On the Implicit Reward Overfitting and the Low-rank Dynamics in RLVR

Anonymous Author(s)
Affiliation
Address
email

Abstract

Recent extensive research has demonstrated that the enhanced reasoning capabilities acquired by models through Reinforcement Learning with Verifiable Rewards (RLVR) are primarily concentrated within the rank-1 components. Predicated on this observation, we employed Periodic Rank-1 Substitution and identified a counterintuitive phenomenon: RLVR may exhibit implicit reward overfitting to the training dataset. Specifically, the model can achieve satisfactory performance on the test set even when its rewards remain relatively low during the training process. Furthermore, we characterize three distinct properties of RL training: (1) The effective rank-1 component in RLVR don't maintain other model knowledge except mathematical reasoning capability. (2) RLVR fundamentally functions by optimizing a specific singular spectrum. The distribution of singular values of almost all linear layers in RLVR-trained model behaves like heavy-tailed distribution. (3) the left singular vectors associated with rank-1 components demonstrate a stronger alignment tendency during training, which echoes the discovery that RLVR is optimizing sampling efficiency in essence. Taken together, our findings and analysis further reveal how RLVR shapes model parameters and offer potential insights for improving existing RL paradigms or other training paradigms to implement continual learning. Source code is available at https://anonymous.4open.science/r/implicit-reward-overfitting-0BF7/

1 Introduction

In recent years, Reinforcement Learning (RL) [27, 28, 24, 2] has emerged as a core paradigm for aligning Large Language Models (LLMs) [35] with human preferences and, more importantly, for eliciting complex reasoning capabilities within these models. From early methods like RLHF [21] to recent reasoning-oriented models such as DeepSeek-R1 [11] and Kimi-K2 [34], RL has demonstrated remarkable potential. However, despite the community's immense success in engineering practices, our understanding of how RL actually alters model parameters remains underdeveloped. Existing research [29, 14] predominantly focuses on macroscopic training dynamics, interaction to agents or reward design, while few studies delve into the microscopic level to investigate the specific structure and physical significance of RL updates ΔW within the parameter space.

This lack of insight into microscopic mechanisms directly contributes to our confusion regarding the boundaries of RL capabilities. Recently, a substantial body of researches [40] have observed that LLMs exhibit low-rank characteristics when adapting to downstream tasks, and RL appears to be no exception. A recent study [3] suggests that the reasoning improvements conferred by RL are primarily concentrated within the Rank-1 component of weight updates. Closely related work has boldly proposed the universal weight subspace hypothesis [17]. Furthermore, another disruptive study [39] recently found that even without RL training, the pass@k [5] metric of models continues to improve as k increases. This implies that RL may not be imparting new underlying reasoning logic

Submitted to 40th Conference on Neural Information Processing Systems (NeurIPS 2026). Do not distribute.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

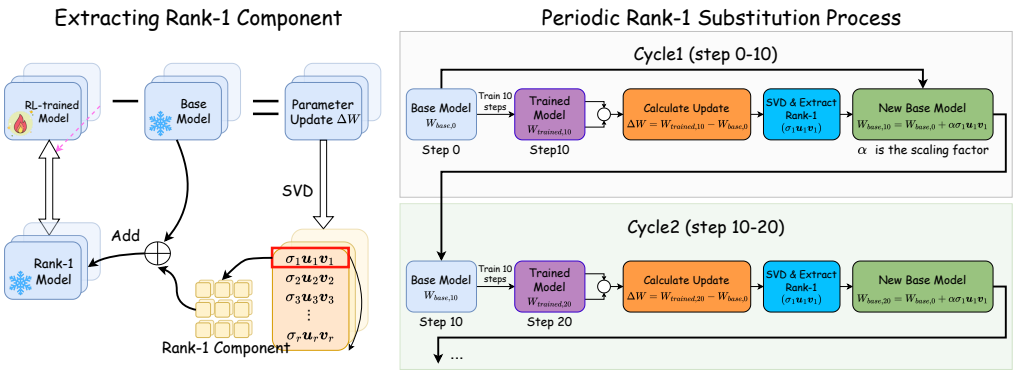

Figure 1: **Left**: The process of extracting Rank-1 component from RL-trained model. We pick out the rank-1 matrix corresponding to the greatest singular value. **Right**: The process of periodic Rank-1 substitution. base model is repeatedly trained for a short interval, and only a rank-1 approximation of its weight update is kept.

to the model, but rather optimizing its sampling strategy to efficiently elicit latent correct answers. This perspective has triggered profound questioning within the community regarding the true nature of RLVR: is RLVR genuinely learning reasoning, or is it merely fitting the reward?

Another point of interest is the low-rank property of RLVR. A research from Thinking Machines [26] found that LoRA performs equivalently to full fine-tuning for reinforcement learning even with small ranks. They find that RL requires very low capacity, a result based on information-theoretical arguments. This observation echoes the fact mentioned above that the rank-1 component of parameter update obtained in RLVR contains most of the reasoning capability [3]. Therefore, it's significant to further explore how LoRA learns this rank-1 component under the hood.

To unravel these mysteries, this paper conducts an in-depth empirical study and theoretical analysis starting from the spectral properties of weight updates. Our work reveals three critical phenomena during the RLVR training process:

First, we identify that RLVR may suffer from Implicit Reward Overfitting. By introducing a technique called Periodic Rank-1 Substitution, we discover that a vast number of non-Rank-1 components in RLVR (e.g., GRPO [30], DAPO [38], GSPO [43]) do not contribute to the improvement of reasoning capabilities. By periodically eliminating these components, we find that on a significant portion of datasets, the model's performance on the test set remains comparable, even though its training reward is lower than that of Full RL Fine-tuning. This indicates that while non-Rank-1 components significantly boost the reward on the training set, they fail to translate into generalized performance on the test set. Furthermore, due to the nature of the reward function in the RL process, the model may be overfitting to the training data.

However, if the rank-1 component is enough to obtain equivalent reasoning improvement, what's the side-effect of discarding non-rank-1 component during training? Thus, we further observed the effect of non-rank-1 components and corroborated that non-rank-1 components maintain the model's out-of-domain capability, such as world knowledge, instruction-following ability and safety, etc. Besides, we analyzed the singular spectra (the distribution of singular values) of all linear layers in RLVR-tuned models and found that they follow a RL-induced pattern: a leading spike followed by a heavy-tailed distribution.

Third, we explain the geometric dynamics of LoRA in RL: the Geometric Asymmetry of Input-Output Subspaces. We find that weight updates of almost all linear layers induced by RLVR exhibit strong Rank-1 characteristics on the output side (Output Space), which easily align with the left singular vectors (corresponding to adjustments in sampling strategy) of the greatest singular value. However, on the input side (Input Space) , updates involve complex reorganization of contextual features, which are difficult to capture with simple low-rank matrices. This observation also reflects the fact that RLVR is inherently optimizing sampling(output) efficiency.

To summarize, the contributions of our work are:

- We reveal **potential implicit overfitting** in RLVR, and we give a plausible explanation to a disruptive discovery that reinforcement learning don't incentivize reasoning capacity in LLMs beyond the base model [39].
- We find that non-rank-1 components in RLVR training maintain the overall knowledge and ability of model except reasoning ability.
- We discover that the singular spectrum of the parameter update of a RL-tuned model follow a consistent pattern: a leading spike followed by a heavy-tailed distribution.
- We provide an in-depth analysis of the training dynamics of LoRA in RLVR: the asymmetry of the input-output space.

2 Rank-1 Dominance in RL: Decoupling underlying Reasoning from Implicit Reward Overfitting

2.1 Preliminary: The Dominance of Rank-1 Subspace

Suppose ΔW is the parameter updates after RL training. We perform SVD on ΔW:

$$\Delta W = \sum_{i=1}^{r} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^T$$

where σ_i are singular values in descending order and $\boldsymbol{u}_i$,$\boldsymbol{v}_i$ are left and right singular vectors. The **Rank-1 Component** (or **Rank-1 Subspace**) denotes the rank-1 matrix obtained from the greatest singular value:

$$\Delta W^{(1)} = \sigma_1 \boldsymbol{u}_1 \boldsymbol{v}_1^T$$

To ensure consistency in update strength, this Rank-1 component should be rescaled by Frobenius norm to match the magnitude of the original update:

$$\Delta \hat{W}^{(1)} = \frac{\|\Delta W\|_F}{\|\Delta W^{(1)}\|_F} \Delta W^{(1)}$$

The Rank-1 model is obtained by adding $\Delta \hat{W}^{(1)}$ to base model. Existing study found that the rank-1 model boasts virtually equivalent reasoning capability compared to model fully fine-tuned by RL [3] and ablation study demonstrated that cumulative rank-k components performs almost equally to the rank-1 component as k grows.

2.2 Periodic Rank-1 Substitution: Eliminating Non-rank-1 Component

Extensive empirical evidence from prior studies suggests that the enhancement in reasoning capabilities elicited by RLVR is predominantly encapsulated within the Rank-1 component of the weight updates. This observation naturally induces a critical question: within the RL weight update matrix ΔW, do the residual non-Rank-1 components merely constitute stochastic noise inherent to the training process?

To empirically interrogate this noise hypothesis, we conducted comparative training utilizing both standard GRPO and a variant we term GRPO with **Periodic Rank-1 Substitution** on the Countdown-3to4 dataset. As illustrated in Figure 1, our proposed Periodic Rank-1 Substitution mechanism operates as follows: at intervals of 10 training steps, we perform Singular Value Decomposition (SVD) on the accumulated weight updates relative to the base model. Subsequently, the extracted Rank-1 component is added onto the base parameters, serving as the initialization for the subsequent 10-step training window. The remaining training steps proceeds likewise.

2.3 Divergence between Training Reward and Test Generalization

Figure 2 presents the trajectories of the training mean reward alongside the evaluation metrics on the test set across varying training steps for both strategies. We observe a counterintuitive and prominent mismatch between the training reward and testing performance: while standard GRPO exhibits a consistent upward trend and achieves significantly higher training rewards compared to the Periodic

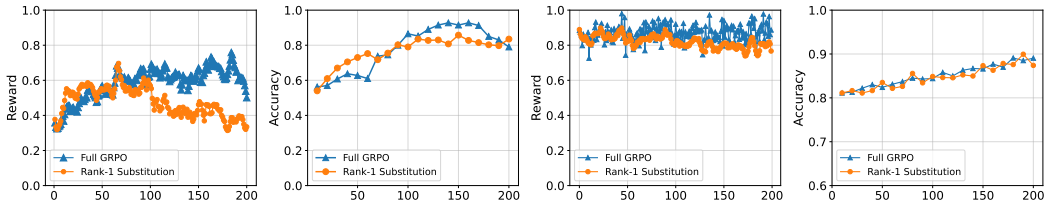

Figure 2: **Left**: The mean reward within each batch during GRPO training for Qwen2.5-7B-Instruct [23]. **Mid-left**: Test-set accuracy of the leftmost figure. **Mid-right**: The mean reward within each batch during GRPO training for Llama3.1-8B-Instruct [9]. **Right**: Test-set accuracy of the Mid-right figure. The horizontal axes of four subfigures are training steps. We don't use Qwen3 as post-training has enabled the lightweight models in the Qwen3 family to acquire reasoning capabilities distilled from larger, RL-trained models, as described by the Qwen3 tech reports [37], which makes it harder to measure what is being learned only during RLVR.

Rank-1 Substitution variant, this advantage does not translate into a superiority in test performance, where only a marginal gap is observed.

A recent study posits that RLVR does not genuinely elicit the model's underlying reasoning ability. Rather, it primarily optimizes the sampling strategy to efficiently prioritize correct solutions. Consequently, due to the reduced diversity in sampling post-RL, the model may fail to answer a subset of queries that the base model could originally solve, particularly when evaluated under larger k values for pass@k.

Our empirical findings provide robust support for this assertion: satisfactory evaluation metrics on the test set do not preclude the possibility of overfitting to the training data. Specifically, existing RLVR paradigms are potentially prone to implicit reward overfitting. To put it in another perspective, there exists a potential risk of over-optimizing the sampling strategy during the RL process. This elucidates why, as Pass@k increases, the performance of the standard RL model may unexpectedly fall short of the base model [8]. By extracting the weight components responsible for reasoning and filtering out non-Rank-1 noise, our Periodic Rank-1 Substitution strategy effectively mitigates this implicit reward overfitting while retaining the desired target weight updates for reasoning performance, thereby preserving generalization capabilities. From the perspective of learning dynamics, eliminating the noise that artificially inflates training rewards forces the model to concentrate on core reasoning mechanisms. Even with a weaker reward signal, the model successfully acquires the critical logic for problem-solving through calibration along these principal directions.

3 Non-rank-1 Components are Better Than Nothing

3.1 Non-rank-1 Components are NOT Functionally Redundant

As illustrated in Figure 3, we evaluated the performance of both the fully RLVR-tuned model and the model fine-tuned by RLVR with periodic rank-1 substitution on the SafetyBench benchmark. We observed a prominent degradation in safety metrics following the Rank-1 extraction. This suggests that while the non-Rank-1 components contribute minimally to the enhancement of reasoning capabilities, they may encode other behavioral shifts acquired during the RL process.

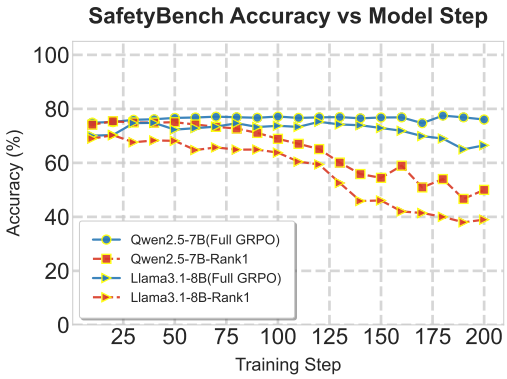

Figure 3: Obvious performance degradation in safety after RLVR with periodic rank-1 subsitution. Models are the same as those in Figure 2

To further investigate the role of non-rank-1 components during the RLVR process, and taking into account that the results shown in Figure 3 might be attributed to frequent manual parameter

Model	Benchmark			
	IF-Eval	MMLU	MMLU-Pro	SafetyBench
Instruction-tuned or RLVR-tuned Model				
Qwen2.5-7B-Instruct	78.66 ± 1.2	74.64 ± 1.8	45.01 ± 1.4	78.34 ± 1.9
Qwen2.5-7B-R1	78.18 ± 1.1	74.29 ± 1.5	52.93 ± 2.1	77.41 ± 1.7
DeepMath-Zero-7B	38.73 ± 1.9	77.95 ± 1.2	60.07 ± 1.4	82.32 ± 1.1
Qwen2.5-32B-DAPO	53.36 ± 0.7	86.51 ± 0.8	68.32 ± 1.1	86.61 ± 0.9
Qwen3-8B-DAPO-Math	52.04 ± 1.4	76.55 ± 2.0	62.40 ± 1.6	54.87 ± 1.3
Llama3.1-8B-Thinking-R1	60.91 ± 1.6	61.54 ± 1.7	35.38 ± 1.5	45.22 ± 2.2
Rank-1 Model				
Qwen2.5-7B-R1-rank1	74.64 ± 1.3	71.15 ± 1.6	49.57 ± 1.8	73.22 ± 1.9
DeepMath-Zero-7B-rank1	31.57 ± 1.8	74.81 ± 1.4	57.24 ± 1.6	78.50 ± 1.3
Qwen2.5-32B-DAPO-rank1	46.73 ± 0.8	83.43 ± 0.9	64.50 ± 1.2	82.16 ± 1.1
Qwen3-8B-DAPO-Math-rank1	55.04 ± 1.5	73.94 ± 1.4	58.89 ± 1.3	51.08 ± 1.8
Llama3.1-8B-Thinking-R1-rank1	57.67 ± 1.1	67.79 ± 1.9	41.88 ± 1.2	66.14 ± 2.0

Table 1: Out-of-domain ability of models before/after rank-1 extraction, including instruction-following(IF-Eval), world knowledge(MMLU, MMLU-Pro) and safety(SafetyBench). Note that DeepMath-Zero-7B [12] performs especially bad on IF-Eval as the post-training of it utilized Open-Reasoner [15] chat template instead of default chat template, according to its technical report.

updates applied to the model during training, we conducted an additional comparative experiment. To ensure generalizability, we analyzed the performance of LLMs with varying parameter sizes, from different model families, trained on diverse datasets using various reinforcement learning methods. Specifically, we evaluated their performance on non-mathematical reasoning benchmarks before and after rank-1 extraction. Here, the rank-1 extraction is only performed once right after the whole training process. For example, Llama3.1-8B-Thinking-R1-rank1 is obtained by adding rank-1 component to Llama3.1-8B-Instruct [9]. As shwon in Table 1, there are two key observations:

Non-rank-1 components maintain out-of-domain ability. In most cases, models after rank-1 extraction exhibit varying degrees of significant performance degradation on non-mathematical reasoning tasks. This indicates that even a one-time rank-1 extraction can impair the model's capabilities and knowledge. Therefore, although we find that periodic rank-1 substitution can mitigate potential implicit reward overfitting in RLVR, it cannot serve as an effective improvement strategy for RLVR. In other words, non-rank-1 components in RLVR play a crucial role in preserving the model's non-reasoning abilities, thereby preventing catastrophic forgetting akin to that observed in supervised fine-tuning (SFT).

Llama3.1-8B-Thinking-R1 seems to be a counterexample. After rank-1 extraction, Llama3.1-8B-Thinking-R1-rank1 not only avoids performance degradation but also exhibits a significant performance improvement compared to Llama3.1-8B-Thinking-R1 [16]. This is because it is the only model in the table that has not been trained exclusively via RLVR. In fact, Llama3.1-8B-Thinking-R1 is derived from Llama-3.1-8B-Instruct through a three-stage training pipeline comprising cold-start SFT, GRPO, and CoT distillation SFT. Consequently, since SFT induces substantial shifts in the model's parameter space, the extracted rank-1 component no longer encapsulates solely the reasoning capabilities attributable to pure RLVR, but also incorporates other knowledge. We posit that, under such circumstances, rank-1 extraction may help mitigate catastrophic forgetting. However, it remains challenging to guarantee that the model can preserve the reasoning abilities achieved through RL.

3.2 The Universal Spectral Pattern of ΔW in RLVR

We performed SVD on the weight difference ΔW before and after RL training, sorting the singular values in descending order to obtain the singular spectrum shown in Figure 4. Through extensive verification, we identified a universal pattern in the singular spectrum of ΔW: the first singular value σ_1 is notably large, followed by a rapid initial decay, while the subsequent singular values exhibit a uniform, quasi-linear rate of decay. This indicates that the tail singular vectors are systematically induced by the training process rather than being artifacts of random stochastic noise.

Furthermore, we observe that although the improvement in reasoning capability is primarily concentrated in the Rank-1 component, the effective rank of ΔW remains substantial. This implies that structurally, or numerically, the rank-1 component does not dominate the composition of the matrix relative to the non-rank-1 parts(in other words, rank-1 component doesn't bear the most energy of the weight). Synthesizing these findings, we propose a novel perspective: RL does not merely learn a strictly low-rank update but rather induces a highly structured spectral shape. Specifically, a dominant Rank-1 direction acts as a switch to activate reasoning capabilities, while RL systematically generates a series of linearly decaying secondary singular values. These non-Rank-1 components, characterized by low amplitude but consistent structure across layers and models, are responsible for safety, robustness, and subtle feature alignment, jointly refining the model's overall competence.

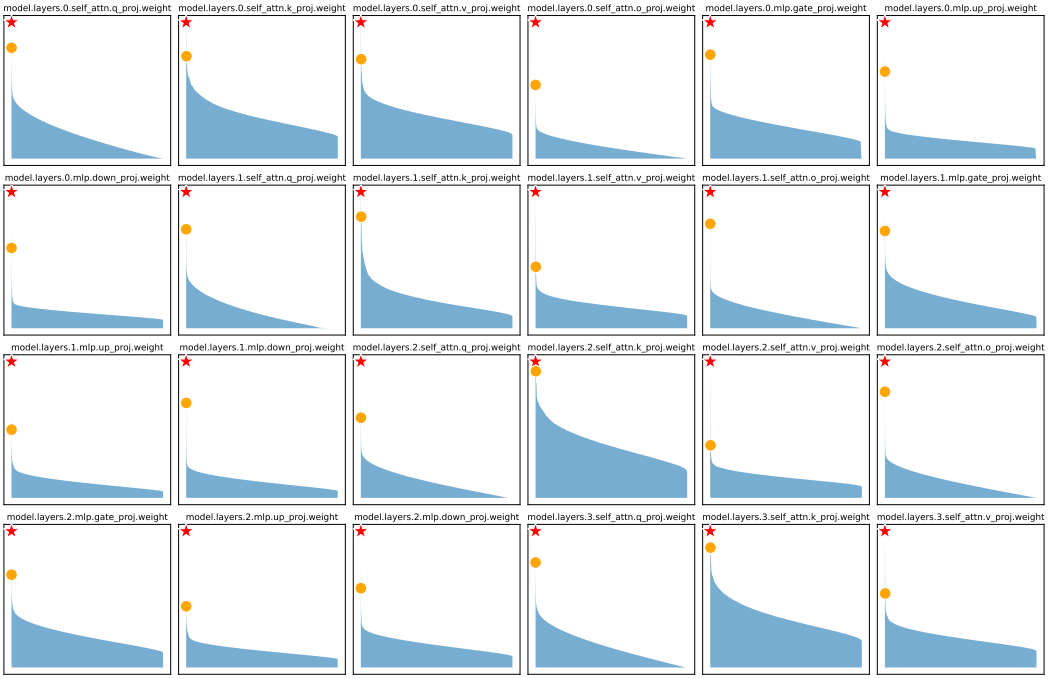

Figure 4: The distribution of the singular values of each linear layer update. The greatest singular value is marked by a red star, the second largest singular value is marked by a yellow circle. It's obvious that the heavy-tail part of non-rank-1 components take almost all energy of the parameter update, which echoes the need of the scaling factor in Section 2. Without scaling, the magnitude of the rank-1 component would be too small to come into effect. In another perspective, rank-1 component captures the desired direction within parameter space.

3.3 A Theoretical Perspective

An outstanding study at ICLR [25] lends strong support to the fact that the gradient of all RL objective functions can be formulated as $A * \nabla_\theta \log \pi_\theta$ in essence. In RLVR context, the model ultimately produces a policy $\pi_\theta(o_{i,t} \mid q, o_{i,<t})$. For on-policy RL methods (GRPO/DAPO/GSPO family), the core term generally contains (we leave out unnecessary clip, min, etc.)

$$J(\theta) = \mathbb{E}_{(q,a)\sim\mathcal{D},\{o_i\}\sim\pi_\theta}\left[\hat{A}_{i,t}\pi_\theta(o_{i,t} \mid q, o_{i,<t})\right].$$

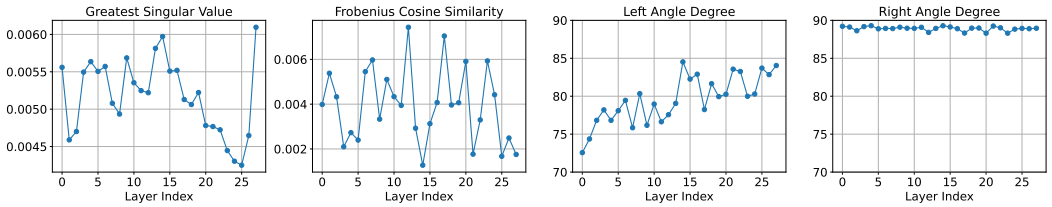

Figure 5: Layer-wise alignment analysis between ΔW_{LoRA} and $\Delta W^{(1)}$. From left to right: greatest singular value, Frobenius cosine similarity, left principal angle, and right principal angle across layers. The Frobenius cosine similarity remains near zero, indicating negligible global correlation in parameter space. In contrast, the subspace angle distributions show consistently smaller left principal angles than right ones, suggesting that the LoRA adapter, particularly its output-side component, tends to align more strongly with the left singular subspace during training.

The gradient of the weights for this layer (via chain rule) becomes

$$\nabla_W J = \mathbb{E}_{(q,a)\sim\mathcal{D},\{o_i\}\sim\pi_\theta}\left[\hat{A}_{i,t}\nabla_W \pi_\theta(o_{i,t} \mid q, o_{i,<t})\right]$$

Let the layer input be $\boldsymbol{x} \in \mathbb{R}^{d_{\text{in}}}$, weights $W \in \mathbb{R}^{d_{\text{out}}\times d_{\text{in}}}$, and the output hidden state be $\boldsymbol{h} = W\boldsymbol{x}$. Define the backpropagated vector to the layer output as

$$\boldsymbol{\delta} \triangleq \frac{\partial \pi_\theta(o_{i,t} \mid q, o_{i,<t})}{\partial \boldsymbol{h}} \in \mathbb{R}^{d_{\text{out}}}.$$

Given $\boldsymbol{h} = W\boldsymbol{x}$, matrix differential gives $\mathrm{d}\boldsymbol{h} = \mathrm{d}W\boldsymbol{x}$. Thus

$$\mathrm{d}\pi_\theta(o_{i,t} \mid q, o_{i,<t}) = \boldsymbol{\delta}^T\mathrm{d}\boldsymbol{h} = \boldsymbol{\delta}^T(\mathrm{d}W\boldsymbol{x}) = \mathrm{tr}(\boldsymbol{\delta}^T\mathrm{d}W\boldsymbol{x}) = \mathrm{tr}(\boldsymbol{x}\boldsymbol{\delta}^T\mathrm{d}W)$$

$$\implies \nabla_W \pi_\theta(o_{i,t} \mid q, o_{i,<t}) = \boldsymbol{\delta}\boldsymbol{x}^T$$

Substituting into J:

$$\nabla_W J = \mathbb{E}_{(q,a)\sim\mathcal{D},\{o_i\}\sim\pi_\theta}\left[\hat{A}_{i,t}\boldsymbol{\delta}\boldsymbol{x}^T\right]$$

For each token level probability, the gradient backpropagated to the weight matrix is a rank-1 matrix. The overall gradient is the average of these rank one matrices, which supports the rank-1 dominance in RLVR.

4 LoRA Unveils the Training Dynamics of RLVR

4.1 Experimental Perspective: Alignment in Output Space

It comes as a natural idea to use LoRA on RLVR because the weight update formulation in LoRA bears a striking formal resemblance to the Rank-1 structure:

$$\Delta W_{LoRA} = \alpha A_{m\times r}B_{r\times n} \qquad \Delta W^{(1)} = \sigma_1 \boldsymbol{u_1}\boldsymbol{v_1}^T$$

By setting $r = 1$, the LoRA adapter matrices degenerate into vectors, rendering the two formulations mathematically identical. Consequently, we trained LoRA adapters(with $r = 1$) using GRPO on the same dataset in Figure 1. To deeply probe the degree of alignment between LoRA updates and the true Rank-1 subspace, we employ two intrinsic metrics to quantify the approximation:

Metric 1: Frobenius Cosine Similarity, which evaluates the approximation of the weight update matrix from a holistic perspective:

$$\text{Alignment} = \frac{\|\Delta W_{LoRA} \odot \Delta W^{(1)}\|_F}{\|\Delta W_{LoRA}\|_F \cdot \|\Delta W^{(1)}\|_F}$$

Metric 2: Principal Angles (Left and Right), which measure the alignment between the constituent subspaces (components) of the update matrices.

$$\theta_{\text{left}} = \arccos\left|\frac{A^T_{m\times 1}\boldsymbol{u_1}}{\|A_{m\times 1}\|_F}\right|$$

$$\theta_{\text{right}} = \arccos\left|\frac{B_{1\times n}\boldsymbol{v_1}}{\|B_{1\times n}\|_F}\right|$$

As illustrated in Figure 5, we find that from a holistic view, the Frobenius cosine similarity between ΔW_{LoRA} and $\Delta W^{(1)}$ is negligible, suggesting a lack of correlation in the parameter space. However, the distribution of subspace angles reveals a noteworthy phenomenon: the left principal angle is consistently smaller than the right. This indicates that during training, the LoRA adapter (specifically the output-side component A) exhibits a stronger tendency to align with the left singular vectors compared to the input side.

We attribute this to the distinct semantic information encoded within the input and output spaces. Geometrically, for a given weight matrix W, the left singular vectors(orthonormal eigenvectors of WW^T) form an orthonormal basis for its column space, constituting a representation of the layer's output, which is essentially the model's expressivity or sampling policy. Since RLVR fundamentally optimizes the sampling strategy or output representation, the shared semantic coordinate system on the output side is more easy to alignment. Conversely, the right singular vectors reside in the input-side feature combination space, which is highly contingent upon upstream activations and the contextual distribution. RL updates on the input side involve complex feature reorganization rather than simple directional adjustments, resulting in a dispersed input space that is difficult to align via simple low-rank matrices. In short, the left and right singular vectors exhibit asymmetric alignability in RLVR.

4.2 Theoretical Proof

This can also be explained from a theoretical perspective on gradient analysis. Consider a linear layer

$$\boldsymbol{y} = W\boldsymbol{x}, \quad W \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}, \boldsymbol{x} \in \mathbb{R}^{d_{\text{in}}}$$

When LoRA of rank-1 is applied, the weight of this linear layer is:

$$W' = W + \Delta W, \qquad \Delta W = \alpha \boldsymbol{a}\boldsymbol{b}^T$$

where $a \in \mathbb{R}^{d_{\text{out}}}, \quad b \in \mathbb{R}^{d_{\text{in}}}, \quad \alpha > 0$. The output $\boldsymbol{y}$ of this layer is:

$$\boldsymbol{y}' = (W + \alpha \boldsymbol{a}\boldsymbol{b}^T)x = Wx + \alpha \boldsymbol{a}(\boldsymbol{b}^T x).$$

Suppose the loss function is $\mathcal{L}$, we define the backpropagation vector $\boldsymbol{g}$ to this layer:

$$\boldsymbol{g} \triangleq \frac{\partial \mathcal{L}}{\partial \boldsymbol{y}'} \in \mathbb{R}^{d_{\text{out}}}.$$

Deriving gradients for the two vectors of rank-1 LoRA. First, differentiate with respect to ΔW:

$$\frac{\partial \mathcal{L}}{\partial \Delta W} = \frac{\partial \mathcal{L}}{\partial \boldsymbol{y}'} \frac{\partial \boldsymbol{y}'}{\partial \Delta W} = \boldsymbol{g}\boldsymbol{x}^T$$

Then, using the chain rule for $\Delta W = \alpha \boldsymbol{a}\boldsymbol{b}^T$:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{a}} = \alpha \frac{\partial \mathcal{L}}{\partial \Delta W}\boldsymbol{b} = \alpha(\boldsymbol{g}\boldsymbol{x}^T)\boldsymbol{b} = \alpha \boldsymbol{g}(\boldsymbol{x}^T\boldsymbol{b})$$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{b}} = \alpha \left(\frac{\partial \mathcal{L}}{\partial \Delta W}\right)^T \boldsymbol{a} = \alpha(\boldsymbol{x}\boldsymbol{g}^T)\boldsymbol{a} = \alpha \boldsymbol{x}(\boldsymbol{g}^T\boldsymbol{a})$$

These are the root cause of the geometric asymmetry in gradients. The direction of $\nabla_{\boldsymbol{a}}$ is primarily determined by the output-side backpropagated error $\boldsymbol{g}$, scaled only by a scalar $(\boldsymbol{x}^T\boldsymbol{b})$. In other words, the information of $\boldsymbol{x}$ is compressed to a scalar here. The direction of $\nabla_{\boldsymbol{b}}$ is primarily determined by the input activation $\boldsymbol{x}$, scaled only by a scalar $(\boldsymbol{g}^T\boldsymbol{a})$. Therefore, as long as the RL signal causes g to exhibit strong alignable directions during training, a will be easier to align, whereas x is often highly context-dependent, widely distributed, and directionally scattered, making b difficult to align , especially in low-rank settings.

5 Experiments

Experiment of Figure 2. We trained Qwen2.5-7B-Instruct on Countdown-3to4 [22] dataset using GRPO method. At each training step, we sample 8 model responses for each question and there are

32 questions in a rollout batch. During rollout, top_p for sampling is 0.95. Learning rate is 5e-6. We use AdamW [1, 20] optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. It takes 6 hours to train on a single RTX Pro 6000 GPU. We sampled the last 1000 question out as the test set. For experiment in mid-right figure, we trained Llama3.1-8B-Instruct [9] on GSM8K [6] dataset. At each training step, we sample 8 model responses also for each question. However, gradient accumulation is employed here. There are only 2 questions in a rollout batch thus we perform a gradient update every 16 rollout steps. top_p is also 0.95. A gradient clipping of 1.0 is applied. It takes approximately 20 hours on a single RTX Pro 6000 GPU. Note that both two experiment nearly consumed all VRAM, so at least 96GB VRAM is needed to reproduce the result.

Experiment of Figure 3. We sampled the first 10k questions in SafetyBench-test-en.json [41], and evaluate models on a single NVIDIA H100 GPU using VLLM engine [18]. Hyperparameters are top_p=0.95, temperature=0.9, batch_size=256, max_tokens=10000.

Experiment of Table 1. For IF-Eval [44], the accuracy in the table is the instruction-level accuracy under strict mode. We set temperature=top_p=0.95, and max_tokens=10000. For MMLU [13], We set temperature=top_p=0.95, and max_tokens=10000. For MMLU-Pro [36], we set temperature=0.9, top_p=0.95, max_tokens=5000. For SafetyBench [41], we sampled the first 10k questions and we set temperature=0.9, top_p=0.95. Evaluation takes a single NVIDIA H100 GPU using VLLM engine [18].

Experiment of Figure 5 . We follow the configuration of [26], training Qwen2.5-7B-Instruct [23] on a subset of OpenThought-1.2M [10] using Verl [32] framework. The training takes 4 NVIDIA H100 80GB GPUs.

6 Related Works

Reinforcement Learning for LLM. Before the emergence of reasoning-capable models such as OpenAI's o1, reinforcement learning was primarily employed in RLHF to improve instruction-following and alignment with human preferences. More recently, RL with Verifiable Rewards has been proposed as an effective strategy to enhance reasoning in domains such as mathematics and programming. OpenAI's ol was the first to demonstrate that RL can incentivize large-scale reasoning, inspiring subsequent models such as DeepSeek-R1[11], and Qwen3 [37] . Building on these advances, later approaches such as Dr.GRPO [19], CISPO [4], GFPO [33], GMPO [42], etc. have further broadened the landscape of RL-based reasoning.

Interpreting Reinforcement learning. A recent study [7] identified the phenomenon of **entropy collapse** in reinforcement learning, where rapid early convergence causes the model to become overly confident, prematurely degrading its exploratory capacity. A related study [31] observed in chain-of-thought reasoning that high-entropy tokens often act as branching points defining multiple potential reasoning paths.

7 Conclusion and Limitations

This study observes, through Periodic Rank-1 Substitution, that while RLVR enhances training rewards, it does not necessarily yield corresponding improvements in test-set generalization, suggesting the presence of implicit reward overfitting. Furthermore, we analyze RLVR from the perspective of parameter spectral structure and find that its reasoning gains are primarily concentrated along the Rank-1 update direction. In contrast, the Rank-1 component predominantly serves to preserve general capabilities, such as safety, knowledge retention, and instruction-following. We further substantiate this finding using LoRA, demonstrating that vectors on the output side are more easy to alignment.

The limitations of our work are: due to limited computing resources, we have no opportunity to observe the behavior of models with more parameters. Furthermore, we don't explore whether our findings hold ture for new model architectures. These questions require further study to solve.

References

[1] Kingma DP Ba J Adam et al. A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 1412(6), 2014.

[2] Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting reinforce style optimization for learning from human feedback in llms. *arXiv preprint arXiv:2402.14740*, 2024.

[3] Yuchen Cai, Ding Cao, Xin Xu, Zijun Yao, Yuqing Huang, Zhenyu Tan, Benyi Zhang, Guiquan Liu, and Junfeng Fang. On predictability of reinforcement learning dynamics for large language models. *arXiv preprint arXiv:2510.00553*, 2025.

[4] Aili Chen, Aonian Li, Bangwei Gong, Binyang Jiang, Bo Fei, Bo Yang, Boji Shan, Changqing Yu, Chao Wang, Cheng Zhu, et al. Minimax-m1: Scaling test-time compute efficiently with lightning attention. *arXiv preprint arXiv:2506.13585*, 2025.

[5] Zhipeng Chen, Xiaobo Qin, Youbin Wu, Yue Ling, Qinghao Ye, Wayne Xin Zhao, and Guang Shi. Pass@ k training for adaptively balancing exploration and exploitation of large reasoning models. *arXiv preprint arXiv:2508.10751*, 2025.

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

[7] Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Haozhan Li, Yuchen Fan, Huayu Chen, Weize Chen, Zhiyuan Liu, Hao Peng, Lei Bai, Wanli Ouyang, Yu Cheng, Bowen Zhou, and Ning Ding. The entropy mechanism of reinforcement learning for reasoning language models, 2025. URL `https://arxiv.org/abs/2505.22617`.

[8] Xingyu Dang, Christina Baek, J Zico Kolter, and Aditi Raghunathan. Assessing diversity collapse in reasoning. In *Scaling Self-Improving Foundation Models without Human Supervision*, 2025.

[9] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407, 2024.

[10] Etash Guha, Ryan Marten, Sedrick Keh, Negin Raoof, Georgios Smyrnis, Hritik Bansal, Marianna Nezhurina, Jean Mercat, Trung Vu, Zayne Sprague, et al. Openthoughts: Data recipes for reasoning models. *arXiv preprint arXiv:2506.04178*, 2025.

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

[12] Zhiwei He, Tian Liang, Jiahao Xu, Qiuzhi Liu, Xingyu Chen, Yue Wang, Linfeng Song, Dian Yu, Zhenwen Liang, Wenxuan Wang, et al. Deepmath-103k: A large-scale, challenging, decontaminated, and verifiable mathematical dataset for advancing reasoning. *arXiv preprint arXiv:2504.11456*, 2025.

[13] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

[14] Alexandre Heuillet, Fabien Couthouis, and Natalia Díaz-Rodríguez. Collective explainable ai: Explaining cooperative strategies and agent contribution in multiagent reinforcement learning with shapley values. *IEEE Computational Intelligence Magazine*, 17(1):59–71, 2022.

[15] Jingcheng Hu, Yinmin Zhang, Qi Han, Daxin Jiang, Xiangyu Zhang, and Heung-Yeung Shum. Open-reasoner-zero: An open source approach to scaling up reinforcement learning on the base model. *arXiv preprint arXiv:2503.24290*, 2025.

[16] Jackrong. Llama3.1-8B-Thinking-R1. https://huggingface.co/Jackrong/Llama3.1-8B-Thinking-R1, 2025. Accessed: 2026-05-07.

[17] Prakhar Kaushik, Shravan Chaudhari, Ankit Vaidya, Rama Chellappa, and Alan Yuille. The universal weight subspace hypothesis. *arXiv preprint arXiv:2512.05117*, 2025.

[18] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*, pages 611–626, 2023.

[19] Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*, 2025.

[20] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

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

[22] Jiayi Pan, Junjie Zhang, Xingyao Wang, Lifan Yuan, Hao Peng, and Alane Suhr. Tinyzero. https://github.com/Jiayi-Pan/TinyZero, 2025. Accessed: 2025-01-24.

[23] Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

[24] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

[25] Yi Ren and Danica J Sutherland. Learning dynamics of llm finetuning. In *The Thirteenth International Conference on Learning Representations*, 2025.

[26] John Schulman and Thinking Machines Lab. Lora without regret. *Thinking Machines Lab: Connectionism*, 2025. doi: 10.64434/tml.20250929. https://thinkingmachines.ai/blog/lora/.

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

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

[29] Pedro Sequeira and Melinda Gervasio. Interestingness elements for explainable reinforcement learning: Understanding agents' capabilities and limitations. *Artificial Intelligence*, 288:103367, 2020.

[30] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

[31] Idan Shenfeld, Jyothish Pari, and Pulkit Agrawal. Rl's razor: Why online reinforcement learning forgets less, 2025. URL https://arxiv.org/abs/2509.04259.

[32] Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, pages 1279–1297, 2025.

[33] Vaishnavi Shrivastava, Ahmed Awadallah, Vidhisha Balachandran, Shivam Garg, Harkirat Behl, and Dimitris Papailiopoulos. Sample more to think less: Group filtered policy optimization for concise reasoning. *arXiv preprint arXiv:2508.09726*, 2025.

[34] Kimi Team, Yifan Bai, Yiping Bao, Guanduo Chen, Jiahao Chen, Ningxin Chen, Ruijue Chen, Yanru Chen, Yuankun Chen, Yutian Chen, Zhuofu Chen, Jialei Cui, Hao Ding, Mengnan Dong, Angang Du, Chenzhuang Du, Dikang Du, Yulun Du, Yu Fan, Yichen Feng, Kelin Fu, Bofei Gao, Hongcheng Gao, Peizhong Gao, Tong Gao, Xinran Gu, Longyu Guan, Haiqing Guo, Jianhang Guo, Hao Hu, Xiaoru Hao, Tianhong He, Weiran He, Wenyang He, Chao Hong, Yangyang Hu, Zhenxing Hu, Weixiao Huang, Zhiqi Huang, Zihao Huang, Tao Jiang, Zhejun Jiang, Xinyi Jin, Yongsheng Kang, Guokun Lai, Cheng Li, Fang Li, Haoyang Li, Ming Li, Wentao Li, Yanhao Li, Yiwei Li, Zhaowei Li, Zheming Li, Hongzhan Lin, Xiaohan Lin, Zongyu Lin, Chengyin Liu, Chenyu Liu, Hongzhang Liu, Jingyuan Liu, Junqi Liu, Liang Liu, Shaowei Liu, T. Y. Liu, Tianwei Liu, Weizhou Liu, Yangyang Liu, Yibo Liu, Yiping Liu, Yue Liu, Zhengying Liu, Enzhe Lu, Lijun Lu, Shengling Ma, Xinyu Ma, Yingwei Ma, Shaoguang Mao, Jie Mei, Xin Men, Yibo Miao, Siyuan Pan, Yebo Peng, Ruoyu Qin, Bowen Qu, Zeyu Shang, Lidong Shi, Shengyuan Shi, Feifan Song, Jianlin Su, Zhengyuan Su, Xinjie Sun, Flood Sung, Heyi Tang, Jiawen Tao, Qifeng Teng, Chensi Wang, Dinglu Wang, Feng Wang, Haiming Wang, Jianzhou Wang, Jiaxing Wang, Jinhong Wang, Shengjie Wang, Shuyi Wang, Yao Wang, Yejie Wang, Yiqin Wang, Yuxin Wang, Yuzhi Wang, Zhaoji Wang, Zhengtao Wang, Zhexu Wang, Chu Wei, Qianqian Wei, Wenhao Wu, Xingzhe Wu, Yuxin Wu, Chenjun Xiao, Xiaotong Xie, Weimin Xiong, Boyu Xu, Jing Xu, Jinjing Xu, L. H. Xu, Lin Xu, Suting Xu, Weixin Xu, Xinran Xu, Yangchuan Xu, Ziyao Xu, Junjie Yan, Yuzi Yan, Xiaofei Yang, Ying Yang, Zhen Yang, Zhilin Yang, Zonghan Yang, Haotian Yao, Xingcheng Yao, Wenjie Ye, Zhuorui Ye, Bohong Yin, Longhui Yu, Enming Yuan, Hongbang Yuan, Mengjie Yuan, Haobing Zhan, Dehao Zhang, Hao Zhang, Wanlu Zhang, Xiaobin Zhang, Yangkun Zhang, Yizhi Zhang, Yongting Zhang, Yu Zhang, Yutao Zhang, Yutong Zhang, Zheng Zhang, Haotian Zhao, Yikai Zhao, Huabin Zheng, Shaojie Zheng, Jianren Zhou, Xinyu Zhou, Zaida Zhou, Zhen Zhu, Weiyu Zhuang, and Xinxing Zu. Kimi k2: Open agentic intelligence, 2025. URL `https://arxiv.org/abs/2507.20534`.

[35] Qwen Team et al. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2(3), 2024.

[36] Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37:95266–95290, 2024.

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

[38] Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, et al. Dapo: An open-source llm reinforcement learning system at scale, 2025. *URL https://arxiv. org/abs/2503.14476*, 2025.

[39] Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? *arXiv preprint arXiv:2504.13837*, 2025.

[40] Michael Zhang, Simran Arora, Rahul Chalamala, Alan Wu, Benjamin Spector, Aaryan Singhal, Krithik Ramesh, and Christopher Ré. Lolcats: On low-rank linearizing of large language models. *arXiv preprint arXiv:2410.10254*, 2024.

[41] Zhexin Zhang, Leqi Lei, Lindong Wu, Rui Sun, Yongkang Huang, Chong Long, Xiao Liu, Xuanyu Lei, Jie Tang, and Minlie Huang. Safetybench: Evaluating the safety of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15537–15553, 2024.

[42] Yuzhong Zhao, Yue Liu, Junpeng Liu, Jingye Chen, Xun Wu, Yaru Hao, Tengchao Lv, Shaohan Huang, Lei Cui, Qixiang Ye, et al. Geometric-mean policy optimization. *arXiv preprint arXiv:2507.20673*, 2025.

[43] Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, et al. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*, 2025.

[44] Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models, 2023. URL `https://arxiv.org/abs/2311.07911`.

A Why RLVR Algorithms Are the Same in Essence

Let

$$\hat{A}_i = \frac{R_i - \mu_R}{\sigma_R}, \qquad \mu_R = \frac{1}{G}\sum_{j=1}^{G} R_j.$$

GRPO objective:

$$J_{\mathrm{GRPO}}(\theta) = \mathbb{E}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|y_i|}\sum_{t=1}^{|y_i|}\min\Big(r_{i,t}(\theta)\hat{A}_i,\ \mathrm{clip}(r_{i,t}(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_i\Big) - \beta D_{\mathrm{KL}}(\pi_\theta\|\pi_{\mathrm{ref}})\right],$$

$$r_{i,t}(\theta) = \frac{\pi_\theta(y_{i,t} \mid x, y_{i,<t})}{\pi_{\mathrm{old}}(y_{i,t} \mid x, y_{i,<t})}.$$

DAPO:

$$J_{\mathrm{DAPO}}(\theta) = \mathbb{E}\left[\frac{1}{\sum_j |y_j|}\sum_{i=1}^{G}\sum_{t=1}^{|y_i|}\min\Big(r_{i,t}(\theta)\hat{A}_i,\ \mathrm{clip}(r_{i,t}(\theta), 1-\epsilon_{\mathrm{low}}, 1+\epsilon_{\mathrm{high}})\hat{A}_i\Big)\right],$$

Here, dynamic sampling per se is not a loss, it's a sampling strategy.

$$0 < \Big|\{\, y_i \mid \text{is_equivalent}(a, y_i)\,\}\Big| < G.$$

overlong shaping:

$$R_i = R_{\mathrm{task}}(y_i, a) + R_{\mathrm{len}}(y_i),$$

$$R_{\mathrm{len}}(y) = \begin{cases} 0, & |y| \le L_{\max} - L_{\mathrm{cache}} \\ \dfrac{L_{\max} - L_{\mathrm{cache}} - |y|}{L_{\mathrm{cache}}}, & L_{\max} - L_{\mathrm{cache}} < |y| \le L_{\max} \\ -1, & |y| > L_{\max}. \end{cases}$$

GSPO objective:

$$s_i(\theta) = \left(\frac{\pi_\theta(y_i \mid x)}{\pi_{\mathrm{old}}(y_i \mid x)}\right)^{1/|y_i|} = \exp\left(\frac{1}{|y_i|}\sum_{t=1}^{|y_i|}\log\frac{\pi_\theta(y_{i,t} \mid x, y_{i,<t})}{\pi_{\mathrm{old}}(y_{i,t} \mid x, y_{i,<t})}\right),$$

$$J_{\mathrm{GSPO}}(\theta) = \mathbb{E}\left[\frac{1}{G}\sum_{i=1}^{G}\min\Big(s_i(\theta)\hat{A}_i,\ \mathrm{clip}(s_i(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_i\Big)\right].$$

Regardless of the clip,and treat π_{old} and $\hat{A}_i$ as stop-grad constants, then

$$\nabla_\theta r_{i,t}(\theta) = r_{i,t}(\theta)\nabla_\theta \log \pi_\theta(y_{i,t} \mid x, y_{i,<t}),$$

$$\nabla_\theta s_i(\theta) = s_i(\theta)\frac{1}{|y_i|}\sum_{t=1}^{|y_i|}\nabla_\theta \log \pi_\theta(y_{i,t} \mid x, y_{i,<t}).$$

all gradients update can be formulated as:

$$\nabla_\theta J \approx \mathbb{E}\left[\sum_{i,t}\omega_{i,t}\,\nabla_\theta \log \pi_\theta(y_{i,t} \mid x, y_{i,<t})\right],$$

In other words, this approach performs signed weighted Supervised Fine-Tuning (SFT) on the sampled tokens or responses. The primary distinction lies in the definition of the weight ω:

- GRPO defines the weight as the token-level probability ratio multiplied by the advantage function.
- DAPO also operates at the token level but introduces an asymmetric clipping mechanism. Furthermore, it shifts the normalization for long samples from sample-average to token-mean, while reformulating R_i through dynamic sampling and length-based rewards.
- GSPO, in contrast, elevates the probability ratio to the sequence level. Consequently, all tokens within a single response share the same weight s_i, which explains why GSPO more closely resembles whole-sequence weighted SFT.

B More Examples of Singular Spectrum

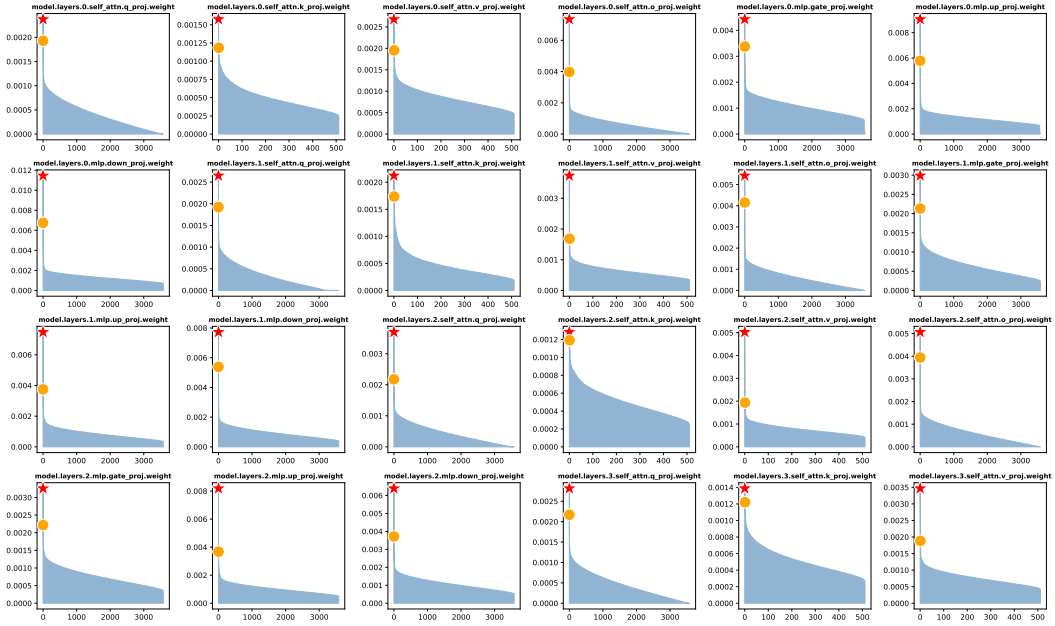

Figure 6: step10

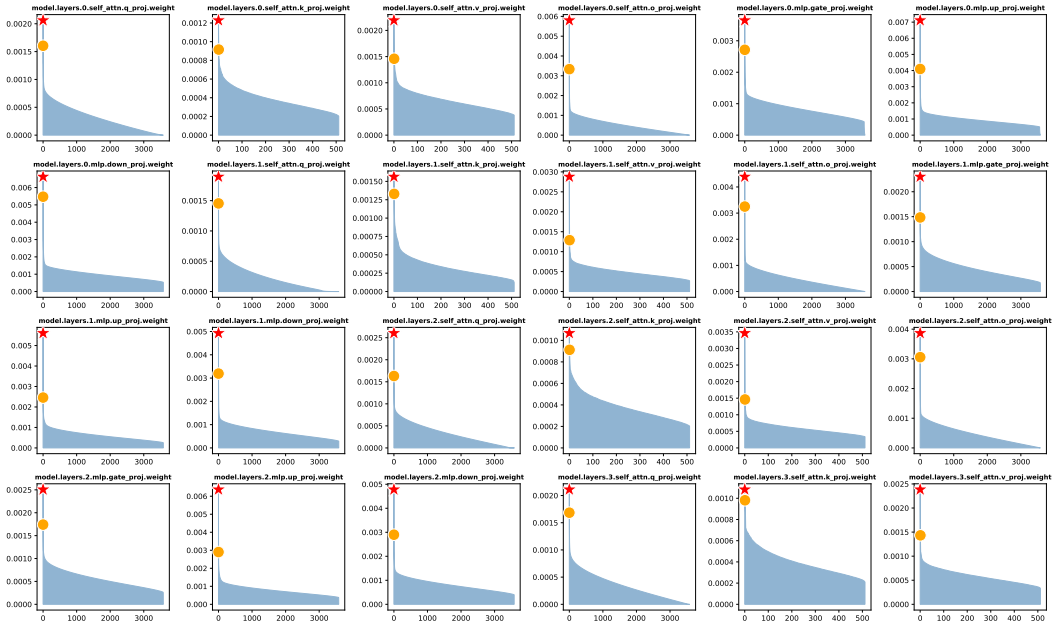

Figure 7: step20

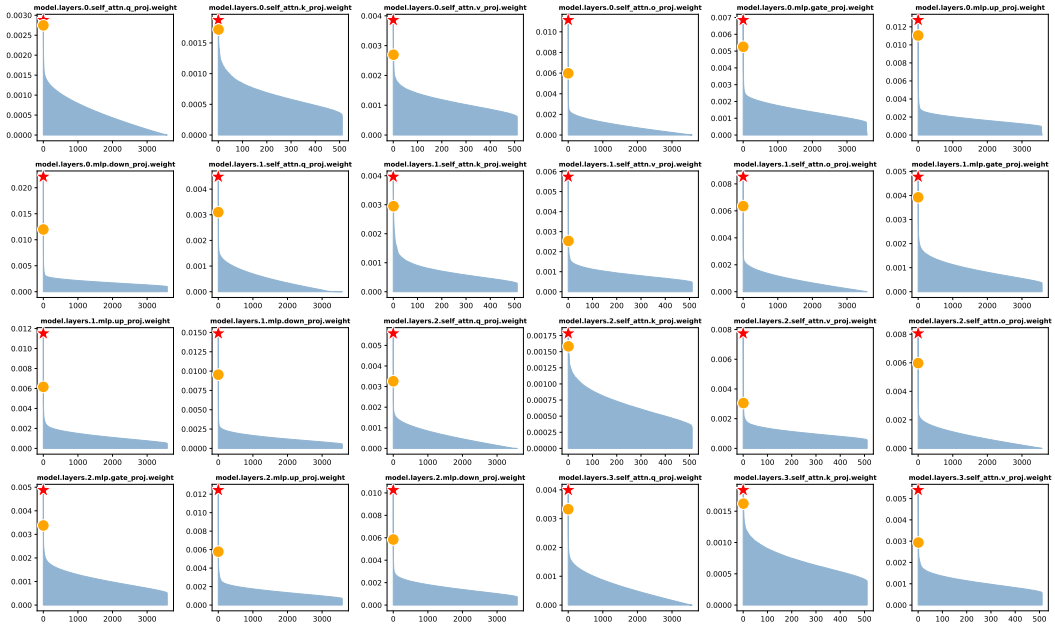

Figure 8: step30

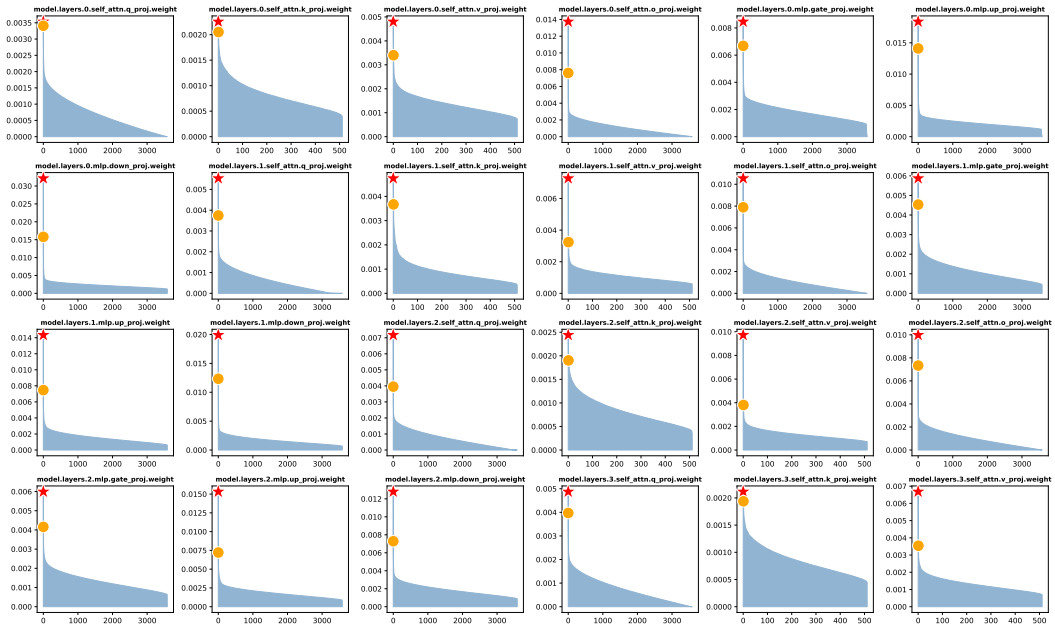

Figure 9: step40

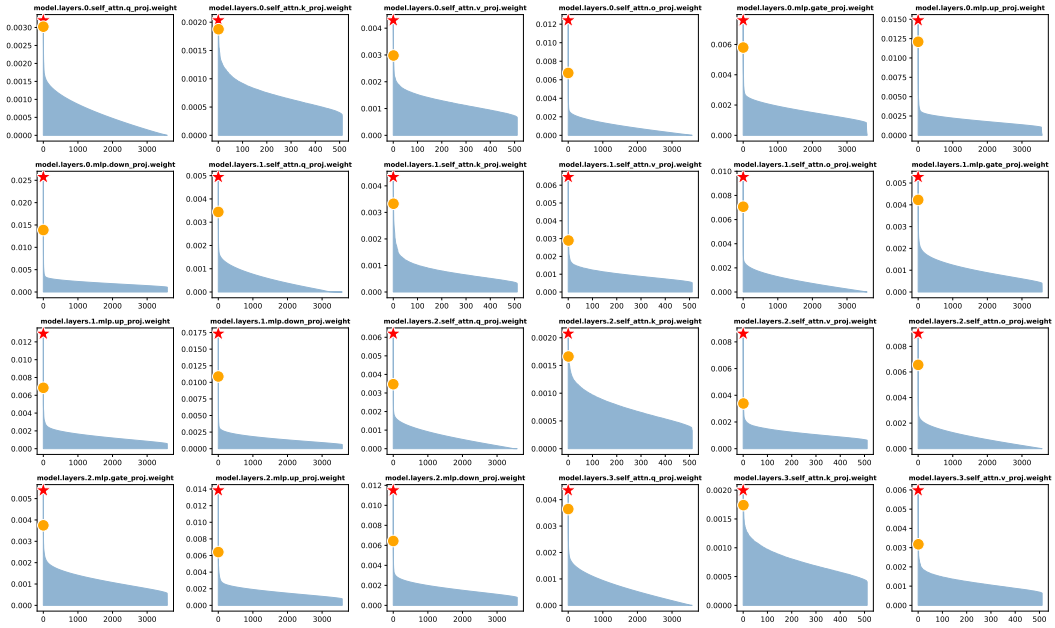

Figure 10: step50

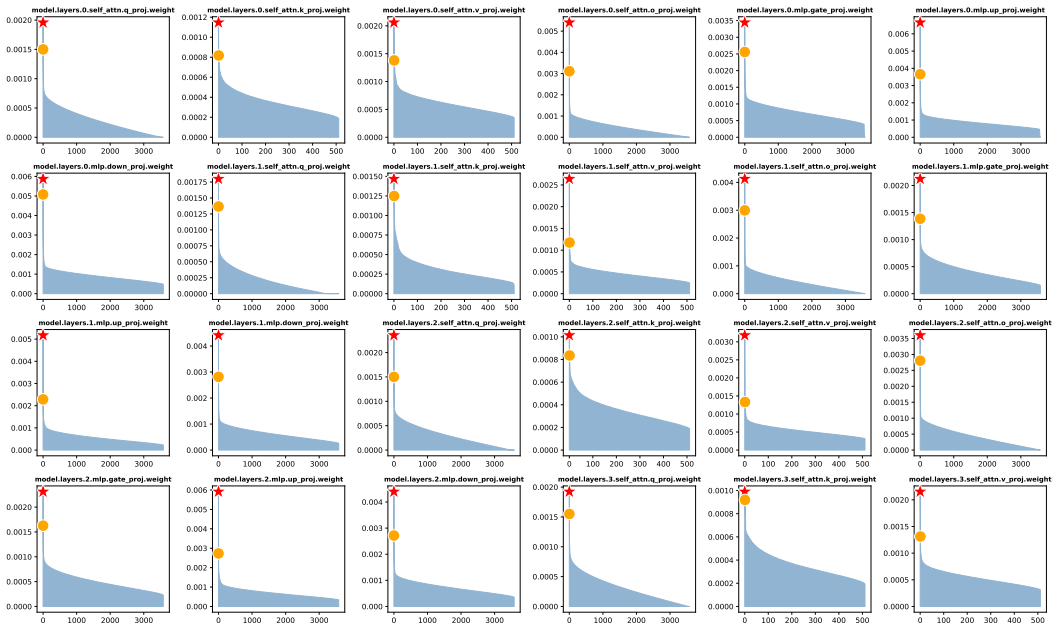

Figure 11: step60

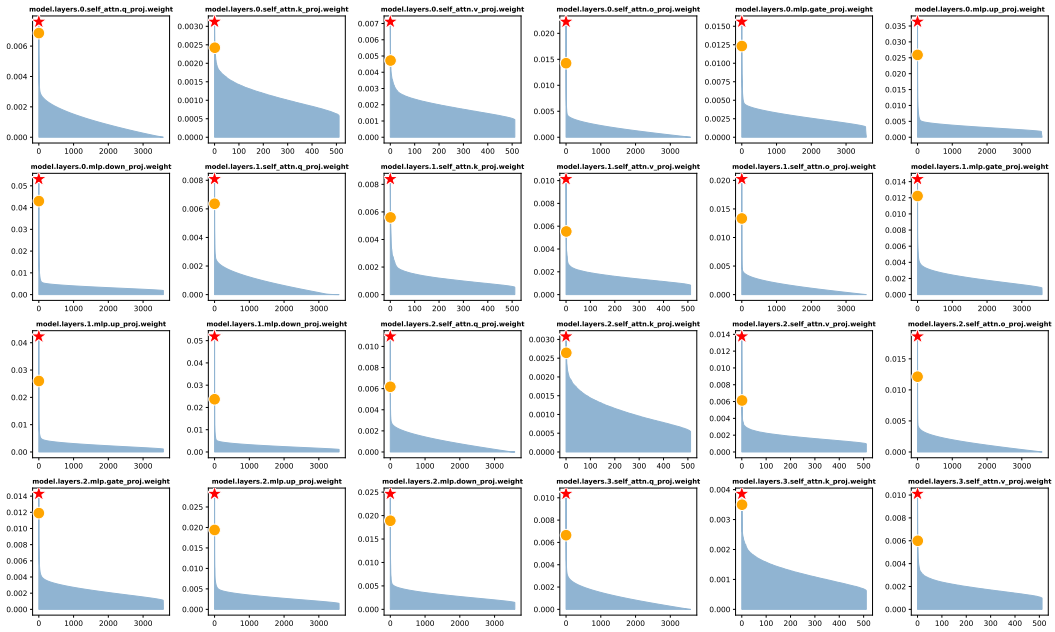

Figure 12: step70

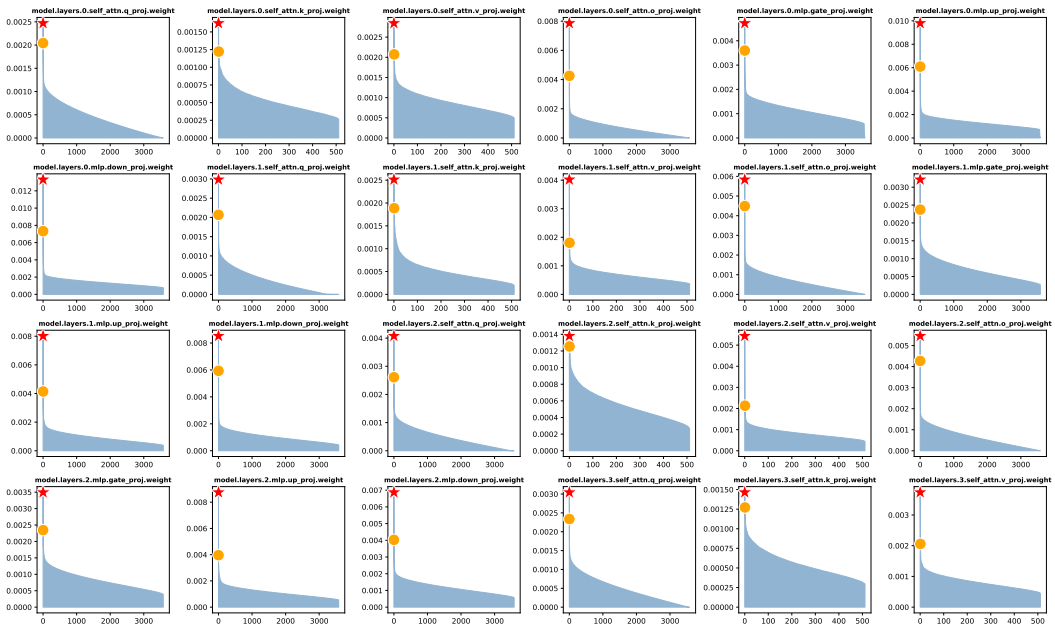

Figure 13: step80

C More Results for Singular Spectra

As shown in Table 2, deep layers(near the output side) tend to have clearer pattern of "leading spike followed by a heavy tail".

Layer	σ_1	σ_2	σ_3	σ_4	σ_5	σ_6	σ_7
`layers.0.mlp.gate_proj`	100.00%	42.40%	41.85%	41.58%	41.43%	40.63%	39.84%
`layers.0.mlp.up_proj`	100.00%	63.97%	63.85%	63.75%	63.66%	63.61%	63.58%
`layers.1.mlp.up_proj`	100.00%	44.48%	40.70%	39.56%	38.76%	36.81%	36.62%
`layers.2.mlp.up_proj`	100.00%	45.38%	44.93%	44.85%	44.77%	44.68%	44.62%
`layers.6.mlp.up_proj`	100.00%	64.69%	64.64%	64.58%	64.53%	64.45%	64.34%
`layers.19.mlp.up_proj`	100.00%	67.90%	67.24%	65.72%	65.41%	65.37%	65.32%
`layers.21.mlp.up_proj`	100.00%	69.08%	67.27%	67.25%	67.16%	67.09%	67.05%
`layers.22.mlp.down_proj`	100.00%	64.60%	57.21%	56.70%	56.48%	55.97%	55.74%
`layers.22.mlp.up_proj`	100.00%	68.26%	67.39%	67.13%	67.04%	67.02%	66.91%
`layers.22.self_attn.o_proj`	100.00%	68.23%	66.25%	66.02%	65.63%	64.81%	64.46%
`layers.23.mlp.down_proj`	100.00%	63.83%	58.42%	58.27%	58.20%	58.15%	58.08%
`layers.23.self_attn.o_proj`	100.00%	49.44%	48.39%	47.25%	47.18%	46.55%	46.32%
`layers.24.mlp.down_proj`	100.00%	61.17%	59.56%	59.51%	59.45%	59.38%	59.34%
`layers.24.self_attn.o_proj`	100.00%	42.64%	39.82%	39.71%	39.71%	39.62%	39.53%
`layers.25.mlp.down_proj`	100.00%	68.28%	66.29%	66.07%	65.76%	65.54%	65.50%
`layers.25.self_attn.o_proj`	100.00%	54.14%	48.65%	46.65%	46.48%	46.43%	46.34%
`layers.26.self_attn.o_proj`	100.00%	43.54%	41.27%	40.71%	40.64%	40.53%	40.43%
`layers.27.mlp.down_proj`	100.00%	48.49%	44.82%	43.14%	39.28%	38.82%	37.78%
`layers.27.mlp.gate_proj`	100.00%	65.37%	61.14%	60.30%	57.03%	56.24%	55.29%
`layers.27.mlp.up_proj`	100.00%	62.04%	61.90%	61.67%	61.09%	60.56%	60.43%

Table 2: Linear layers of Qwen2.5-7B-R1 with $\sigma_2/\sigma_1 \leq 0.70$; entries are percentages normalized by each layer's largest singular value.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: We elaborate our contributions at the end of the Introduction.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: We discussed the limitations in the Conclusion.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

Justification: Our mathematical reasoning don't need strong assumption, the input vector can be any vector during training.

Guidelines:

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

4. **Experimental result reproducibility**

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

Answer: [Yes]

Justification: We disclose all source codes needed to reproduce the experiment result.

Guidelines:

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

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

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

Answer: [Yes]

Justification: Our anonymous repository includes a README.md to offer guidance on reproducing the main results.

Guidelines:

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

6. **Experimental setting/details**

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

Answer: [Yes]

Justification: We detailed our experimental setup in the main text.

Guidelines:

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

7. **Experiment statistical significance**

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

Answer: [Yes]

Justification: In Table 1, we give our results with statistical errors.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: As detailed in Experiment part, our experiment requires a GPU with more than 96GB VRAM.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: Our source code is in an anonymous repository.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [No]

    Justification: Our work is focused on reinforcement learning, which don't entail societal impacts.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [N/A]

    Justification: This paper poses no such risks.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: We correctly cited all models and datasets used.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [N/A]

    Justification: We don't release new assets.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: This paper doesn't involve crowdsourcing nor research with human subjects.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: This paper doesn't involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

Answer: [N/A]

Justification: Our ideas are totally original without the intervention of LLM.

Guidelines:

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