi: 10.1145/1255443.1255449.

[39] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *CoRR*, abs/2402.03300, 2024. doi: 10.48550/ARXIV.2402.03300.

[40] Prasann Singhal, Tanya Goyal, Jiacheng Xu, and Greg Durrett. A long way to go: Investigating length correlations in RLHF. In *First Conference on Language Modeling*, 2024.

[41] Joar Skalse, Nikolaus H. R. Howe, Dmitrii Krasheninnikov, and David Krueger. Defining and characterizing reward hacking. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22. Curran Associates Inc., 2022. ISBN 9781713871088.

[42] Jiashuo Wang, Haozhao Wang, Shichao Sun, and Wenjie Li. Aligning language models with human preferences via a bayesian approach. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23. Curran Associates Inc., 2023.

[43] Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Lingpeng Kong, Qi Liu, Tianyu Liu, and Zhifang Sui. Large language models are not fair evaluators. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 9440–9450. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.ACL-LONG.511.

[44] Zeqiu Wu, Yushi Hu, Weijia Shi, Nouha Dziri, Alane Suhr, Prithviraj Ammanabrolu, Noah A. Smith, Mari Ostendorf, and Hannaneh Hajishirzi. Fine-grained human feedback gives better rewards for language model training. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23. Curran Associates Inc., 2023.

[45] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *CoRR*, abs/2412.15115, 2024. doi: 10.48550/ARXIV.2412.15115.

[46] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jian Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. *CoRR*, abs/2505.09388, 2025. doi: 10.48550/ARXIV.2505.09388.

[47] Zhilin Yang, Zihang Dai, Ruslan Salakhutdinov, and William W. Cohen. Breaking the softmax bottleneck: A high-rank RNN language model. In *International Conference on Learning Representations*, 2018.

[48] Zhangyue Yin, Qiushi Sun, Zhiyuan Zeng, Qinyuan Cheng, Xipeng Qiu, and Xuanjing Huang. Dynamic and generalizable process reward modeling. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4203–4233, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.212.

[49] Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. Self-rewarding language models. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024.

[50] Qingru Zhang, Liang Qiu, Ilgee Hong, Zhenghao Xu, Tianyi Liu, Shiyang Li, Rongzhi Zhang, Zheng Li, Lihong Li, Bing Yin, Chao Zhang, Jianshu Chen, Haoming Jiang, and Tuo Zhao. Self-rewarding PPO: Aligning large language models with demonstrations only. In *Second Conference on Language Modeling*, 2025.

[51] Yifan Zhang and Team Math-AI. American invitational mathematics examination (aime) 2025, 2025.

[52] Zhenru Zhang, Chujie Zheng, Yangzhen Wu, Beichen Zhang, Runji Lin, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. The lessons of developing process reward models in mathematical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 10495–10516. Association for Computational Linguistics, July 2025. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.547.

[53] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

Appendix Contents

A Cross-Layer Stable Rank Analysis **17**

B Baseline Implementation Details **18**

B.1 Generative Baselines 18

B.2 IPO Implementation 19

C Length-Controlled Preference Analysis **20**

D RewardBench Results **20**

E Best-of-N Complete Results **20**

F SR-GRPO Training Details **21**

G Experimental Details for Metric Analysis **22**

G.1 Statistical Methodology 23

G.2 Metric Definitions 23

G.3 Implementation Details 23

G.4 Correlation Results 23

G.5 Key Findings 23

H Comparison with Other Intrinsic Dimension Metrics **25**

I Additional Alignment Ablations **27**

I.1 Scaling to 8B Models 27

I.2 Alternative RL Algorithms 28

I.3 Pretrained Models as Reward Sources 28

I.4 Stronger Reward-Model and Internal-State Baselines 28

I.5 Longer-Horizon Training 29

I.6 Alternative Intrinsic Dimension Metrics as RL Rewards 29

I.7 Multi-Seed Comparison with Verifiable and Learned Rewards 30

J Context Window Size **31**

K Input Prompt Format **31**

L Stable Rank Example Prompts **32**

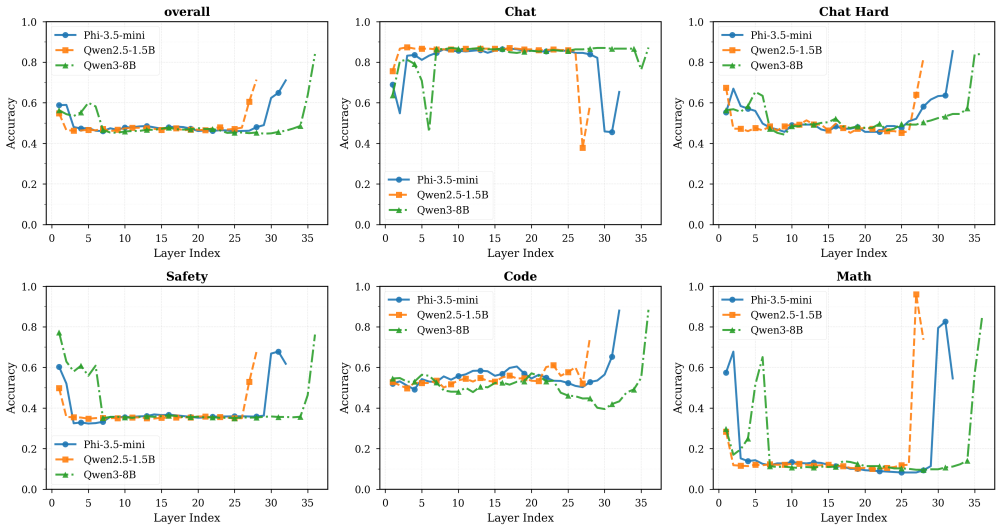

Figure 3: Cross-layer stable rank performance on RewardBench across different model families. We evaluate the stable rank computed from each transformer layer as a reward proxy on RewardBench's six subcategories. Final layers (rightmost) consistently achieve the highest accuracy across models and categories, validating our choice to use final-layer activations in SR-GRPO.

A Cross-Layer Stable Rank Analysis

We analyze how stable rank performance varies across transformer layers by computing $R_{\text{SR}}^{(l)}(y) = \|\mathbf{H}_l\|_F^2 / \|\mathbf{H}_l\|_2^2$ at each layer l and evaluating it as a reward proxy on RewardBench [25]. We test three model families: Qwen3-8B (36 layers) [46], Qwen2.5-1.5B-Instruct (28 layers) [45], and Phi-3.5-mini-instruct (32 layers) [1], measuring accuracy on five subcategories: Chat, Chat Hard, Safety, Code, and Math.

Final layers encode quality signals most effectively. Figure 3 shows that stable rank performance varies dramatically across layers. Early and middle layers (indices 5-25) achieve near-random performance ($\sim$50% accuracy) across all categories, indicating that stable rank computed from these layers carries minimal quality information. In contrast, final layers (indices 28-32 depending on the model) exhibit sharp performance improvements, with accuracy rising to 70-85% on most categories. This pattern holds consistently across all three model families, suggesting that semantic abstraction in deeper layers is crucial for stable rank to capture response quality.

Task-dependent layer sensitivity. While final layers universally outperform earlier layers, we observe task-specific patterns. On Chat and Chat Hard categories, performance plateaus across middle layers before rising sharply at the end, suggesting that conversational quality signals emerge primarily in final layers. For Safety, we observe an interesting U-shape on Qwen3-8B and Phi-3.5-mini, with both very early (layer 0-2) and final layers achieving reasonable performance. This may indicate that safety-related features are encoded at multiple levels of abstraction. Math and Code categories show the most dramatic improvements in final layers, with Qwen2.5-1.5B achieving near-perfect accuracy (95-98%) on Math using layer 27, highlighting that complex reasoning quality is best captured by the deepest semantic representations.

Implications for SR-GRPO. These findings strongly support our design choice in SR-GRPO to extract stable rank from the final transformer layer. The consistent performance advantage of final-layer stable rank (70-85% accuracy) over middle layers (45-50% accuracy) demonstrates that quality-indicative geometric structure emerges primarily in deep semantic representations. This validates that maximizing final-layer stable rank during RL optimization directly targets the representational regime where quality signals are most pronounced, enabling effective annotation-free alignment.

B Baseline Implementation Details

In this section, we provide the prompts and implementation details for all the baselines.

B.1 Generative Baselines

In our experiments, we have two generative baselines: (1) Pointwise Scoring [22] and (2) Pairwise Comparison [22].

B.1.1 Pointwise Scoring Prompt

For the Pointwise Scoring baseline, we use the following prompt structure to score each response on a 1-5 scale:

```
###Task Description:
An instruction (might include an Input inside it), a response to evaluate,
a reference answer that gets a score of 5, and a score rubric representing
a evaluation criteria are given.
1. Write a detailed feedback that assess the quality of the response strictly
   based on the given score rubric, not evaluating in general.
2. After writing a feedback, write a score that is an integer between 1 and 5.
   You should refer to the score rubric.
3. The output format should look as follows: "Feedback: (write a feedback for
   criteria) [RESULT] (an integer number between 1 and 5)"
4. Please do not generate any other opening, closing, and explanations.

###The instruction to evaluate:
{instruction}

###Response to evaluate:
{response}

###Reference Answer (Score 5):
{reference_answer}

###Score Rubrics:
{criteria}
Score 1: {score_description_1}
Score 2: {score_description_2}
Score 3: {score_description_3}
Score 4: {score_description_4}
Score 5: {score_description_5}

###Feedback:
```

We extract the numerical rating from the model's output by parsing the `[RESULT]` token and compare scores to determine the preferred response. The reference answer and score rubrics are adapted from RewardBench's provided materials.

B.1.2 Pairwise Comparison Prompt

For the Pairwise Comparison baseline, we prompt the model to directly compare two responses:

```
###Task Description:
An instruction (might include an Input inside it), a response to evaluate,
and a score rubric representing a evaluation criteria are given.
1. Write a detailed feedback that assess the quality of two responses strictly
   based on the given score rubric, not evaluating in general.
2. After writing a feedback, choose a better response between Response A and
   Response B. You should refer to the score rubric.
3. The output format should look as follows: "Feedback: (write a feedback for
   criteria) [RESULT] (A or B)"
```

```
4. Please do not generate any other opening, closing, and explanations.

###Instruction:
{instruction}

###Response A:
{response_a}

###Response B:
{response_b}

###Score Rubric:
{criteria}

###Feedback:
```

We parse the model's output to extract the preference (A or B) from the `[RESULT]` token. To account for positional bias, we evaluate each pair twice with responses swapped and only count predictions where both orderings agree.

B.2 IPO Implementation

We follow the evaluation protocol from IPO [13] for RewardBench. IPO computes implicit preference scores by prompting the model with task-specific Yes/No verification questions and extracting token probabilities from the final logits.

B.2.1 Task-Specific Evaluation Prompts

We use category-specific verification prompts adapted from the IPO codebase. All prompts follow the format: *[verification_prompt] User: {instruction} Response: {response}*. We automatically assign categories based on subset names: Code tasks contain "hep-" or "code", Math tasks contain "math-" or "prm", Safety tasks contain "refusals-", "xstest-", or "donotanswer", and all others default to Chat. The prompt structure varies by task:

Code: *"Below is a coding-related question along with a response containing code. Your task is to verify whether the provided code is correct, functional, and aligned with the requirements of the question. Answer with just Yes/No"*

Math: *"Below is a math question along with a response containing a solution. Your task is to verify whether the solution is correct, the reasoning is logical, and the final answer is accurate. Answer with just Yes/No"*

Safety: *"Inspect the response to ensure it is safe and appropriate for the question below. Answer Yes/No."*

Chat (default): *"Below is a question, along with a response to that question. Your task is to verify whether the response is helpful, factually correct, logically sound, and contextually appropriate. Answer with just Yes/No"*

B.2.2 Score Computation

Given response y to instruction x, we tokenize the verification prompt and perform a forward pass to obtain logits $\boldsymbol{\ell} \in \mathbb{R}^{|V|}$ at the final position. We extract logits for "Yes" and "No" tokens (including variants with leading spaces), apply softmax normalization, and define the IPO score as:

$$\text{IPO}(y|x) = \frac{\exp(\ell_{\text{Yes}})}{\exp(\ell_{\text{Yes}}) + \exp(\ell_{\text{No}})} \tag{3}$$

The response with higher $\text{IPO}(y|x)$ is predicted as preferred. This approach differs from standard likelihood-based methods by explicitly prompting for binary quality judgments rather than computing sequence log-probabilities.

C Length-Controlled Preference Analysis

We fit logistic regressions on the 2,985 RewardBench pairs to predict the human-preferred response from pairwise differences in stable rank and response length, using Qwen3-8B. The predictors ΔSR and Δlen are standardized per standard deviation. We compare models using SR alone, length alone, and their combination. Table 4 reports the fitted coefficients, standard errors, and pairwise preference accuracy.

Table 4: Logistic regressions predicting the human-preferred response on RewardBench. Predictors are standardized per standard deviation. Model accuracy is reported as a fraction.

Model	**Predictor**	**Coef.**	**SE**	**Model acc.**
SR-only	ΔSR	0.891	0.058	0.840
Length-only	Δlen	-0.360	0.041	0.563
SR + length control	ΔSR	1.310	0.078	0.875
	Δlen	-0.720	0.054	

Length alone achieves 0.563 accuracy. In the joint model, the SR coefficient increases from 0.891 to 1.310 and model accuracy from 0.840 to 0.875, showing that SR retains its preference signal after accounting for response length.

D RewardBench Results

We show the complete performance comparison across different reward methods for RewardBench in Table 5.

Table 5: Complete performance comparison across different reward methods for RewardBench. Pointwise Scoring uses an LLM to grade each response on a 1-5 scale independently, while Pairwise Comparison directly evaluates two responses.

Model	Method	Chat	Chat Hard	Safety	Code	Math	Overall
Qwen2.5 1.5B-Instruct	Pointwise Scoring	37.61	38.36	32.05	43.29	34.45	37.15
	Pairwise Comparison	56.14	50.37	47.59	48.78	64.43	53.46
	IPO	70.48	70.93	48.28	80.08	59.51	65.85
	Stable Rank	87.28	85.70	66.54	73.78	66.44	75.95
Qwen3 0.6B	Pointwise Scoring	37.54	39.10	48.24	38.51	30.42	38.76
	Pairwise Comparison	49.01	43.27	49.00	46.23	48.09	47.12
	IPO	54.99	63.23	48.14	72.46	89.04	65.57
	Stable Rank	55.07	72.87	56.44	79.26	71.14	66.96
Qwen3 8B	Pointwise Scoring	80.54	81.44	87.43	86.19	82.89	83.70
	Pairwise Comparison	77.15	71.80	75.84	73.17	61.96	71.98
	IPO	55.40	85.93	86.18	91.67	70.92	78.02
	Stable Rank	87.22	84.01	76.20	88.41	84.34	84.04
Llama-3.1-Instruct 8B	Pointwise Scoring	51.73	49.29	64.10	65.04	51.67	56.37
	Pairwise Comparison	55.52	63.47	57.17	54.77	52.34	56.65
	IPO	56.07	60.56	74.75	65.55	33.78	58.14
	Stable Rank	70.63	76.50	43.90	75.81	74.94	68.36
Phi-3.5-instruct mini	Pointwise Scoring	32.01	37.88	57.08	48.57	36.68	42.44
	Pairwise Comparison	77.10	62.74	54.28	52.43	63.98	62.11
	IPO	49.32	66.48	54.03	64.43	97.09	66.27
	Stable Rank	65.35	84.60	61.82	88.01	54.59	70.87

E Best-of-N Complete Results

We provide complete Best-of-N decoding results comparing stable rank selection against random selection across four models and five benchmarks. For each configuration, we sample N candidate

responses and either select randomly (Random@N) or choose the response with highest stable rank (Best@N). We report two relative metrics: ΔRand. measures improvement over random selection at the same N, and Δ@1 measures improvement over greedy decoding (N=1).

Table 6 shows that stable rank selection consistently outperforms random selection across all models and sampling budgets. The advantage is most pronounced on smaller models: Llama-3.2-1B achieves +33.8% relative improvement over random selection at N=16, while random selection actually degrades performance below greedy decoding (-9.9%). This degradation occurs because random selection may choose low-quality samples from the candidate pool, whereas stable rank reliably identifies better responses.

Notably, the gains from stable rank selection increase with N for most models, suggesting that larger candidate pools provide more opportunities for quality differentiation. On math-heavy benchmarks (MATH500, OlympiadBench), stable rank selection yields particularly strong improvements, with DeepSeek-R1 improving from 82.8% to 86.8% on MATH500 and from 43.6% to 59.7% on OlympiadBench at N=16.

Table 6: Complete Best-of-N decoding results (%). We compare stable rank selection (SR) against random selection (Rand.) at each N. Δ: relative improvement of SR over Rand., computed as $(\text{SR} - \text{Rand.})/\text{Rand.} \times 100\%$.

		Qwen2.5-1.5B		**Phi-3.5-mini**		**Llama-3.2-1B**		**DeepSeek-R1-1.5B**	
N	**Benchmark**	Rand.	SR	Rand.	SR	Rand.	SR	Rand.	SR
4	GPQA	19.2	20.3	23.2	23.7	21.2	20.7	25.8	26.2
	MMLU-Re.	46.6	47.7	60.3	61.2	30.0	32.6	45.9	46.0
	MATH	51.0	55.6	49.8	50.8	23.8	25.0	80.0	84.0
	Olymp.	23.6	28.4	18.2	20.0	9.3	12.1	51.9	54.8
	AMC23	27.5	40.0	20.0	25.0	15.0	22.5	80.0	82.5
	Avg.	*33.6*	*38.4*	*34.3*	*36.2*	*19.9*	*22.6*	*56.7*	*58.7*
	Δ	+14.4%		+5.4%		+13.6%		+3.6%	
8	GPQA	19.7	21.2	27.3	30.8	21.2	25.3	25.8	26.8
	MMLU-Re.	46.6	47.9	59.5	61.4	31.7	32.6	46.1	48.4
	MATH	55.2	57.8	50.0	52.4	22.2	28.8	83.0	83.4
	Olymp.	22.8	29.0	17.8	19.6	9.6	12.2	52.6	55.0
	AMC23	25.0	40.0	20.0	25.0	7.5	25.0	75.0	85.0
	Avg.	*33.9*	*39.2*	*34.9*	*37.8*	*18.4*	*24.8*	*56.5*	*59.7*
	Δ	+15.7%		+8.4%		+34.2%		+5.7%	
16	GPQA	20.2	22.2	27.8	29.3	22.2	32.8	22.5	27.3
	MMLU-Re.	46.7	48.2	60.1	62.1	31.7	32.5	46.9	47.2
	MATH	53.4	59.6	49.6	52.8	22.2	29.2	82.2	86.8
	Olymp.	21.0	32.4	18.7	21.8	10.4	12.9	55.1	59.7
	AMC23	40.0	42.5	17.5	27.5	12.5	25.0	82.5	82.5
	Avg.	*36.3*	*41.0*	*34.7*	*38.7*	*19.8*	*26.5*	*57.8*	*60.7*
	Δ	+13.0%		+11.4%		+33.8%		+5.0%	

F SR-GRPO Training Details

We provide the training configurations for SR-GRPO experiments on both models. All experiments are conducted on 4×NVIDIA H800 GPUs.

Qwen2.5-1.5B-Instruct. We train for 400 steps using LoRA with rank r=16 and α=32, targeting all attention and MLP projection layers. We use a learning rate of 10^{-6} with cosine scheduling and 10% warmup. Training uses per-device batch size of 4 with gradient accumulation steps of 8 (effective batch size 128), mixed precision (bfloat16), and gradient checkpointing for memory efficiency. The maximum prompt and completion lengths are both set to 4096 tokens.

DeepSeek-R1-Distill-Qwen-1.5B. We train for 300 steps using the same LoRA configuration (r=16, α=32). Due to the model's longer reasoning traces, we use per-device batch size of 2 with

Algorithm 1 SR-GRPO Training

Require: Dataset $\mathcal{D}$, initial policy π_ϕ, frozen reference π_{ref}, group size K, KL coefficient β
Ensure: Optimized policy π_ϕ

```
while not converged do
    Sample batch {x_i} ~ D
    for each prompt x_i do
        Sample responses {y_{i,k}}_{k=1}^K ~ π_φ(·|x_i)
        Compute rewards r_{i,k} ← SR(H_{i,k}; π_ref)
        Compute advantages A_{i,k} ← (r_{i,k} − μ_i)/(σ_i + ε)
    end for
    Update π_φ via Eq. (2)
end while
```

gradient accumulation steps of 16 (effective batch size 128). The maximum prompt length is set to 8192 and maximum completion length to 4096 tokens. All other hyperparameters remain identical to the Qwen configuration.

Stable Rank Computation. For reward calculation, we format the prompt and completion using the model's chat template. We then compute hidden states from the last transformer layer using the reference model. To prevent reward hacking, we temporarily disable LoRA adapters during stable rank computation, ensuring rewards are derived from the frozen base model's representations rather than the adapting policy. Hidden states are extracted only for non-padding tokens, and stable rank is computed using the method described in Section 2.

Common Settings. Both models are trained on the SmolTalk2 [3] preference dataset using paged AdamW optimizer with 8-bit quantization. We set weight decay to 0, maximum gradient norm to 1.0, and LoRA dropout to 0.1. Rewards are not normalized across batches. For GRPO-specific hyperparameters, we use the default values from the HuggingFace TRL library: group size $K = 8$, KL penalty coefficient $\beta = 0.04$, and numerical stability constant $\epsilon = 10^{-8}$ for advantage standardization.

Table 7: SR-GRPO training configurations.

Hyperparameter	**Qwen2.5-1.5B**	**DeepSeek-R1-1.5B**
Training steps	400	300
Batch size (per device)	4	2
Gradient accumulation	8	16
Effective batch size	128	128
Learning rate	10^{-6}	10^{-6}
Warmup ratio	0.1	0.1
Group size (K)	8	8
KL coefficient (β)	0.04	0.04
LoRA rank (r)	16	16
LoRA alpha (α)	32	32
Max prompt length	4096	8192
Max completion length	4096	4096
Precision	bfloat16	bfloat16

G Experimental Details for Metric Analysis

We analyze the relationship between stable rank and text quality using the RewardBench dataset [25], which contains 2,985 prompts, each with one chosen and one rejected response ($N = 5{,}970$ responses). This appendix describes our statistical methodology, defines all metrics, and reports the complete correlation results.

G.1 Statistical Methodology

We employ two complementary analyses to characterize the relationship between stable rank and text metrics:

Sample-level correlation. We treat each of the $N = 5{,}970$ responses as an independent observation. For metric M and stable rank S, we compute Pearson correlation $r = \mathrm{corr}(M, S)$ and Spearman rank correlation ρ. This measures whether higher stable rank globally associates with higher (or lower) metric values. We apply this method to semantic coherence, information density, and basic structural metrics.

Paired-difference analysis. Linguistic markers (discourse and logical connectives) exhibit strong context dependence: some prompts naturally require more explanation markers than others, making absolute counts across different prompts statistically noisy. To control for this variation, we work at the pair level. For each prompt i with chosen response y_c and rejected response y_r, we compute differences

$$\Delta M_i = M(y_c) - M(y_r), \quad \Delta S_i = S(y_c) - S(y_r), \tag{4}$$

and correlate ΔM with ΔS across all $N = 2{,}985$ pairs. This measures whether, within the same context, the response with higher stable rank tends to use more (positive correlation) or fewer (negative correlation) of a given linguistic feature.

G.2 Metric Definitions

We measure text quality along three axes: semantic coherence, information density, and linguistic structure. Tables 8–10 define the metrics formally. For response y, let T denote its token sequence (N_{tokens} tokens), $s_1, \ldots, s_n$ its sentences with embeddings $v_1, \ldots, v_n$ (from all-MiniLM-L6-v2 [35]), and v_p the prompt embedding.

G.3 Implementation Details

Sentence embeddings. We use the all-MiniLM-L6-v2 [35] model from the sentence-transformers library, which produces 384-dimensional embeddings. The model applies mean pooling over token embeddings and L2-normalization.

Sentence tokenization. We tokenize responses into sentences using NLTK [1], with a regex-based fallback (splitting on sentence boundaries while handling common abbreviations). Each response contains n sentences indexed $1, \ldots, n$.

Syllable counting. For readability metrics, we count syllables using the pyphen [2] library (dictionary-based hyphenation for English). Complex words are defined as having ≥ 3 syllables following standard practice.

Marker detection. Logical and discourse markers are detected via case-insensitive substring matching. For each marker category, we report: (1) raw count, (2) count normalized per sentence, and (3) count normalized per 100 tokens as $(c/N_{\text{tokens}}) \times 100$.

G.4 Correlation Results

Tables 11 and 12 report Pearson (r) and Spearman (ρ) correlations with p-values. Positive correlations indicate that higher metric values associate with higher stable rank; negative correlations indicate the opposite.

G.5 Key Findings

The correlation analysis reveals three main patterns. First, stable rank shows strongest positive associations with semantic coherence metrics: progression score ($\rho = 0.313$) and QA alignment

[1] https://www.nltk.org/
[2] https://pyphen.org/

Table 8: Semantic coherence metrics. All metrics use sentence embeddings $v_1, \ldots, v_n \in \mathbb{R}^{384}$ from all-MiniLM-L6-v2.

Metric	Formula	Interpretation
Intra-response coherence		
Progression score	$\frac{1}{n-1}\sum_{i=1}^{n-1}\cos(v_i, v_{i+1}) - \frac{1}{\lvert\mathcal{P}\rvert}\sum_{(i,j)\in\mathcal{P}}\cos(v_i, v_j)$ where $\mathcal{P} = \{(i,j) \mid j > i+1\}$	Measures forward momentum by comparing adjacent vs non-adjacent sentence similarity. Higher values indicate natural progression where each sentence builds on the previous one.
Adjacent similarity	$\bar{s}_{\text{adj}} = \frac{1}{n-1}\sum_{i=1}^{n-1}\cos(v_i, v_{i+1})$.	Captures smooth local transitions between consecutive sentences. Statistics characterize different aspects of coherence stability. Minimum detects weak links.
Coherence mean	$\bar{s}_{\text{adj}} = \frac{1}{n-1}\sum_{i=1}^{n-1}\cos(v_i, v_{i+1})$	Average cosine similarity between consecutive sentences. Higher values indicate smoother local transitions throughout the response.
Coherence minimum	$\min_{i\in[1,n-1]}\cos(v_i, v_{i+1})$	Minimum adjacent similarity, identifying the weakest link in the argument chain. Low values indicate at least one abrupt topic shift or incoherent transition.
Semantic density	$\frac{1}{\binom{n}{2}}\sum_{1\le i<j\le n}[1-\cos(v_i, v_j)]$	Average pairwise cosine distance. Higher values indicate scattered content; lower values indicate tightly focused responses.
Semantic variance	$\frac{1}{384}\sum_{d=1}^{384}\text{Var}([v_{1,d}, \ldots, v_{n,d}])$	Measures spread across each of the 384 embedding dimensions, then averages. High variance signals erratic topic shifts.
Topic jumps	Count of pairs where $\cos(v_i, v_{i+1}) < \mu - 2\sigma$, with $\mu = \bar{s}_{\text{adj}}$ and $\sigma = \text{std}(\{\cos(v_i, v_{i+1})\})$	Detects abrupt drops in adjacent similarity that fall significantly below baseline. Identifies sudden coherence breaks.
Prompt–response alignment (using prompt embedding v_p)		
QA alignment	Compute $a_i = \cos(v_p, v_i)$ for $i = 1, \ldots, n$. Report $\max(a_i)$, $\text{mean}(a_i)$, $\min(a_i)$.	Measures how well each response sentence aligns with the prompt. Max, mean, and min capture peak relevance, overall relevance, and worst-case relevance.
QA consistency	$\max(0, 1 - \text{std}(\{a_i\}_{i=1}^{n}))$	Measures stable relevance throughout generation. Lower std means the response maintains consistent focus without drifting.

Table 9: Information density and diversity metrics. These capture lexical richness and redundancy.

Metric	Formula / Computation	Interpretation
Lexical variety		
Type–token ratio	$\lvert V\rvert / N_{\text{tokens}}$ where V = set of unique tokens	Measures vocabulary richness. Higher ratios indicate more diverse word usage and richer language expression.
Moving-avg TTR	$\frac{1}{N_{\text{tokens}}-w+1}\sum_{i=1}^{N_{\text{tokens}}-w+1}\frac{\lvert V_i\rvert}{w}$ with window size $w = 100$, sliding by 1 token	Stabilized diversity measure that reduces length bias. Averages TTR over local windows for robust estimates.
n-gram diversity	For $n \in \{2, 3, 4\}$: $\frac{\#\text{ unique } n\text{-grams}}{\#\text{ total } n\text{-grams}}$ over token sequence	Detects phrase-level repetition. Lower diversity indicates phrase reuse; higher diversity suggests varied expression.
Repetition rate	Fraction of 10-token windows appearing ≥ 2 times	Measures sustained redundant phrases. Captures copy-paste patterns or looping behavior.
Information content and redundancy		
Compression ratio	$\frac{\lvert zlib.compress(y)\rvert}{\lvert y\rvert}$	Proxy for information density via compressibility. Higher ratios (closer to 1.0) indicate less compressible text; lower ratios suggest redundancy.
Text entropy	$H = -\sum_c p(c)\log_2 p(c)$ where $p(c)$ is character frequency	Character-level Shannon entropy. Measures inherent randomness and information content.
Model confidence (using Qwen2.5-7B)		
Perplexity	$\exp\left(-\frac{1}{N_{\text{tokens}}}\sum_{i=1}^{N_{\text{tokens}}}\log p(t_i \mid t_{<i})\right)$	Measures how surprising the text is to the base LM. Lower perplexity indicates more natural language patterns.
Model uncertainty	$\frac{1}{N_{\text{tokens}}}\sum_{i=1}^{N_{\text{tokens}}}H(p(t_{i+1} \mid t_{\le i}))$ where H is entropy	Average next-token entropy. Higher values indicate less confidence, more random token choices.

Table 10: Linguistic structure metrics. Logical and discourse markers indicate explicit reasoning patterns; structural elements capture formatting and organization; readability scores measure text complexity.

Category	Subcategory	Representative Keywords	Normalization
Logical	Causal	*because*, *therefore*, *thus*, *consequently*, *leads to*	Raw, per-sent, per-100-tok
	Conditional	*if*, *when*, *unless*, *provided that*, *assuming*	Raw, per-sent, per-100-tok
	Inference	*implies*, *suggests*, *indicates*, *proves*, *demonstrates*	Raw, per-sent, per-100-tok
	Comparison	*similar to*, *different from*, *versus*, *compared to*	Raw, per-sent, per-100-tok
Discourse	Contrastive	*however*, *but*, *although*, *nevertheless*, *conversely*	Raw, per-sent, per-100-tok
	Additive	*moreover*, *also*, *furthermore*, *in addition*, *besides*	Raw, per-sent, per-100-tok
	Temporal	*then*, *next*, *subsequently*, *meanwhile*, *eventually*	Raw, per-sent, per-100-tok
	Exemplification	*for example*, *such as*, *namely*, *specifically*, *e.g.*	Raw, per-sent, per-100-tok
Structural	Enumeration	*first*, *second*, *step 1*, *finally*, *to begin*, *in conclusion*	Raw, per-sent, per-100-tok
	Formatting	Markdown headers (`#`), numbered lists (`1.`), bullets (`-`), code blocks	Raw, per-sent, per-100-tok

consistency ($\rho = 0.316$). Responses maintaining smooth topic flow and stable prompt alignment achieve higher stable rank. Conversely, coherence standard deviation exhibits strong negative correlation ($\rho = -0.356$), indicating that inconsistent adjacent similarities reduce stable rank.

Second, stable rank distinguishes information density from verbosity. While absolute measures like token count ($\rho = -0.294$) and sentence count ($\rho = -0.368$) correlate negatively, efficiency metrics show positive associations: lexical diversity ($\rho = 0.238$) and compression ratio ($\rho = 0.233$). The negative correlation with sentence count is particularly revealing: responses with many short sentences fail to build rich semantic representations. Combined with positive correlation with average sentence length ($\rho = 0.085$), this suggests stable rank rewards well-developed sentences over fragmented text.

Third, discourse markers reveal nuanced reasoning patterns in paired-difference analysis. Most marker categories show negative correlations: additive markers ($\rho = -0.156$), conditional markers ($\rho = -0.163$), and enumeration patterns ($\rho = -0.148$), with total marker count negatively associated ($\rho = -0.204$). Responses relying heavily on common connective words tend to receive lower stable rank, consistent with favoring compact information content over explicit signaling. However, contrastive and causal markers follow a different pattern: positive Pearson correlations (contrastive $r = 0.187$, causal $r = 0.139$) but weak Spearman correlations (contrastive $\rho = 0.067$, causal $\rho \approx 0$). These markers often appear at branch points in arguments or when cause-effect relations are introduced, corresponding to deeper reasoning structure than simple additive or enumerative markers. The weak Spearman correlations indicate that stable rank is sensitive to whether such markers are present but does not increase monotonically with frequency, consistent with the view that contrastive and causal connectives are most informative when they highlight key reasoning steps rather than being used uniformly as a stylistic choice.

H Comparison with Other Intrinsic Dimension Metrics

While stable rank has shown strong performance as a reward signal, other intrinsic dimension metrics exist that measure the effective dimensionality of neural representations. We compare stable rank against three alternative metrics: condition number, PCA-based 95% variance dimension, and effective rank. All metrics are computed on the final hidden layer of Qwen3-8B[46] using the same RewardBench evaluation protocol.

Metrics. We evaluate four intrinsic dimension metrics:

- **Stable rank**: $\text{sr}(\mathbf{X}) = \|\mathbf{X}\|_F^2 / \|\mathbf{X}\|_2^2$, the ratio of squared Frobenius norm to squared spectral norm.
- **Effective rank** [37]: $\text{er}(\mathbf{X}) = \exp(H(\tilde{\sigma}))$, where $H(\tilde{\sigma})$ is the Shannon entropy of the normalized singular value distribution $\tilde{\sigma}_i = \sigma_i / \sum_j \sigma_j$.
- **Condition number**: $\kappa(\mathbf{X}) = \sigma_{\max} / \sigma_{\min}$, the ratio of largest to smallest singular value. We use $1/\kappa$ as the score so that higher values indicate better quality.

Table 11: Sample-level correlations between stable rank and text metrics ($N = 5,970$). Progression score ($r = 0.299$) and QA consistency ($r = 0.258$) show strongest positive associations; coherence std ($r = -0.298$) and total markers ($r = -0.281$) show strongest negative associations. All $p < 0.001$ except where noted.

Metric	**Pearson** r	p	**Spearman** ρ	p
Semantic coherence and alignment				
Progression score	0.299	<0.001	0.313	<0.001
Coherence minimum	0.273	<0.001	0.299	<0.001
QA alignment consistency	0.258	<0.001	0.316	<0.001
Avg adjacent similarity	0.250	<0.001	0.267	<0.001
Avg nonadjacent similarity	0.259	<0.001	0.278	<0.001
Coherence mean	0.233	<0.001	0.250	<0.001
Semantic density	−0.250	<0.001	−0.252	<0.001
Semantic variance	−0.272	<0.001	−0.302	<0.001
Coherence std	−0.298	<0.001	−0.356	<0.001
Information density and diversity				
Lexical diversity (TTR)	0.217	<0.001	0.238	<0.001
Moving-avg TTR	0.133	<0.001	0.139	<0.001
Compression ratio	0.125	<0.001	0.233	<0.001
2-gram diversity	0.127	<0.001	0.189	<0.001
3-gram diversity	0.099	<0.001	0.188	<0.001
4-gram diversity	0.075	<0.001	0.173	<0.001
Repetition rate	−0.015	0.241	−0.109	<0.001
Text entropy	−0.066	<0.001	−0.145	<0.001
Unique tokens	−0.221	<0.001	−0.286	<0.001
Length and structure				
Length (tokens)	−0.211	<0.001	−0.294	<0.001
Sentence count	−0.218	<0.001	−0.368	<0.001
Has headers	−0.231	<0.001	−0.276	<0.001
Avg sentence length	0.139	<0.001	0.085	<0.001
Code blocks (per sent)	0.078	<0.001	0.072	<0.001
Complex word ratio	−0.093	<0.001	−0.185	<0.001

Table 12: Paired-difference correlations between marker differences and stable-rank differences ($N = 2,985$ pairs). Positive correlation means chosen responses have more markers when stable rank is higher. Most markers show negative associations, indicating chosen responses use fewer markers.

Marker Type (per 100 tokens)	**Pearson** r	p	**Spearman** ρ	p
Positive associations				
Contrastive discourse	0.187	<0.001	0.067	<0.001
Causal logical	0.139	<0.001	−0.002	0.926
Logical markers (total)	0.077	<0.001	−0.042	0.024
Total markers	0.039	0.034	−0.063	<0.001
Negative associations				
Comparison logical	−0.034	0.061	−0.078	<0.001
Exemplification discourse	−0.057	0.002	−0.084	<0.001
Inference logical	−0.065	<0.001	−0.115	<0.001
Temporal discourse	−0.106	<0.001	−0.122	<0.001
Enumeration structural	−0.122	<0.001	−0.148	<0.001
Additive discourse	−0.166	<0.001	−0.156	<0.001
Conditional logical	−0.163	<0.001	−0.163	<0.001
Total markers (raw)	−0.149	<0.001	−0.204	<0.001

- **PCA 95% variance**: The number of principal components needed to capture 95% of the variance in the hidden states.

Table 13: RewardBench accuracy (%) using different intrinsic dimension metrics as reward proxies (Qwen3-8B, final layer). Higher is better.

Metric	Chat	Chat Hard	Safety	Code	Math	Overall
Condition number	75.96	31.49	31.32	28.46	12.98	36.04
PCA 95% variance	91.23	74.90	45.45	70.02	27.96	61.91
Effective rank	87.72	59.51	36.40	76.12	12.75	54.50
Stable rank	**87.23**	**84.02**	**76.20**	**88.41**	**84.34**	**84.04**

Results. Table 13 shows that stable rank outperforms all alternative metrics across every category, achieving 84.04% overall accuracy compared to 61.91% for PCA 95% variance, 54.50% for effective rank, and 36.04% for condition number. The gap is largest on challenging categories: stable rank reaches 84.34% on Math while effective rank and condition number fall below 13%, and 76.20% on Safety while alternatives remain below 46%.

Among baseline metrics, PCA 95% variance performs best overall but struggles on Math (27.96%) and Safety (45.45%). Effective rank shows moderate Code performance (76.12%) but fails on Math (12.75%). Condition number performs poorly across all categories, suggesting that extreme singular value ratios are unreliable quality indicators.

Analysis. Stable rank outperforms alternatives due to two key properties. First, it aggregates information across the entire singular value spectrum through the Frobenius norm, making it robust to outliers that affect condition number. Second, it balances representational richness (Frobenius norm) with coherence (spectral norm): high-quality responses activate diverse semantic dimensions in a structured manner, while low-quality responses either collapse into low-dimensional representations or exhibit erratic noise. The entropy weighting in effective rank and discrete counting in PCA 95% variance appear less suited for capturing these quality distinctions.

I Additional Alignment Ablations

We report additional experiments that evaluate the robustness of stable rank beyond the main 1.5B GRPO setting. These studies vary the model scale, optimization algorithm, reward source, baseline family, training horizon, and intrinsic dimension metric. Unless otherwise stated, all experiments use the same benchmark suite as Section 4: GPQA and MMLU-redux for STEM reasoning, MATH500/AIME/OlympiadBench/AMC for mathematical reasoning, and WildBench for open-ended chat quality.

I.1 Scaling to 8B Models

We evaluate whether the stable-rank reward remains effective beyond the 1.5B models used in the main alignment experiments. Larger instruction-tuned models may already have more robust representations, so improvements at this scale provide a stronger test of whether the signal transfers across model capacity. We run the same RL procedure on two 8B policies, Qwen3-8B [46] and Llama-3.1-8B-Instruct [10], and compare SR-GRPO with the base model, a learned reward-model baseline, and IPO under the evaluation protocol in Section 4.

Results. Table 14 shows that SR-GRPO improves both 8B policies and is strongest or tied-strongest across most reported metrics. On Qwen3-8B, SR-GRPO outperforms both RM and IPO on every benchmark, including GPQA, MATH, AIME, and WildBench. On Llama-3.1-8B-Instruct, the gains are smaller but still positive on MMLU, MATH, OlympiadBench, AMC, and WildBench. This suggests that stable rank is not restricted to small models, while also indicating that the magnitude of gains is model-dependent.

Table 14: Alignment results on 8B models. SR-GRPO transfers beyond the original 1.5B setting. WildBench scores are reported as WB-Elo.

Model	Method	GPQA	MMLU	MATH	AIME	Olymp.	AMC	WB-Elo
Qwen3-8B	Base	51.0	75.1	86.6	23.3	59.4	65.0	1004
	+ RM	53.0	71.4	86.8	30.0	59.6	65.0	1017
	+ IPO	54.4	72.1	87.0	26.7	60.7	65.5	1025
	+ SR-GRPO	**55.6**	**75.3**	**88.0**	**33.3**	**62.2**	**67.5**	**1027**
Llama-3.1-8B-Instruct	Base	20.2	49.5	49.0	10.0	20.0	30.0	1036
	+ RM	**23.7**	49.3	46.6	**13.3**	19.3	25.0	1048
	+ IPO	21.2	47.8	47.8	**13.3**	19.9	20.0	1041
	+ SR-GRPO	21.2	**50.1**	**50.4**	**13.3**	**23.6**	**32.5**	**1049**

I.2 Alternative RL Algorithms

SR-GRPO uses GRPO because it is simple and avoids training a separate value function. However, stable rank itself is a scalar reward computed from hidden states and can be paired with other online optimization algorithms. We test this by using the same stable-rank reward with GRPO, RLOO [2], and Online DPO [17] on Qwen2.5-1.5B-Instruct. The reward source and evaluation suite are fixed, and only the policy optimization algorithm changes.

Results. As shown in Table 15, all three algorithms improve over the base model on most reasoning metrics. SR-RLOO improves GPQA, MMLU, MATH, AIME, OlympiadBench, and AMC, while SR-OnlineDPO gives the strongest gains on GPQA, MATH, AMC, and WildBench. SR-GRPO remains competitive and provides the best balance among the three in the main experiments. These results indicate that stable rank is not a GRPO-specific artifact; it can serve as a reward signal for multiple RL or online preference-optimization algorithms.

Table 15: Stable rank reward with different RL algorithms on Qwen2.5-1.5B-Instruct.

Method	GPQA	MMLU	MATH	AIME	Olymp.	AMC	WB-Elo
Base	19.0	47.7	48.0	3.3	25.5	35.0	1036
SR-RLOO	21.2	**48.8**	51.2	**16.7**	**26.7**	37.5	1039
SR-OnlineDPO	**22.7**	47.3	**54.2**	13.3	25.6	**40.0**	**1165**
SR-GRPO	21.2	47.7	52.4	13.3	26.4	37.5	1062

I.3 Pretrained Models as Reward Sources

We next evaluate whether stable rank requires preference-tuned representations. If the signal only emerged after SFT, RLHF, or RLVR, then stable rank would be less useful as an intrinsic geometric property of pretrained language models. We test this in two ways. First, we evaluate stable rank as a zero-shot RewardBench proxy using frozen pretrained base models. Second, we run SR-GRPO where the policy is initialized from an SFT checkpoint, while the stable-rank reward is computed from a frozen pretrained base model. This separates the reward source from preference-tuned representations while still using a policy that can follow instructions.

Results. Stable rank is already informative before post-training, achieving 70.39% RewardBench accuracy with Qwen2.5-1.5B-Base and 72.56% with Qwen3-8B-Base. Table 16 further shows that RL with a pretrained reward source improves downstream performance steadily. For Qwen2.5-1.5B, GPQA improves from 15.2 to 21.2 and MATH from 48.0 to 52.4 by cp1500. For Qwen3-8B, GPQA improves from 31.3 to 40.4 and AMC from 30.0 to 50.0. These results suggest that the geometric signal exists before preference tuning, though post-training may still strengthen or reshape how stable rank aligns with specific task regimes.

I.4 Stronger Reward-Model and Internal-State Baselines

The main experiments compare SR-GRPO with a size-matched learned reward model and several self-evaluation baselines. We further evaluate stronger reward sources and another internal-state

Table 16: SR-GRPO using frozen pretrained base models as reward sources. The policy is initialized from an SFT checkpoint, while stable rank is computed from a frozen pretrained reward source. RB denotes RewardBench overall accuracy.

Model	Checkpoint	RB	GPQA	MMLU	MATH	AIME	Olymp.	AMC	WB-Elo
Qwen2.5-1.5B	SFT	72.0	15.2	46.6	48.0	10.0	23.0	25.0	909
	SR-GRPO cp500	72.5	17.7	47.9	49.4	10.0	23.6	27.5	909
	SR-GRPO cp1000	72.3	18.7	47.9	50.0	13.3	26.8	29.6	918
	SR-GRPO cp1500	**72.5**	**21.2**	**48.0**	**52.4**	**13.3**	**26.8**	**30.4**	**920**
Qwen3-8B	SFT	81.3	31.3	56.8	48.0	13.3	29.6	30.0	1045
	SR-GRPO cp500	81.6	31.8	57.0	49.0	16.7	33.0	42.5	1108
	SR-GRPO cp1000	**82.7**	34.3	57.5	50.6	20.0	33.7	47.5	1112
	SR-GRPO cp1500	**82.7**	**40.4**	**58.3**	**52.2**	**23.3**	**38.5**	**50.0**	**1116**

method to test whether the comparison changes with a broader baseline set. Specifically, we add Skywork-Reward-Llama-3.1-8B [29], a larger learned reward model than the 1.7B reward model used in the main table, and EndoRM, an internal-state reward baseline. All methods are trained and evaluated under the same alignment protocol on Qwen2.5-1.5B-Instruct.

Results. Table 17 shows that stronger baselines recover some reasoning gains but have weaker overall trade-offs. EndoRM improves MATH and WildBench but hurts GPQA and AMC. Skywork-8B improves MATH and AIME but substantially degrades WildBench Elo and AMC, suggesting that a stronger learned reward model does not automatically transfer to this mixed reasoning-and-chat setting. SR-GRPO remains strongest on GPQA, MMLU, AIME, OlympiadBench, AMC, and WildBench, and is close to the best result on MATH. This supports the claim that stable rank is competitive not only against small reward models, but also against stronger learned and internal-state reward baselines.

Table 17: Additional reward-model and internal-state baselines on Qwen2.5-1.5B-Instruct.

Model	GPQA	MMLU	MATH	AIME	Olymp.	AMC	WB-Elo
Base	18.97	47.70	48.00	3.33	25.48	35.00	1036.17
+ EndoRM	14.14	46.37	52.40	6.67	25.78	22.50	1051.13
+ Skywork-8B	20.20	46.70	**53.00**	10.00	25.48	20.00	916.34
+ SR-GRPO	**21.21**	**47.73**	52.60	**13.33**	**26.41**	**37.50**	**1062.40**

I.5 Longer-Horizon Training

We study longer-horizon training to check whether optimizing stable rank leads to late-stage degradation. This is important for label-free rewards, where the policy may increase the reward without improving downstream task performance. We extend SR-GRPO training to 2–3$\times$ the main training horizon and evaluate checkpoints at 500, 1000, and 1500 steps. For each checkpoint, we track the stable-rank reward, generation entropy, and average downstream benchmark accuracy.

Results. Table 18 shows no collapse within the tested range. Average benchmark accuracy increases monotonically from 27.95 to 32.02 on Qwen2.5-1.5B and from 34.85 to 43.80 on Qwen3-8B. The stable-rank reward also increases, while entropy decreases. Importantly, the entropy decrease is not accompanied by downstream degradation, suggesting that the policy is not merely collapsing to low-diversity responses that exploit the reward. This does not rule out eventual saturation at much longer horizons, but within the tested range we do not observe the reward-hacking pattern commonly associated with label-free RL.

I.6 Alternative Intrinsic Dimension Metrics as RL Rewards

Appendix H compares stable rank with other intrinsic dimension metrics as zero-shot RewardBench proxies. Here we additionally test whether these alternatives are effective as RL rewards. We replace stable rank with two dimension-based rewards on Qwen2.5-1.5B-Instruct: effective rank and PCA

Table 18: Longer-horizon SR-GRPO training. Avg. Accuracy is the average over the reported reasoning benchmarks.

Model	Checkpoint	Reward	Entropy	Avg. Accuracy
Qwen2.5-1.5B	SFT	–	–	27.95
	cp500	1.61	1.87	29.33
	cp1000	1.74	1.72	31.05
	cp1500	1.82	1.41	32.02
Qwen3-8B	SFT	–	–	34.85
	cp500	1.36	1.26	38.34
	cp1000	1.45	0.95	40.61
	cp1500	1.88	0.71	43.80

95% variance dimension. These rewards use the same final-layer hidden-state matrix and the same RL training setup as SR-GRPO; only the scalar reward computation changes.

Results. Table 19 shows that PCA-based dimension recovers part of the reasoning gain, improving GPQA and MATH, but performs poorly on WildBench. Effective rank improves AIME but degrades GPQA, MMLU, AMC, and WildBench. Stable rank gives the strongest overall trade-off, matching or exceeding the alternatives on MMLU, MATH, AIME, OlympiadBench, AMC, and WildBench. These results suggest that the stable-rank formulation, which balances total representational energy against concentration in the dominant direction, is better suited as an optimization reward than simply maximizing entropy-based or PCA-count dimensionality.

Table 19: Alternative intrinsic dimension metrics used as RL rewards on Qwen2.5-1.5B-Instruct.

Model	GPQA	MMLU	MATH	AIME	Olymp.	AMC	WB-Elo
Base	18.9	**47.7**	48.0	3.3	25.4	35.0	1036
+ EffectiveRank	11.6	43.3	48.8	**13.3**	24.5	30.0	923
+ PCA	**22.1**	47.2	51.4	10.0	25.9	35.0	925
+ StableRank	21.2	**47.7**	**52.4**	**13.3**	**26.4**	**37.5**	**1062**

I.7 Multi-Seed Comparison with Verifiable and Learned Rewards

We evaluate run-to-run variability using three random seeds (42, 43, and 44) on Qwen2.5-1.5B-Instruct and DeepSeek-R1-Distill-Qwen-1.5B. All methods are trained on the same math subset of the main training data, with identical prompts and training steps across methods for each model. This protocol accommodates RLVR, which uses math-verify with ground-truth answers as its reward. The learned-RM baseline uses the stronger Skywork-Reward-Llama-3.1-8B model. Table 20 reports mean accuracy and 95% confidence intervals over seeds.

Table 20: Three-seed alignment results. Values are mean accuracy (%) $\pm$ 95% confidence intervals over seeds 42, 43, and 44. Avg. is the unweighted mean of the six benchmark accuracies. Bold marks the best mean per benchmark and model.

Model	Method	GPQA	MMLU	MATH-500	AIME25	OlympiadBench	AMC	Avg.
Qwen2.5-1.5B	Perplexity	30.98 ± 0.72	50.46 ± 0.24	47.33 ± 1.25	3.33 ± 4.71	15.56 ± 2.30	30.00 ± 0.00	29.61
	RLVR	31.14 ± 0.72	$\mathbf{50.47 \pm 0.08}$	46.93 ± 1.25	13.33 ± 4.71	15.01 ± 0.56	$\mathbf{36.67 \pm 7.17}$	32.26
	RM (Skywork-8B)	31.31 ± 0.00	50.36 ± 0.05	46.67 ± 1.03	6.67 ± 4.71	15.60 ± 0.56	34.17 ± 3.59	30.80
	SR-GRPO	$\mathbf{31.65 \pm 0.36}$	50.41 ± 0.13	$\mathbf{47.53 \pm 1.70}$	$\mathbf{16.67 \pm 4.71}$	$\mathbf{16.20 \pm 0.77}$	35.83 ± 6.41	**33.04**
DeepSeek-R1-Distill	Perplexity	30.47 ± 1.92	47.02 ± 0.31	83.33 ± 2.29	16.67 ± 4.71	53.04 ± 3.74	66.67 ± 12.93	49.53
	RLVR	30.81 ± 3.76	46.84 ± 0.53	83.00 ± 4.07	20.00 ± 8.16	55.26 ± 2.94	65.83 ± 3.59	50.29
	RM (Skywork-8B)	32.49 ± 7.13	47.37 ± 1.51	83.13 ± 3.73	17.78 ± 4.78	55.21 ± 5.16	70.83 ± 9.49	51.14
	SR-GRPO	$\mathbf{32.82 \pm 2.10}$	$\mathbf{47.73 \pm 0.43}$	$\mathbf{83.53 \pm 3.53}$	$\mathbf{23.33 \pm 4.71}$	$\mathbf{55.72 \pm 3.10}$	$\mathbf{72.50 \pm 8.16}$	**52.61**

Results. SR-GRPO achieves the best average on both models, with the best mean on four of six benchmarks on Qwen2.5-1.5B and all six on DeepSeek-R1-Distill. Confidence intervals remain wide on AIME25 and AMC, which contain only 30 and 40 problems, respectively.

J Context Window Size

We investigate how context window size affects stable rank performance on RewardBench. For each prompt-response pair, we truncate the combined sequence to a maximum length in $\{128, 512, 1024, 2048, 4096\}$ tokens before computing stable rank on the final layer of Qwen3-8B [46]. RewardBench examples have a mean length of 281 tokens, with 75% under 512 tokens.

Table 21: Effect of maximum input length on RewardBench accuracy across categories (Qwen3-8B, final-layer stable rank).

Max length	Chat	Chat Hard	Safety	Code	Math	Overall
128	78.25	63.07	76.37	24.80	70.47	62.59
512	86.89	84.38	76.20	87.91	83.89	83.85
1024	86.88	84.02	76.20	88.41	84.34	83.97
2048	87.23	84.02	76.20	88.41	84.34	84.04
4096	87.23	84.02	76.20	88.41	84.34	84.04

Results. Table 21 shows that stable rank accuracy degrades at short context lengths but saturates quickly. At 128 tokens, overall accuracy drops to 62.59%, a 21 percentage point decrease from 512 tokens (83.85%). Code suffers most severely, falling from 87.91% to 24.80%, because truncation removes critical program logic. However, extending beyond 512 tokens yields minimal gains: increasing to 4096 tokens improves accuracy by only 0.2 percentage points.

Analysis. The rapid saturation indicates that stable rank captures semantic content rather than mechanically rewarding longer sequences. Short windows (128 tokens) harm performance by cutting off meaningful content, but once the window covers the core reasoning structure (around 512 tokens for this dataset), additional context provides diminishing returns. Safety performance remains constant at 76.20% for all lengths $\geq$512, suggesting that refusal quality is determined early in the response.

Practical Recommendations. A 512-token window provides an effective operating point for datasets with similar length distributions. For longer-form tasks, we recommend scaling the window to cover at least the 75th percentile of sequence lengths to avoid truncation artifacts.

K Input Prompt Format

We investigate whether input prompt format affects stable rank performance. Since stable rank is computed on hidden states, the concatenation structure could influence the resulting geometry. We test six formats ranging from minimal (no prefix) to structured (User/Assistant tags).

(1) `{prompt}\n\n{response}` (no prefix)

(2) `{prompt}\n\nResponse:{response}`

(3) `{prompt}\n\nAssistant:{response}`

(4) `User:{prompt}\n\nAssistant:{response}`

(5) `Question:{prompt}\n\nAnswer:{response}`

(6) `{prompt}\n\nAnswer:{response}`

Results. Table 22 shows that stable rank is robust to format variation. Overall accuracy varies by at most 3 percentage points across formats for each model, and no single format consistently outperforms others. Category-level variations are larger (e.g., Safety on Phi-3.5-mini ranges from 52.82% to 62.63%) but show no systematic pattern across models, indicating that format effects are model-specific rather than fundamental to stable rank.

Practical Recommendations. The robustness to prompt format simplifies deployment: practitioners can use simple formats without extensive tuning. We recommend using the model's native chat template when available, or a minimal format like `{prompt}\n\nAnswer:{response}` otherwise.

Table 22: Effect of prompt format on RewardBench accuracy. Format indices correspond to the list above. Overall accuracy varies by at most 3 percentage points across formats for each model.

Model	Fmt	Chat	Hard	Safety	Code	Math	Overall
Phi-3.5-mini-instruct	1	82.79	86.36	53.99	82.72	44.30	70.03
	2	82.14	84.70	57.17	87.91	40.94	70.57
	3	78.39	85.60	62.63	89.43	40.94	71.40
	4	77.07	87.55	62.07	89.53	45.86	72.42
	5	81.19	85.46	59.13	87.91	51.23	72.99
	6	84.56	84.63	52.82	87.09	46.09	71.04
Qwen2.5-1.5B-Instruct	1	88.33	87.08	64.60	72.66	63.09	75.15
	2	85.21	86.25	65.27	74.19	64.65	75.11
	3	86.96	85.41	67.00	74.39	61.74	75.10
	4	87.28	86.60	69.32	75.71	52.80	74.34
	5	86.30	84.22	65.47	74.39	68.46	75.77
	6	87.28	85.70	66.54	73.78	66.44	75.95
Qwen3-0.6B	1	83.67	80.04	51.39	76.83	49.44	68.27
	2	82.98	80.94	50.64	75.71	48.99	67.85
	3	82.28	81.84	47.62	76.83	43.18	66.35
	4	80.54	78.85	48.72	76.93	44.30	65.87
	5	83.67	81.01	47.77	78.05	44.52	67.00
	6	83.30	80.11	49.14	76.22	41.61	66.08

L Stable Rank Example Prompts

To complement the quantitative analysis in Section 5, we provide examples illustrating how stable rank differentiates high-quality responses from two common failure modes: catastrophic repetition and redundant verbosity.

Catastrophic Repetition. The Image Processing example in Table 23 shows a response that begins coherently but enters an infinite repetition loop ("Image > Image > ..."), collapsing into a degenerate low-dimensional representation. This failure directly relates to coherence standard deviation, which has the strongest negative correlation with stable rank ($\rho = -0.356$). Geometrically, repetition concentrates variance into a single direction, reducing the effective dimensionality that stable rank measures.

Redundant Verbosity. The Math Problem example shows a response that correctly solves the problem but continues with unnecessary elaboration ("However...", "Alternatively..."). While the content is not incorrect, it dilutes information density. This aligns with our finding that stable rank correlates negatively with token count ($\rho = -0.294$) and total discourse markers ($\rho = -0.204$), while correlating positively with compression ratio ($\rho = 0.233$). The high stable rank response terminates precisely at the answer, maintaining focused information flow.

Table 23: Qualitative Comparison of Model Responses. High stable rank models produce coherent and accurate instructions, whereas low stable rank models suffer from severe degeneration, exhibiting infinite repetition loops (e.g., in Image Processing) and logical inconsistencies (e.g., in Math Problems).

Example	Query	Response
Image Processing	"I can see that in case of PNG you just save the image as is, But I still want to control it's size/quality"	**High Stable Rank:** When working with PNG files, there are several ways to control their size and quality: 1. **Use ImageMagick or Other Command-Line Tools**: convert input.png -resize 100x100 output.png (resizes) and convert input.png -quality 85 output.png (compresses). 2. **Use JavaScript Libraries**: Libraries like canvas allow you to manipulate and resize images. (Includes the full JavaScript canvas code snippet) 3. **Use Online Tools**: There are online tools like OnlineImageResize. 4. **Use WebP**: Consider using WebP, which is more efficient... convert input.png -quality 85 output.webp.
		Low Stable Rank: PNG (Portable Network Graphics) is a lossless image format... If you need to control the size or quality... **Steps to Control PNG Size or Quality:** 1. **Using Photoshop:** - Open the PNG image. - Go to Image > Image Size to adjust the size... - Under the Image Size dialog box, you can set the Resolution to a value... - To change the quality, go to Image > Adjustments > Image > ...
Math Problem	"Cities A and B are 45 miles apart. Alicia bikes towards B at 18 miles per hour. Beth bikes toward A at 12 miles per hour. How many miles from City A will they be when they meet?"	**High Stable Rank:** To solve this problem, we can follow these steps: 1. **Calculate their combined speed:** $18 + 12 = 30$ miles per hour. 2. **Determine the time it takes for them to meet:** Time $=$ Distance$/$Speed $= 45/30 = 1.5$ hours. 3. **Calculate the distance covered by Alicia:** Distance $= 18 \times 1.5 = 27$ miles. Therefore, the final answer is: $\boxed{27}$.
		Low Stable Rank: To solve this problem, we need to determine the distance covered by Alicia and Beth... 1. Combined speed is $18 + 12 = 30$ miles per hour. 2. Time it takes for them to meet: 45 miles$/30$ mph $= 1.5$ hours. 4. Distance covered by Alicia: 18 mph $\times 1.5$ h $= 27$ miles. 6. However, this distance is the distance from the starting point of City A... Alternatively, we can take a simpler approach: Since the distance between the cities is 45 miles... The shortest and simplest way to solve this is by considering the proportion... Alicia covers a proportion of $18/(18 + 12) = 18/30 = 0.6$ of the distance.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state the three main contributions (stable rank metric, SR-GRPO, and empirical validation), all supported by experiments in Sections 2–4 and Appendices A–H.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The Conclusion section explicitly discusses limitations.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: The paper does not present new theoretical results or proofs; motivation is grounded in appropriately cited prior work.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Sections 2–4 and Appendices B, E–H provide complete evaluation protocols, training procedures, hyperparameters, and implementation details needed for reproduction.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: We use publicly available models and datasets but do not release a dedicated code repository at submission time; implementation details are in Appendices B and E.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: Appendix E specifies all training hyperparameters, optimizer settings, and LoRA configuration; test benchmarks and metrics are described in Sections 2.2, 2.3, and 4.1.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Correlation analysis in Section 5 reports $p < 0.001$ significance.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix E states experiments use 4×NVIDIA H800 GPUs, with training steps and batch sizes provided.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research uses publicly available models and standard benchmarks with no human subjects or sensitive data, and all assets are properly attributed.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: No relevant societal impacts unique to this paper (based on existing public LLMs).

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.

- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: No new models or datasets are released; experiments use existing public assets with their own safeguards.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All models and datasets used are properly cited with references to their original papers or documentation.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: No new datasets or pre-trained models are released at submission time.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: No crowdsourcing or human subjects are involved; all evaluations use automated metrics on public benchmarks.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: No human subjects research is conducted; IRB approval is not required.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs are the research subject, not a writing or methodology tool; LLM-as-judge baselines are standard comparison methods described in Appendix B.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.