

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

Yixuan Tang<sup>1</sup> Yi Yang<sup>1</sup>

## 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 Reward-Bench and improves task accuracy by an average of 11.3 percentage points over 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) (Ouyang et al., 2022; Bai et al., 2022). Despite its success, this paradigm depends heavily on external supervision. Human judgments

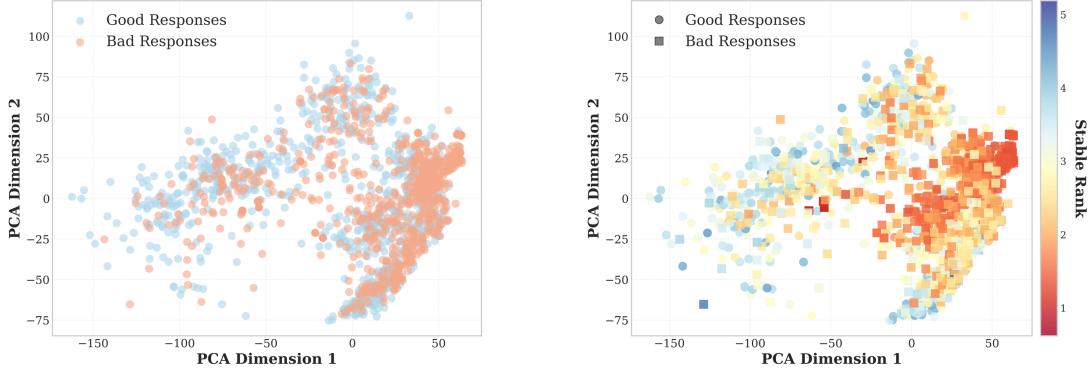
<sup>1</sup>The Hong Kong University of Science and Technology. Correspondence to: Yixuan Tang <ytangch@connect.ust.hk>, Yi Yang <imiyang@ust.hk>.

are subjective and context-dependent, making it difficult to train robust reward models (Chakraborty et al., 2024; Wang et al., 2023), and learned proxies are susceptible to reward hacking (Casper et al., 2023). The sparsity of annotations further limits the model’s ability to fine-grained behaviors (Wu et al., 2023).

These limitations have motivated various attempts to reduce annotation dependence. Direct Preference Optimization (Rafailov et al., 2023) eliminates explicit reward modeling but still requires preference datasets. Verifiable rewards (DeepSeek-AI, 2025; Lambert et al., 2024) leverage ground-truth outcomes but only apply where automatic verification is feasible. Self-evaluation methods (Yuan et al., 2024; Lee et al., 2024; Garg et al., 2025) suffer from prompt sensitivity and systematic biases (Zheng et al., 2023; Wang et al., 2024). 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 (He et al., 2024c); when it hallucinates, different activation patterns emerge (Chen et al., 2024). This raises a natural question: *Can 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** (Rudelson & Vershynin, 2007), 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



*Figure 1.* Hidden states visualization for responses in RewardBench (Lambert et al., 2025) based on Qwen3-8B (Yang et al., 2025). 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.

high-rank (Yang et al., 2018; Godey et al., 2024). Second, literature also shows that when representations collapse into a narrow cone, expressiveness is severely limited and generation quality degrades (Gao et al., 2019). 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 (Lambert et al., 2025) with Qwen3-8B (Yang et al., 2025), 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 (Yang et al., 2024), SR-GRPO outperforms GRPO with learned reward models by 10 to 19% on reasoning tasks at zero annotation cost, and surpasses self-reward baselines by 8 to 9%. Consistent gains on DeepSeek-R1-Distill-Qwen-1.5B (DeepSeek-AI, 2025) 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 (Rudelson & Vershynin, 2007) 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  $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_{\min(T,d)} \geq 0$  denote the singular values of  $\mathbf{H}$ . The stable rank is defined as:

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

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 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 (Yang et al., 2018; Godey et al., 2024) implies that expressive language modeling requires high-rank representations, while representation collapse is a known symptom of degraded generation (Gao et al., 2019). 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 (Lambert et al., 2025), 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.) (Kim et al., 2024): prompting the LLM to score each response on a 1-5 scale; (2) Pairwise Comparison (Pair.) (Kim et al., 2024): prompting the LLM to directly compare two responses; (3) IPO (Garg et al., 2025): using implicit preference scores derived from Yes/No token probabilities. We evaluate five models: Qwen2.5-1.5B-Instruct (Yang et al., 2024), Qwen3-0.6B (Yang et al., 2025), Qwen3-8B (Yang et al., 2025), Llama-3.1-8B-Instruct (Dubey et al., 2024), and Phi-3.5-mini-Instruct (Abdin et al., 2024). 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. 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.* Overall accuracy comparison across different reward methods on RewardBench. Pointwise and Pairwise refer to LLM-as-judge baselines using 1-5 scoring and direct comparison, respectively. SR denotes Stable Rank. The detailed results are demonstrated in the Appendix C.

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

## 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 (Rein et al., 2024), MMLU-redux (Gema

et al., 2025)) and mathematical reasoning tasks (MATH500 (Lightman et al., 2023), OlympiadBench (He et al., 2024a), AMC23 (knoveleng, 2023)) across four models: Qwen2.5-1.5B-Instruct (Yang et al., 2024), Phi-3.5-mini-Instruct (Abdin et al., 2024), Llama-3.2-1B-Instruct (Dubey et al., 2024), and DeepSeek-R1-Distill-Qwen-1.5B (DeepSeek-AI, 2025). 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 D.

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

#### 3.2. Training Procedure

We build on Group Relative Policy Optimization (GRPO) (Shao et al., 2024), which compares responses within a sampled group to obtain relative quality signals, avoiding the need for a learned value function. This 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

---

#### Algorithm 1 SR-GRPO Training

---

**Require:** Dataset  $\mathcal{D}$ , initial policy  $\pi_\phi$ , frozen reference  $\pi_{\text{ref}}$ , group size  $K$ , KL coefficient  $\beta$   
**Ensure:** Optimized policy  $\pi_\phi$

- 1: **while** not converged **do**
- 2:   Sample batch  $\{x_i\} \sim \mathcal{D}$
- 3:   **for** each prompt  $x_i$  **do**
- 4:     Sample responses  $\{y_{i,k}\}_{k=1}^K \sim \pi_\phi(\cdot|x_i)$
- 5:     Compute rewards  $r_{i,k} \leftarrow \text{SR}(\mathbf{H}_{i,k}; \pi_{\text{ref}})$
- 6:     Compute advantages  $A_{i,k} \leftarrow (r_{i,k} - \mu_i)/(\sigma_i + \epsilon)$
- 7:   **end for**
- 8:   Update  $\pi_\phi$  via Eq. (2)
- 9: **end while**

---

policy and compute their stable rank on the frozen reference model  $\pi_{\text{ref}}$ . 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  $\mu$  and  $\sigma$  are the group mean and standard deviation. The training objective is:

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

where  $\rho_k = \pi_\phi(y_k|x)/\pi_{\phi_{\text{old}}}(y_k|x)$  is the importance ratio,  $\phi_{\text{old}}$  denotes the parameters before the current update, and  $\beta$  controls the KL penalty. Algorithm 1 summarizes the procedure.

**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 H).

## 4. Alignment Experiments

### 4.1. Experiment Setup

We evaluate SR-GRPO on two models: Qwen2.5-1.5B-Instruct (Yang et al., 2024) and DeepSeek-R1-Distill-Qwen-1.5B (DeepSeek-AI, 2025). All methods are trained on prompts from SmolTalk2 (Allal et al., 2025), 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 (Hu et al., 2022) and compute stable rank on the base model without LoRA adapters, which also serves as the frozen reference.

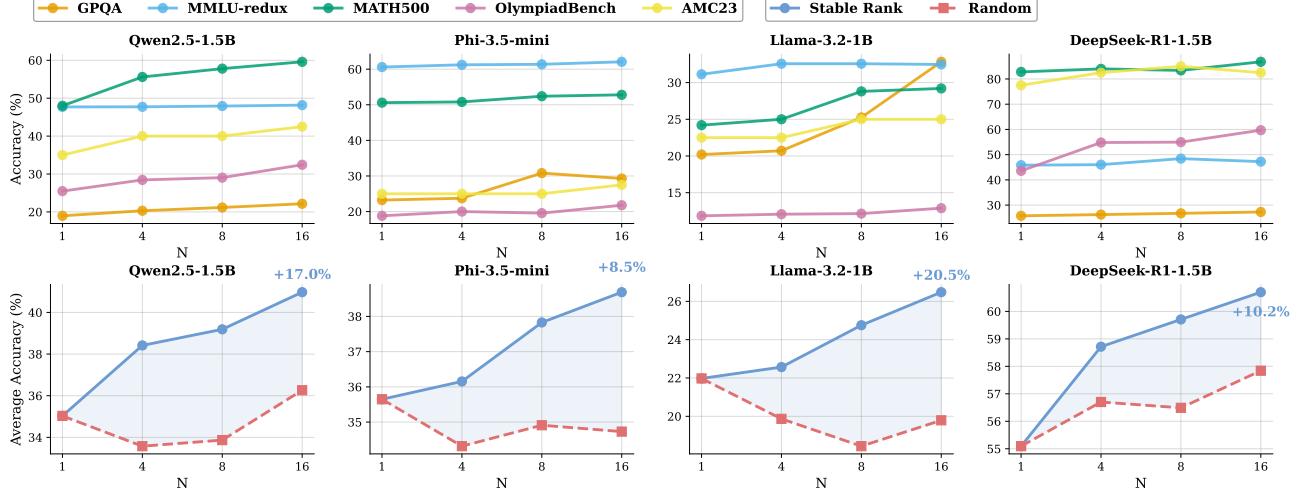


Figure 2. Best-of-N decoding with stable rank selection. **Top:** Performance on individual benchmarks as  $N$  increases. **Bottom:** Average accuracy comparing stable rank selection (solid) versus random selection (dashed). Annotations show relative improvement over greedy decoding ( $N=1$ ). Stable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

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.

Model	Method	STEM			Math					Chat
		GPQA	MMLU	Avg.	MATH	AIME	Olymp.	AMC	Avg.	WB-Elo
Qwen2.5-1.5B-Instruct	Base	19.0	<b>47.7</b>	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	<b>13.3</b>	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	<b>47.7</b>	32.4	49.2	3.3	24.0	27.5	26.0	1037.7
	+ SR-GRPO	<b>21.2</b>	<b>47.7</b>	<b>34.5</b>	<b>52.4</b>	<b>13.3</b>	<b>26.4</b>	<b>37.5</b>	<b>32.4</b>	<b>1062.4</b>
DeepSeek-R1-Distill-Qwen-1.5B	Base	25.8	45.8	35.8	82.8	30.0	43.6	<b>77.5</b>	58.5	913.5
	+ RM	24.2	46.3	35.3	85.4	<b>36.7</b>	56.2	72.5	62.7	918.4
	+ Self-Reward	23.7	45.8	34.8	85.2	<b>36.7</b>	57.2	67.5	61.6	919.2
	+ Perplexity	24.2	46.3	35.3	<b>86.2</b>	<b>36.7</b>	55.0	<b>77.5</b>	63.8	917.0
	+ IPO	28.8	45.8	37.3	86.0	26.7	56.7	<b>77.5</b>	61.7	922.4
	+ SR-GRPO	<b>30.3</b>	<b>46.6</b>	<b>38.4</b>	<b>86.2</b>	<b>36.7</b>	<b>58.5</b>	<b>77.5</b>	<b>64.7</b>	<b>932.5</b>

**Benchmarks.** We evaluate on three categories: (1) *STEM* tasks: GPQA (Rein et al., 2024), a graduate-level science QA benchmark requiring expert knowledge, and MMLU-redux (Gema et al., 2025), a curated subset of MMLU with corrected labels; (2) *Mathematical reasoning*: MATH500 (Lightman et al., 2023), a subset of competition mathematics problems, AIME25 (Zhang & Math-AI, 2025), problems from the 2025 American Invitational Mathematics Examination, OlympiadBench (He et al., 2024b), olympiad-level math and physics problems, and AMC23 (knoveleng, 2023), problems from the 2023 American Mathematics Competition; (3) *General chat*: WildBench (Lin et al., 2025), which evaluates open-ended conversation quality using GPT-4o mini (OpenAI, 2025) as a judge. For all benchmarks except WildBench, we report Pass@1 accuracy. WildBench (Lin et al., 2025) 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 (Liu et al., 2025), 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 (Yuan et al., 2024): The model eval-

ates its own completions using the pointwise scoring prompt (Appendix B), generating a 1-5 score that serves as the reward signal.

- Perplexity: We use the negative log-likelihood of the completion as the reward, where lower perplexity indicates higher fluency.
- IPO (Garg et al., 2025): The model acts as a binary classifier, determining response quality by generating “Yes” or “No”. We extract logits for both tokens from the first output position and compute their probabilities via softmax. The probability of “Yes” serves as the reward signal.

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

## 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 DeepSeek-R1, substantially outperforming all baselines. This indicates that stable rank captures qualities valued in general conversation, such as coherence

Table 3. Semantic coherence and information density correlations with stable rank (Spearman  $\rho$ ). Mini-bars visualize correlation magnitude and direction (scale: -1 to 1). All  $p < 0.001$ .

Metric	$\rho$	Mini-bar
<i>Semantic Coherence</i>		
Coherence std (adjacent sim.)	-0.356	
QA alignment consistency	0.316	
Progression score	0.313	
Semantic variance	-0.272	
Adjacent similarity (mean)	0.250	
<i>Information Density</i>		
Sentence count	-0.368	
Token count	-0.294	
Unique token count	-0.286	
Lexical diversity (TTR)	0.238	
Compression ratio	0.233	

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 F; here we summarize key findings.

### 5.1. 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.

### 5.2. Stable Rank Distinguishes Density from Verbosity

Contrary to common reward hacking, where models maximize length, stable rank correlates negatively with token

**Table 4.** Discourse marker correlations with stable rank (paired-difference analysis,  $N = 2,985$ ). Negative values indicate fewer markers in higher-rank responses. Both parametric (Pearson  $r$ ) and non-parametric (Spearman  $\rho$ ) correlations shown; see Table 12 for full results. \*\*\* indicates  $p < 0.001$

Category	Pearson $r$	Spearman $\rho$	Example keywords
Contrastive	+0.187***	+0.067***	however, but, although
Causal	+0.139***	-0.002	because, therefore, thus
Additive	-0.166***	-0.156***	moreover, furthermore
Conditional	-0.163***	-0.163***	if, when, unless
Enumeration	-0.122***	-0.148***	first, second, step 1
Temporal	-0.106***	-0.122***	then, next, meanwhile
Inference	-0.065***	-0.115***	implies, suggests
Total (raw count)	-0.149***	-0.204***	<i>all markers</i>

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.

### 5.3. 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 F) 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 4 shows that most marker categories, especially additive ( $\rho = -0.156$ ), conditional ( $\rho = -0.163$ ), and enumeration ( $\rho = -0.148$ ), have consistently negative correlations with stable rank, and the total marker count is also negatively associated ( $\rho = -0.204$ ). Responses that rely heavily on common connective words such as “moreover”, “first”, and “then” tend to receive a lower stable rank, which is consistent with Section 5.2 where stable rank disfavors verbose, pattern-like exposition in favor of compact information content.

Contrastive and causal markers follow a different pattern. Their Pearson correlations with stable rank are positive (contrastive  $r = 0.187$ , causal  $r = 0.139$ ), but their Spearman correlations are small. This divergence suggests that the presence of these markers matters more than their frequency: a few well-placed “however” or “because” phrases benefit stable rank, but overusing them does not. This aligns with

linguistic intuition: contrastive and causal connectives are most informative when they mark key reasoning steps, not when they appear as formulaic filler.

### 5.4. Summary

Together, these correlation patterns align with established findings in text quality research. The strong association with semantic coherence metrics mirrors neural coherence models, which show that smooth semantic transitions predict text quality (Mesgar & Strube, 2018; Cui et al., 2017). The negative correlation with length while favoring information density addresses a well-documented failure mode in RLHF: length bias, where reward models favor longer responses regardless of quality (Singhal et al., 2024). Unlike learned reward models that often exploit this spurious correlation, stable rank’s geometric formulation inherently penalizes verbose, low-information content.

While these correlations are moderate in magnitude ( $|\rho| \approx 0.2\text{--}0.4$ ), they are consistent in direction across metrics and statistically significant ( $p < 0.001$ ), suggesting stable rank captures genuine quality signals rather than noise. These findings indicate that stable rank correlates with multiple dimensions of text quality, including coherence, density, and reasoning structure, suggesting it captures a meaningful aggregate signal rather than a single superficial feature. Qualitative examples illustrating how stable rank differentiates high-quality reasoning from repetition and verbosity are provided in Appendix J.

## 6. Ablation Studies

We conduct three ablation studies to validate our design choices.

**Alternative intrinsic dimension metrics.** We compare stable rank against three alternatives: condition number, which measures the ratio of largest to smallest singular value; PCA 95% variance, which counts principal components needed to capture 95% of variance; and effective rank (Roy & Vetterli, 2007), which uses entropy over the singular value distribution. On RewardBench, stable rank achieves 84.04% overall accuracy, outperforming PCA 95% variance (61.91%), effective rank (54.50%), and condition number (36.04%). The gap widens on harder categories such as Math and Safety. Stable rank succeeds because it aggregates information across the entire singular value spectrum through the Frobenius norm, making it robust to outliers. Condition number is sensitive to extreme singular values. Effective rank’s entropy weighting and PCA’s discrete component counting appear less suited for capturing quality distinctions. Full results are provided in Appendix G.

**Context length.** We examine how sequence truncation affects performance. Accuracy drops from 83.85% at 512 to

kens 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 H.

**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 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) (Ouyang et al., 2022; Bai et al., 2022). 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 (Dubey et al., 2024), but it inherits well-known issues: preference data are expensive and noisy, and reward models are vulnerable to reward hacking and distribution shift (Casper et al., 2023; Skalse et al., 2022). These limitations motivate supervised preference-based methods such as Direct Preference Optimization (DPO) (Rafailov et al., 2023), KTO (Ethayarajh et al., 2024), Bayesian RLHF (Wang et al., 2023), and MaxMin-RLHF (Chakraborty et al., 2024), 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 (Zhang et al., 2025b; Yin et al., 2025). 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 (DeepSeek-AI, 2025) and TÜLU 3 (Lambert et al., 2024) 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 (Yuan et al., 2024) and RLAIF (Lee et al., 2024) generate AI feedback instead of human labels, while IPO (Garg et al., 2025) and Self-Rewarding PPO (Zhang et al., 2025a) derive preference signals from token-level probabilities. Closer to our setting, work on internal activations such as Factoscope (He et al., 2024c) and INSIDE (Chen et al., 2024) 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. (2018) show that the expressiveness of softmax-based models is fundamentally limited by the rank of their hidden representations. Godey et al. (2024) 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 (Gao et al., 2019). Together, these results establish a link between representational dimensionality and generation quality. Empirical work supports this connection. Garrido et al. (2023) 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 (Mikolov et al., 2013), and intermediate representations encode factuality (He et al., 2024c) and hallucination risk (Chen et al., 2024), 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 demonstrates 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 84.04% accuracy on RewardBench and improves Best-of-N sampling by 11.3 percentage points on average across reasoning benchmarks. Building on these insights, we introduce SR-GRPO, which uses stable rank as a novel reward signal for reinforcement learning without external supervision. SR-

GRPO improves Qwen2.5-1.5B-Instruct by 10% on STEM tasks and 19% on mathematical reasoning, outperforming both learned reward models and self-evaluation baselines. Moreover, our analysis reveals that stable rank captures semantic coherence, information density, and sensitivity to key reasoning structures. Collectively, these results suggest that internal representation geometry provides sufficient signal for effective LLM alignment, offering a scalable alternative to annotation-dependent approaches.

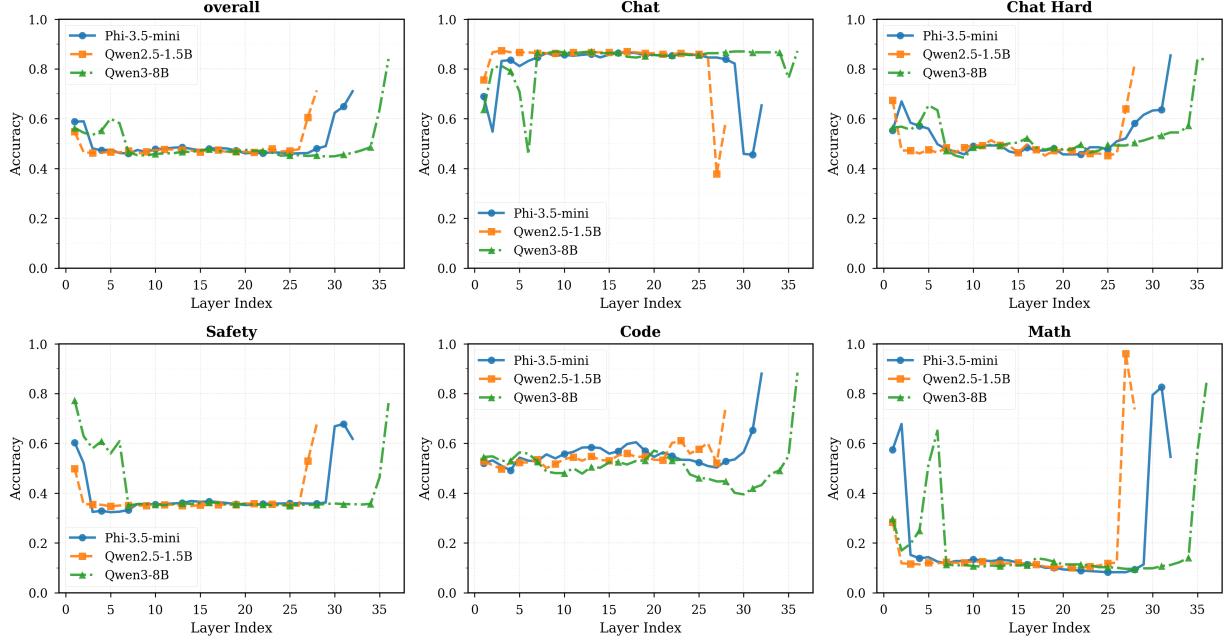
## References

- Abdin, M. I., Jacobs, S. A., Awan, A. A., Aneja, J., Awadallah, A., Awadalla, H., Bach, N., Bahree, A., Bakhtiari, A., Behl, H. S., Benhaim, A., Bilenko, M., Bjorck, J., Bubeck, S., Cai, M., Mendes, C. C. T., Chen, W., Chaudhary, V., Chopra, P., Giorno, A. D., de Rosa, G., Dixon, M., Eldan, R., Iter, D., Garg, A., Goswami, A., Gunasekar, S., Haider, E., Hao, J., Hewett, R. J., Huynh, J., Javaheripi, M., Jin, X., Kauffmann, P., Karampatziakis, N., Kim, D., Khademi, M., Kurilenko, L., Lee, J. R., Lee, Y. T., Li, Y., Liang, C., Liu, W., Lin, E., Lin, Z., Madan, P., Mitra, A., Modi, H., Nguyen, A., Norick, B., Patra, B., Perez-Becker, D., Portet, T., Pryzant, R., Qin, H., Radmilac, M., Rosset, C., Roy, S., Ruwase, O., Saarikivi, O., Saied, A., Salim, A., Santacroce, M., Shah, S., Shang, N., Sharma, H., Song, X., Tanaka, M., Wang, X., Ward, R., Wang, G., Witte, P. A., Wyatt, M., Xu, C., Xu, J., Yadav, S., Yang, F., Yang, Z., Yu, D., Zhang, C., Zhang, C., Zhang, J., Zhang, L. L., Zhang, Y., Zhang, Y., Zhang, Y., and Zhou, X. Phi-3 technical report: A highly capable language model locally on your phone. *CoRR*, abs/2404.14219, 2024. doi: 10.48550/ARXIV.2404.14219.
- Allal, L. B., Lozhkov, A., Bakouch, E., Blázquez, G. M., Penedo, G., Tunstall, L., Marafioti, A., Kydlícek, H., Lajarín, A. P., Srivastav, V., et al. Smollm2: When smol goes big-data-centric training of a small language model. *CoRR*, 2025.
- Bai, Y., Jones, A., Ndousse, K., Askell, A., Chen, A., Das-Sarma, N., Drain, D., Fort, S., Ganguli, D., Henighan, T., Joseph, N., Kadavath, S., Kernion, J., Conerly, T., Showk, S. E., Elhage, N., Hatfield-Dodds, Z., Hernandez, D., Hume, T., Johnston, S., Kravec, S., Lovitt, L., Nanda, N., Olsson, C., Amodei, D., Brown, T. B., Clark, J., McCandlish, S., Olah, C., Mann, B., and Kaplan, J. Training a helpful and harmless assistant with reinforcement learning from human feedback. *CoRR*, abs/2204.05862, 2022.
- Casper, S., Davies, X., Shi, C., Gilbert, T. K., Scheurer, J., Rando, J., Freedman, R., Korbak, T., Lindner, D., Freire, P., Wang, T. T., Marks, S., Segerie, C.-R., Carroll, M., Peng, A., Christoffersen, P. J., Damani, M., Slocum, S., Anwar, U., Siththanjan, A., Nadeau, M., Michaud, E. J., Pfau, J., Krasheninnikov, D., Chen, X., Langosco, L., Hase, P., Biyik, E., Dragan, A., Krueger, D., Sadigh, D., and Hadfield-Menell, D. Open problems and fundamental limitations of reinforcement learning from human feedback. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856.
- Chakraborty, S., Qiu, J., Yuan, H., Koppel, A., Manocha, D., Huang, F., Bedi, A., and Wang, M. 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*, pp. 6116–6135. PMLR, 21–27 Jul 2024.
- Chen, C., Liu, K., Chen, Z., Gu, Y., Wu, Y., Tao, M., Fu, Z., and Ye, J. INSIDE: LLMs’ internal states retain the power of hallucination detection. In *The Twelfth International Conference on Learning Representations*, 2024.
- Cui, B., Li, Y., Zhang, Y., and Zhang, Z. Text coherence analysis based on deep neural network. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, CIKM ’17, pp. 2027–2030, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450349185. doi: 10.1145/3132847.3133047.
- DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *CoRR*, abs/2501.12948, 2025. doi: 10.48550/ARXIV.2501.12948.
- Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Yang, A., Fan, A., Goyal, A., Hartshorn, A., Yang, A., Mitra, A., Sravankumar, A., Korenev, A., Hinsvark, A., Rao, A., Zhang, A., Rodriguez, A., Gregerson, A., Spataru, A., Rozière, B., Biron, B., Tang, B., Chern, B., Caucheteux, C., Nayak, C., Bi, C., Marra, C., McConnell, C., Keller, C., Touret, C., Wu, C., Wong, C., Ferrer, C. C., Nikolaidis, C., Al-lonsius, D., Song, D., Pintz, D., Livshits, D., Esiobu, D., Choudhary, D., Mahajan, D., Garcia-Olano, D., Perino, D., Hupkes, D., Lakomkin, E., AlBadawy, E., Lobanova, E., Dinan, E., Smith, E. M., Radenovic, F., Zhang, F., Synnaeve, G., Lee, G., Anderson, G. L., Nail, G., Mialon, G., Pang, G., Cucurell, G., Nguyen, H., Korevaar, H., Xu, H., Touvron, H., Zarov, I., Ibarra, I. A., Kloumann, I. M., Misra, I., Evtimov, I., Copet, J., Lee, J., Geffert, J., Vranes, J., Park, J., Mahadeokar, J., Shah, J., van der Linde, J., Billock, J., Hong, J., Lee, J., Fu, J., Chi, J., Huang, J., Liu, J., Wang, J., Yu, J., Bitton, J., Spisak, J., Park, J., Rocca, J., Johnstun, J., Saxe, J., Jia, J., Alwala, K. V., Upasani, K., Plawiak, K., Li, K., Heafield, K., Stone, K., and et al. The llama 3 herd of models. *CoRR*, abs/2407.21783, 2024.
- Ethayarajh, K., Xu, W., Muennighoff, N., Jurafsky, D., and Kiela, D. Model alignment as prospect theoretic optimiza-

- tion. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.
- Gao, J., He, D., Tan, X., Qin, T., Wang, L., and Liu, T. Representation degeneration problem in training natural language generation models. In *International Conference on Learning Representations*, 2019.
- Garg, S., Singh, A., Singh, S., and Chopra, P. 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, pp. 19425–19441. Association for Computational Linguistics, 2025.
- Garrido, Q., Balestrieri, R., Najman, L., and LeCun, Y. Rankme: assessing the downstream performance of pre-trained self-supervised representations by their rank. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.
- Gema, A. P., Leang, J. O. J., Hong, G., Devoto, A., Mancino, A. C. M., Saxena, R., He, X., Zhao, Y., Du, X., Ghasemi Madani, M. R., Barale, C., McHardy, R., Harris, J., Kaddour, J., Van Krieken, E., and Minervini, P. 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)*, pp. 5069–5096. Association for Computational Linguistics, April 2025. ISBN 979-8-89176-189-6. doi: 10.18653/v1/2025.nacl-long.262.
- Godey, N., de la Clergerie, É. V., and Sagot, B. Why do small language models underperform? studying language model saturation via the softmax bottleneck. In *First Conference on Language Modeling*, 2024.
- He, C., Luo, R., Bai, Y., Hu, S., Thai, Z., Shen, J., Hu, J., Han, X., Huang, Y., Zhang, Y., Liu, J., Qi, L., Liu, Z., and Sun, M. 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)*, pp. 3828–3850. Association for Computational Linguistics, August 2024a. doi: 10.18653/v1/2024.acl-long.211.
- He, C., Luo, R., Bai, Y., Hu, S., Thai, Z., Shen, J., Hu, J., Han, X., Huang, Y., Zhang, Y., Liu, J., Qi, L., Liu, Z., and Sun, M. 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)*, pp. 3828–3850. Association for Computational Linguistics, August 2024b. doi: 10.18653/v1/2024.acl-long.211.
- He, J., Gong, Y., Lin, Z., Wei, C., Zhao, Y., and Chen, K. LLM factoscope: Uncovering LLMs' factual discernment through measuring inner states. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 10218–10230. Association for Computational Linguistics, August 2024c.
- Hu, E. J., yelong shen, Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.
- Kim, S., Suk, J., Longpre, S., Lin, B. Y., Shin, J., Welleck, S., Neubig, G., Lee, M., Lee, K., and Seo, M. 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*, pp. 4334–4353. Association for Computational Linguistics, November 2024. doi: 10.18653/v1/2024.emnlp-main.248.
- knoveleng. Amc-23. Hugging Face Dataset, 2023. URL <https://huggingface.co/datasets/knoveleng/AMC-23>.
- Lambert, N., Morrison, J., Pyatkin, V., Huang, S., Ivison, H., Brahman, F., Miranda, L. J. V., Liu, A., Dziri, N., Lyu, S., Gu, Y., Malik, S., Graf, V., Hwang, J. D., Yang, J., Bras, R. L., Tafjord, O., Wilhelm, C., Soldaini, L., Smith, N. A., Wang, Y., Dasigi, P., and Hajishirzi, H. Tülu 3: Pushing frontiers in open language model post-training. *CoRR*, abs/2411.15124, 2024. doi: 10.48550/ARXIV.2411.15124.
- Lambert, N., Pyatkin, V., Morrison, J., Miranda, L. J. V., Lin, B. Y., Chandu, K. R., Dziri, N., Kumar, S., Zick, T., Choi, Y., Smith, N. A., and Hajishirzi, H. Reward-bench: 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*, pp. 1755–1797. Association for Computational Linguistics, 2025.
- Lee, H., Phatale, S., Mansoor, H., Mesnard, T., Ferret, J., Lu, K., Bishop, C., Hall, E., Carbune, V., Rastogi, A., and Prakash, S. RLAI 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*. Open-Review.net, 2024.
- Lightman, H., Kosaraju, V., Burda, Y., Edwards, H., Baker, B., Lee, T., Leike, J., Schulman, J., Sutskever, I., and Cobbe, K. Let's verify step by step. *arXiv preprint arXiv:2305.20050*, 2023.

- Lin, B. Y., Deng, Y., Chandu, K., Ravichander, A., Pyatkin, V., Dziri, N., Bras, R. L., and Choi, Y. Wildbench: Benchmarking LLMs with challenging tasks from real users in the wild. In *The Thirteenth International Conference on Learning Representations*, 2025.
- Liu, C. Y., Zeng, L., Xiao, Y., He, J., Liu, J., Wang, C., Yan, R., Shen, W., Zhang, F., Xu, J., Liu, Y., and Zhou, Y. Skywork-reward-v2: Scaling preference data curation via human-ai synergy. *arXiv preprint arXiv:2507.01352*, 2025.
- Mesgar, M. and Strube, M. A neural local coherence model for text quality assessment. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 4328–4339, Brussels, Belgium, October–November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1464.
- Mikolov, T., Yih, W.-t., and Zweig, G. 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*, pp. 746–751, Atlanta, Georgia, June 2013. Association for Computational Linguistics.
- OpenAI. Gpt-4o mini: A more efficient multimodal model, 2025. URL <https://openai.com/blog/gpt-4o-mini>.
- Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P. F., Leike, J., and Lowe, R. 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.
- Rafailov, R., Sharma, A., Mitchell, E., Manning, C. D., Ermon, S., and Finn, C. 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.
- Reimers, N. and Gurevych, I. 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.
- Rein, D., Hou, B. L., Stickland, A. C., Petty, J., Pang, R. Y., Dirani, J., Michael, J., and Bowman, S. R. GPQA: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*, 2024.
- Roy, O. and Vetterli, M. The effective rank: A measure of effective dimensionality. In *2007 15th European signal processing conference*, pp. 606–610. IEEE, 2007.
- Rudelson, M. and Vershynin, R. 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.
- Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Zhang, M., Li, Y. K., Wu, Y., and Guo, D. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *CoRR*, abs/2402.03300, 2024. doi: 10.48550/ARXIV.2402.03300.
- Singhal, P., Goyal, T., Xu, J., and Durrett, G. A long way to go: Investigating length correlations in RLHF. In *First Conference on Language Modeling*, 2024.
- Skalse, J., Howe, N. H. R., Krasheninnikov, D., and Krueger, D. 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.
- Wang, J., Wang, H., Sun, S., and Li, W. 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.
- Wang, P., Li, L., Chen, L., Cai, Z., Zhu, D., Lin, B., Cao, Y., Kong, L., Liu, Q., Liu, T., and Sui, Z. 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, pp. 9440–9450. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.ACL-LONG.511.
- Wu, Z., Hu, Y., Shi, W., Dziri, N., Suhr, A., Ammanabrolu, P., Smith, N. A., Ostendorf, M., and Hajishirzi, H. 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.
- Yang, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Li, C., Liu, D., Huang, F., Wei, H., Lin, H., Yang, J., Tu, J., Zhang, J., Yang, J., Yang, J., Zhou, J., Lin, J., Dang, K., Lu, K., Bao, K., Yang, K., Yu, L., Li, M., Xue, M., Zhang, P., Zhu, Q., Men, R., Lin, R., Li, T., Xia, T., Ren, X., Ren, X., Fan, Y., Su, Y., Zhang, Y., Wan, Y., Liu, Y., Cui, Z., Zhang, Z., and Qiu, Z. Qwen2.5 technical report.

- CoRR*, abs/2412.15115, 2024. doi: 10.48550/ARXIV.2412.15115.
- Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.
- Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Gao, C., Huang, C., Lv, C., Zheng, C., Liu, D., Zhou, F., Huang, F., Hu, F., Ge, H., Wei, H., Lin, H., Tang, J., Yang, J., Tu, J., Zhang, J., Yang, J., Yang, J., Zhou, J., Lin, J., Dang, K., Bao, K., Yang, K., Yu, L., Deng, L., Li, M., Xue, M., Li, M., Zhang, P., Wang, P., Zhu, Q., Men, R., Gao, R., Liu, S., Luo, S., Li, T., Tang, T., Yin, W., Ren, X., Wang, X., Zhang, X., Ren, X., Fan, Y., Su, Y., Zhang, Y., Zhang, Y., Wan, Y., Liu, Y., Wang, Z., Cui, Z., Zhang, Z., Zhou, Z., and Qiu, Z. Qwen3 technical report. *CoRR*, abs/2505.09388, 2025. doi: 10.48550/ARXIV.2505.09388.
- Yang, Z., Dai, Z., Salakhutdinov, R., and Cohen, W. W. Breaking the softmax bottleneck: A high-rank RNN language model. In *International Conference on Learning Representations*, 2018.
- Yin, Z., Sun, Q., Zeng, Z., Cheng, Q., Qiu, X., and Huang, X. Dynamic and generalizable process reward modeling. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 4203–4233, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.212.
- Yuan, W., Pang, R. Y., Cho, K., Li, X., Sukhbaatar, S., Xu, J., and Weston, J. Self-rewarding language models. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024.
- Zhang, Q., Qiu, L., Hong, I., Xu, Z., Liu, T., Li, S., Zhang, R., Li, Z., Li, L., Yin, B., Zhang, C., Chen, J., Jiang, H., and Zhao, T. Self-rewarding PPO: Aligning large language models with demonstrations only. In *Second Conference on Language Modeling*, 2025a.
- Zhang, Y. and Math-AI, T. American invitational mathematics examination (aime) 2025, 2025.
- Zhang, Z., Zheng, C., Wu, Y., Zhang, B., Lin, R., Yu, B., Liu, D., Zhou, J., and Lin, J. The lessons of developing process reward models in mathematical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 10495–10516. Association for Computational Linguistics, July 2025b. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.547.
- Zheng, L., Chiang, W., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E. P., Zhang, H., Gonzalez, J. E., and Stoica, I. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Advances in Neural Information Processing Systems 36: Annual*



**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 (Lambert et al., 2025). We test three model families: Qwen3-8B (36 layers) (Yang et al., 2025), Qwen2.5-1.5B-Instruct (28 layers) (Yang et al., 2024), and Phi-3.5-mini-instruct (32 layers) (Abdin et al., 2024), 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 (Kim et al., 2024) and (2) Pairwise Comparison (Kim et al., 2024).

#### 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 (Garg et al., 2025) 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  $\ell \in \mathbb{R}^{|V|}$  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:

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

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. RewardBench Results

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

## Stable Rank as Intrinsic Reward Signal

---

*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

## D. 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:  $\Delta\text{Rand}$ . measures improvement over random selection at the same N, and  $\Delta@1$  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.

## E. 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  $\alpha=32$ , 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 (bfloating16), 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 ( $r=16$ ,  $\alpha=32$ ). Due to the model’s longer reasoning traces, we use per-device batch size of 2 with 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

## Stable Rank as Intrinsic Reward Signal

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

N	Benchmark	Qwen2.5-1.5B		Phi-3.5-mini		Llama-3.2-1B		DeepSeek-R1-1.5B	
		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. $\Delta$	33.6 +14.4%	38.4	34.3 +5.4%	36.2	19.9 +13.6%	22.6	56.7 +3.6%	58.7
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. $\Delta$	33.9 +15.7%	39.2	34.9 +8.4%	37.8	18.4 +34.2%	24.8	56.5 +5.7%	59.7
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. $\Delta$	36.3 +13.0%	41.0	34.7 +11.4%	38.7	19.8 +33.8%	26.5	57.8 +5.0%	60.7

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 (Allal et al., 2025) 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.

## F. Experimental Details for Metric Analysis

We analyze the relationship between stable rank and text quality using the RewardBench dataset (Lambert et al., 2025), 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.

### F.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  $r = \text{corr}(M, S)$  and Spearman rank correlation  $\rho$ . 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.

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 ( $\beta$ )	0.04	0.04
LoRA rank ( $r$ )	16	16
LoRA alpha ( $\alpha$ )	32	32
Max prompt length	4096	8192
Max completion length	4096	4096
Precision	bfloat16	bfloat16

**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

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

and correlate  $\Delta M$  with  $\Delta S$  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.

## F.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 ( $N_{\text{tokens}}$  tokens),  $s_1, \dots, s_n$  its sentences with embeddings  $v_1, \dots, v_n$  (from all-MiniLM-L6-v2 (Reimers & Gurevych, 2019)), and  $v_p$  the prompt embedding.

## F.3. Implementation Details

**Sentence embeddings.** We use the all-MiniLM-L6-v2 (Reimers & Gurevych, 2019) 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<sup>1</sup>, with a regex-based fallback (splitting on sentence boundaries while handling common abbreviations). Each response contains  $n$  sentences indexed  $1, \dots, n$ .

**Syllable counting.** For readability metrics, we count syllables using the pyphen<sup>2</sup> library (dictionary-based hyphenation for English). Complex words are defined as having  $\geq 3$  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  $(c/N_{\text{tokens}}) \times 100$ .

## F.4. Correlation Results

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

<sup>1</sup><https://www.nltk.org/>

<sup>2</sup><https://pyphen.org/>

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

Metric	Formula	Interpretation
<i>Intra-response coherence</i>		
Progression score	$\frac{1}{n-1} \sum_{i=1}^{n-1} \cos(v_i, v_{i+1}) - \frac{1}{ \mathcal{P} } \sum_{(i,j) \in \mathcal{P}} \cos(v_i, v_j)$ 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	$\frac{1}{\binom{n}{2}} \sum_{1 \leq i < j \leq n} [1 - \cos(v_i, v_j)]$	Average pairwise cosine distance. Higher values indicate scattered content; lower values indicate tightly focused responses.
Semantic variance	$\frac{1}{384} \sum_{d=1}^{384} \text{Var}([v_{1,d}, \dots, v_{n,d}])$	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.
<i>Prompt-response alignment (using prompt embedding <math>v_p</math>)</i>		
QA alignment	Compute $a_i = \cos(v_p, v_i)$ for $i = 1, \dots, n$ . 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.

## Stable Rank as Intrinsic Reward Signal

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

Metric	Formula / Computation	Interpretation
<i>Lexical variety</i>		
Type–token ratio	$ V /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	$\frac{1}{N_{\text{tokens}} - w + 1} \sum_{i=1}^{N_{\text{tokens}} - w + 1} \frac{ V_i }{w}$ 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 $\geq 2$ times	Measures sustained redundant phrases. Captures copy-paste patterns or looping behavior.
<i>Information content and redundancy</i>		
Compression ratio	$\frac{ zlib.compress(y) }{ y }$	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.
<i>Model confidence (using Qwen2.5-7B)</i>		
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	$\frac{1}{N_{\text{tokens}}} \sum_{i=1}^{N_{\text{tokens}}} H(p(t_{i+1} \mid t_{\leq i}))$ where $H$ is entropy	Average next-token entropy. Higher values indicate less confidence, more random token choices.

### F.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 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.

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

## G. Ablation: 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(Yang et al., 2025) 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** (Roy & Vetterli, 2007):  $\text{er}(\mathbf{X}) = \exp(H(\tilde{\sigma}))$ , 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.
- **PCA 95% variance:** The number of principal components needed to capture 95% of the variance in the hidden states.

**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.

## H. Ablation: 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

*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 $\rho$	$p$
<i>Semantic coherence and alignment</i>				
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
<i>Information density and diversity</i>				
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
<i>Length and structure</i>				
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 $\rho$	$p$
<i>Positive associations</i>				
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
<i>Negative associations</i>				
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

## Stable Rank as Intrinsic Reward Signal

---

*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	<b>87.23</b>	<b>84.02</b>	<b>76.20</b>	<b>88.41</b>	<b>84.34</b>	<b>84.04</b>

rank on the final layer of Qwen3-8B (Yang et al., 2025). RewardBench examples have a mean length of 281 tokens, with 75% under 512 tokens.

*Table 14.* 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 14 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.

## I. Ablation: 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}

## Stable Rank as Intrinsic Reward Signal

---

*Table 15.* 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

**Results.** Table 15 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.

## J. 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 16 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 16.* 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"	<p><b>High Stable Rank:</b> When working with PNG files, there are several ways to control their size and quality: 1. <b>Use ImageMagick or Other Command-Line Tools:</b> convert input.png -resize 100x100 output.png (resizes) and convert input.png -quality 85 output.png (compresses). 2. <b>Use JavaScript Libraries:</b> Libraries like canvas allow you to manipulate and resize images. (Includes the full JavaScript canvas code snippet) 3. <b>Use Online Tools:</b> There are online tools like OnlineImageResize. 4. <b>Use WebP:</b> Consider using WebP, which is more efficient... convert input.png -quality 85 output.webp.</p> <p><b>Low Stable Rank:</b> PNG (Portable Network Graphics) is a lossless image format... If you need to control the size or quality... <b>Steps to Control PNG Size or Quality:</b> 1. <b>Using Photoshop:</b> - Open the PNG image. - Go to Image &gt; 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 &gt; Adjustments &gt; Image &gt; ...</p>
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?"	<p><b>High Stable Rank:</b> To solve this problem, we can follow these steps: 1. <b>Calculate their combined speed:</b> <math>18 + 12 = 30</math> miles per hour. 2. <b>Determine the time it takes for them to meet:</b> <math>\text{Time} = \text{Distance}/\text{Speed} = 45/30 = 1.5</math> hours. 3. <b>Calculate the distance covered by Alicia:</b> <math>\text{Distance} = 18 \times 1.5 = 27</math> miles. Therefore, the final answer is: <b>27</b>.</p> <p><b>Low Stable Rank:</b> To solve this problem, we need to determine the distance covered by Alicia and Beth... 1. Combined speed is <math>18 + 12 = 30</math> miles per hour. 2. Time it takes for them to meet: <math>45 \text{ miles}/30 \text{ mph} = 1.5</math> hours. 4. Distance covered by Alicia: <math>18 \text{ mph} \times 1.5 \text{ h} = 27</math> 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 <math>18/(18 + 12) = 18/30 = 0.6</math> of the distance.</p>