# Structure Over Scale: Rethinking Adaptation for Reinforcement Learning with Verifiable Rewards

**Anonymous authors**
**Paper under double-blind review**

## Abstract

The standard justification for Full Fine-Tuning (FFT) in Reinforcement Learning with Verifiable Rewards (RLVR) rests on a reasonable intuition: reasoning requires expressive weight updates that Low-Rank Adaptation (LoRA) cannot provide. We show this intuition identifies the wrong variable. Through a systematic rank sweep under GRPO, we document *rank collapse*—a discontinuous performance cliff where increasing LoRA rank beyond a threshold causes catastrophic, irrecoverable policy failure, a phenomenon absent from the SFT literature. A batch-size ablation shows that this failure is not rescued by larger batches under the same one-epoch cold-start GRPO protocol: LoRA ranks 128 and 256 remain near floor across batch sizes 64, 128, and 256, while rank 64 itself falls from 73.1% at batch size 64 to 8.7% and 6.0% at batch sizes 128 and 256. This failure is not generic undertraining: LoRA $r = 8$, DoRA $r = 16$, and QuanTA $d = 3$ remain trainable under the same larger-batch regimes. Spectral analysis suggests a mechanism: collapsed high-rank adapters concentrate update energy into a small number of singular directions, consistent with degenerate optimization rather than distributed reasoning improvement. FFT shows a milder version of the same spectral concentration pattern, achieving lower effective rank than structured adapters despite updating far more parameters. Expressivity alone is therefore not the bottleneck; the structure of the update manifold is. Structured adapters that constrain which high-rank solutions are reachable by gradient descent outperform LoRA and FFT on our primary DeepMath-Hard comparison and remain more robust under the larger-batch stress tests. Across three 8B base models, the relative behavior of low-rank and structured high-rank adapters also correlates with frozen-weight spectral structure and reported pre-training scale, a pattern we term the Model Maturity Hypothesis. We present this as a falsifiable hypothesis rather than a causal law: architecture, tokenizer, and data mixture remain confounded with pre-training scale in the current model set. The operative question for RLVR is not simply whether to use LoRA or FFT, but what structure to impose over the update manifold under a given model, task, and optimization budget.

## 1 Introduction

### 1.1 The RLVR Efficiency Problem

The dominant recipe for training reasoning-capable large language models (LLMs) pairs a base model pre-trained on trillions of tokens with a Reinforcement Learning with Verifiable Rewards (RLVR) post-training stage (DeepSeek-AI, 2025). Within this stage, Group Relative Policy Optimization (GRPO) (Shao et al., 2024) has emerged as the standard method for reasoning: it eliminates a separate value-function critic by normalizing rewards within a group of sampled rollouts, halving the memory footprint compared with PPO.

Despite GRPO's efficiency, the field faces a persistent dichotomy when choosing how many model parameters to update. Leading laboratories default to *Full Fine-Tuning* (FFT) (DeepSeek-AI, 2025), accepting massive memory and gradient-synchronization overhead under the assumption that reasoning updates require high intrinsic dimensionality. Academic work instead adopts *Low-Rank Adaptation* (LoRA) (Hu et al., 2021), citing evidence that rank $r$=16 achieves performance parity with FFT when hyperparameters are carefully

optimized (Schulman and Lab, 2025). The field has thus settled into a binary choice: expensive correctness or affordable approximation.

We present experimental evidence hypothesizing that *both* assumptions underlying this binary are incorrect, and that the resolution lies in structured high-rank tensor adaptation.

### 1.2 Two Surprising Findings

**Finding 1 — Rank Collapse in RLVR.** The SFT literature predicts that increasing LoRA rank yields diminishing returns (Hu et al., 2021; Albert et al., 2025). Our LoRA rank sweep on Qwen 3 8B trained with GRPO on DeepMath-Hard reveals a qualitatively different phenomenon:

| Nominal rank $r$ | 8 | 16 | 32 | 64 | 128 | 256 |
|---|---|---|---|---|---|---|
| Accuracy | 78.1% | 76.2% | 77.3% | 73.1% | **4.7%** | **2.3%** |

Performance peaks at $r = 8$ and collapses catastrophically at $r \geq 128$ under the original batch-size-64 protocol. A batch-size ablation shows that this is not a small-batch artifact: $r = 128$ and $r = 256$ remain near floor across batch sizes 64, 128, and 256, while $r = 64$ itself falls from 73.1% at batch size 64 to 8.7% and 6.0% at batch sizes 128 and 256. At the same larger-batch regimes, LoRA $r = 8$, DoRA $r = 16$, and QuanTA $d = 3$ remain trainable, showing that the failure is rank- and structure-dependent rather than generic undertraining. SVD analysis (Figure 3) identifies a consistent spectral signature of this failure: despite their high nominal rank, the $r = 128$ and $r = 256$ checkpoints achieve near-zero effective rank (participation ratio), with update energy concentrated into fewer than ten singular directions. This suggests that unconstrained high-rank parameters can create destructive optimization pathways in cold-start RLVR rather than simply increasing useful expressivity. This behavior contrasts with the diminishing-returns pattern reported in the SFT literature we compare against.

**Finding 2 — The Model Maturity Hypothesis.** Alongside rank collapse, we observe a systematic divergence in which adapter family performs best, and this divergence correlates cleanly with base-model pre-training scale. Table 1 summarizes the pattern across three architecturally independent models.

Table 1: Performance split across three base models. All results on DeepMath-Hard (Seed 42). QuanTA is competitive on ∼15T-token models but is the clear winner on the 36T-token model.

| Model | Pre-training | Best LoRA | Best QuanTA | Δ |
|---|---|---|---|---|
| Apertus 8B | ∼15T | 2.3% | 3.5% | +1.2% |
| Llama 3.1 8B | ∼15.6T | 2.7% | 3.9% | +1.2% |
| Qwen 3 8B | ∼36T | 78.1% | 84.0% | +5.9% |

The two ∼15T models not only agree in direction but also exhibit a qualitative signature absent from Qwen 3. We hypothesize that this behavioral bifurcation is related to pre-training scale. The 36T-token model has markedly flatter singular value spectra in its frozen weights, suggesting more distributed pre-trained representations. Under this interpretation, adaptation may benefit from updates spread across many singular directions—precisely the regime targeted by QuanTA's Matrix Product Operator (MPO) structure. We present this as a well-supported hypothesis rather than an established causal law: the two groups are cleanly separated, but architecture, tokenizer, and data mixture co-vary with pre-training scale and cannot be fully disentangled with the current model set.

### 1.3 Why Structure, Not Rank Alone

Both findings point to the same organizing principle. Rank collapse shows that unconstrained high-rank updates can be harmful in cold-start RLVR under our fixed-pass GRPO protocol. The model-maturity results suggest that constrained low-rank updates may also become insufficient for sufficiently mature models. The

resolution we study is structured high-rank adaptation: adapters that retain distributed update capacity while restricting which high-rank solutions gradient descent can reach.

### 1.4 Contributions

1. **Rank Collapse as a Rank–Batch Stability Boundary.** The first systematic documentation of catastrophic LoRA failure in cold-start GRPO: at batch size 64, collapse appears at $r \geq 128$, while a batch-size ablation shows that increasing batch size to 128 or 256 does not rescue high-rank LoRA and instead moves the observed safe-rank boundary downward.

2. **The Full Fine-Tuning Effective-Rank Paradox.** Evidence that FFT achieves lower effective rank in learned $\Delta W$ matrices than structured adapters using less than 0.6% of the parameters, suggesting that structural inductive bias—not parameter count alone—is a central factor in this RLVR regime.

3. **Comprehensive Empirical Study with Spectral Diagnostics.** The first head-to-head comparison of LoRA (rank sweep $r \in \{8, 16, 32, 64, 128, 256\}$), DoRA, QuanTA, and FFT under a cold-start GRPO pipeline, with multi-seed validation and SVD-based spectral analysis.

4. **The Model Maturity Hypothesis.** A falsifiable hypothesis linking base-model pre-training scale to the relative behavior of low-rank and structured high-rank adapters under cold-start GRPO, substantiated by behavioral replication across three architecturally independent models spanning $\sim$15T to 36T pre-training tokens and by a frozen-weight spectral signature that is consistent with the observed adaptation behavior before any fine-tuning updates are applied.

5. **Practical Guidance for Efficient RLVR.** We show that the industry FFT-vs-LoRA binary is a false dichotomy, that unconstrained high-rank LoRA is unsafe in cold-start GRPO under fixed-pass training, that larger batches should not be assumed to rescue collapse, and that structured adapters widen the stable optimization region.

## 2 Background and Related Work

### 2.1 Reinforcement Learning with Verifiable Rewards

RLVR uses ground-truth outcomes—such as mathematically verified answers—as the reward signal, avoiding the over-optimization risks of learned reward models (DeepSeek-AI, 2025). Shao et al. (2024) introduced Group Relative Policy Optimization (GRPO), which eliminates the memory cost of a separate critic by computing per-output advantages within a group of $G$ completions sampled from the current policy:

$$A_i = \frac{r_i - \text{mean}(r_1, \ldots, r_G)}{\text{std}(r_1, \ldots, r_G) + \epsilon}, \tag{1}$$

where $r_i \in \{0, 1\}$ is the binary correctness reward for the $i$-th completion. The resulting high variance in advantage estimates—inherent to sparse binary rewards—is a defining property of the RLVR optimization landscape and, as our experiments in Section 4 suggest, an important condition under which rank collapse can emerge in unconstrained adapters. We operate in the cold-start (Base-to-RL) regime (Liu et al., 2025), in which the model must simultaneously learn output structure and reasoning logic from the reward signal alone, without any SFT warmup. We adopt the Dr. GRPO token-level loss normalization (Liu et al., 2025) to prevent penalizing longer Chain-of-Thought completions.

### 2.2 Parameter-Efficient Fine-Tuning

**LoRA.** Hu et al. (2021) freezes the pre-trained weight matrix $W_0 \in \mathbb{R}^{d \times k}$ and injects a trainable low-rank update $\Delta W = BA$, where $B \in \mathbb{R}^{d \times r}$, $A \in \mathbb{R}^{r \times k}$, and $r \ll \min(d, k)$:

$$W = W_0 + \Delta W, \qquad \Delta W = BA. \tag{2}$$

The method is grounded in the Intrinsic Dimension Hypothesis (Aghajanyan et al., 2020), which posits that downstream adaptation lies in a low-dimensional subspace—a characterization established in SFT settings whose validity in RLVR our experiments directly challenge.

**DoRA.** Liu et al. (2024) decompose each weight matrix into a magnitude scalar and a directional component, applying a LoRA update only to the direction:

$$W = \|W_0\|_c \cdot \frac{W_0 + \Delta W}{\|W_0 + \Delta W\|_c}, \tag{3}$$

where $\|\cdot\|_c$ denotes column-wise norm. By forcing the optimizer to treat magnitude and direction as independent degrees of freedom, this decomposition acts as a structural constraint on the update manifold—one that, as our spectral analysis shows, produces substantially higher effective rank than LoRA at identical nominal rank. DoRA is therefore not simply a LoRA variant with better hyperparameters: it is a structurally constrained adapter whose inductive bias places it in the same mechanistic category as QuanTA, achieved via a different parameterization.

**QuanTA.** Chen et al. (2024) parameterize $\Delta W$ as a Matrix Product Operator (MPO), a tensor network structure adapted from quantum many-body physics. For a weight matrix with reshaped local indices $\mathbf{i} = (i_1, \ldots, i_n)$ and $\mathbf{j} = (j_1, \ldots, j_n)$, the update takes the form:

$$\Delta W(\mathbf{i}, \mathbf{j}) = \sum_{\boldsymbol{\alpha}} \prod_{k=1}^{n} C^{(k)}_{\alpha_{k-1}\alpha_k}(i_k, j_k), \tag{4}$$

where $C^{(k)} \in \mathbb{R}^{\chi \times \chi \times d_k \times d_k}$ are learnable core tensors and $\chi$ is the bond dimension. The MPO structure decouples adaptation rank from parameter count: $\Delta W$ can approximate a full-rank transformation while parameters scale linearly in the number of modes $n$. We evaluate two configurations: a parameter efficient setting ($d = 4$, decomposition $[16, 8, 8, 4]$, 7.7M parameters) and a high-capacity setting ($d = 3$, decomposition $[16, 16, 16]$, 35.3M parameters). BOFT (Liu et al., 2023) and MoRA (Jiang et al., 2024) impose alternative structural constraints; we defer a detailed theoretical, not empirical, comparison to Appendix K.

### 2.3 Related Work

**Rank and expressivity in SFT.** The SFT literature consistently finds that increasing LoRA rank yields diminishing returns, with performance plateauing well below the full-rank regime (Hu et al., 2021; Liu et al., 2024)—a pattern explained by the Intrinsic Dimension Hypothesis (Aghajanyan et al., 2020). Our findings establish a qualitative distinction: in cold-start RLVR, higher rank without structural constraint does not yield diminishing returns, it yields catastrophic policy collapse.

**Stability and scheduling in RLVR.** Schulman and Lab (2025) report that Base-to-RL fine-tuning with LoRA $r = 16$ achieves parity with FFT under optimized hyperparameters, that constant learning-rate schedules are stable at large batch sizes, and that high-rank LoRA does not collapse in their setup. Our results show that these conclusions do not transfer unchanged to our fixed-pass cold-start GRPO protocol: constant schedules collapse at batch size 64, LoRA parity depends on the structured-adapter baseline, and increasing batch size alone does not rescue high-rank LoRA under our fixed protocol. We therefore attribute the discrepancy to the joint effects of batch size, update horizon, learning-rate schedule, and optimization protocol rather than to a single universal LoRA rank threshold.

**PEFT for RL reasoning.** Liu et al. (2025) analyze cold-start RLVR failure modes and motivate the Dr. GRPO loss correction we adopt; DeepSeek-AI (2025) demonstrate strong reasoning via RLVR but rely on FFT throughout. To our knowledge, no prior work has conducted a systematic comparison of structured high-rank adapters against LoRA and FFT in the same RLVR pipeline, nor provided the spectral mechanistic analysis we present here.

# 3 Methodology

## 3.1 Base Models and Pre-training Scale

We study three dense transformer models spanning a wide range of pre-training scale (Table 2), using the *base* checkpoint throughout. Base checkpoints ensure the model must learn both output formatting and reasoning logic from the reward signal alone—the cold-start regime in which adapter expressivity is most likely to be a binding constraint.

Table 2: **Base models evaluated.** All models are dense transformers at the 8B parameter scale; pre-training token counts are taken from the respective technical reports.

| Model | Family | Pre-training Scale | Vocab Size |
|---|---|---|---|
| Apertus 8B | Independent | ∼15T tokens | 131K |
| Llama 3.1 8B | Meta | ∼15.6T tokens | 128K |
| Qwen 3 8B | Alibaba | ∼36T tokens | 150K |

## 3.2 Adapter Configurations

Table 3 summarizes all configurations evaluated, applied to the same set of linear projection modules in `bfloat16` throughout. The LoRA rank sweep ($r \in \{8, 16, 32, 64, 128, 256\}$) and DoRA rank sweep ($r \in \{8, 16, 32, 64\}$) are conducted on Qwen 3 8B only; FFT is similarly restricted to Qwen 3 8B due to compute constraints. We additionally evaluate QuanTA with target-module ablations (excluding key/value projections and excluding only the value projection); full ablation results are in Appendix G.

Table 3: **Adapter configurations.** Trainable parameter counts reported for Qwen 3 8B. "All linear"$^\dagger$ denotes all attention projection and feed-forward linear modules with KV excluded.

| Method | Configuration | Target Modules | Trainable Params |
|---|---|---|---|
| LoRA | $r \in \{8, 16, 32, 64, 128, 256\}$ | All linear | 22M–∼700M |
| DoRA | $r \in \{8, 16, 32, 64\}$ | All linear | 23M–180M |
| QuanTA $d = 4$ | $[16, 8, 8, 4]$ | All linear$^\dagger$ | 7.7M |
| QuanTA $d = 3$ | $[16, 16, 16]$ | All linear$^\dagger$ | 35.3M |
| FFT | Full model | All params | ∼8B |

## 3.3 Benchmark Selection and Data Curation

We train and evaluate on five benchmarks; full curation details are in Appendix B. **DeepMath-Hard** applies a difficulty filter ($\geq 8.5/10$) to DeepMath-103K (He et al., 2025), yielding 5,399 hardest problems concentrated in abstract mathematics; this is our primary discriminative benchmark. This hard-example construction is also motivated by recent evidence that, under fixed annotation budgets, GRPO post-training benefits most from examples that are difficult for the base model rather than easy, medium, or randomly selected examples (Pikus et al., 2025). **Skywork-Hard** retains the 6,702 problems from Skywork-OR1 (Zeng et al., 2024) on which DeepSeek-R1-Distill-Qwen-32B achieves a pass rate below 19% (Score $\geq 13$), providing a frontier stress test. **Enigmata** (Chen et al., 2025) is a logic and puzzle reasoning benchmark. We draw 10,000 problems while preserving the dataset's original difficulty distribution, and evaluate the model on the official published evaluation set. **MATH** (Hendrycks et al., 2021) serves as a standard reference benchmark. We additionally probe out-of-distribution generalization by evaluating DeepMath-Hard-trained checkpoints on the merged **AIME 2025–2026** (Art of Problem Solving, 2025a;b; 2026a;b) competition set (60 problems, Avg@32).

### 3.4 Reinforcement Learning Protocol

Unless otherwise stated, we train models with GRPO (Shao et al., 2024) using group size $G = 8$, KL coefficient $\beta = 0$, the Dr. GRPO token-level loss (Liu et al., 2025), and global batch size 64; vLLM (Kwon et al., 2023) accelerates generation via PagedAttention. The batch-size ablation in Section 4.2 varies the global batch size over $\{64, 128, 256\}$ while holding the learning-rate schedule, reward function, group size, optimizer, and one-epoch protocol fixed. The reward is a composite $R_{\text{total}} = R_{\text{acc}} + R_{\text{fmt}}$: $R_{\text{acc}}$ uses depth-aware `\boxed{}` extraction with `math_verify` semantic equivalence (1.0 correct, 0.0 otherwise), and $R_{\text{fmt}}$ applies shaped penalties for structural degeneration (empty reasoning chains, repetition, hallucinated turns), bounded below $R_{\text{acc}}$ to prevent reward hacking. Full reward weights and hyperparameters are in Appendix A.

**Scheduling and cold-start stability.** Schulman and Lab (2025) report that constant learning-rate schedules are stable in large-batch Base-to-RL training. In our batch-size-64 main protocol, constant schedules cause immediate policy collapse across all methods (Appendix F). A cosine decay schedule with warmup ratio 0.1 and minimum learning-rate ratio 0.15 eliminates this failure mode in the batch-size-64 main protocol, so we use it as a fixed component of all experiments in this paper. We do not claim that this schedule is universally optimal across batch sizes or update budgets.

## 4 Experiments and Results

### 4.1 Rank Collapse: The LoRA Rank Sweep in RLVR

Table 4 presents a systematic LoRA rank sweep on Qwen 3 8B trained on DeepMath-Hard at global batch size 64. Performance peaks at $r = 8$ (78.1%) and collapses catastrophically between $r = 64$ (73.1%) and $r = 128$ (4.7%); at $r = 256$, the policy learns nothing (2.3%). This is qualitatively distinct from the SFT literature's prediction of diminishing returns: under this RLVR protocol, performance does not merely plateau at high rank but falls to near-floor accuracy. Section 4.5 provides a spectral analysis showing that these collapsed runs concentrate update energy into a small number of singular directions.

**From rank threshold to stability boundary.** The batch-size ablation in Section 4.2 refines this finding. At batch size 64, the observed collapse boundary lies between $r = 64$ and $r = 128$. However, this threshold is not a property of nominal rank alone: under the same one-epoch GRPO protocol, increasing global batch size to 128 or 256 does not rescue high-rank LoRA under our fixed protocol and instead moves the observed safe-rank boundary downward. Thus, rank collapse should be understood as a rank–batch–optimization stability boundary for unconstrained LoRA, rather than as a fixed universal rank cutoff.

Table 4: **LoRA rank sweep on Qwen 3 8B, DeepMath-Hard (seed 42, batch size 64).** Performance peaks at $r = 8$ and collapses at $r \geq 128$, a discontinuous cliff unlike the diminishing-returns pattern reported in the SFT literature.

| Method | Nominal Rank | Trainable Params | DeepMath-Hard |
|---|---|---|---|
| LoRA | $r = 8$ | 22M | 78.1% |
| LoRA | $r = 16$ | 43.6M | 76.2% |
| LoRA | $r = 32$ | 87.2M | 77.3% |
| LoRA | $r = 64$ | 174M | 73.1% |
| LoRA | $r = 128$ | 348M | 4.7% |
| LoRA | $r = 256$ | 696M | 2.3% |

### 4.2 Batch-Size Scaling Does Not Rescue High-Rank LoRA Under Our Fixed Protocol

A natural explanation for the collapse in Table 4 is that global batch size 64 produces high-variance GRPO advantage estimates, and that larger batches should stabilize unconstrained high-rank LoRA. We test this directly by increasing the global batch size to 128 and 256 while preserving the same one-epoch cold-start

GRPO protocol, learning-rate schedule, reward function, group size, and optimizer. Because DeepMath-Hard contains 5,399 training examples, one epoch corresponds to 674, 337, and 168 optimizer steps at batch sizes 64, 128, and 256, respectively.

Table 5 shows that increasing batch size does not rescue high-rank LoRA under our fixed protocol. LoRA $r = 128$ and $r = 256$ remain near floor across all tested batch sizes. More surprisingly, LoRA $r = 64$, which reaches 73.1% at batch size 64, falls to 8.7% and 6.0% at batch sizes 128 and 256. This failure is not explained by reduced update count alone: at batch size 128, LoRA $r = 8$, DoRA $r = 16$, and QuanTA $d = 3$ still reach 64.0%, 67.6%, and 72.2%, respectively. Thus, larger batches do not eliminate rank collapse; under a fixed-pass GRPO protocol, they move the observed LoRA stability boundary downward.

Table 5: **Batch-size ablation on Qwen 3 8B, DeepMath-Hard (seed 42).** All runs use one epoch and the same learning-rate schedule. Increasing batch size reduces the number of optimizer steps but does not rescue high-rank LoRA under our fixed protocol. Structured adapters, especially QuanTA $d = 3$, remain trainable in the same larger-batch regimes.

| **Method** | **Config** | **BS=64 674 steps** | **BS=128 337 steps** | **BS=256 168 steps** |
|---|---|---|---|---|
| LoRA | $r = 8$ | 78.1% | 64.0% | 20.3% |
| LoRA | $r = 64$ | 73.1% | 8.7% | 6.0% |
| LoRA | $r = 128$ | 4.7% | 4.1% | 3.3% |
| LoRA | $r = 256$ | 2.3% | 3.2% | 3.5% |
| DoRA | $r = 16$ | 84.0% | 67.6% | 22.5% |
| QuanTA | $d = 3$ | 84.0% | 72.2% | 65.8% |

Figure 1 visualizes this phase boundary for vanilla LoRA. At batch size 64, both $r = 8$ and $r = 64$ remain functional, while $r = 128$ and $r = 256$ collapse. At batch sizes 128 and 256, the safe region contracts: $r = 8$ remains trainable, but $r = 64$ joins the near-floor regime. This shows that the relevant object is not nominal rank alone, but the interaction between rank, batch size, optimization horizon, and the geometry of the update manifold.

Train-time diagnostics at batch size 256 further distinguish slow learning from non-ignition (Appendix D). LoRA $r = 8$ and DoRA $r = 16$ improve steadily but remain update-limited after 168 optimizer steps, whereas LoRA $r = 64$, $r = 128$, and $r = 256$ remain near floor throughout training. QuanTA $d = 3$ enters the learning regime early and reaches 65.8%, providing additional evidence that structure widens the stable optimization region in cold-start GRPO.

These results also refine the comparison to large-batch LoRA reports. Schulman and Lab (2025) report no rank collapse at $r = 128$ using the same base model. Our ablation shows that simply increasing batch size is not, by itself, a generic stabilizer for unconstrained high-rank LoRA under a one-pass cold-start GRPO protocol. Larger batches reduce per-update sampling noise, but they also reduce the number of optimizer updates and alter the temporal structure of exploration. We therefore interpret the discrepancy as evidence that rank collapse depends on a rank–batch–optimization boundary, not as a contradiction over a single universal rank threshold.

### 4.3 Main Comparison: LoRA, DoRA, QuanTA, and Full Fine-Tuning

Table 6 presents results for all primary methods on Qwen 3 8B (seed 42); multi-seed DeepMath-Hard and Skywork-Hard results are in Appendix E.

Three findings stand out. **(i)** Both QuanTA $d = 3$ and DoRA $r = 16$ reach **84.0%** on DeepMath-Hard, outperforming FFT by 6.7 percentage points and LoRA $r = 8$ by 5.9 points—with QuanTA using 22% fewer parameters than DoRA (35.3M vs. 45.0M) and leading on Skywork-Hard (19.4% vs. 17.0%), where the inversion of the DoRA–QuanTA ordering is consistent with QuanTA's higher effective rank providing a clearer advantage as task difficulty increases. That two structurally distinct adapters converge to identical

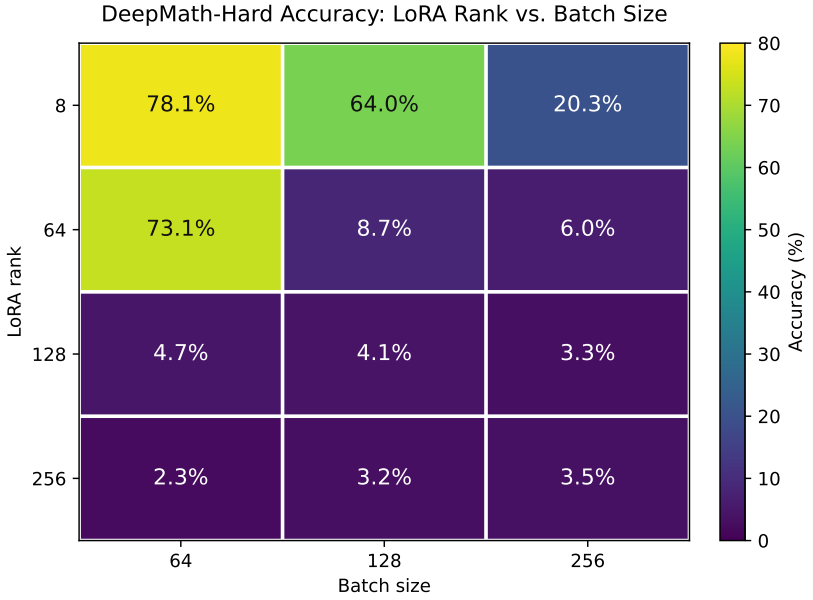


Figure 1: **LoRA rank–batch phase boundary on DeepMath-Hard.** All runs use Qwen 3 8B, seed 42, one epoch, and the same learning-rate schedule. Larger batch sizes reduce optimizer steps from 674 to 337 and 168. Increasing batch size does not rescue high-rank LoRA under our fixed protocol: $r = 128$ and $r = 256$ remain near floor, and $r = 64$ fails at batch sizes 128 and 256 despite being functional at batch size 64.

Table 6: **Main results on Qwen 3 8B (seed 42).** †QuanTA $d = 3$ uses the `no_kv` target-module configuration; see Appendix G for full ablation. Best result per benchmark is **bolded**.

| Method | Config | Params | MATH | DeepMath-Hard | Skywork-Hard |
|---|---|---|---|---|---|
| LoRA | $r = 8$ | 22M | **78.9%** | 78.1% | 19.0% |
| LoRA | $r = 16$ | 43.6M | 78.1% | 76.2% | 15.2% |
| DoRA | $r = 8$ | 23.2M | 78.1% | 81.9% | 16.9% |
| DoRA | $r = 16$ | 45.0M | 75.6% | **84.0%** | 17.0% |
| QuanTA† | $d = 3$ | 35.3M | **78.9%** | **84.0%** | **19.4%** |
| FFT | — | ∼8B | **78.9%** | 77.3% | 15.9% |

performance on DeepMath-Hard is itself evidence that the operative variable is the structural constraint they share, not any property specific to either parameterization. **(ii)** FFT scores 77.3% on DeepMath-Hard, below both structured adapters and LoRA $r = 8$ despite ∼200× more trainable parameters; the spectral explanation is in Section 4.5. On MATH, all methods saturate near 78.9% with the exception of DoRA $r = 16$ (75.6%); given that multi-seed results are unavailable for this configuration on MATH, we do not draw conclusions from this single-seed anomaly. Within vanilla LoRA, $r = 8$ outperforms $r = 16$ across both MATH and DeepMath-Hard in this seed, suggesting that the instability associated with higher rank may begin to affect performance even below the catastrophic collapse threshold. **(iii)** The batch-size ablation shows that this advantage is not limited to the original batch-size setting. At batch sizes 128 and 256, the strongest structured configuration, QuanTA $d = 3$, remains trainable and substantially outperforms vanilla LoRA at higher ranks. At batch size 256, QuanTA $d = 3$ reaches 65.8% after only 168 optimizer steps, while LoRA $r = 64$, $r = 128$, and $r = 256$ remain near floor. This supports the central claim that structure widens the stable optimization region in cold-start GRPO.

Multi-seed averages (seeds 42–44) show meaningful run-to-run variance: DoRA $r = 16$ achieves $74.4\% \pm 10.9\%$ and QuanTA $d = 3$ no_kv achieves $75.1\% \pm 9.4\%$ on DeepMath-Hard. We therefore do not claim a statistically

clean ordering between the two structured adapters. The primary multi-seed signal is that structured adapters remain competitive with, and on average stronger than, LoRA and FFT on the primary DeepMath-Hard comparison (Appendix E). Out-of-distribution generalization to AIME weakly follows the same trend, although the small absolute differences make this evidence suggestive rather than conclusive; full results are deferred to Appendix H.

### 4.4 The Model Maturity Hypothesis

Table 1 presents results across all three base models. Both ∼15T models score near floor on DeepMath-Hard under any adapter configuration; Qwen 3 8B reaches 78–84%. The behavioral contrast extends beyond absolute accuracy: on MATH, Llama 3.1 8B exhibits seed-level instability under QuanTA $d = 4$ absent from LoRA—seed 43 collapses entirely (0.8%), while LoRA $r = 16$ is stable across all three seeds (30.2% ± 3.7%); full multi-seed results are in Appendix E. Qwen 3 8B shows no such instability under any structured adapter on any benchmark.

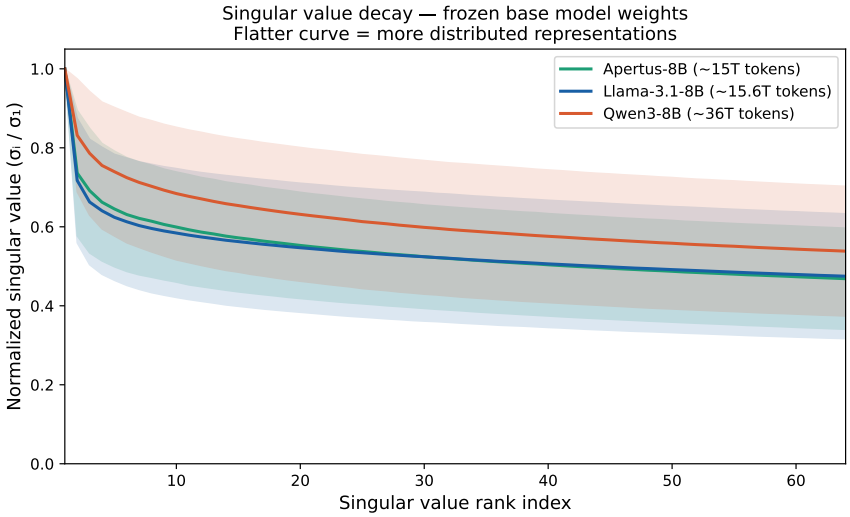


Figure 2: **Singular value decay of frozen base model weights, averaged across sampled linear layers using the top 64 singular values per layer.** Shaded regions denote one standard deviation across layers. The two ∼15T models decay steeply and overlap closely despite differing architectures, while Qwen 3 8B has a markedly flatter curve. This frozen-weight spectral difference is consistent with the observed adaptation behavior: the model with flatter spectra is also the model where structured high-rank adaptation provides the clearest advantage.

Figure 2 offers a structural correlate that precedes any fine-tuning. The singular value decay of frozen linear layers reveals a monotonic ordering: both ∼15T models decay steeply and nearly identically, while Qwen 3 8B's curve sits markedly flatter throughout. This pattern is consistent with pre-training data volume being an important driver, although architecture, tokenizer, and data mixture remain confounded. We hypothesize that flatter singular value spectra indicate more distributed pre-trained representations, making adaptation benefit from updates spread across many singular directions. This interpretation is consistent with QuanTA's advantage on Qwen 3 8B, but controlled validation with matched architectures and intermediate pre-training checkpoints is needed to establish causality. More details can be found in Appendix I.

### 4.5 Spectral Analysis: Why Structure Beats Scale

Figure 3 reports the mean effective rank (participation ratio $\rho = \exp(H(\mathbf{p}))$, where $\mathbf{p}$ is the distribution over squared singular values of $\Delta W$ and $H$ is Shannon entropy) averaged across all target layers for every method.

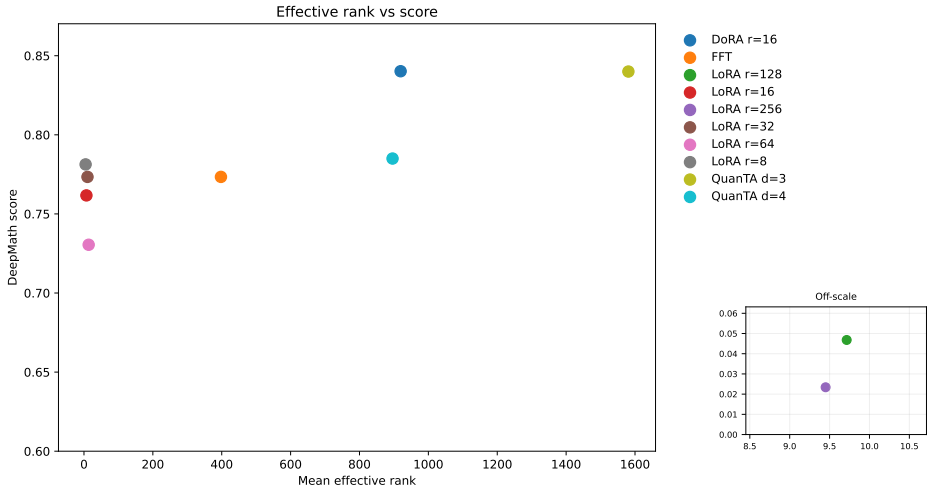


Figure 3: **Mean effective rank versus DeepMath-Hard accuracy (Qwen 3 8B, seed 42).** Performance tracks structured high-rank adaptation rather than trainable parameter count. DoRA and QuanTA achieve higher effective rank via structurally distinct mechanisms yet converge to identical accuracy, supporting effective rank as the operative variable rather than any specific parameterization.

Three findings emerge. (i) Rank collapse is visible at the weight level: collapsed LoRA runs ($r \geq 128$) achieve near-zero effective rank despite high nominal rank, indicating that update energy is concentrated into a handful of singular directions. (ii) Parameter count alone does not explain effective rank: FFT updates all ∼8B parameters but achieves mean effective rank around 400, below DoRA $r = 16$ (∼900) and QuanTA $d = 3$ (∼1580), each using less than 0.6% of the parameters. (iii) Structural constraint is the common feature of the strongest adapters: DoRA's magnitude-direction decomposition and QuanTA's MPO tensor network are mechanistically distinct, yet both produce substantially higher effective rank than vanilla LoRA and FFT in this setting. These results support effective rank, induced by structural bias, as a central explanatory variable in RLVR adaptation, while leaving open whether causal intervention on singular directions would reproduce the same behavior.

### 4.6 Domain Generalization: Enigmata

To assess whether the advantage of structured adaptation extends beyond mathematics, we train all primary methods on Enigmata 10K under the same GRPO protocol and evaluate on the held-out Enigmata evaluation set (Table 7).

Table 7: **Enigmata results (Qwen 3 8B).** Models trained on Enigmata 10K and evaluated on the disjoint Enigmata evaluation set (pass@1).

| Method | Config | Enigmata 10K (train-time) | Enigmata Eval (pass@1) |
|---|---|---|---|
| LoRA | $r = 8$ | 3.3% | 11.8% |
| DoRA | $r = 16$ | 7.7% | 16.6% |
| FFT | — | 9.8% | 19.1% |
| QuanTA | $d = 3$ (no_kv) | 12.7% | **20.6%** |

The held-out ordering—QuanTA (20.6%) > FFT (19.1%) > DoRA (16.6%) > LoRA (11.8%)—is consistent across both training-time and evaluation signals, suggesting that the structured-adapter advantage can extend beyond mathematical reasoning. QuanTA's advantage over DoRA is larger here (4.0pp) than on DeepMath-Hard (0.0pp at seed 42), consistent with the prediction that higher effective rank provides a clearer advantage when the task ceiling is less binding. The intermediate FFT result (19.1%) falls between

the two structured adapters; given that Enigmata results reflect a single seed on a novel and difficult task, we are cautious about interpreting this ordering as meaningful—multi-seed validation would be required to distinguish a genuine effect from the run-to-run variance documented on comparably difficult benchmarks (Appendix E).

## 5 Discussion and Limitations

**Why RL optimization may cause rank collapse.** Sparse binary rewards normalized within groups of $G = 8$ completions produce high-variance advantage estimates. Our spectral diagnostics suggest that, in this setting, unconstrained high-rank adapters can concentrate update energy into a small number of singular directions rather than distributing it across the available rank. This provides a plausible explanation for why increasing LoRA rank eventually becomes harmful under cold-start GRPO. Structural constraints—whether DoRA's magnitude-direction decomposition or QuanTA's MPO tensor network—appear to regularize which high-rank solutions are reachable by gradient descent. This interpretation unifies the three spectral observations in Section 4.5: collapsed LoRA runs have low effective rank, FFT has lower effective rank than structured adapters despite far more trainable parameters, and two distinct structured parameterizations both produce higher effective rank and stronger primary-benchmark performance. We view this as mechanistic evidence from spectral diagnostics, not as a causal intervention on the update subspace.

The batch-size ablation refines this explanation. Larger batches reduce per-update sampling noise, but under a fixed one-epoch GRPO protocol they also reduce the number of optimizer updates and alter the temporal structure of exploration. Empirically, simply increasing global batch size from 64 to 128 or 256 does not rescue high-rank LoRA under our fixed protocol: $r = 128$ and $r = 256$ remain near floor, and $r = 64$ fails to enter the learning regime at larger batches. This is not generic undertraining, since LoRA $r = 8$, DoRA $r = 16$, and QuanTA $d = 3$ remain trainable under the same larger-batch regimes. We therefore interpret rank collapse as a rank–batch–optimization stability boundary shaped by update geometry, rather than as a fixed rank threshold or a pure small-batch artifact. This also sharpens the comparison to Schulman and Lab (2025): the absence of collapse in their setting likely reflects a different point in the joint space of batch size, update horizon, learning-rate schedule, and optimization protocol, rather than a contradiction over a universal LoRA rank cutoff.

**Model Maturity Hypothesis: evidence and limits.** Two lines of evidence support the hypothesis. Behaviorally, Apertus and Llama 3.1—architecturally independent models with similar reported pre-training data volume—produce similar signatures: near-floor DeepMath-Hard accuracy and seed-level instability under QuanTA. Spectrally, frozen-weight singular value decay reveals the ordering Apertus $\approx$ Llama 3.1 $<$ Qwen 3 before any fine-tuning updates are applied (Appendix I), matching the qualitative partition observed after adaptation. This reduces the likelihood that the pattern is purely family-specific, but it does not causally isolate pre-training scale from architecture, tokenizer, or data mixture. The hypothesis therefore warrants controlled validation using matched architectures and intermediate pre-training checkpoints.

**Practical takeaways.** (1) **Replace the FFT-vs-LoRA binary** with the question of which structured adapter suits the model maturity, task difficulty, and optimization budget. (2) **Avoid unconstrained high-rank LoRA** in cold-start GRPO: at batch size 64, collapse appears at $r \geq 128$, while under the same one-epoch protocol at batch sizes 128 and 256 even $r = 64$ falls to near-floor performance. (3) **Do not assume larger batches rescue high-rank LoRA in one-epoch cold-start GRPO.** In our fixed-pass ablation, larger batches leave high-rank LoRA collapsed while structured adapters remain trainable, with QuanTA $d = 3$ retaining 65.8% at batch size 256 after only 168 optimizer steps. (4) **Cosine scheduling with warmup is recommended** at batch size 64; constant schedules cause immediate policy collapse regardless of adapter choice.

**Limitations.**

- **Scale and architecture.** All experiments are at 8B dense transformer scale; MoE architectures and 70B+ models remain untested.

- **Maturity confounds.** Pre-training scale co-varies with architecture, tokenizer, and data mixture; the three-model comparison cannot disentangle these effects, and the hypothesis remains correlational.

- **Batch-size and update-budget boundary.** We find that increasing global batch size from 64 to 128 or 256 does not rescue high-rank LoRA under the same one-epoch GRPO protocol. LoRA $r = 128$ and $r = 256$ remain near floor across all tested batch sizes, and $r = 64$ fails to enter the learning regime at larger batches. Because larger batches also reduce the number of optimizer steps per epoch, we interpret this as a fixed-pass practical result rather than a claim that batch size alone causes failure. Disentangling batch size, optimizer-update count, learning-rate scaling, and total sampled rollouts remains future work.

- **Single-seed batch-size ablation.** The batch-size ablation is run at seed 42. Its effect sizes are large and the conclusions are supported by matched-regime controls—LoRA $r = 8$, DoRA $r = 16$, and QuanTA $d = 3$ remain trainable where high-rank LoRA fails—but multi-seed replication would be required to estimate the precise variance of the rank–batch boundary.

## 6 Conclusion

The field has long framed the choice between Full Fine-Tuning and LoRA as a tradeoff between performance and compute, but our results suggest this framing misidentifies the operative variable. What determines the quality of a learned reasoning policy in RLVR is not how many parameters are updated, but what structure is imposed over the update manifold: unconstrained optimization—whether via high-rank LoRA or full fine-tuning—does not produce maximally expressive updates under sparse rewards, it produces degenerate ones. The batch-size ablation strengthens this conclusion: increasing global batch size under the same one-pass GRPO protocol does not rescue high-rank LoRA, and the observed safe-rank boundary moves downward rather than disappearing. At the same larger-batch regimes, low-rank and structured controls remain trainable, showing that the failure is not simply fewer optimizer steps but a rank–batch–optimization instability specific to unconstrained LoRA.

That two structurally distinct adapters exhibit higher effective rank and strong primary-benchmark performance suggests that the relevant property is not the specific tensor decomposition, but the presence of an inductive bias that restricts which high-rank updates gradient descent can reach. This reframing does not eliminate the need to choose between FFT, LoRA, and structured adapters; instead, it makes the choice conditional on the model, task, and optimization budget.

Our evidence also suggests that this choice may depend on base-model maturity. In the current three-model comparison, the two ∼15T-token models are better explained by low-rank regularization, while the 36T-token Qwen 3 8B model shows the clearest benefit from structured high-rank adaptation. Because pre-training scale remains confounded with architecture, tokenizer, and data mixture, we present this as a falsifiable hypothesis rather than a causal law. The spectral diagnostic framework introduced here provides a practical way to test this hypothesis on future models before running a full RLVR sweep. **Anonymized code can be found** `here`.

## References

Armen Aghajanyan, Luke Zettlemoyer, and Sonal Gupta. Intrinsic dimensionality explains the effectiveness of language model fine-tuning, 2020. URL `https://arxiv.org/abs/2012.13255`.

Paul Albert, Frederic Z. Zhang, Hemanth Saratchandran, Cristian Rodriguez-Opazo, Anton van den Hengel, and Ehsan Abbasnejad. Randlora: Full-rank parameter-efficient fine-tuning of large models. *arXiv preprint arXiv:2502.00987*, 2025.

Art of Problem Solving. 2025 AIME I problems. `https://artofproblemsolving.com/wiki/index.php/2025_AIME_I_Problems`, 2025a. AoPS Wiki. American Invitational Mathematics Examination I problems and answer key.

Art of Problem Solving. 2025 AIME II problems. https://artofproblemsolving.com/wiki/index.php/2025_AIME_II_Problems, 2025b. AoPS Wiki. American Invitational Mathematics Examination II problems and answer key.

Art of Problem Solving. 2026 AIME I problems. https://artofproblemsolving.com/wiki/index.php/2026_AIME_I_Problems, 2026a. AoPS Wiki. American Invitational Mathematics Examination I problems and answer key.

Art of Problem Solving. 2026 AIME II problems. https://artofproblemsolving.com/wiki/index.php/2026_AIME_II_Problems, 2026b. AoPS Wiki. American Invitational Mathematics Examination II problems and answer key.

Jiangjie Chen, Qianyu He, Siyu Yuan, Aili Chen, Zhicheng Cai, Weinan Dai, Hongli Yu, Qiying Yu, Xuefeng Li, Jiaze Chen, Hao Zhou, and Mingxuan Wang. Enigmata: Scaling logical reasoning in large language models with synthetic verifiable puzzles, 2025. URL https://arxiv.org/abs/2505.19914.

Zhuo Chen, Rumen Dangovski, Charlotte Loh, Owen Dugan, Di Luo, and Marin Soljačić. QuanTA: Efficient high-rank fine-tuning of llms with quantum-informed tensor adaptation. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 92210–92245. Curran Associates, Inc., 2024. doi: 10.52202/079017-2928. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/a7c17115db36193f6b83b71b0fe1d416-Paper-Conference.pdf.

DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

Zhiwei He, Yuehua Chen, Ting Liang, et al. Deepmath-103k: A large-scale, challenging, decontaminated, and verifiable mathematical dataset for advancing reasoning, 2025. URL https://arxiv.org/abs/2504.11456.

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. *CoRR*, abs/2103.03874, 2021. URL https://arxiv.org/abs/2103.03874.

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, 2021. URL https://arxiv.org/abs/2106.09685.

Ting Jiang, Shaohan Huang, Shengyue Luo, et al. MoRA: High-rank updating for parameter-efficient fine-tuning, 2024. URL https://arxiv.org/abs/2405.12130.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention, 2023. URL https://arxiv.org/abs/2309.06180.

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, pages 32100–32121. PMLR, 2024. URL https://proceedings.mlr.press/v235/liu24bn.html.

Weiyang Liu, Zeju Qiu, Yao Feng, et al. Parameter-efficient orthogonal finetuning via butterfly factorization, 2023. URL https://arxiv.org/abs/2311.06243.

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, 2025. URL https://arxiv.org/abs/2503.20783.

Benjamin Pikus, Pratyush Ranjan Tiwari, and Burton Ye. Hard examples are all you need: Maximizing grpo post-training under annotation budgets, 2025. URL https://arxiv.org/abs/2508.14094.

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

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

Liang Zeng, Liangjun Zhong, Liang Zhao, et al. Skywork-Math: data scaling laws for mathematical reasoning in large language models — the story goes on, 2024. URL https://arxiv.org/abs/2407.08348.

# A Implementation Details

To ensure complete reproducibility of our "Cold Start" RLVR experiments, we detail the exact computational environment, hyperparameters, and reward definitions used.

## A.1 Infrastructure and Environment

All experiments were conducted on 4xH100 node. We used the Axolotl training framework with the GRPO trainer from TRL integrated into Axolotl. To maintain consistency across runs, we utilized a custom Singularity container. The precise hardware and software stack specifications are listed in Table 8.

Table 8: **Computational Infrastructure.**

| Component | Specification |
|---|---|
| **Compute Hardware** | 4× NVIDIA H100 (64GB) node |
| **Operating System** | Ubuntu 24.04 LTS (via Singularity) |
| **CUDA Driver** | Version 12.8.0 |
| **Precision** | `bfloat16` (Training & Inference) |
| **Deep Learning Framework** | PyTorch 2.9.0 (with Custom `torchao` 0.13.0) |
| **Training Orchestration** | Axolotl 0.13.0 |
| **Inference Engine** | vLLM 0.11.1 (PagedAttention enabled) |
| **Acceleration Kernels** | FlashAttention 2.8.3, Xformers 0.0.33 |
| **Distributed Strategy** | PyTorch DDP (DistributedDataParallel) |

## A.2 Hyperparameter Configuration

We utilized the Axolotl training framework with GRPO trainer by TRL integrated into Axolotl. Table 9 details the global optimization parameters held constant across runs to ensure fair comparison. Table 10 details the specific configurations for the adaptation methods and task-specific adjustments.

Table 9: **Global RL training hyperparameters for the main protocol.** Unless otherwise stated, experiments use global batch size 64. The batch-size ablation varies global batch size over $\{128, 256\}$ while keeping all other entries fixed.

| Parameter | Value |
|---|---|
| **RL Algorithm** | Group Relative Policy Optimization (GRPO) |
| **Loss Function** | Dr. GRPO (Token-level unbiased) |
| **Group Size ($G$)** | 8 generations per prompt |
| **KL Coefficient ($\beta$)** | 0.0 (Pure Outcome-Based RL) |
| **Reward Scaling** | `False` (Raw scores used) |
| **Global Batch Size** | 64; $\{128, 256\}$ in batch-size ablation |
| **Optimizer** | AdamW (Fused) |
| **Weight Decay** | 0.0 |
| **Scheduler** | Cosine Decay |
| **Warmup Ratio** | 0.1 (10% of steps) |
| **Min LR Ratio** | 0.15 |
| **Num Epochs** | 1 |

## A.3 Reward Definitions and Verification

In the "Cold Start" regime, the model must simultaneously learn formatting (XML tags) and reasoning logic. We utilized a composite reward function $R_{total} = R_{format} + R_{accuracy}$ with equal weighting ($w = 1.0$).

Table 10: **Method-Specific and Task-Specific Configurations.** We highlight the drastic difference in parameter count between LoRA and QuanTA.

| Configuration | LoRA / DoRA ($r = 16$) | QuanTA ($d = 4$) | QuanTA ($d = 3$) |
|---|---|---|---|
| **Rank / Dims** | $r = 16, \alpha = 32$ | $d = 4$, FEATS $= [16, 8, 8, 4]$ | $d = 3$, FEATS $= [16, 16, 16]$ |
| **Target Modules** | ALL LINEAR | ALL LINEAR | ALL LINEAR |
| **Dropout** | 0.0 | 0.0 | 0.0 |
| **Bias** | NONE | NONE | NONE |
| **Trainable Params (LLAMA 3.1 8B)** | 41.9M | 6.9M | N/A |
| **Trainable Params (Qwen 3 8B)** | 43.6M | 7.7M | 35.3M |
| *TASK-SPECIFIC LEARNING RATES & CONTEXT* | | | |
| **MATH / DeepMath / Skywork** | LR: $1 \times 10^{-5}$ \| MAX LENGTH: 8,192 | | |

### A.3.1 Accuracy Reward ($R_{accuracy}$)

To evaluate correctness, we strip all formatting tags (e.g., `<think>`) and extract the final answer using a hierarchical strategy:

1. **Robust Box Extraction:** We scan for the **last** occurrence of `\boxed{...}`, utilizing depth-aware parsing to correctly capture mathematical expressions containing nested braces.

2. **Semantic Verification:** We utilize the `math_verify` library to compare the extracted answer against the ground truth. This handles symbolic equivalences (e.g., $\frac{1}{2} = 0.5$, $x + y = y + x$) that strict string matching would miss.

3. **Score:** Returns **1.0** if semantically equivalent, **0.0** otherwise.

### A.3.2 Format Shaping Reward ($R_{\text{fmt}}$)

To induce Chain-of-Thought reasoning without Supervised Fine-Tuning, we impose structural constraints to penalize degeneration (e.g., empty thoughts, infinite loops). The shaping reward is computed as a weighted sum of satisfied constraints:

$$R_{\text{fmt}}(y) = \sum_i w_i \cdot \mathbb{I}(C_i(y)) \tag{5}$$

**Constraints ($C_i$) and Weights ($w_i$):**

- **Structure ($+0.375$):** The output contains a valid closing `</think>` tag.
- **Answer Existence ($+0.375$):** Non-empty content exists after the reasoning block.
- **Hallucination ($-0.75$):** The model generates simulated user turns (e.g., "User:").
- **Lazy Thinking ($-0.6$):** The content inside `<think>...</think>` is empty or null.
- **Repetition ($-0.5$):** Line-level n-gram repetition exceeds the degeneration threshold.
- **Malformed XML ($-0.3$):** Tags are missing or unclosed.

*Note: The maximum cumulative format bonus ($0.75$) is bounded below the accuracy reward ($1.0$) to prevent reward hacking, ensuring the optimizer prioritizes correctness over mere template compliance.*

# B Data Curation & "Hardcore" Filtering Strategy

## B.1 DeepMath-Hard Construction

We utilized the `DeepMath-103K` dataset (He et al., 2025), which aggregates high-difficulty problems from sources like NuminaMath and MathStackExchange. The original dataset contains a difficulty metadata field ranging from 0.0 to 10.0.

**Filtering Logic.** We applied a strict threshold of $\geq 8.5$, discarding the vast majority of the dataset to focus exclusively on the "long tail" of complexity. As illustrated in Figure 4, this reduces the dataset from $\sim 103,000$ to 5,399 samples.

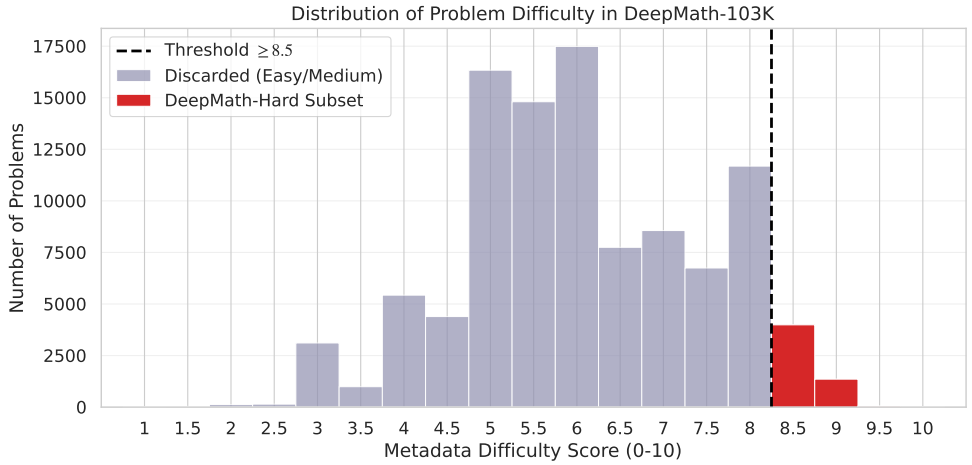


Figure 4: **Difficulty Distribution of DeepMath-103K.** The dashed line represents our cut-off at difficulty score 8.5. We discard the high-volume "easy-medium" mode (grey) to train exclusively on the high-complexity tail (red).

**Impact on Topic Distribution.** Filtering for difficulty implicitly shifts the topic distribution toward abstract mathematical domains. As shown in Figure 5, the "Hardcore" subset (red) is enriched for fields like *Differential Geometry*, *Abstract Algebra (Group/Field Theory)*, and *Topology*, while simpler procedural topics like standard Calculus are de-emphasized compared to the original distribution (purple). This suggests that the filter shifts the training distribution toward problems requiring more abstract mathematical reasoning.

## B.2 Skywork-Hard and the 32B Judge

For the `Skywork-OR1-RL-Data` Zeng et al. (2024), we utilized the `train-math-deepscaler` subset. Unlike DeepMath, this dataset includes metadata on how larger models performed on each specific problem.

**The 32B Judge Protocol.** We utilized the `model_difficulty` metadata field for the `DeepSeek-R1-Distill-Qwen-32B` model. This field reports a score inversely proportional to the pass rate. We selected a cutoff score of $\geq 13$, in the Skywork metadata schema, higher scores indicate higher failure rates for the 32B model.

**Distributional Cutoff.** Figure 6 visualizes this exact cutoff. The filtering removes the massive volume of "trivial" problems and isolates a cluster of 6,702 problems on which the 32B judge model has a low reported pass rate.

## B.3 Enigmata 10k Train and Eval

We construct a reduced Enigmata training set from the original Enigmata-Data training split released by BytedTsinghua-SIA. The full training split contains 217,541 examples across 38 task names, with a highly

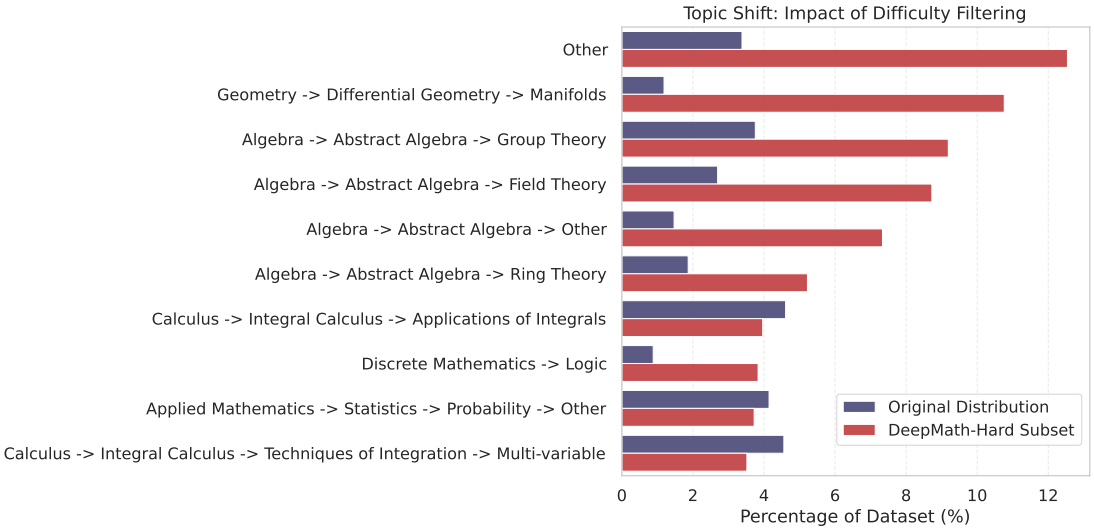


Figure 5: **Topic Shift Analysis.** Filtering for difficulty ($\geq 8.5$) fundamentally alters the dataset composition, enriching for abstract reasoning domains like Manifolds and Field Theory while suppressing procedural calculus.

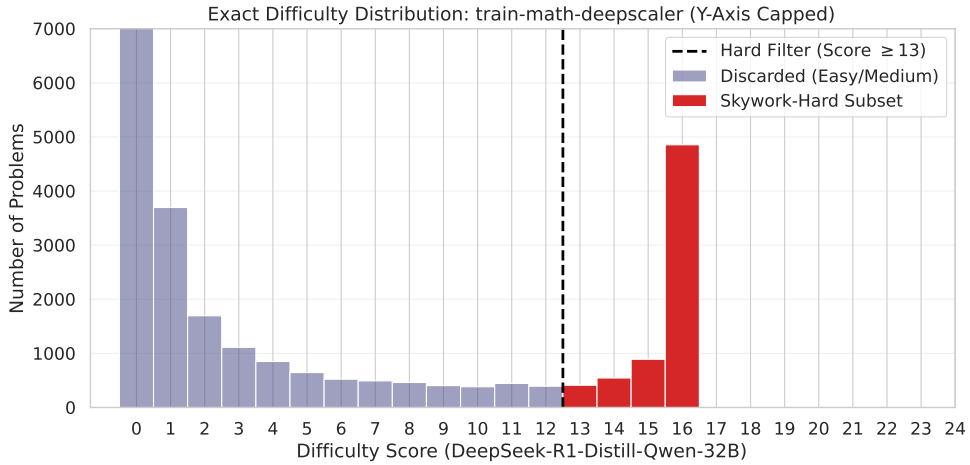


Figure 6: **The 32B Judge Filter on Skywork-Math.** The y-axis is capped to visualize the tail. The vast majority of problems (Scores 0-12) are discarded. We retain only the subset (red) where the 32B judge model has a low reported pass rate (Score $\geq 13$).

non-uniform task distribution. Most tasks contain 6,000 examples each, but several deviate substantially from this mode: `full_crosswords` contains 19,000 examples, `arc_agi` 11,116, `pattern_recognition` 9,000, `hitori`, `light_up`, `natural_language_navigation`, and `tic_tac_toe` 4,000 each, `knights_and_knaves` 3,414, `checkmate_in_one` 2,500, `big_bench_symbolic` 2,000, `hamiltonian_path` 1,095, and `symbolic_hard` 416. Difficulty labels are similarly imbalanced: among rows with valid difficulty annotations, the original training split contains 56,477 easy, 77,451 medium, and 83,197 hard examples.

**Shared-task restriction.** To ensure that training and evaluation measure performance over the same task family, we first restrict the training pool to tasks shared between the Enigmata training split and the official Enigmata evaluation split. This removes the train-only tasks `arc_agi`, `pattern_recognition`, and

`symbol_pattern`. Conversely, the eval-only task `FOLIO` cannot be included because it is absent from the training split. The resulting candidate pool therefore consists of the 35 task names common to both training and evaluation.

**Stratified 10k construction.** From this shared-task pool, we sample 10,000 examples using task-by-difficulty strata. The sampling procedure enforces two constraints. First, the marginal task distribution of the reduced set matches the relative task frequencies of the original Enigmata training split after restricting to shared tasks. Second, the marginal difficulty distribution matches that of the official evaluation set, rather than the more imbalanced difficulty distribution of the original training split. Concretely, the final Enigmata 10k subset contains 2,930 easy, 3,381 medium, and 3,689 hard examples.

This construction preserves the original train-task mixture as closely as possible while aligning the reduced training set with the evaluation set's difficulty profile. As a result, the Enigmata experiments test generalization over the same task families present at evaluation time without allowing train-only tasks to distort the reduced training distribution. All Enigmata models are trained on this 10k subset and evaluated on the disjoint official Enigmata evaluation set.

## C Full Rank Sweep Results

### C.1 LoRA Rank Sweep with Spectral Diagnostics

Table 11 extends the main-body rank sweep (Table 4) with mean effective rank measurements for each configuration, enabling direct comparison between nominal and effective rank.

Table 11: **Full LoRA rank sweep on Qwen 3 8B, DeepMath-Hard (seed 42).** Mean effective rank is the participation ratio $\rho = \exp(H(\mathbf{p}))$ averaged across all target modules. Note the divergence between nominal and effective rank at $r \geq 128$: despite 128 and 256 nominal singular directions, the optimizer concentrates nearly all update energy into fewer than 10 effective directions.

| Method | Nominal Rank | Trainable Params | Mean Eff. Rank | DeepMath-Hard |
|---|---|---|---|---|
| LoRA | $r = 8$ | 22M | 4.7 | 78.1% |
| LoRA | $r = 16$ | 43.6M | 7.1 | 76.2% |
| LoRA | $r = 32$ | 87.2M | 10.2 | 77.3% |
| LoRA | $r = 64$ | 174M | 13.6 | 73.1% |
| LoRA | $r = 128$ | 348M | 9.7 | 4.7% |
| LoRA | $r = 256$ | 696M | 9.5 | 2.3% |

Two aspects of Table 11 are worth emphasizing. First, effective rank increases modestly but consistently from $r = 8$ to $r = 64$ ($4.7 \rightarrow 13.6$), yet performance is non-monotonic over this range. This shows that effective rank alone does not determine policy quality; the structure of the update manifold also matters. Second, collapsed runs ($r = 128$, $r = 256$) achieve lower effective rank than the best non-collapsed configurations despite their higher nominal rank. The optimizer therefore concentrates update energy into a small number of singular directions rather than using the available rank, matching the spectral signature discussed in Section 5.

### C.2 DoRA Rank Sweep

Table 12 presents the full DoRA rank sweep on Qwen 3 8B. Unlike LoRA, DoRA does not exhibit catastrophic collapse at any rank tested: performance peaks at $r = 16$ (84.0%) and degrades gracefully to 75.0% at $r = 64$, with no discontinuous cliff.

The asymmetry between LoRA and DoRA at high rank is mechanistically informative. DoRA's decomposition of each weight matrix into magnitude and direction components introduces a structural constraint absent from vanilla LoRA. Under our protocol, this constraint is associated with graceful degradation rather than

Table 12: **DoRA rank sweep on Qwen 3 8B, DeepMath-Hard (seed 42).** Graceful degradation at high rank contrasts sharply with LoRA's catastrophic collapse, suggesting that DoRA's magnitude-direction decomposition acts as a structural regularizer under this RLVR protocol.

| Method | Nominal Rank | Trainable Params | DeepMath-Hard |
|---|---|---|---|
| DoRA | $r = 8$ | 23.2M | 81.9% |
| DoRA | $r = 16$ | 45.0M | **84.0%** |
| DoRA | $r = 32$ | 90.0M | 80.9% |
| DoRA | $r = 64$ | 180M | 75.0% |

catastrophic collapse. We interpret this as evidence that DoRA's parameterization regularizes the update manifold and reduces the tendency toward singular-direction concentration observed in high-rank LoRA.

### C.3 Efficiency Summary

QuanTA $d = 3$'s speed advantage over both $d = 4$ and DoRA despite its larger parameter count reflects the greater parallelism of the $[16, 16, 16]$ MPO tensor contraction relative to DoRA's per-column normalization overhead.

Table 13: **Efficiency comparison, Qwen 3 8B on DeepMath-Hard (seed 42).** Training time reflects a full single-epoch run on 4×H100. In this seed-42 comparison, QuanTA $d = 3$ is the strongest efficiency point among the tested configurations: it matches DoRA $r = 16$ on DeepMath-Hard while using fewer trainable parameters and less wall-clock time.

| Method | Config | Trainable Params | Training Time | DeepMath-Hard |
|---|---|---|---|---|
| LoRA | $r = 8$ | 22.0M | 13.1h | 78.1% |
| LoRA | $r = 16$ | 43.6M | 12.4h | 76.2% |
| DoRA | $r = 16$ | 45.0M | 15.6h | 84.0% |
| QuanTA | $d = 4$ | 7.7M | 12.9h | 78.5% |
| QuanTA | $d = 3$ | 35.3M | 7.5h | 84.0% |

## D Batch-Size 256 Train-Time Dynamics

Figure 7 reports train-time verifier accuracy curves for the batch-size-256 runs from Section 4.2. This setting is intentionally update-limited: under the one-epoch DeepMath-Hard protocol, batch size 256 yields only 168 optimizer steps, compared with 674 steps at batch size 64.

The curves separate slow learning from non-ignition. LoRA $r = 8$ and DoRA $r = 16$ improve steadily throughout training and reach 20.3% and 22.5%, respectively, suggesting that these runs are primarily limited by the reduced number of optimizer updates. In contrast, LoRA $r = 64$, $r = 128$, and $r = 256$ remain near floor for the entire run, indicating failure to enter the learning regime rather than merely slower convergence. QuanTA $d = 3$ exhibits a qualitatively different trajectory: it enters the learning regime early and reaches 65.8% despite the same 168-step budget. This supports the interpretation that structured adapters can widen the stable optimization region in cold-start GRPO, while unconstrained high-rank LoRA can fail to enter the learning regime under the same fixed-pass protocol.

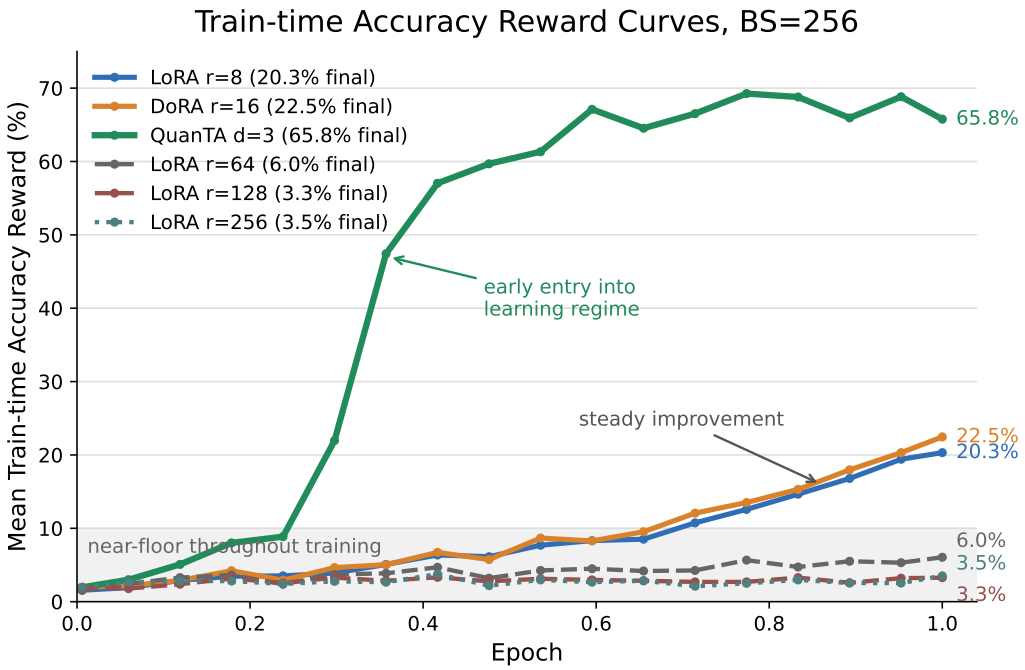


Figure 7: **Train-time verifier accuracy at batch size 256.** All curves are Qwen 3 8B on DeepMath-Hard, seed 42, one epoch. LoRA $r = 8$ and DoRA $r = 16$ show steady improvement but remain update-limited after 168 optimizer steps. QuanTA $d = 3$ enters the learning regime early and reaches 65.8%. High-rank LoRA remains near floor throughout training, indicating non-ignition rather than slow learning.

## E Multi-Seed Results

All results in this appendix use seeds 42, 43, and 44. Standard deviations reflect genuine stochasticity in cold-start GRPO at batch size 64, not measurement noise; single-seed comparisons from the main body should be interpreted in light of the variance reported here.

### E.1 Qwen 3 8B — DeepMath-Hard

Table 14: **Multi-seed DeepMath-Hard results, Qwen 3 8B.** All structured adapter configurations show substantial run-to-run variance, consistent with the difficulty of cold-start GRPO on a highly filtered corpus. No method achieves a statistically clean win on mean accuracy; the primary signal is that the structured-adapter configurations remain in the same performance band and substantially above the collapsed high-rank LoRA runs reported in Appendix C.

| Method | Config | Seed 42 | Seed 43 | Seed 44 | Mean | Std |
|---|---|---|---|---|---|---|
| DoRA | $r = 16$ | 84.0% | 76.6% | 62.5% | 74.4% | 10.9% |
| DoRA | $r = 16$ (no_kv) | 80.9% | 77.0% | 60.9% | 72.9% | 10.4% |
| QuanTA | $d = 3$ (linear) | 78.5% | 79.3% | 64.5% | 74.1% | 8.4% |
| QuanTA | $d = 3$ (no_v_proj) | 80.1% | 78.5% | 63.7% | 74.1% | 9.0% |
| QuanTA | $d = 3$ (no_kv) | 84.0% | 76.2% | 65.2% | 75.1% | 9.6% |

QuanTA $d = 3$ (no_kv) achieves the highest mean (75.1%) and is the Pareto-optimal configuration in the main body results, though its advantage over DoRA $r = 16$ (74.4%) and the other QuanTA variants (74.1%) is well within one standard deviation. Seed 44 is consistently the lowest-performing seed across the

structured-adapter configurations, suggesting that part of the variance is shared across methods rather than being specific to a single parameterization.

### E.2 Qwen 3 8B — Skywork-Hard

Table 15: **Multi-seed Skywork-Hard results, Qwen 3 8B.** High problem difficulty produces substantial reward variance throughout training, which propagates into evaluation variance. Differences between methods are within noise; the primary signal is that all methods remain functional on this benchmark, unlike the catastrophic collapses observed at high LoRA rank.

| Method | Config | Seed 42 | Seed 43 | Seed 44 | Mean | Std |
|---|---|---|---|---|---|---|
| LoRA | $r = 8$ | 19.0% | 13.2% | 21.4% | 17.9% | 4.3% |
| QuanTA | $d = 4$ | 19.6% | 11.2% | 18.5% | 16.4% | 4.6% |
| QuanTA | $d = 3$ | 19.4% | 14.5% | 21.7% | 18.5% | 3.6% |

QuanTA $d = 3$ achieves the highest mean (18.5%) and the lowest standard deviation (3.6%) among the tested configurations, but the differences are small relative to the observed variance. We therefore interpret Skywork-Hard mainly as evidence that all tested non-collapsed configurations remain functional on this harder benchmark.

### E.3 Llama 3.1 8B — MATH

Table 16 shows a sharp seed-level instability for QuanTA $d = 4$ on Llama 3.1 8B: seed 43 collapses to 0.8%, while LoRA $r = 16$ remains stable across all three seeds. We do not discard this run as an outlier, because such instability is informative under the Model Maturity Hypothesis. One interpretation is that, for less mature models, structured high-rank adaptation may expose optimization directions that are not reliably useful under sparse binary rewards, whereas LoRA's lower-rank constraint provides stronger regularization. This remains an interpretation rather than a causal conclusion; more models and seeds would be required to determine whether this is a general property of less mature base models.

Table 16: **Multi-seed MATH results, Llama 3.1 8B.** LoRA $r = 16$ is stable across all seeds; QuanTA $d = 4$ collapses entirely on seed 43 (0.8%), producing high variance and a substantially lower mean. This asymmetry—stable under LoRA, volatile under structured high-rank adaptation—is one behavioral signature motivating the Model Maturity Hypothesis for less mature models.

| Method | Config | Seed 42 | Seed 43 | Seed 44 | Mean | Std |
|---|---|---|---|---|---|---|
| LoRA | $r = 16$ | 28.9% | 34.3% | 27.3% | 30.2% | 3.7% |
| QuanTA | $d = 4$ | 26.6% | 0.8% | 21.9% | 16.4% | 13.7% |

## F Constant Learning Rate Collapse

Figure 8 shows the training reward curve of QuanTA $d = 3$ on Qwen-3 8B trained on MATH under a constant learning-rate schedule, with all other hyperparameters identical to the main experiments. The policy initially learns, peaking near step 200, before collapsing to near-zero reward by step 280. We interpret this failure as arising from the interaction between sparse rewards, batch size 64, and the absence of learning-rate decay. Without decay, the optimizer continues to take large updates late in training, when advantage estimates can become unstable, and the policy fails to recover. A cosine schedule with warmup ratio 0.1 and minimum learning-rate ratio 0.15 avoids this failure mode in our main protocol, so we adopt it as a fixed component rather than a tuned hyperparameter.

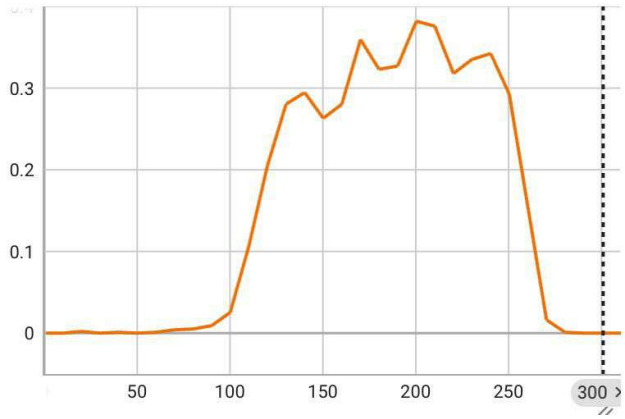


Figure 8: Training reward under a constant learning rate schedule (QuanTA $d = 3$, Qwen-3 8B, MATH dataset). The policy collapses catastrophically near step 280 despite healthy early progress, illustrating why cosine decay with warmup is suggested in cold-start GRPO at batch size 64. The dashed vertical line marks the early stoppage of training.

## G QuanTA Target-Module Ablation

Table 17 reports DeepMath-Hard results for QuanTA across three target-module configurations: `target_linear` (all linear modules), `no_v_proj` (all linear modules except the value projection), and `no_kv` (all linear modules except key and value projections).

Table 17: **QuanTA target-module ablation on Qwen 3 8B, DeepMath-Hard.** Excluding key and value projections consistently produces the best or near-best single-seed performance for $d = 3$, and the best mean across seeds. $d = 4$ multi-seed results are not available; seed-42 values are reported for reference.

| Config | Modules | Seed 42 | Seed 43 | Seed 44 | Mean | Std |
|---|---|---|---|---|---|---|
| $d = 3$ | target_linear | 78.5% | 79.3% | 64.5% | 74.1% | 8.4% |
| $d = 3$ | no_v_proj | 80.1% | 78.5% | 63.7% | 74.1% | 9.0% |
| $d = 3$ | no_kv | **84.0%** | 76.2% | 65.2% | **75.1%** | 9.6% |
| $d = 4$ | target_linear | 79.7% | — | — | — | — |
| $d = 4$ | no_kv | 78.5% | — | — | — | — |

The no_kv configuration for $d = 3$ achieves both the best seed-42 result and the best mean across seeds, although its mean advantage over the other $d = 3$ variants is small relative to the standard deviation. This suggests that excluding key and value projections does not harm performance in this setting and may provide a mild regularizing effect. One possible explanation is that key and value projections encode content-retrieval representations that are already useful from pre-training, so adapting them aggressively during reasoning alignment can be unnecessary or mildly harmful. Under this interpretation, query projections and feed-forward layers provide a more useful locus for reasoning-oriented adaptation in the no_kv configuration. For $d = 4$, the target_linear configuration marginally outperforms no_kv on seed 42 (79.7% vs. 78.5%), but the difference is small and multi-seed validation is unavailable; we therefore avoid making a definitive target-module recommendation for $d = 4$.

## H AIME 2025–2026 Generalization

FFT's near-floor AIME performance (1.35%) requires careful interpretation because it is not straightforwardly explained by in-distribution overfitting. If FFT had simply memorized DeepMath-Hard's surface features, we

Table 18: **AIME 2025–2026 out-of-distribution generalization (Avg@32), Qwen 3 8B.** All models trained on DeepMath-Hard only (∼5,400 samples). Full fine-tuning collapses to near-random performance despite achieving competitive in-distribution accuracy on DeepMath-Hard (77.3%, Table 6).

| Method | Config | AIME Avg@32 |
|---|---|---|
| LoRA | $r = 16$ | 9.95% |
| DoRA | $r = 16$ | **11.04%** |
| QuanTA | $d = 3$ (no_kv) | 10.36% |
| FFT | — | 1.35% |

would expect it to score substantially *higher* than LoRA in-distribution; instead, it scores comparably (77.3% vs. LoRA $r = 8$ at 78.1%), while collapsing far more severely out-of-distribution.

FFT's near-floor AIME performance (1.35%) is notable because the same checkpoint remains competitive in-distribution on DeepMath-Hard (77.3%, Table 6). This suggests that FFT's learned update may transfer poorly to the AIME distribution despite fitting the DeepMath-Hard training/evaluation regime. However, AIME contains only 60 problems in our merged 2025–2026 set, so we treat this result as suggestive rather than definitive.

A possible explanation is the effective-rank pattern documented in Section 4.5. FFT updates all parameters but produces lower effective-rank updates than the structured adapters. Without an explicit structural prior, these updates may become more tied to DeepMath-Hard-specific features, such as topic distribution, phrasing, or solution style, and less robust to AIME's shorter competition-style problems. LoRA, DoRA, and QuanTA all score in a narrow 10–11% range, so the main AIME signal is not a clean ordering among adapters; rather, it is that FFT generalizes unusually poorly in this OOD probe. This is consistent with the broader claim that unconstrained optimization under sparse rewards can produce brittle updates, but additional OOD benchmarks and seeds would be needed to establish the mechanism.

## I Frozen Weight Spectral Analysis

Figure 9 plots the normalized singular value decay of frozen linear layers across all three base models, measured before any fine-tuning occurs. This analysis provides a training-free spectral correlate of the behavioral partition documented in Section 4.4.

Three observations follow from Figure 9.

**The maturity ordering is visible in frozen weights.** Qwen 3 8B's decay curve sits consistently above the two ∼15T models across all 64 singular-value indices, indicating flatter spectra in its pre-trained weights. Because this measurement is taken before fine-tuning, it provides a candidate pre-training diagnostic associated with the later adaptation behavior.

**Apertus and Llama are spectrally similar.** Despite differing in architecture, tokenizer, and training data mixture, the two ∼15T models produce overlapping decay curves with nearly identical means throughout. This reduces the likelihood that the observed behavioral partition is purely family-specific, but it does not rule out all confounds: architecture, tokenizer, data mixture, and pre-training data volume still co-vary in the current model set.

**Proposed mechanism.** We hypothesize that steep singular value decay corresponds to more concentrated pre-trained representations, for which low-rank updates can provide useful regularization. Conversely, flatter spectra may indicate that useful representations are distributed across more directions, making adaptation benefit from updates spread across a broader singular subspace. Under this interpretation, Qwen 3 8B benefits more clearly from structured high-rank adaptation because QuanTA can express distributed updates while still constraining the reachable update manifold. This mechanism is consistent with the observed performance

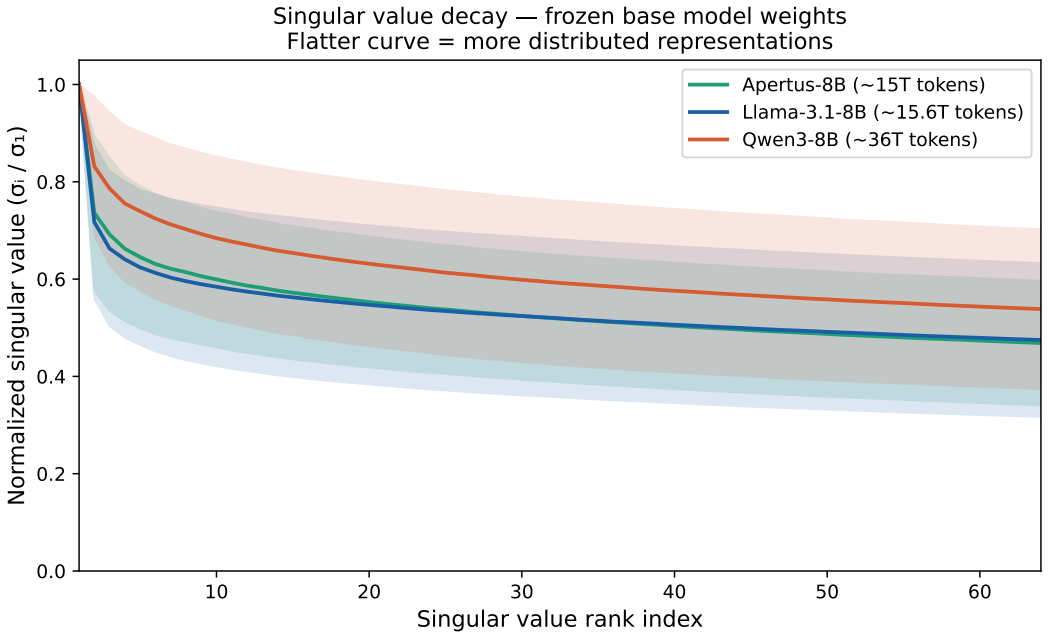


Figure 9: **Singular value decay of frozen base model weights, averaged across sampled linear layers using the top 64 singular values per layer.** Shaded regions denote one standard deviation across layers. Qwen 3 8B (∼36T tokens) exhibits markedly flatter decay than both ∼15T models. Apertus-8B and Llama 3.1-8B produce closely overlapping decay curves despite differing model families. This frozen-weight spectral signature is consistent with the behavioral partition observed in Section 4.4, but does not by itself causally isolate pre-training scale from other model differences.

gap on Qwen 3 8B and the absence of a comparable gap on the two ∼15T models, but controlled experiments with matched architectures and intermediate pre-training checkpoints are needed to establish causality.

## J Qualitative Analysis: An Illustrative Reasoning Failure

To illustrate the kinds of reasoning differences that can accompany the quantitative gaps, we analyze one Chain-of-Thought trace generated by Qwen 3 8B on a complex counting problem from Skywork-Hard. This case study is not intended as standalone evidence for the mechanism; rather, it provides a concrete example of the failure modes that motivate further qualitative analysis.

We compare a LoRA $r = 16$ checkpoint against the QuanTA $d = 3$ checkpoint sampled from the final 5% of the training run.

### J.1 Case Study: Combinatorial Counting

**Problem:** *In an Olympic 100-meter final, there are 10 sprinters competing, among which 4 are Americans. The gold, silver, and bronze medals are awarded to first, second, and third place, respectively. Calculate the number of ways the medals can be awarded if at most two Americans are to receive medals.*

**Correct Answer:** 588.

### J.2 Analysis of the Failure

The LoRA trace retrieves the high-level strategy—case analysis by number of American medalists—but fails in the local combinatorial accounting. In Case 2, it refers to "4 positions" even though there are only three medals. In Case 3, it defaults to a simple multiplication heuristic and fails to account for the dependency

| **LoRA ($r = 16$) Trace [FAILURE]** | **QuanTA ($d = 3$) Trace [SUCCESS]** |
|---|---|
| *... (Case 1: No Americans is correct) ...* | *... (Case 1: No Americans is correct) ...* |
| **Case 2: Exactly one American.**<br>"We have 4 choices for which American receives a medal. For that American, we have 4 choices (since they can get any of the 4 positions). [ERROR: There are only 3 medal positions, not 4] ... The number of ways is $4 \times 4 \times 6 \times 5 = 480$." | **Case 2: Exactly one American.**<br>"In this case, we have to choose which of the three medals the American will receive, then choose which one of the 4 Americans will receive that medal... The number of ways is: $3 \times 4 \times P(6, 2) = 3 \times 4 \times 30 = \mathbf{360}$." |
| **Case 3: Exactly two Americans.**<br>"...For the remaining medal, we have 6 choices. The number of ways to award the medals in this case is: $6 \times 6 \times 6 = 216$." [ERROR: Failed to account for permutations of the specific Americans selected] | **Case 3: Exactly two Americans.**<br>"...Choose which two of the three medals the Americans will receive, then choose which two of the 4 Americans... The number of ways is: $3 \times \binom{4}{2} \times 6 = 3 \times 6 \times 6 = \mathbf{108}$." |
| **Final Calculation:**<br>$120 + 480 + 216 = \mathbf{816}$ | **Final Calculation:**<br>$120 + 360 + 108 = \mathbf{588}$ |

Table 19: **Illustrative reasoning failure.** LoRA (left) identifies the correct case structure but makes local counting errors in Case 2 and Case 3. QuanTA (right) correctly applies the relevant permutation and binomial factors for this example. This single trace illustrates a possible state-tracking failure mode, but should not be read as a statistically representative qualitative evaluation.

between selected Americans and medal slots. These errors are consistent with a loss of state precision during multi-step reasoning.

The QuanTA trace preserves the relevant state variables in this example: which medals are assigned, which Americans are selected, and how many non-American sprinters remain. This is qualitatively consistent with the hypothesis that structured high-rank adaptation can better support distributed state tracking. However, because this appendix presents a single example, we treat it as illustrative rather than conclusive evidence. A larger qualitative analysis would be needed to determine whether this failure mode is systematically more common for LoRA.

## K PEFT Theoretical Comparison

Our spectral analysis suggests two properties that may be useful for strong performance in mature-model RLVR: (a) high effective rank in the learned $\Delta W$, and (b) structural restriction of the update manifold so that unconstrained singular-direction concentration is less likely. We analyze several PEFT methods through this lens. This appendix is theoretical and hypothesis-generating: except for LoRA, DoRA, QuanTA, and FFT, the methods discussed here are not empirically evaluated in our RLVR pipeline.

**LoRA.** Constraining $\Delta W = BA$ to rank at most $r$ bounds the maximum rank, but it does not force the optimizer to use all available directions. In our RLVR experiments, high-rank LoRA checkpoints ($r \geq 128$) concentrate update energy into a small number of singular directions despite large nominal rank. LoRA therefore satisfies neither property in this framework: low-rank LoRA lacks high effective rank, while high-rank LoRA lacks a structural constraint that prevents singular-direction concentration.

**DoRA.** DoRA decomposes each weight into magnitude and direction components, with LoRA applied to the directional component. In our experiments, this parameterization is associated with substantially higher effective rank than vanilla LoRA at comparable nominal rank and avoids the catastrophic collapse observed in high-rank LoRA. We interpret this as evidence that the magnitude-direction decomposition acts as a regularizer on the reachable update manifold, though the precise causal pathway remains to be established.

**QuanTA.** Parameterizing $\Delta W$ as an MPO tensor network constrains updates to a tensor-structured manifold $\mathcal{T}_\chi$ whose elements can be high-rank as matrices. Unlike the LoRA rank-$r$ manifold $\mathcal{M}_r$, this parameterization can distribute update energy across many mode indices while still restricting the set of reachable matrices. In our experiments, QuanTA $d = 3$ achieves the highest measured effective rank among the tested methods and matches DoRA on the primary DeepMath-Hard seed-42 result. This supports the view that structured high-rank adaptation, rather than tensor decomposition specifically, is the important property.

**BOFT.** BOFT restricts updates through butterfly-factored orthogonal transformations. Because orthogonal updates are norm-preserving, this structure may regularize some forms of unstable rescaling. However, the same constraint may also limit the ability to change feature magnitudes, which our spectral interpretation suggests could matter for mature-model RLVR. Since we do not evaluate BOFT empirically, we treat this as a geometric prediction rather than a conclusion.

**MoRA.** MoRA sandwiches a learnable square matrix between fixed non-learnable compression operators, constraining updates to a subspace determined before training. This can in principle produce higher-rank updates than standard LoRA, but the fixed compression structure may be less adaptive to the particular directions needed during cold-start GRPO. Because MoRA is not evaluated in our experiments, we present this as a hypothesis about its geometry rather than a claim about its empirical RLVR performance.

**Summary of the proposed geometric lens.**

| Method | (a) High effective rank expected | (b) Structural restriction expected |
|---|---|---|
| LoRA | × | × |
| DoRA | ✓ | ✓ |
| QuanTA | ✓ | ✓ |
| BOFT | × | ✓ |
| MoRA | ∼ | × |
| FFT | × | × |

The resulting prediction is testable: PEFT methods that combine high effective rank with a structured reachable-update manifold should be more stable than unconstrained high-rank LoRA in mature-model cold-start GRPO. DoRA and QuanTA provide two empirical examples consistent with this prediction, but additional methods such as BOFT and MoRA would need to be evaluated directly before assigning them a definitive performance tier.