REWARD-GRADIENT ALIGNED LOW-RANK ADAPTATION FOR POST-TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement Learning with Verifiable Rewards (RLVR) has become a prominent post-training paradigm for enhancing the reasoning capabilities of large language models. Low-Rank Adaptation (LoRA) offers a natural way to reduce its training cost. Intuitively, training both low-rank factors should allow LoRA to adapt along the directions that matter for the reward. However, we observe that the input-side factor of LoRA barely changes during RLVR. As a result, the input subspace remains close to its random initialization throughout training. We prove that under standard initialization this subspace rotates only at second order in the cumulative gradient, one order slower than the output-side factor. The random initialization therefore persistently constrains subsequent optimization. Motivated by this finding, we propose **RE**ward-Gradient **A**ligned **L**ow-Rank Adaptation (REAL), which aligns the input subspace with task-specific reward gradients at initialization. REAL contrasts the gradients of correct and incorrect rollouts for each prompt and selects the subspace that retains the most energy of these contrasts across prompts. Since exact selection requires a full weight gradient for every prompt, REAL estimates the subspace by two-sided random compression with memory linear in the layer width. Extensive experiments on the Qwen3 series show that REAL outperforms seven PEFT baselines and exceeds LoRA by up to 5.90% in average accuracy. With a length-aware selection reward, REAL also shortens the responses of distilled reasoning models by up to 44.9% while improving their accuracy.

1 INTRODUCTION

Large reasoning models (LRMs) have made substantial advances in competition-level mathematics, code generation, and agentic reasoning (OpenAI, 2024; DeepSeek-AI et al., 2025; 2026). A central driver of this progress is reinforcement learning with verifiable rewards (RLVR), which optimizes the model policy using rewards derived from answer correctness, code execution, or formal verification (DeepSeek-AI et al., 2025; Yu et al., 2025). As test-time scaling increases reasoning length, RLVR incurs greater computational costs in both rollout generation and backpropagation (Snell et al., 2025; Tang et al., 2026). Full-parameter optimization additionally maintains gradients and optimizer states, further increasing memory consumption, which motivates the use of parameter-efficient fine-tuning (PEFT) for scalable LRMs post-training (Lialin et al., 2023).

PEFT reduces training overhead by freezing pretrained weights and optimizing only a small set of task-specific parameters (Lialin et al., 2023). Existing approaches achieve efficient adaptation through inserted adapter modules, optimizing continuous prefixes, or low-rank parameterizations of weight updates (Houlsby et al., 2019; Li & Liang, 2021; Hu et al., 2022). Among them, Low-Rank Adaptation (LoRA) parameterizes the update to a linear layer as $\Delta W = BA$, where $B \in \mathbb{R}^{d_{\text{out}} \times r}$, $A \in \mathbb{R}^{r \times d_{\text{in}}}$, and $r \ll \min(d_{\text{in}}, d_{\text{out}})$, reducing the number of trainable parameters per layer from $d_{\text{out}} d_{\text{in}}$ to $r(d_{\text{in}} + d_{\text{out}})$ (Hu et al., 2022). Recent studies further show that effective weight updates induced by RLVR often exhibit pronounced low-rank structure, providing empirical support for applying LoRA to reasoning-oriented reinforcement learning (Ye et al., 2026). However, under comparable training budgets, LoRA still consistently underperforms full-parameter fine-tuning on complex reasoning tasks (Yin et al., 2025; Zhang et al., 2026a). This gap suggests that, *despite the low-rank structure of RLVR updates, existing low-rank adaptation schemes do not yet fully exploit the effective updates that drive improvements in reasoning capability.*

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.

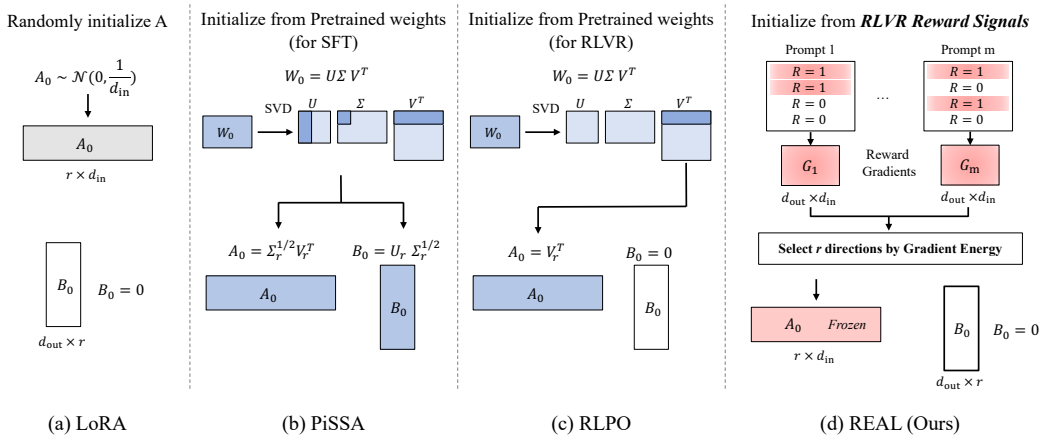

Figure 1: Initialization of the two low-rank factors in (a) LoRA, (b) PiSSA, (c) RLPO, and (d) REAL. LoRA draws A_0 at random, while PiSSA and RLPO derive it from the top-r right singular vectors of the pretrained weight W_0. REAL computes a reward-contrast gradient G_p for each prompt from the verified rollouts of the initial policy and builds A_0 from the r input directions that retain the most energy of these gradients.

To understand this unrealized potential, we further analyze the optimization dynamics of LoRA under RLVR. Empirically, we find that the input subspace induced by the input-side factor A remains nearly unchanged throughout training, and freezing A has little effect on final performance. Our theoretical analysis attributes this behavior to the asymmetric dynamics of the two LoRA factors: under standard LoRA initialization with $B_0 = 0$, the output-side factor B changes at first order in the cumulative gradient, whereas the input subspace deviates from its initialization only at second order. Consequently, the input directions available to LoRA are largely determined by the randomly initialized A, making input-subspace selection critical to RLVR optimization. Gradient-aligned initialization offers a natural way to select this subspace, and LoRA-GA (Wang et al., 2024) has shown that aligning the initial low-rank update with the full-parameter gradient of a sampled batch improves supervised fine-tuning. RLVR, however, derives its learning signal from relative rewards within each prompt group, so each group contributes a gradient that contrasts its correct and incorrect responses. These contrasts are prompt-specific, and different groups can require different output-side corrections on the same input directions. Aggregating their gradients before subspace selection can therefore cancel group-specific components and obscure their individual contributions. We instead seek a shared input subspace that preserves reward-relevant directions across groups.

Building on the above analysis, we propose Reward-Gradient Aligned Low-Rank Adaptation (REAL), a reward-gradient aligned adaptation method tailored to RLVR. Before formal training, REAL uses the initial policy to sample and verify a small number of rollouts, from which it constructs reward-contrast gradients that capture the relative reward structure within each prompt group. Rather than directly aggregating gradients across groups, REAL evaluates how much input-side gradient energy a candidate subspace retains for each group and selects the shared input subspace that maximizes the average retained energy across groups. This subspace is then used to initialize and fix the input-side factor, while formal RLVR training optimizes only the output-side factor, thereby constraining subsequent updates to reward-relevant input directions. Figure 1 contrasts this initialization with those of LoRA, PiSSA, and RLPO, which draw the input subspace at random or from the singular vectors of the pretrained weights. To avoid explicitly constructing full weight gradients and high-dimensional input-side second-moment matrices, we further propose two-sided random compression for estimating the target subspace from streaming low-dimensional representations. In this way, REAL directly incorporates the group-level reward structure of RLVR into LoRA subspace design, enabling limited low-rank capacity to preserve optimization signals that are more relevant to reward improvement. Experiments on Qwen3-1.7B, 4B, and 8B Base models show that REAL outperforms seven PEFT baselines in average accuracy across six reasoning benchmarks, exceeding LoRA by 3.69%, 5.90%, and 3.19%. On DeepSeek-R1-Distill-Qwen-1.5B and 7B, REAL improves the average accuracy of the original models across 12 benchmarks by 1.89% and 0.52% while reducing mean completion length by 44.9% and 37.5%, respectively.

2 PRELIMINARIES

2.1 REINFORCEMENT LEARNING WITH VERIFIABLE REWARDS

RLVR optimizes policy model π_θ using rewards computed by task-specific verifiers. For a prompt $q \sim \mathcal{D}$ and response o, the binary reward $R(q, o) \in \{0, 1\}$ indicates whether the final answer is correct. GRPO computes advantages by standardizing rewards within each prompt group (Shao et al., 2024). For each prompt q, it independently samples K responses $\{o_1, \ldots, o_K\}$ from the old policy $\pi_{\theta_{\text{old}}}$ and standardizes their rewards $R_i = R(q, o_i)$ to obtain

$$\widehat{A}_i = \begin{cases} (R_i - \bar{R})/\sigma, & \sigma > 0, \\ 0, & \sigma = 0, \end{cases} \qquad \bar{R} = \frac{1}{K}\sum_{i=1}^{K} R_i, \qquad \sigma^2 = \frac{1}{K}\sum_{i=1}^{K}(R_i - \bar{R})^2. \tag{1}$$

Each token in response o_i shares the advantage $\widehat{A}_i$. $\rho_{i,t}(\theta) = \frac{\pi_\theta(o_{i,t}|q,o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t}|q,o_{i,<t})}$ is the probability ratio between current and old policies. The optimization objective is

$$\mathbb{E}\left[\frac{1}{K}\sum_{i=1}^{K}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\min\Big\{\rho_{i,t}(\theta)\widehat{A}_i, \text{clip}(\rho_{i,t}(\theta), 1-\epsilon, 1+\epsilon)\,\widehat{A}_i\Big\}\right], \tag{2}$$

where $\epsilon > 0$ controls clipping.

2.2 LOW-RANK ADAPTATION

LoRA freezes the pretrained weight matrix and parameterizes its update with two trainable low-rank factors (Hu et al., 2022). For adapted layer ℓ, the effective weight is

$$W^\ell = W_0^\ell + \Delta W^\ell, \qquad \Delta W^\ell = \alpha_\ell B^\ell A^\ell, \qquad B^\ell \in \mathbb{R}^{d_{\text{out}}^\ell \times r}, \quad A^\ell \in \mathbb{R}^{r \times d_{\text{in}}^\ell}, \tag{3}$$

where $W_0^\ell \in \mathbb{R}^{d_{\text{out}}^\ell \times d_{\text{in}}^\ell}$ is frozen, $r \ll \min(d_{\text{in}}^\ell, d_{\text{out}}^\ell)$ is the adapter rank, and $\alpha_\ell > 0$ is the adapter scaling factor. Standard LoRA initializes A_0^ℓ randomly and sets $B_0^\ell = 0$, preserving the pretrained policy at the start of training. For an input activation $x \in \mathbb{R}^{d_{\text{in}}^\ell}$, the adapter computes $\alpha_\ell B^\ell(A^\ell x)$. The input-side factor A^ℓ maps x into r coordinates. The weight update satisfies $\text{row}(\Delta W^\ell) \subseteq \text{row}(A^\ell)$. Thus, the row space of A^ℓ determines the input directions available to the low-rank update. We refer to this row space as the ***LoRA input subspace***.

2.3 OBSERVATION: LORA'S INPUT SUBSPACE REMAINS NEARLY FIXED IN RLVR

Recent work shows that much of the reasoning gain from RLVR is preserved by a low-rank approximation of the full-parameter weight update (Cai et al., 2026), indicating that LoRA is, in principle, expressive enough to capture the updates responsible for these gains. Nevertheless, LoRA still underperforms full fine-tuning on RLVR reasoning benchmarks (Yin et al., 2025). This gap suggests that the bottleneck lies less in LoRA's representational capacity than in how its low-rank update is learned. We study this learning process through the LoRA input subspace. Since this subspace determines the input directions on which ΔW^ℓ can act, exploiting new input directions requires *rotating* $\text{row}(A^\ell)$. Accordingly, we track both how much each LoRA factor moves and how far its associated subspace rotates during training.

Setup. We train Qwen3-1.7B-Base on Big-Math with rank-16 LoRA under the standard initialization, jointly optimizing both factors. For each of the 196 adapted matrices, we track two quantities: (i) the relative Frobenius displacement of each factor, which measures how much it changes; and (ii) the largest principal angle between its current and reference subspaces, which measures how far $\text{row}(A)$ or $\text{col}(B)$ rotates. We measure A relative to its initialization, i.e. $\|A_t - A_0\|_F / \|A_0\|_F$ and the angle between $\text{row}(A_t)$ and $\text{row}(A_0)$. Since $B_0 = 0$, we measure B relative to step 100 in the same way. We report medians across all adapted matrices.

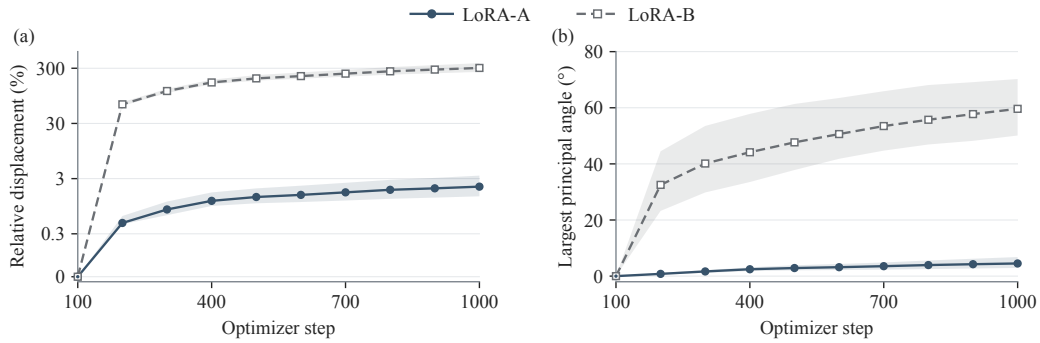

Figure 2: Asymmetric dynamics of the two LoRA factors under RLVR on Qwen3-1.7B-Base. We track (a) the relative Frobenius displacement of each factor and (b) the largest principal angle of its subspace, reporting medians over the 196 adapted matrices and measuring A against its initialization and B against step 100. By step 1000 the output-side factor has moved 303.97% and its subspace has rotated $59.62°$, whereas A has moved only 2.15% and its subspace has rotated $4.51°$, leaving the LoRA input subspace close to its random initialization throughout training.

Results. As shown in Figure 2, the two factors exhibit a pronounced asymmetry. By step 1000, the median relative displacement is 2.15% for A, compared with 303.97% for B. The subspaces show the same pattern: the median largest principal angle is only $4.51°$ for $\mathrm{row}(A)$, versus $59.62°$ for $\mathrm{col}(B)$. Thus, while the output-side factor continues to adapt in both scale and direction, LoRA's input subspace remains close to its random initialization throughout training, confining the weight update to nearly the same r-dimensional input subspace. As this subspace originates from a task-agnostic random initialization and rotates little thereafter, the initialization itself largely determines which input directions LoRA can exploit. This motivates using the task's reward signal to select the r-dimensional input subspace at initialization.

3 Methodology

REAL selects the LoRA input subspace from task rewards before RLVR, freezes A, and trains only B. Section 3.1 shows that the input subspace rotates only at second order during training, so its initialization determines the input directions available to RLVR. Section 3.2 derives the subspace that retains the most energy of per-group reward-contrast gradients, Section 3.3 estimates it with two-sided random compression, and Section 3.4 summarizes the algorithm.

3.1 Near-Invariance of the LoRA Input Subspace

We fix an adapted layer and omit the index ℓ. Let $\mathcal{J}(\theta)$ denote the GRPO objective in Equation 2 and $G = \nabla_W \mathcal{J}$ its gradient with respect to the effective weight $W = W_0 + \alpha BA$. The two factors receive the gradients

$$\nabla_A \mathcal{J} = \alpha B^\top G, \qquad \nabla_B \mathcal{J} = \alpha G A^\top, \tag{4}$$

so the gradient of each factor is scaled by the other. Under the standard initialization $B_0 = 0$, A receives no gradient until B has grown. We quantify this coupling for the gradient ascent iterates $A_{t+1} = A_t + \eta_t \alpha B_t^\top G_t$ and $B_{t+1} = B_t + \eta_t \alpha G_t A_t^\top$ with arbitrary gradients G_t, using the cumulative gradient scale $u_T = \sum_{t<T} \eta_t \alpha \|G_t\|_2$.

Theorem 1 (Near-invariance of the input subspace)**.** *Let $B_0 = 0$, let the singular values of A_0 lie in $[\underline{\nu}, \overline{\nu}]$, and let $\Delta_T = \overline{\nu}(\cosh u_T - 1) < \underline{\nu}$. Then $\|B_T\|_2 \le \overline{\nu} \sinh u_T$, and the largest principal angle ϕ_T between $\mathrm{row}(A_T)$ and $\mathrm{row}(A_0)$ satisfies*

$$\sin \phi_T \le \frac{\Delta_T}{\underline{\nu} - \Delta_T} = \frac{\overline{\nu}}{2\underline{\nu}} u_T^2 + O(u_T^4). \tag{5}$$

Theorem 1 (proved in Appendix C) bounds $\|B_T\|_2$ at first order in u_T but the rotation of the input subspace at second order. Small learning rates and clipped policy updates in RLVR limit the growth of u_T, consistent with the $4.51°$ median rotation in Section 2.3.

The input subspace also fixes the updates that LoRA can express. For the compact SVD $A = U\Sigma P^\top$, we have $BA = (BU\Sigma)P^\top$, so a fixed A expresses exactly the updates $ZP^\top$ with $Z \in \mathbb{R}^{d_{\text{out}} \times r}$, which depend on A only through $\text{row}(A) = \text{range}(P)$. Initializing A therefore amounts to choosing the r-dimensional input subspace that RLVR uses throughout training.

3.2 Reward-Aligned Input Subspace

Since the input subspace barely rotates, REAL freezes A after initialization and trains only B. Prior work freezes a randomly initialized A (Zhang et al., 2023a; Zhu et al., 2024), whereas REAL selects A from task rewards. We write $A = \nu P^\top$ with $P^\top P = I_r$ and $\nu > 0$, so that P spans the input subspace.

Let $\mathcal{J}_p$ be the objective of prompt group p and $G_p = \nabla_W \mathcal{J}_p$. With A frozen, $\nabla_B \mathcal{J}_p = \alpha\nu G_p P$, so a gradient step of size η on B increases $\mathcal{J}_p$ by $\eta\alpha^2\nu^2\|G_pP\|_F^2$ to first order. The retained energy $\|G_pP\|_F^2$ thus measures how much the group can improve through the adapter, and REAL chooses P to maximize its average over groups.

REAL computes these gradients at the initial policy π_{θ_0} on a set $\mathcal{I}$ of m prompt groups, each with at least two correct and two incorrect responses. At θ_0, every ratio $\rho_{i,t}$ equals 1 and clipping is inactive, so a group with n_p^+ correct and n_p^- incorrect responses has the gradient

$$G_p = \frac{1}{K}\sum_{i=1}^{K} \widehat{A}_i S_i = \frac{\sqrt{n_p^+ n_p^-}}{K}\left(\overline{S}_p^+ - \overline{S}_p^-\right), \tag{6}$$

where $S_i = |o_i|^{-1}\nabla_W \log \pi_{\theta_0}(o_i \mid q)$ is the length-normalized score of response o_i, and $\overline{S}_p^+$ and $\overline{S}_p^-$ average S_i over correct and incorrect responses. We call G_p the *reward-contrast gradient*. Averaging the retained energy over $\mathcal{I}$ gives

$$\gamma(P) = \frac{1}{m}\sum_{p\in\mathcal{I}} \|G_pP\|_F^2 = \text{tr}\big(P^\top C P\big), \qquad C = \frac{1}{m}\sum_{p\in\mathcal{I}} G_p^\top G_p. \tag{7}$$

Proposition 1 (Optimal input subspace)**.** *Over $P^\top P = I_r$, $\gamma(P)$ is maximized by $P^\star$, whose columns are the top-r eigenvectors of C, and the maximum equals the sum of the r largest eigenvalues of C.*

Proposition 1 (proved in Appendix D) follows from the Ky Fan maximum principle (Fan, 1949). The optimal input subspace thus consists of the r directions in which reward-contrast gradients concentrate their energy.

This criterion differs from gradient-aligned initialization for supervised fine-tuning (Wang et al., 2024), which aligns the adapter with the aggregated gradient $\overline{G} = m^{-1}\sum_{p\in\mathcal{I}} G_p$:

$$\gamma(P) - \|\overline{G}P\|_F^2 = \frac{1}{2m^2}\sum_{p,p'\in\mathcal{I}} \|(G_p - G_{p'})P\|_F^2 \geq 0. \tag{8}$$

Aggregation cancels the components on which groups disagree; for example, two groups with $G_{p'} = -G_p$ vanish from $\overline{G}$, although each retains its energy in $\gamma(P)$. GRPO computes advantages within each group, and different groups can require different output-side corrections on the same input directions. REAL therefore accumulates the energy of each group separately.

3.3 Efficient Subspace Estimation

Computing $P^\star$ exactly requires every group gradient G_p and the second moment C, whose sizes are quadratic in the layer width. REAL avoids both matrices with two-sided random compression. It draws Gaussian matrices $\Psi_p \in \mathbb{R}^{d_{\text{out}} \times s_{\text{out}}}$, independent across groups and scaled so that

$\mathbb{E}[\Psi_p \Psi_p^\top] = I$, and a standard Gaussian matrix $\Omega \in \mathbb{R}^{d_{\text{in}} \times s_{\text{in}}}$ shared by all groups. It then accumulates

$$H_p = \Psi_p^\top G_p, \qquad Y = \frac{1}{m} \sum_{p \in \mathcal{I}} H_p^\top (H_p \Omega) = \widetilde{C}\Omega, \tag{9}$$

where $\widetilde{C} = m^{-1} \sum_{p \in \mathcal{I}} H_p^\top H_p$ is an unbiased estimate of C.

Both sketches are computed in a streaming manner. Since G_p sums the outer products of the layer's output gradients and inputs over tokens, H_p is accumulated token by token during backpropagation and discarded after updating Y. After all groups are processed, the Nyström approximation $\widetilde{C} \approx FF^\top$ with $F = Y(\Omega^\top Y)^{\dagger/2}$ (Halko et al., 2011; Tropp et al., 2017) gives the estimate $\widehat{P}$ as the top-r left singular vectors of F.

3.4 Overall Pipeline

REAL consists of an observation phase that selects the input subspace and an RLVR phase that trains the adapter within it. In the observation phase, the initial policy samples and verifies K responses per prompt until $\mathcal{I}$ contains m groups, and one backward pass per group compresses its reward-contrast gradient into the sketches of all adapted layers $\ell \in \mathcal{L}$. These sketches occupy memory linear in the layer width and yield the estimate $\widehat{P}^\ell$ for each layer. REAL then initializes $A^\ell = \nu(\widehat{P}^\ell)^\top$ and $B^\ell = 0$. In the RLVR phase, each A^ℓ stays fixed and every policy update acts on the input directions in which reward-contrast gradients concentrate their energy. Algorithm 1 in Appendix B gives the complete procedure.

4 Experiments

4.1 Experimental Setup

Models and baselines. We evaluate REAL in two settings. First, we apply RLVR to Qwen3-1.7B-Base, Qwen3-4B-Base, and Qwen3-8B-Base (Yang et al., 2025) and compare REAL with 7 PEFT methods: LoRA (Hu et al., 2022); VeRA (Kopiczko et al., 2024), which freezes random low-rank matrices shared across layers and trains only scaling vectors; rsLoRA (Kalajdzievski, 2023) and DoRA (Liu et al., 2024), which rescale and reparameterize the update; and PiSSA (Meng et al., 2024) and MiLoRA (Wang et al., 2025), which initialize the factors from the principal and minor singular components of the pretrained weights, respectively. We further include RLPO (Zhang et al., 2026c), which initializes the adapter with geometry-preserving orthonormal factors for RLVR. Second, we apply REAL to DeepSeek-R1-Distill-Qwen-1.5B and DeepSeek-R1-Distill-Qwen-7B (DS-1.5B and DS-7B; DeepSeek-AI et al., 2025), which already produce long reasoning traces, and compare the adapted models with the original models in both accuracy and generation length.

Datasets. Training uses mathematical problems with verifiable reference answers: difficulty-filtered subsets of Big-Math (Albalak et al., 2025) for Qwen3-1.7B and Qwen3-4B, and DAPO-Math-17k (Yu et al., 2025) for Qwen3-8B and the distilled models. We evaluate the Qwen3 models on GSM8K (Cobbe et al., 2021), MATH-500 (Hendrycks et al., 2021; Lightman et al., 2024), AMC 2023, Minerva (Lewkowycz et al., 2022), OlympiadBench (He et al., 2024), and GPQA-Diamond (Rein et al., 2024). For the distilled models, we further include AIME 2024, the 2025 editions of AIME, BRUMO, CMIMC, and HMMT (Balunović et al., 2025), and LSAT (Zhong et al., 2024), yielding 12 benchmarks that span mathematical, scientific, and logical reasoning.

Training. We apply adapters to all attention and MLP projections and train on four NVIDIA H100 GPUs with a constant learning rate of 10^{-6} following GeoRA (Zhang et al., 2026b), a global batch size of 128, and $K = 8$ rollouts per prompt sampled at temperature 1.0. For the Qwen3 models, all methods train for 600 steps with rank $r = 16$ ($r = 256$ for VeRA) and optimize binary correctness rewards using GSPO sequence-level importance ratios (Zheng et al., 2025) without a KL penalty. For the distilled models, we use $r = 64$ and GSPO-token importance ratios, train for 300 steps, and set the KL coefficient to 0.01 to keep the policy close to the original model. In both settings, REAL spends the first 64 training steps on observation without parameter updates, with sketch sizes $s_{\text{out}} = s_{\text{in}} = 4r$ and scale $\nu = 1$.

Table 1: REAL achieves the highest average accuracy at every model size. Accuracy (%) on six reasoning benchmarks across three Qwen3 model sizes, comparing REAL against LoRA, VeRA, rsLoRA, DoRA, PiSSA, MiLoRA and RLPO. Within each model size the best result is shown in **bold** and the second-best is underlined, and the blue figure reports REAL's gain over LoRA.

	Method	GSM8K	MATH-500	AMC 2023	Minerva	OlympiadBench	GPQA	Avg.
Qwen3-1.7B	LoRA	80.99	63.85	33.44	26.01	29.15	27.02	43.41
	VeRA	76.00	57.85	34.22	22.70	23.52	26.58	40.15
	rsLoRA	83.04	65.45	37.34	28.68	30.01	**28.60**	45.52
	DoRA	80.72	64.45	35.47	**28.86**	28.93	26.83	44.21
	PiSSA	83.11	66.80	34.38	26.56	30.23	28.47	44.93
	MiLoRA	81.82	65.65	34.84	26.10	29.08	26.07	43.93
	RLPO	82.66	65.30	38.91	26.01	29.12	28.09	45.01
	REAL	**83.98**	**69.05**	**40.31**	28.77	**33.31**	27.15	**47.10** +3.69
Qwen3-4B	LoRA	84.00	73.05	47.81	30.70	38.95	38.01	52.09
	VeRA	80.76	69.80	47.50	29.78	37.76	36.81	50.40
	rsLoRA	91.85	77.65	51.72	35.57	40.69	39.14	56.10
	DoRA	88.27	73.40	47.50	30.97	38.46	38.32	52.82
	PiSSA	91.49	77.70	52.03	33.27	42.66	37.56	55.79
	MiLoRA	91.07	77.05	49.53	33.46	41.21	38.07	55.06
	RLPO	90.90	76.65	52.97	33.55	41.51	39.27	55.81
	REAL	**92.19**	**78.35**	**53.28**	**39.15**	**44.14**	**40.85**	**57.99** +5.90
Qwen3-8B	LoRA	92.25	79.55	57.81	37.50	43.69	42.23	58.84
	VeRA	89.56	75.35	53.12	34.10	41.32	40.28	55.62
	rsLoRA	93.12	79.75	58.91	40.07	46.18	43.56	60.26
	DoRA	92.63	77.65	59.22	37.78	44.92	41.79	59.00
	PiSSA	92.97	80.20	60.47	**41.54**	45.73	42.49	60.57
	MiLoRA	93.01	79.15	60.00	39.71	45.59	**44.44**	60.32
	RLPO	93.31	79.70	60.00	39.25	45.14	42.11	59.92
	REAL	**93.92**	**81.55**	**62.03**	**41.54**	**48.74**	44.38	**62.03** +3.19

Evaluation. All models decode with temperature 0.6, top-p 0.95, and a maximum length of 32,768 tokens. We report accuracy averaged over 4 to 16 samples per problem and mean completion length in tokens, weighting benchmarks equally (Appendix E).

4.2 Comparison with PEFT Methods

REAL outperforms all PEFT baselines at every scale. As shown in Table 1, REAL achieves the highest average accuracy on all three Qwen3 models, reaching 47.10%, 57.99%, and 62.03% on 1.7B, 4B, and 8B and exceeding LoRA by 3.69%, 5.90%, and 3.19%, respectively. It surpasses the strongest baseline at each scale by 1.46–1.89% and improves over LoRA on every benchmark, including GPQA, although training uses only mathematical problems. VeRA, which trains scaling vectors on frozen random matrices, falls below LoRA, whereas PiSSA and MiLoRA, which initialize the factors from the singular components of the pretrained weights, improve over LoRA but remain below REAL. RLPO, the only baseline designed for RLVR, trails REAL by 2.09%, 2.18%, and 2.11% on 1.7B, 4B, and 8B. Selecting the input subspace from reward-contrast gradients is therefore more effective than using random or weight-derived subspaces, and REAL achieves this with about half of LoRA's trainable parameters.

4.3 Subspace-Guided Efficient Reasoning

The subspace REAL selects during observation shapes the entire training run. Once A is fixed, every subsequent policy update acts on the input directions chosen from the observation rewards. Reasoning models tend to overthink (Chen et al., 2025), and the resulting long traces increase both rollout and inference costs. Since REAL selects the input subspace from rewards observed before

Table 2: Accuracy (%) and mean completion length (tokens) on 12 reasoning benchmarks for two distilled reasoning models. The shaded columns report REAL, where Δ gives its change relative to BASE, measured in percentage points for accuracy and as a relative change for length; these changes are shown in blue when they favour REAL and in orange otherwise. For each model the better of the two methods is shown in **bold**.

	DeepSeek-R1-Distill-Qwen-1.5B						DeepSeek-R1-Distill-Qwen-7B					
	BASE		REAL				BASE		REAL			
Benchmark	Acc.↑	Len.↓	Acc.↑	Δ	Len.↓	Δ%	Acc.↑	Len.↓	Acc.↑	Δ	Len.↓	Δ%
GSM8K	85.56	2562	**85.75**	+0.19	**1545**	-39.7	92.29	1664	**92.93**	+0.64	**1306**	-21.5
MATH-500	86.10	5785	**86.80**	+0.70	**3338**	-42.3	**94.05**	4120	**94.05**	+0.00	**2971**	-27.9
Minerva	32.08	7866	**35.02**	+2.94	**4256**	-45.9	43.29	5722	**45.04**	+1.75	**3748**	-34.5
OlympiadBench	54.19	11660	**55.93**	+1.74	**6233**	-46.5	69.14	8826	**69.77**	+0.63	**5686**	-35.6
AMC 2023	70.16	10124	**75.00**	+4.84	**5098**	-49.6	**90.47**	6453	89.53	-0.94	**4358**	-32.5
AIME 2024	30.62	17055	**31.04**	+0.42	**9571**	-43.9	54.38	13667	**54.58**	+0.20	**8673**	-36.5
AIME 2025	22.92	16081	**23.75**	+0.83	**8824**	-45.1	**38.96**	14395	38.12	-0.84	**8865**	-38.4
BRUMO 2025	31.46	15985	**34.79**	+3.33	**8560**	-46.5	51.04	13169	**56.46**	+5.42	**7689**	-41.6
CMIMC 2025	12.50	19232	**16.25**	+3.75	**10450**	-45.7	26.88	16917	**28.91**	+2.03	**9996**	-40.9
HMMT 2025	**14.38**	18181	13.96	-0.42	**9934**	-45.4	24.17	16772	**25.00**	+0.83	**10173**	-39.4
GPQA	**35.35**	10162	34.53	-0.82	**6143**	-39.6	**49.49**	8933	48.80	-0.69	**5563**	-37.7
LSAT	22.72	14865	**27.88**	+5.16	**8439**	-43.2	**53.04**	10826	50.27	-2.77	**6906**	-36.2
Average	41.50	12463	**43.39**	+1.89	**6866**	-44.9	57.27	10122	**57.79**	+0.52	**6328**	-37.5

RLVR, it can align the subspace with objectives beyond correctness while leaving the RL objective unchanged. We use this property on DS-1.5B and DS-7B: during observation, REAL adds a relative length utility with weight 0.5 to the reward of each correct response, which favors shorter correct responses to the same prompt (Appendix E). Groups with non-constant observation rewards then contribute to C through the first equality in Equation 6. The utility affects only the choice of A, and all policy updates use binary correctness rewards.

As shown in Table 2, REAL raises average accuracy from 41.50% to 43.39% on DS-1.5B and from 57.27% to 57.79% on DS-7B, while reducing mean completion length by 44.9% (from 12,463 to 6,866 tokens) and 37.5% (from 10,122 to 6,328 tokens). Responses become shorter on all 12 benchmarks at both scales. Accuracy improves on 10 benchmarks for DS-1.5B and 7 for DS-7B, and all remaining changes stay within 1% except on LSAT for DS-7B. The gains extend to competition problems: REAL raises AMC 2023 from 70.16% to 75.00% and CMIMC 2025 from 12.50% to 16.25% on DS-1.5B, and BRUMO 2025 from 51.04% to 56.46% on DS-7B, each with 41–50% fewer tokens. These results indicate that the input subspace initialized by REAL during observation aligns to some extent with the direction of concise and accurate reasoning, enabling training to optimize the model toward efficiency without an explicit length penalty.

4.4 Analysis

Gains across difficulty levels. To better understand the performance gains of REAL, we analyze its accuracy gains over six baselines across the five difficulty levels of MATH-500. As shown in Figure 3, the gains concentrate on the hardest problems: the gain over LoRA peaks at Level 5 on all three models, reaching 5.97%, 8.58%, and 3.92% on Qwen3-1.7B, 4B, and 8B, respectively. Although Level 5 contains only 26.8% of MATH-500 problems, it accounts for 30.8%, 43.4%, and 52.5% of the overall gain of REAL over LoRA. REAL ranks first at every level on Qwen3-1.7B and at Levels 3–5 on Qwen3-8B, exceeding PiSSA, the strongest MATH-500 baseline, by 3.92% at Level 5. REAL also matches or exceeds RLPO at 14 of the 15 model–level pairs. These results show that the reward-aligned subspace benefits most on the problems that demand the most reasoning.

Reward-gradient coverage. We further examine how well the input subspace learned by each method captures the reward signal. For each Qwen3-4B model trained for 600 steps, we measure the fraction of the base model's reward-weighted gradient energy on held-out MATH-500 problems

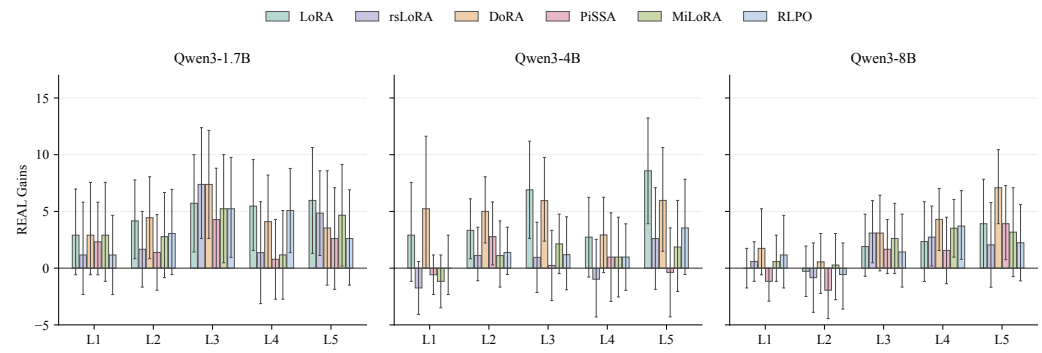

Figure 3: Accuracy gains (%) of REAL over LoRA, rsLoRA, DoRA, PiSSA, MiLoRA, and RLPO on MATH-500, grouped by difficulty level (L1–L5), for Qwen3 series models. Error bars show paired 95% bootstrap intervals.

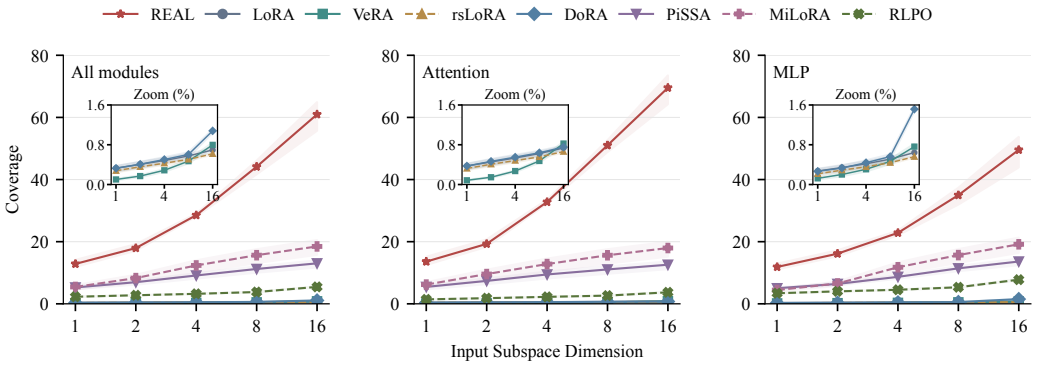

Figure 4: Reward-gradient coverage of the input subspaces learned on Qwen3-4B, evaluated on held-out MATH-500 problems as a function of the subspace dimension k and averaged over all adapted modules, attention modules, and MLP modules. Shaded regions show 95% bootstrap intervals, and insets magnify the baselines with coverage below 1.6%.

retained by the top-k right singular vectors of its weight update (Appendix E). As shown in Figure 4, REAL retains far more reward-gradient energy than every baseline at all dimensions and in both attention and MLP modules. At $k = 16$, REAL captures 61.0% of the energy across all modules, compared with 18.5% for MiLoRA, 13.0% for PiSSA, 5.4% for RLPO, and at most 1.1% for LoRA, VeRA, rsLoRA, and DoRA, exceeding the strongest baseline by 42.5%. Its leading direction alone captures 12.9%, on par with the entire 16-dimensional subspace of PiSSA. The subspace REAL selects during observation therefore stays aligned with the reward on unseen problems, whereas the updates of LoRA, VeRA, rsLoRA, and DoRA remain nearly orthogonal to it.

5 Conclusion

LoRA offers an appealing route to efficient RLVR, as training both factors is expected to discover the directions that matter. Our analysis shows otherwise: the input subspace barely moves from its random initialization, which therefore decides which input directions RLVR can exploit.

This observation points to a simple remedy: let the reward choose the subspace before training. REAL fixes the input subspace that retains the most reward-contrast gradient energy and trains only the output-side factor, outperforming seven PEFT baselines with about half of LoRA's trainable parameters and producing shorter yet more accurate reasoning without an explicit length penalty. By letting the reward decide where to adapt, this work invites a reconsideration of how low-rank adaptation is designed for reinforcement learning.

AI USE STATEMENT

We used large language models (LLMs) only to support routine aspects of this research: helping with programming tasks, correcting grammatical errors, and checking the formatting of the manuscript. These tools were intended solely to improve the quality and presentation of the paper. The central ideas, the experimental design, and the full writing of the paper are our own. We did not directly adopt any LLM-generated content. Everything was carefully reviewed and checked by us, and we take full responsibility for all content in this paper.

REPRODUCIBILITY STATEMENT

To ensure reproducibility, we provide the complete algorithm, full proofs of our theoretical results, and detailed experimental settings in the main text and appendix. All models, datasets, and benchmarks used in this work are publicly available. Upon acceptance, we will release our code, training configurations, and evaluation scripts to facilitate the reproduction of all reported results.

REFERENCES

Pranjal Aggarwal and Sean Welleck. L1: Controlling how long a reasoning model thinks with reinforcement learning. In *Conference on Language Modeling*, 2025.

Alon Albalak, Duy Phung, Nathan Lile, Rafael Rafailov, Kanishk Gandhi, Louis Castricato, Anikait Singh, Chase Blagden, Violet Xiang, Dakota Mahan, and Nick Haber. Big-Math: A large-scale, high-quality math dataset for reinforcement learning in language models, 2025.

Daman Arora and Andrea Zanette. Training language models to reason efficiently. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Mislav Balunović, Jasper Dekoninck, Ivo Petrov, Nikola Jovanović, and Martin Vechev. MathArena: Evaluating LLMs on uncontaminated math competitions. In *Advances in Neural Information Processing Systems*, volume 38. Curran Associates, Inc., 2025. doi: 10.52202/085713-0679.

Dan Biderman, Jacob Portes, Jose Javier Gonzalez Ortiz, Mansheej Paul, Philip Greengard, Connor Jennings, Daniel King, Sam Havens, Vitaliy Chiley, Jonathan Frankle, Cody Blakeney, and John Patrick Cunningham. LoRA learns less and forgets less. *Transactions on Machine Learning Research*, 2024.

Yuchen Cai, Ding Cao, Xin Xu, Zijun Yao, Yuqing Huang, Benyi Zhang, Zhenyu Tan, Guiquan Liu, and Junfeng Fang. On predictability of reinforcement learning dynamics for large language models. In *International Conference on Learning Representations*, volume 2026, pp. 45906–45953, 2026.

Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. Do NOT think that much for 2+3=? On the overthinking of long reasoning models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*. PMLR, 2025.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021.

DeepSeek-AI et al. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025. doi: 10.48550/arXiv.2501.12948.

DeepSeek-AI et al. DeepSeek-V4: Towards highly efficient million-token context intelligence, 2026.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. QLoRA: Efficient finetuning of quantized LLMs. In *Advances in Neural Information Processing Systems*, volume 36, pp. 10088–10115. Curran Associates, Inc., 2023. doi: 10.52202/075280-0441.

Ky Fan. On a theorem of Weyl concerning eigenvalues of linear transformations I. *Proceedings of the National Academy of Sciences of the United States of America*, 35(11):652–655, 1949. doi: 10.1073/pnas.35.11.652.

Nathan Halko, Per-Gunnar Martinsson, and Joel A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53 (2):217–288, 2011. doi: 10.1137/090771806.

Soufiane Hayou, Nikhil Ghosh, and Bin Yu. The impact of initialization on LoRA finetuning dynamics. In *Advances in Neural Information Processing Systems*, volume 37, 2024a.

Soufiane Hayou, Nikhil Ghosh, and Bin Yu. LoRA+: Efficient low rank adaptation of large models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*. PMLR, 2024b.

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

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. *arXiv preprint arXiv:2103.03874*, 2021. doi: 10.48550/arXiv.2103.03874.

Bairu Hou, Yang Zhang, Jiabao Ji, Yujian Liu, Kaizhi Qian, Jacob Andreas, and Shiyu Chang. ThinkPrune: Pruning long chain-of-thought of LLMs via reinforcement learning. *Transactions on Machine Learning Research*, 2026.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2790–2799. PMLR, 2019.

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

Damjan Kalajdzievski. A rank stabilization scaling factor for fine-tuning with LoRA, 2023.

Dawid Jan Kopiczko, Tijmen Blankevoort, and Yuki M. Asano. VeRA: Vector-based random matrix adaptation. In *International Conference on Learning Representations*, 2024.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James Validad Miranda, Alisa Liu, Nouha Dziri, Xinxi Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Christopher Wilhelm, Luca Soldaini, Noah A. Smith, Yizhong Wang, Pradeep Dasigi, and Hannaneh Hajishirzi. Tülu 3: Pushing frontiers in open language model post-training. In *Conference on Language Modeling*, 2025.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 3045–3059. Association for Computational Linguistics, November 2021. doi: 10.18653/v1/2021.emnlp-main.243.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. Solving quantitative reasoning problems with language models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 3843–3857. Curran Associates, Inc., 2022. doi: 10.52202/068431-0278.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 4582–4597. Association for Computational Linguistics, August 2021. doi: 10.18653/v1/2021.acl-long.353.

Vladislav Lialin, Vijeta Deshpande, Xiaowei Yao, and Anna Rumshisky. Scaling down to scale up: A guide to parameter-efficient fine-tuning, 2023.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, 2024.

Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. DoRA: Weight-decomposed low-rank adaptation. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 32100–32121. PMLR, 2024.

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. In *Conference on Language Modeling*, 2025.

Fanxu Meng, Zhaohui Wang, and Muhan Zhang. PiSSA: Principal singular values and singular vectors adaptation of large language models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 121038–121072. Curran Associates, Inc., 2024. doi: 10.52202/079017-3846.

John X. Morris, Niloofar Mireshghallah, Mark Ibrahim, and Saeed Mahloujifar. Learning to reason in 13 parameters, 2026.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: Simple test-time scaling, 2025.

Sagnik Mukherjee, Lifan Yuan, Dilek Hakkani-Tür, and Hao Peng. Reinforcement learning finetunes small subnetworks in large language models. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

OpenAI. Learning to reason with LLMs, September 2024. URL `https://openai.com/index/learning-to-reason-with-llms/`.

Fabian Paischer, Lukas Hauzenberger, Thomas Schmied, Benedikt Alkin, Marc Deisenroth, and Sepp Hochreiter. Parameter efficient fine-tuning via explained variance adaptation. In *Advances in Neural Information Processing Systems*, volume 38, pp. 51637–51689, 2025. doi: 10.52202/085713-1539.

Zhenting Qi, Mingyuan Ma, Jiahang Xu, Li Lyna Zhang, Fan Yang, and Mao Yang. Mutual reasoning makes smaller LLMs stronger problem-solver. In *International Conference on Learning Representations*, 2025.

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

John Schulman and Thinking Machines Lab. LoRA without regret. *Thinking Machines Lab: Connectionism*, September 2025. doi: 10.64434/tml.20250929. URL `https://thinkingmachines.ai/blog/lora/`.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017.

Zhihong Shao et al. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024. doi: 10.48550/arXiv.2402.03300.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling parameters for reasoning. In *International Conference on Learning Representations*, 2025.

Xinyu Tang, Zhenduo Zhang, Yurou Liu, Wayne Xin Zhao, Zujie Wen, Zhiqiang Zhang, and Jun Zhou. Towards high data efficiency in reinforcement learning with verifiable reward. In *International Conference on Learning Representations*, volume 2026, pp. 32611–32635, 2026.

Joel A. Tropp, Alp Yurtsever, Madeleine Udell, and Volkan Cevher. Fixed-rank approximation of a positive-semidefinite matrix from streaming data. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Hanqing Wang, Yixia Li, Shuo Wang, Guanhua Chen, and Yun Chen. MiLoRA: Harnessing minor singular components for parameter-efficient LLM finetuning. 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. 4823–4836. Association for Computational Linguistics, April 2025. doi: 10.18653/v1/2025.naacl-long.248.

Shangshang Wang, Julian Asilis, Ömer Faruk Akgül, Enes Burak Bilgin, Ollie Liu, and Willie Neiswanger. Tina: Tiny reasoning models via LoRA. In *International Conference on Learning Representations*, 2026.

Shaowen Wang, Linxi Yu, and Jian Li. LoRA-GA: Low-rank adaptation with gradient approximation. In *Advances in Neural Information Processing Systems*, volume 37, pp. 54905–54931. Curran Associates, Inc., 2024. doi: 10.52202/079017-1741.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 24824–24837. Curran Associates, Inc., 2022. doi: 10.52202/068431-1800.

An Yang et al. Qwen3 technical report, 2025.

Yibo Yang, Xiaojie Li, Zhongzhu Zhou, Shuaiwen Leon Song, Jianlong Wu, Liqiang Nie, and Bernard Ghanem. CorDA: Context-oriented decomposition adaptation of large language models for task-aware parameter-efficient fine-tuning. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Hao Ye, Jisheng Dang, Junfeng Fang, Bimei Wang, Yizhou Zhang, Ning Lv, Wencan Zhang, Hong Peng, Bin Hu, and Tat-Seng Chua. On the implicit reward overfitting and the low-rank dynamics in RLVR, 2026.

Qingyu Yin, Yulun Wu, Zhennan Shen, Sunbowen Li, Zhilin Wang, Yanshu Li, Chak Tou Leong, Jiale Kang, and Jinjin Gu. Evaluating parameter efficient methods for RLVR, 2025.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Juncai Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Ru Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Yonghui Wu, and Mingxuan Wang. DAPO: An open-source LLM reinforcement learning system at scale. In *Advances in Neural Information Processing Systems*, volume 38, pp. 113222–113244, 2025. doi: 10.52202/085713-3775.

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? In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Chengqian Zhang, Wei Zhu, and Kyumin Lee. Hybrid-LoRA: Bridging full fine-tuning and low-rank adaptation for post-training, 2026a.

Jiaying Zhang, Lei Shi, Jiguo Li, Jun Xu, Jiuchong Gao, Jinghua Hao, and Renqing He. GeoRA: Geometry-aware low-rank adaptation for RLVR. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 24207–24221. Association for Computational Linguistics, July 2026b. doi: 10.18653/v1/2026.acl-long.1110.

Longteng Zhang, Lin Zhang, Shaohuai Shi, Xiaowen Chu, and Bo Li. LoRA-FA: Memory-efficient low-rank adaptation for large language models fine-tuning, 2023a.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. Adaptive budget allocation for parameter-efficient fine-tuning. In *International Conference on Learning Representations*, 2023b.

Ruijia Zhang, Jiacheng Zhu, Hanqing Zhu, and Laixi Shi. Geometry-preserving orthonormal initialization for low-rank adaptation in RLVR, 2026c.

Yuanhe Zhang, Fanghui Liu, and Yudong Chen. LoRA-One: One-step full gradient could suffice for fine-tuning large language models, provably and efficiently. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*. PMLR, 2025.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, Jingren Zhou, and Junyang Lin. Group sequence policy optimization, 2025.

Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. AGIEval: A human-centric benchmark for evaluating foundation models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pp. 2299–2314. Association for Computational Linguistics, June 2024. doi: 10.18653/v1/2024.findings-naacl.149.

Hanqing Zhu, Zhenyu Zhang, Hanxian Huang, DiJia Su, Zechun Liu, Jiawei Zhao, Igor Fedorov, Hamed Pirsiavash, Zhizhou Sha, Jinwon Lee, David Z. Pan, Zhangyang Wang, Yuandong Tian, and Kai Sheng Tai. The path not taken: RLVR provably learns off the principals, 2025.

Jiacheng Zhu, Kristjan Greenewald, Kimia Nadjahi, Haitz Sáez De Ocáriz Borde, Rickard Brüel Gabrielsson, Leshem Choshen, Marzyeh Ghassemi, Mikhail Yurochkin, and Justin Solomon. Asymmetry in low-rank adapters of foundation models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 62369–62385. PMLR, 2024.

A Related Work

A.1 Reasoning LLMs

Reasoning LLMs such as OpenAI o1 and DeepSeek-R1 have substantially advanced competition-level mathematics, code generation, and scientific reasoning (OpenAI, 2024; DeepSeek-AI et al., 2025). Building on chain-of-thought prompting (Wei et al., 2022), test-time scaling improves accuracy by allocating more inference compute to search or to longer reasoning traces (Qi et al., 2025; Snell et al., 2025; Muennighoff et al., 2025). Reinforcement learning with verifiable rewards (RLVR) has become a central post-training paradigm for such models, replacing learned reward models with verifiers of answer correctness or code execution (Lambert et al., 2025; DeepSeek-AI et al., 2025). GRPO removes the value network of PPO (Schulman et al., 2017) by standardizing rewards within a group of responses to the same prompt (Shao et al., 2024), and subsequent methods refine its normalization, clipping, and loss aggregation (Liu et al., 2025; Yu et al., 2025; Zheng et al., 2025). Because RLVR samples and backpropagates through many long rollouts, its training is computationally expensive (Tang et al., 2026); meanwhile, the tendency of reasoning models to overthink (Chen et al., 2025) has motivated RL objectives that shorten reasoning traces (Aggarwal & Welleck, 2025; Arora & Zanette, 2025; Hou et al., 2026).

Recent analyses further characterize the parameter updates induced by RLVR. The reasoning abilities of RLVR-trained models remain largely bounded by those of the base model (Yue et al., 2025), and RL updates only a small subnetwork, although the resulting update matrices are nearly full-rank (Mukherjee et al., 2025). RLVR updates also avoid the principal directions of the pretrained weights (Zhu et al., 2025), and the top singular subspace of the update recovers over 99% of the reasoning gains (Cai et al., 2026). For LoRA, Ye et al. (2026) observe that the output-side factor aligns with the leading left singular vector of the RLVR update more closely than the input-side factor aligns with the leading right singular vector. REAL is complementary to both lines of work: it leaves the RL objective unchanged and uses reward signals to select the low-rank subspace in which the policy is updated.

A.2 Parameter-Efficient Fine-Tuning

Parameter-efficient fine-tuning (PEFT) adapts a frozen pretrained model by training a small set of parameters (Lialin et al., 2023), for example through inserted adapter modules (Houlsby et al., 2019) or continuous prompts and prefixes (Lester et al., 2021; Li & Liang, 2021). LoRA parameterizes each weight update as the product of two low-rank factors that can be merged into the pretrained weights after training (Hu et al., 2022). Its variants quantize the frozen base model (Dettmers et al., 2023), allocate ranks adaptively across weight matrices (Zhang et al., 2023b), decouple magnitude and direction (Liu et al., 2024), or rescale the adapter by $\alpha/\sqrt{r}$ to stabilize learning at higher ranks (Kalajdzievski, 2023). The two factors play asymmetric roles: B benefits from a larger learning rate than A (Hayou et al., 2024b), initializing B rather than A to zero permits larger learning rates (Hayou et al., 2024a), and a random untrained A performs nearly as well as a trained one (Zhu et al., 2024). Accordingly, LoRA-FA freezes a randomly initialized A (Zhang et al., 2023a), and VeRA shares frozen random matrices across layers while learning only scaling vectors (Kopiczko et al., 2024). Theorem 1 gives a dynamical account of why A changes little under the standard initialization, and REAL selects A from reward-contrast gradients instead of keeping it random.

Another line of work replaces random initialization with the singular components of pretrained weights (Meng et al., 2024; Wang et al., 2025), input-activation statistics (Yang et al., 2024; Paischer et al., 2025), or the full-parameter gradient (Wang et al., 2024; Zhang et al., 2025), primarily for supervised fine-tuning. In RL, LoRA can match full fine-tuning under policy-gradient training even at rank one (Schulman & Thinking Machines Lab, 2025) and enables low-cost reasoning post-training (Wang et al., 2026; Morris et al., 2026); yet, as in supervised fine-tuning (Biderman et al., 2024), a noticeable gap to full fine-tuning remains on complex reasoning tasks (Zhang et al., 2026a). Yin et al. (2025) further find that SVD-based initializations such as PiSSA and MiLoRA suffer from spectral collapse under RLVR, consistent with the observation that RLVR updates avoid the principal directions of pretrained weights (Zhu et al., 2025). Weight- and activation-based criteria ignore the reward, and gradient-based criteria applied to RLVR would align the adapter with an aggregated gradient in which group-specific components cancel (Equation 8); REAL instead selects the input subspace that retains the most energy of per-group reward-contrast gradients.

B ALGORITHM

Algorithm 1 REAL: reward-aligned initialization of the input subspace followed by RLVR

Input: initial policy π_{θ_0}, prompt distribution $\mathcal{D}$, adapted layers $\mathcal{L}$, group size K, number of groups m, rank r, sketch sizes $s_{\text{out}}, s_{\text{in}}$, scale ν

1: Sample K responses per prompt from π_{θ_0} and verify them; keep groups with $n_p^+ \geq 2$ and $n_p^- \geq 2$ until $|\mathcal{I}| = m$
2: For each $\ell \in \mathcal{L}$, draw Ω^ℓ and set $Y^\ell \leftarrow 0$
3: **for** each group $p \in \mathcal{I}$ **do**
4: Draw $\{\Psi_p^\ell\}_{\ell \in \mathcal{L}}$; backpropagate $\mathcal{J}_p$ at θ_0 and accumulate each H_p^ℓ by Equation 9
5: $Y^\ell \leftarrow Y^\ell + m^{-1}(H_p^\ell)^\top (H_p^\ell \Omega^\ell)$ for each $\ell \in \mathcal{L}$; discard H_p^ℓ
6: **end for**
7: **for** each layer $\ell \in \mathcal{L}$ **do**
8: $F^\ell \leftarrow Y^\ell \big((\Omega^\ell)^\top Y^\ell\big)^{\dagger/2}$; $\widehat{P}^\ell \leftarrow$ top-r left singular vectors of F^ℓ
9: $A^\ell \leftarrow \nu(\widehat{P}^\ell)^\top$, $B^\ell \leftarrow 0$; freeze A^ℓ
10: **end for**
11: Run RLVR on $\mathcal{D}$, updating only $\{B^\ell\}_{\ell \in \mathcal{L}}$

C PROOF OF THEOREM 1

Proof. Let

$$a_t = \|A_t\|_2, \qquad b_t = \|B_t\|_2, \qquad g_t = \eta_t \alpha \|G_t\|_2,$$

so that $u_{t+1} = u_t + g_t$.

Step 1: Norms of the two factors. The update rules give

$$a_{t+1} \leq a_t + g_t b_t, \qquad b_{t+1} \leq b_t + g_t a_t.$$

Since $\cosh g \geq 1$ and $\sinh g \geq g$, induction with the addition formulas yields

$$a_t \leq \overline{\nu} \cosh u_t, \qquad b_t \leq \overline{\nu} \sinh u_t,$$

which proves the bound on $\|B_T\|_2$.

Step 2: Displacement of A. Using $\cosh u_{t+1} - \cosh u_t \geq g_t \sinh u_t$, we also obtain

$$\begin{aligned} \|A_T - A_0\|_2 &\leq \sum_{t<T} g_t b_t \\ &\leq \overline{\nu} \sum_{t<T} (\cosh u_{t+1} - \cosh u_t) \\ &= \Delta_T. \end{aligned}$$

Hence, by Weyl's inequality,

$$\sigma_{\min}(A_T) \geq \underline{\nu} - \Delta_T > 0.$$

Step 3: Rotation of the input subspace. Let Π_0 be the orthogonal projection onto $\mathrm{row}(A_0)^\perp$. Since $A_0 \Pi_0 = 0$, we have

$$\|A_T \Pi_0\|_2 \leq \|A_T - A_0\|_2 \leq \Delta_T.$$

Every unit vector $x \in \mathrm{row}(A_T)$ can be written as $x = A_T^\top c / \|A_T^\top c\|_2$, so

$$\begin{aligned} \|\Pi_0 x\|_2 &= \frac{\|\Pi_0 A_T^\top c\|_2}{\|A_T^\top c\|_2} \\ &\leq \frac{\|A_T \Pi_0\|_2}{\sigma_{\min}(A_T)} \\ &\leq \frac{\Delta_T}{\underline{\nu} - \Delta_T}. \end{aligned}$$

Maximizing over x bounds $\sin\phi_T$, and

$$\Delta_T = \frac{\overline{\nu}}{2} u_T^2 + O(u_T^4)$$

gives the expansion. □

D Proof of Proposition 1

Proof. Let

$$C = U\Lambda U^\top, \qquad \Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_{d_{\mathrm{in}}}), \qquad \lambda_1 \geq \cdots \geq \lambda_{d_{\mathrm{in}}},$$

and let $u_1, \ldots, u_{d_{\mathrm{in}}}$ be the columns of U. For $P^\top P = I_r$, let $Z = U^\top P$, so that $Z^\top Z = I_r$ and

$$\gamma(P) = \mathrm{tr}\big(Z^\top \Lambda Z\big) = \sum_{j=1}^{d_{\mathrm{in}}} \lambda_j c_j, \qquad c_j = \|Z_{j,:}\|_2^2.$$

Step 1: Upper bound. The weights c_j are the diagonal entries of the rank-r orthogonal projection $ZZ^\top$, so

$$0 \leq c_j \leq 1, \qquad \sum_j c_j = r.$$

Using $c_j - 1 \leq 0$ and $\lambda_j \geq \lambda_r$ for $j \leq r$, and $c_j \geq 0$ and $\lambda_j \leq \lambda_r$ for $j > r$, we obtain

$$\begin{aligned}
\gamma(P) - \sum_{j=1}^{r} \lambda_j &= \sum_{j \leq r} \lambda_j (c_j - 1) + \sum_{j > r} \lambda_j c_j \\
&\leq \lambda_r \sum_{j \leq r} (c_j - 1) + \lambda_r \sum_{j > r} c_j \\
&= \lambda_r \Big(\sum_j c_j - r \Big) \\
&= 0.
\end{aligned}$$

Step 2: Attainment. The bound is attained by $P^\star = [u_1, \ldots, u_r]$, for which

$$c_j = 1 \ \text{ for } j \leq r, \qquad c_j = 0 \ \text{ for } j > r.$$ □

E Experimental Details

Training configuration. For Qwen3-1.7B and Qwen3-4B, we select Big-Math problems (Albalak et al., 2025) with annotated Llama-8B solve rates in $[0.2, 0.4]$ and $[0.1, 0.3]$, respectively, remove duplicates and problems matching evaluation questions, exclude the HARP and AMC/AIME source subsets, and keep only problems whose reference answers our verifier supports. For Qwen3-8B, we use the processed training split of DAPO-Math-17k (Yu et al., 2025). We implement all methods with TRL and PEFT and generate rollouts with vLLM. We set LoRA alpha to 32 for Qwen3 and 128 for the distilled models, disable adapter dropout. For the max completion length, we set 6,000 tokens for Qwen3-1.7B and Qwen3-4B, 8,000 tokens for Qwen3-8B, and 12,000 for distilled model. For Qwen3, we aggregate the loss as in Equation 2 and clip importance ratios with $\epsilon_{\mathrm{low}} = 0.2$ and $\epsilon_{\mathrm{high}} = 0.28$ (Yu et al., 2025); for the distilled models, we use Dr. GRPO loss aggregation (Liu et al., 2025). PiSSA uses fast SVD initialization with 16 iterations. At rank 16, REAL trains 9.18M, 17.69M, and 22.41M adapter parameters on Qwen3-1.7B, Qwen3-4B, and Qwen3-8B, compared with 17.43M, 33.03M, and 43.65M for LoRA.

Relative length reward during observation. For DS-1.5B and DS-7B, REAL adds a relative length utility to the observation reward. For a prompt with responses $o_1, \ldots, o_K$, correctness rewards $R_i \in \{0, 1\}$, and lengths $L_i = |o_i|$, let n^+ be the number of correct responses and μ_L and σ_L the

mean and population standard deviation of their lengths. When $n^+ \geq 2$ and $\sigma_L > 0$, each correct response receives

$$u_i = \frac{z_i - \overline{z}}{\max\{1, \max_j |z_j - \overline{z}|\}}, \qquad z_i = \tanh\left(\frac{\mu_L - L_i}{\sigma_L + 10^{-6}}\right),$$

where j ranges over the correct responses and $\overline{z}$ is the mean of their scores z_j; all other responses receive $u_i = 0$. The observation reward is $R_i^{\text{obs}} = R_i + \lambda u_i$ with $\lambda = 0.5$, so shorter correct responses receive higher rewards. Since $|u_i| \leq 1$ and the utilities sum to zero, correct responses keep rewards in $[0.5, 1.5]$, above the zero reward of incorrect responses, and the mean reward of each group is unchanged; groups whose responses are all correct still provide a length contrast when their lengths differ. REAL uses R_i^{obs} only to compute the observation advantages for subspace estimation; after initialization, A is fixed and policy training uses the correctness rewards R_i.

Evaluation. All methods share the same benchmark-specific answer extraction and verification. We sample 16 responses per problem on AMC 2023, AIME 2024, AIME 2025, BRUMO 2025, CMIMC 2025, and HMMT 2025; 8 on GPQA and LSAT; and 4 on GSM8K, MATH-500, Minerva, and OlympiadBench. Accuracy is the fraction of correct responses among all samples, and completion length is the mean number of generated tokens over all samples. Averages weight benchmarks equally, accuracy improvements denote absolute differences, and aggregate length reductions are computed from benchmark-averaged lengths.

Reward-gradient coverage. We sample 256 MATH-500 problems, none of which overlap with the training data, and generate $K = 8$ responses per problem with Qwen3-4B-Base at temperature 0.6, top-p 0.95, and an 8,192-token limit, retaining the set $\mathcal{P}$ of 79 problems with two to six correct responses. For method m and adapted module ℓ, let the columns of $V_{m,\ell,k}$ be the top-k right singular vectors of the weight update accumulated over 600 training steps, and let $G_{p,\ell}$ be the gradient of the base model's group-normalized policy objective for problem p. We define the coverage

$$\text{Cov}_{m,\ell}(k) = \frac{\sum_{p\in\mathcal{P}} \|G_{p,\ell} V_{m,\ell,k}\|_F^2}{\sum_{p\in\mathcal{P}} \|G_{p,\ell}\|_F^2}.$$

We average it uniformly over the modules in each group and obtain 95% confidence intervals from 2,000 paired bootstrap resamples over problems.

F Memory Efficiency of Subspace Estimation

Table 3: Memory required to estimate the input subspace of all adapted matrices on three Qwen3 models. Exact estimation stores G_p and C, and two-sided random compression stores Ψ_p, H_p, Ω, and Y with $s_{\text{out}} = s_{\text{in}} = 64$. All buffers use fp32, and 1 GB denotes 10^9 bytes.

Model	Exact (GB)	Compressed (GB)	Ratio	Reduction
Qwen3-1.7B	12.684	0.543	23.35×	95.72%
Qwen3-4B	35.295	1.019	34.63×	97.11%
Qwen3-8B	64.022	1.378	46.47×	97.85%

To quantify the memory saved by two-sided random compression, we count the buffers that subspace estimation keeps for all attention and MLP projections of the Qwen3 models under the sketch sizes $s_{\text{out}} = s_{\text{in}} = 4r = 64$ used in the main experimental settings. Exact estimation keeps $d_{\text{in}}(d_{\text{out}}+d_{\text{in}})$ entries per adapted matrix for the group gradient G_p and the second moment C. Two-sided random compression instead keeps $s_{\text{out}}(d_{\text{out}} + d_{\text{in}}) + 2s_{\text{in}}d_{\text{in}}$ entries for Ψ_p, H_p, Ω, and Y in Equation 9. As shown in Table 3, two-sided random compression reduces the memory of subspace estimation by 95.72% to 97.85% across the three models. On Qwen3-8B, it shrinks the 64.022 GB required by exact estimation to only 1.378 GB. The compression ratio grows from 23.35× on Qwen3-1.7B to 46.47× on Qwen3-8B because the exact memory grows quadratically with the layer width and the compressed memory grows only linearly.

G Ablation Analysis

Each ablation changes one setting of the main experiments and keeps the rest fixed. The rank ablation uses Qwen3-4B, and the observation-step, component, and learning-rate ablations use Qwen3-1.7B.

G.1 Rank Analysis

Table 4: Effect of the adapter rank on Qwen3-4B. All other settings follow Section 4.1. The shaded row is the rank used in the main experiments, and the best value in each column is shown in **bold**.

Rank	GSM8K	MATH-500	AMC 2023	Minerva	OlympiadBench	GPQA	Avg.
4	92.04	77.60	52.66	37.68	42.51	38.51	56.83
8	92.34	77.95	54.84	38.42	**44.62**	39.71	57.98
16	92.19	78.35	53.28	39.15	44.14	40.85	57.99
32	92.23	**79.20**	54.22	**39.25**	43.92	**41.29**	**58.35**
64	**92.40**	79.00	**55.16**	38.61	43.95	39.65	58.13

To analyze how the update capacity affects post-training, we train REAL on Qwen3-4B with ranks from 4 to 64 under the same training configuration and compare the resulting accuracy on six benchmarks. As shown in Table 4, the average accuracy increases only slightly, from 56.83% at rank 4 to a peak of 58.35% at rank 32. REAL therefore achieves most of its performance within a very small subspace, and even at rank 4 it outperforms all rank-16 baselines on Qwen3-4B in Table 1. Further increasing the rank brings little gain, as the average accuracy varies by at most 0.37% from rank 8 to rank 64. This saturation is consistent with the low-rank nature of RLVR updates (Cai et al., 2026).

G.2 Observation Steps Analysis

Table 5: Effect of the number of observation steps on Qwen3-1.7B. The shaded row is the observation budget used in the main experiments, and the best value in each column is shown in **bold**.

Obs. steps	GSM8K	MATH-500	AMC 2023	Minerva	OlympiadBench	GPQA	Avg.
8	83.45	65.90	39.53	27.39	31.53	25.76	45.59
16	83.79	67.75	38.44	**28.95**	32.16	24.12	45.87
32	83.74	68.10	38.75	28.12	32.27	26.77	46.29
64	83.98	69.05	40.31	28.77	**33.31**	**27.15**	47.10
128	84.00	68.90	41.09	28.40	32.34	24.56	46.55
256	**84.10**	**69.60**	**42.50**	28.40	32.46	26.20	**47.21**

The number of observation steps is another key variable of REAL. On training data of suitable difficulty, the observation phase efficiently collects prompt groups with both correct and incorrect responses, but the number of steps determines how many groups enter the second moment C and thus how accurately the subspace is estimated. To analyze this effect, we train REAL on Qwen3-1.7B with observation steps ranging from as few as 8 to 256 under the same training configuration and compare the resulting accuracy on six benchmarks. As shown in Table 5, the average accuracy follows an overall upward trend and rises from 45.59% at 8 steps to 47.10% at 64 steps. This trend indicates that more observation steps provide a more stable initial subspace. The accuracy nevertheless varies by at most 1.62% across all settings and essentially plateaus once the observation phase reaches 64 steps.

G.3 Component Analysis

We compare five variants that differ in how the input subspace is chosen and whether it is trained:

- **Random Frozen** initializes A randomly and freezes it, training only B.
- **REAL Positive** estimates the subspace from the gradients of positive-advantage responses alone, then freezes A and trains only B.
- **REAL Gradient** aggregates the reward gradients across prompt groups and initializes A from the top-r right singular vectors of the aggregated gradient, then trains both factors; this applies the gradient-aligned initialization of LoRA-GA (Wang et al., 2024) to RLVR.
- **REAL Train** initializes A as REAL does but trains both A and B.
- **REAL Frozen** is the full method: A is initialized from the principal subspace of the input-side second moment of per-group reward-contrast gradients and kept fixed, and only B is trained.

Table 6: Component ablation on Qwen3-1.7B. Each variant changes one design choice of REAL. The best value in each column is shown in **bold**.

Variant	GSM8K	MATH-500	AMC 2023	Minerva	OlympiadBench	GPQA	Avg.
Random Frozen	80.61	63.50	36.25	25.09	26.89	26.96	43.22
REAL Positive	83.40	66.70	39.84	25.64	30.53	24.05	45.03
REAL Gradient	83.91	68.15	**40.31**	28.12	32.23	25.32	46.34
REAL Train	83.49	68.50	40.00	**29.41**	**33.57**	26.70	46.95
REAL Frozen	**83.98**	**69.05**	**40.31**	28.77	33.31	**27.15**	**47.10**

As shown in Table 6, freezing a random subspace reaches 43.22%, 0.19% below LoRA, because a randomly initialized A still receives updates during training. Choosing the subspace from reward gradients lifts accuracy by 3.88% over Random Frozen. The criterion matters as well: with both factors trained, the per-group second moment reaches 46.95% against 46.34% for the aggregated gradient of LoRA-GA, consistent with the cancellation in Equation 8. Restricting observation to positive advantages costs 2.07%, confirming that the contrast between correct and incorrect responses carries the reward signal. Freezing A after the REAL initialization adds a further 0.15% over training it, as continued updates rotate the subspace away from the directions selected during observation.

G.4 Learning Rate Analysis

Table 7: Effect of the learning rate on Qwen3-1.7B. Full denotes full fine-tuning with a learning rate of 10^{-6}, and LoRA and REAL use a tenfold larger learning rate of 10^{-5}. Results of LoRA and REAL at 10^{-6} are reported in Table 1. The best value in each column is shown in **bold**.

Method	GSM8K	MATH-500	AMC 2023	Minerva	OlympiadBench	GPQA	Avg.
Full	83.21	66.90	**40.00**	**29.78**	**31.75**	**28.22**	**46.64**
LoRA	83.66	64.05	39.06	26.93	30.12	24.94	44.79
REAL	**84.44**	**67.45**	39.69	28.13	30.64	26.70	46.17

Prior work recommends training LoRA with a learning rate about $10\times$ that of full fine-tuning (Schulman & Thinking Machines Lab, 2025). To analyze how the learning rate affects LoRA and REAL, we raise the learning rate of both methods tenfold to 10^{-5} on Qwen3-1.7B and compare them with full fine-tuning. As shown in Table 7, the larger learning rate improves LoRA by 1.38% over its result at 10^{-6} in Table 1, but LoRA still falls 1.85% below full fine-tuning. REAL does not benefit from this increase, which lowers its accuracy from 47.10% to 46.17%. With the default learning rate of 10^{-6}, REAL outperforms both LoRA at 10^{-5} and full fine-tuning.