MEASURING GRADIENT GEOMETRY IN LANGUAGE-MODEL RLVR WITH SKETCHED GRADIENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning with verifiable rewards (RLVR) has emerged as a dominant paradigm in training large language models to solve a wide range of tasks, including mathematical reasoning. Most RLVR pipelines build on a variant of the standard REINFORCE policy-gradient algorithm. Many methods make claims about the noise of policy-gradient estimators, and about the relative geometry between policy gradients on different prompts: for example, baselines as control variates, and pass@k objectives to shift gradient weight from easier to harder problems. However, to our knowledge, no prior work has directly measured these quantities in language-model RLVR, at scale and across training. We compress per-sequence gradients of online rollouts with CountSketch, a linear operation which allows offline analysis to measure the effect of RLVR objectives, rollout allocations and baselines on policy-gradient estimation. After validating that sketches preserve gradient geometry better than common proxies, such as the gradient through the language-model head, we use them to study within-prompt noise and cross-prompt structure. Per-prompt policy-gradient estimates are noisy: at 16 rollouts per prompt, the median cosine similarity between an estimate and a held-out, high-budget estimate for the same prompt is between 0.02 and 0.20, depending on model and checkpoint. We find that, at the beginning of training, prompts share a common policy-gradient direction; at the reward plateau, per-prompt gradients are easier to estimate, but full-batch gradients become more difficult to estimate as the early shared cross-prompt direction disappears. Further, we show that group-relative baselines typically produce more reliable policy-gradient estimates than fixed baselines, that allocating rollouts by estimated pass rate improves reliability over uniform allocation, and that pass@k gradients that shift weight toward harder prompts are harder to estimate as k grows. We propose this sketched gradient analysis as a tool to measure the effect of various algorithmic choices in RLVR.

1 INTRODUCTION

Reinforcement learning with verifiable rewards (RLVR) optimizes a language model to solve tasks whose answers can be checked programmatically (Lambert et al., 2025; Guo et al., 2025). Typically, RLVR pipelines are built on some variant of the REINFORCE policy-gradient estimator (Williams, 1992). Different RLVR algorithms vary in their choice of advantage baseline (Ahmadian et al., 2024), prompt selection (Yu et al., 2025), optimization objective (Walder & Karkhanis, 2025), trust-region clipping (Schulman et al., 2017), and token-level credit assignment (Liu et al., 2025). Many of these design decisions are motivated by claims about gradient noise and geometry. Baselines are often framed in terms of variance reduction (Greensmith et al., 2004), although that framing has been challenged (Chung et al., 2021). Allocating more rollouts to noisier prompts may reduce the noise of the overall estimate compared to uniform allocation (Nguyen et al., 2026; Zeng et al., 2026; Bamba et al., 2025).

These justifications concern the noise and cross-rollout geometry of the full-model gradient. However, storing such gradients is cost-prohibitive: the naive gradient is the size of the model, and thus existing methods that reason about gradient geometry in such systems typically use proxies like final-layer gradients (Barakat et al., 2026; Guo et al., 2026), LoRA gradients (Chen et al., 2026), or gradients aggregated across tasks (Wu et al., 2026; Cai et al., 2026). Studies of whole-gradient

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.

geometry in reinforcement learning for smaller models do exist: notably, from Ilyas et al. (2020) in the continuous control domain. However, no such direct study exists in the language-model RLVR domain.

We propose to close this gap. Within the backward pass, each per-sequence gradient is compressed with CountSketch (Charikar et al., 2002) and stored with its exact norm. The sketch is linear and the stored gradient is unweighted, allowing the offline analysis of policy-gradient estimators under different baselines, objectives and rollout allocations. We tie our study of within-prompt gradient noise to the gradient noise scale of McCandlish et al. (2018); additionally, we measure cross-task gradient alignment. A batch estimate has estimator noise within each prompt, and a certain structure between the policy gradients of each prompt. Combined, these two factors determine how a fixed budget should be split between prompts and rollouts (Zhou et al., 2026; Tang et al., 2026) and provide a gradient noise scale; we measure the effects of such allocations and the noise scale directly on four models across three stages of training.

Our specific contributions are:

- **Direct measurement of within-prompt noise and cross-prompt structure in RLVR gradients.** In four models, from 0.6B to 8B parameters, estimating per-prompt policy gradients requires fewer rollouts at the plateau stage of training than at the beginning. However, per-prompt policy gradients lose their alignment, making the batch policy gradient harder to estimate later in training.
- **Evaluation of algorithmic choices in terms of direct gradient estimation.** We directly measure how well a low-budget policy gradient estimate can reconstruct its high-budget counterpart; the group-mean is the most reliable, while pass@k estimators are less stable. GRPO's successful and unsuccessful trajectories form policy gradients for the pass@2 and all@2 objectives; we validate this empirically.
- **Validation of sketches as a tool to measure language-model gradient geometry.** On exact gradients across model scales and training checkpoints, sketched whole-model gradients reconstruct relative gradient cosines better than other proxies, while using 500 to 14,000 times less storage. We measure this reconstruction error to validate sketching as a tool for measuring gradient geometry.
- **Trajectory tracing as a diagnostic tool.** During RLVR runs, we sketch every optimizer state and reconstruct checkpoint displacement from the sketches to within 1% of the measured value at intermediate checkpoints. GRPO trajectories only exceed the displacement of a noise null by 2–8% over 1,600 optimizer steps. We propose tracing, with this null, as a standard training metric.

2 SETUP

Policy Gradients. A differentiable policy π_θ maps a prompt q to a distribution over rollout trajectories τ. Given a prompt and a rollout, a verifier assigns a reward $r(\tau, q) \in \mathbb{R}$. We consider the binary-reward setup in which $r(\tau, q) \in \{0, 1\}$; the expected update is invariant to reward shifts for online optimization, and reward scales are absorbed into the learning rate. Let $V^\theta(q) = \mathbb{E}_{\tau\sim\pi_\theta(\tau|q)}[r(\tau, q)]$ be the pass rate of π_θ on prompt q. For each sampled rollout τ, we can compute a score gradient $g_\theta(\tau) = \nabla_\theta \log \pi(\tau|q) \in \mathbb{R}^{d_\theta}$. For some baseline function b that is uncorrelated with the per-prompt reward, we define the *advantage* of rollout τ as $A(\tau, q) = r(\tau, q) - b$. Then, given k rollouts on a prompt $\tau_1, \cdots, \tau_k$, we can define the policy-gradient estimate as the sum of scores, each weighted by its advantage. For online samples, $\mathbb{E}_{\tau\sim\pi_\theta(\tau|q)}[g(\tau)] = 0$, and therefore the expected value of this estimate does not depend on the choice of b (Williams, 1992; Greensmith et al., 2004). At the sequence level, b is often set by a Monte-Carlo estimate from multiple independent rollouts on the same prompt (Kool et al., 2019; Ahmadian et al., 2024; Shao et al., 2024). For our purposes, we will consider the GRPO estimator (Shao et al., 2024), without per-sequence normalization as per Dr. GRPO (Liu et al., 2025).

$$\hat{g}_q = \frac{1}{k}\sum_{i=1}^{k} A(\tau_i, q) g_\theta(\tau_i); \quad \hat{V} = \frac{1}{k}\sum_{i=1}^{k} r(\tau_i, q); \quad g_\theta(q) = \mathbb{E}_{\tau\sim\pi_\theta(\tau|q)}[\hat{g}_q]$$

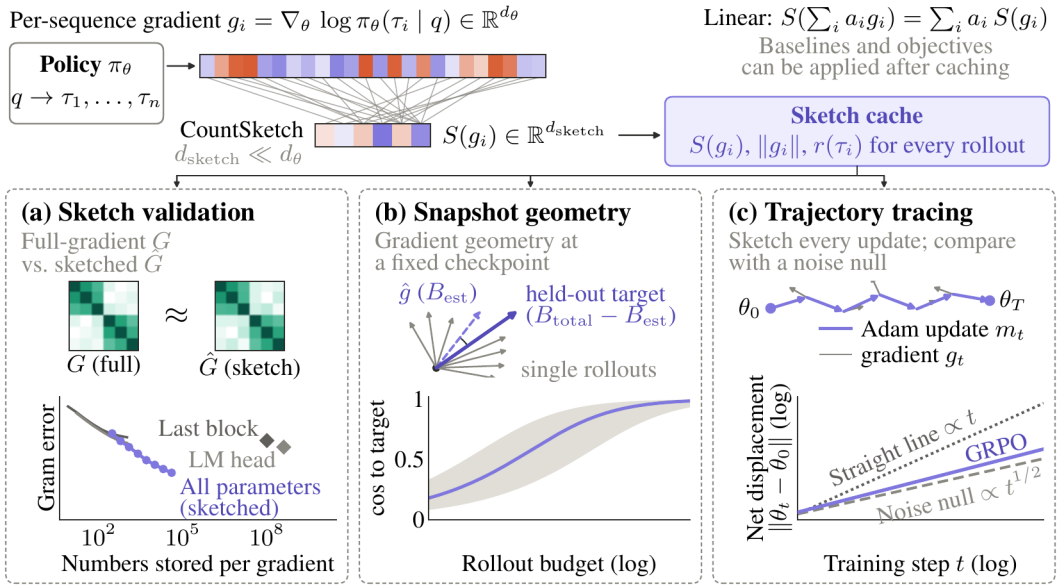


Figure 1: Three-stage pipeline for measuring gradient geometry via gradient sketching.

Targets and Reliability. The target policy gradient of a batch of prompts is simply the linear combination of individual-prompt policy gradients. Per-prompt baselines change the variance of each prompt's policy-gradient estimate, but not its expected direction nor magnitude (beyond trivial factor of $(k-1)/k$ for group-mean baselines). However, one may shift the weights between per-problem policy gradients by changing the objective. For example, pass@k optimization (Walder & Karkhanis, 2025; Chen et al., 2025) has gradient $k(1-V(q))^{k-1}\nabla V(q)$ where $V(q)$ is the pass rate of a given problem. We compare gradient targets via cosine similarity, so as to remove confounds associated with gradient targets of different scales. We follow Ilyas et al. (2020) in defining the reliability of an estimator $\hat{g}^{(k)}$ from k rollouts as $\mathbb{E}[\cos(\hat{g}(k), \hat{g}_{ref})]$ where $\hat{g}_{ref}$ is an estimate that uses a larger, disjoint set of rollouts.

Noise Model. For a given prompt q, let $\bar{g}_q = \mathbb{E}[\hat{g}_q]$ and let σ_q^2 be the trace of the covariance of one rollout's gradient, contributing to $\hat{g}_q$. Then the **noise scale** of the prompt is defined as $B_q = \sigma_q^2 / \|\bar{g}_q\|^2$ (McCandlish et al., 2018), the number of rollouts required for the signal and noise of the estimate to carry equal norm.

CountSketch. CountSketch (Charikar et al., 2002) is a linear map $S : \mathbb{R}^{d_\theta} \to \mathbb{R}^{d_{sketch}}$ consisting of two steps. First, each dimension of the original vector space is mapped to a random bucket via a hash $h : [d_\theta] \to [d_{sketch}]$; additionally, each dimension is mapped to a random sign via $s : [d_\theta] \to \{\pm 1\}$. Then, for an input vector v, CountSketch is defined by $S(v)_i = \sum_{j:h(j)=i} s(j)v_j$.

CountSketch is an unbiased estimator of inner products, with variance decreasing as $1/d_{sketch}$ (Charikar et al., 2002; Weinberger et al., 2009). Furthermore, it is consistent and asymptotically unbiased as an estimator of cosine similarities, with $O(1/d_{sketch})$ bias; to first order, the variance of its cosine similarity also falls as $1/d_{sketch}$.

See subsection A.1 for proofs. We cache exact gradient norms alongside sketched gradients for each component in the model. Because S is linear, any linear recombination of sketched gradients may be applied post-sketch; this allows us to study various algorithmic choices on the sketched dataset.

Models, Checkpoints and Tasks. We train the base-model checkpoints of Qwen3-0.6B, 1.7B and 4B (Yang et al., 2025), as well as Llama-3.1-8B-Instruct (Grattafiori et al., 2024), on DeepMath-103K (He et al., 2026) with GRPO using the SkyRL framework (Griggs et al., 2025). For each model, we analyze three checkpoints: the base model, step 20 (during the initial rise in reward) and at the reward plateau (200 for the Qwen models, 90 for the Llama model). For snapshot studies in section 4, we collect rollouts from these checkpoints on prompts from DAPO-Math-17k (Yu et al., 2025), ensuring that the prompts have not been seen during training. For trajectory tracing studies

Table 1: Relative Gram error of the whole-model sketch and of the exact proxies at every exact-gradient checkpoint with collected data (best per row in bold). All widths: Table 5.

Model	Stage	all param., sketched	LM head, exact	last block, exact	best block, exact
Qwen3-0.6B	base	**0.39**	0.86	1.23	0.62
	step 20	**0.09**	0.37	0.47	0.18
	step 200	**0.50**	1.38	1.90	0.85
	step 200, Adam	**0.21**	2.18	1.28	0.82
Qwen3-1.7B	step 200	0.71	1.45	1.50	**0.52**
Qwen3-4B	step 200	**0.39**	1.40	1.93	0.61
Llama-3.1-8B-Instruct	step 90	**0.46**	1.30	1.83	0.57
numbers stored per gradient, millions		0.032–0.040	156–525	16–218	16–218

in section 5, we train directly on DAPO-Math-17k. Experimental details and hyperparameters are in Appendix B.

3 Validating Sketches for Gradient Measurement

First, we validate that sketching preserves full-gradient geometry better than standard alternatives, and that its fidelity is sufficient to draw the conclusions made in subsequent sections. Sketches are scored in two ways: by their **Gram error**, the RMSE of off-diagonal cosines relative to the cosines of the exact gradients, and by their **path error**, the relative error in the predicted displacement of a checkpoint from the base model using sketched cosines and cached norms. We report the relative Gram error, normalized by the sum of squared target cosine values, so that a null predictor predicting all zeros achieves a relative error of 1. Formulas for these errors are given in subsection C.2.

Setup. We cache exact, full-model, sequence-level gradients for several model-checkpoint combinations: Qwen3-0.6B at steps 0, 20 and 200, Qwen3-4B at steps 0 and 200, Qwen3-1.7B at step 200, and Llama-3.1-8B-Instruct at step 90 (Qwen3-4B base is reported in Table 5). Separately, we compute path reconstruction over the sketched updates of GRPO runs for Qwen3-0.6B, 1.7B and 4B.

We compare three families of compressed gradients, each sketched at a range of widths and also stored exactly: (1) all parameters; (2) the language-model head only, a cheap last-layer proxy (eNTK analyses of fine-tuning (Malladi et al., 2023; Mohamadi et al., 2023) use all parameters); and (3) the last transformer block only.

Results. For all models and checkpoints measured, the whole-model sketches provide better estimates of gradient cosine similarity than the LM head and final-block proxies, and outperform the best single block on all but one tested checkpoint. On plateau checkpoints the LM head and final transformer block underperform the null baseline even when stored exactly (Table 1). With 500 to 14,000 times less storage than the other proxies, the whole-model sketch has the lowest Gram error (Table 5), an error per entry of 0.012–0.030, and a path error of 0.43–0.97%, compared to the LM-head path error of 3.0–27.5%. Cosine-similarity sketch error shrinks as $1/\sqrt{D}$; widening the flat sketch from 4,096 to 113,664 shrinks the spread of the plateau cross-prompt cosines at Qwen3-4B only from 0.033 to 0.031 (Appendix E).

4 Gradient Geometry from Snapshots

For each model and checkpoint, we collect a pool of rollouts and sketch each per-sequence gradient. In our largest collection run, we select 256 prompts from DAPO-Math-17k and collect 2048 rollouts for each prompt from each checkpoint. For checkpoints beyond the base model, we additionally collect the Adam-preconditioned gradient using the checkpoint's optimizer state (Kingma & Ba, 2015); results on Adam-preconditioned gradients follow the same trends (subsection C.4). Prompt and rollout counts are detailed in Appendix B.

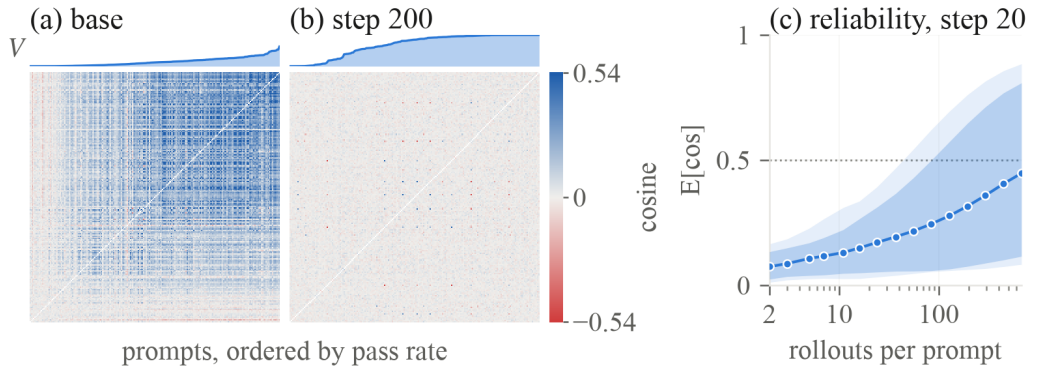


Figure 2: Gram matrix of high-budget policy-gradient directions across prompts for Qwen3-4B-Base at (a) the beginning of training and (b) step 200. The broad agreement across prompts at step 0 is absent at the plateau phase of training. (c) Per-prompt alignment with held-out policy gradient estimate as a function of rollouts; median, 25-75th and 10-90th percentiles shown across prompts.

4.1 Within-Prompt Noise

We begin with a straightforward question: how many rollouts are needed to estimate the policy gradient of a given prompt? More formally, we estimate the gradient noise scale from McCandlish et al. (2018) using our high-budget, offline set of rollouts, for each prompt.

We find that per-prompt noise falls sharply over the early steps of training, then remains similar across the rest of training. This phenomenon largely tracks improved pass rates – a weaker model may require many rollouts to collect enough successes for a stable prediction – but a portion of the decrease in noise remains even after binning problems by pass rate. κ halves for Qwen3-1.7B, Qwen3-4B and Llama-3.1-8B-Instruct, and is unchanged for Qwen3-0.6B (Table 8). We plot the estimated gradient noise scale as a function of pass rate in Figure 8.

Notably, the rollout budgets typically used during LLM RLVR training (Shao et al., 2024; Yu et al., 2025) are not high enough to produce a consistent policy-gradient estimate. We denote the number of rollouts required to reach an estimated cosine similarity of 0.5 between disjoint rollout sets as $n_{0.5}$. For base models, the median $n_{0.5}$ is 1,500–14,000 rollouts; for trained checkpoints, the median $n_{0.5}$ is between 700 and 2,500 rollouts (Table 8).

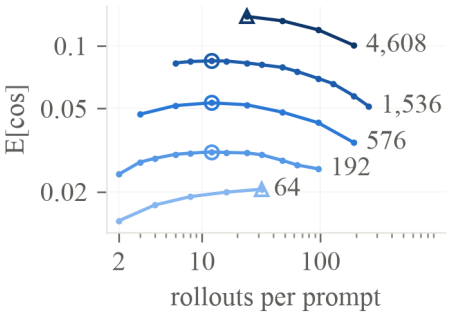
Setup. For each prompt and budget n, reliability is defined as the expected cosine similarity between an n-rollout policy-gradient estimate and a policy-gradient estimate constructed with the prompt's remaining rollouts. Unless otherwise mentioned, we use the group-mean estimate to form the target given that it is nearly unbiased and is the most reliable across trained checkpoints (Table 15).

Results. In all models, median per-prompt gradient noise falls after the start of training. Noise follows the gradient scale of McCandlish et al. (2018), and is inversely proportional to the reward variance $V(1-V)$ (Figure 8).

Takeaways. On a typical prompt, thousands of rollouts are required to converge on a policy gradient direction. This requirement tracks the per-prompt gradient noise scale and the reward variance $V(1-V)$, and decreases after training.

4.2 Cross-Prompt Structure

Beyond noise within a single prompt, we measure the degree to which policy gradients across prompts compose or interfere with one another. To do this, we estimate each prompt's gradient from all of its rollouts and compute the normalized Gram matrix between all prompts' estimated policy gradients. As expected, early in training policy gradients tend to point in the same direction as the policy learns rapidly. Later in training, this broad shared structure vanishes and prompts can



		1,024 rollouts		4,096 rollouts		16,384 rollouts	
Model	Stage	oracle	pilot	oracle	pilot	oracle	pilot
0.6B	base	1.21	0.85	1.14	1.04	1.07	1.01
	final	1.25	1.22	1.16	1.11	1.10	1.03
1.7B	base	1.21	1.23	1.10	1.09	1.05	1.03
	final	1.24	1.21	1.12	1.11	1.07	1.03
4B	base	1.18	1.17	1.04	1.03	1.01	1.00
	final	1.29	1.24	1.13	1.11	1.07	1.03
8B	base	1.12	1.09	1.08	1.05	1.04	1.01
	final	1.26	1.13	1.18	1.11	1.10	1.06
median		1.23	1.19	1.12	1.10	1.07	1.03

Figure 3: (a) Allocation curve for various per-prompt rollout budgets, for Qwen3-4B at step 200. Triangles mark truncated optima, limited by our ceiling of 256 prompts; circles mark interior optima. (b) Cosine similarity to high-budget disjoint GRPO target, relative to uniform-allocation baseline over 256 prompts. Neyman allocation, both with oracle and half-budget estimated V, outperforms uniform GRPO on all but two measured settings.

agree with or oppose one another; largely, updates between prompts are orthogonal, limiting the cross-prompt transfer of policy-gradient estimates for the sake of producing a denoised estimate.

Figure 2 illustrates this phenomenon in Qwen3-4B. At step 0, 85% of policy-gradient pairs have positive cosine similarity, with a mean off-diagonal cosine value of 0.14. By step 200, only 51% of pairs are positive and the mean off-diagonal cosine is $+0.001$. Similar trends hold for other models. At every trained checkpoint, the mean cosine similarity lies between -0.0001 and $+0.0008$, and the positive proportion between 50% and 51%.

For the base model, a naive gradient-noise model that treats each prompt's gradient as orthogonal overpredicts the estimation noise, estimating $n_{0.5}$ to be 1,000–6,200 rollouts per prompt while at the per-batch level it is 28–166 (Table 8). For trained models, however, cross-prompt policy gradients become roughly orthogonal and gradient noise becomes easier to estimate without knowing cross-prompt geometry; even a simple model on the reward variance $\kappa/(V(1-V))$ predicts the Monte-Carlo cosine similarity to the target within 3–15% (Figure 9).

Multi-prompt gradient noise model. The per-prompt noise scales and the cross-prompt Gram combine into a prediction of batch-level gradient noise: with n_q rollouts on prompt q and a group-mean baseline, $\cos(\hat{g}, g) \approx \left(1 + \sum_q \sigma_q^2 (n_q - 1)^{-1} / \|\sum_q \bar{g}_q\|^2\right)^{-1/2}$ (subsection A.5). With the measured Gram matrix, this model predicts the Monte-Carlo agreement to within 5% at every checkpoint, while treating prompts as orthogonal misses by 87 - 94% at the base models (Figure 9, Table 8); the same model approximately gives the optimal group size of subsection 4.3.

Takeaways. Cross-prompt agreement disappears after training; estimating the batch gradient later in training requires covering a larger set of prompts, which we revisit in subsection 4.3.

4.3 Optimal Rollout Allocation

Given a budget of T total rollouts, one may select a number of prompts P and a number of rollouts per prompt R such that $T = PR$. For a fixed T, we can score each choice of R by how well an R-rollout estimate approximates a much higher-budget target estimate constructed from the set of held-out rollouts across all prompts.

Beyond calculating gradient variance, the set of sketched gradients allows for the simulation of various sampling approaches. For example, for a given budget of rollouts and a randomly drawn set of problems, we can sweep through proposed allocations of rollouts per prompt and see which allocations best approximate the high-budget held-out target.

This produces curves that can determine an optimal allocation, as in Figure 3; similar allocation curves are shown in Figure 12. The optimal number of rollouts falls roughly within the range predicted by the noise scale.

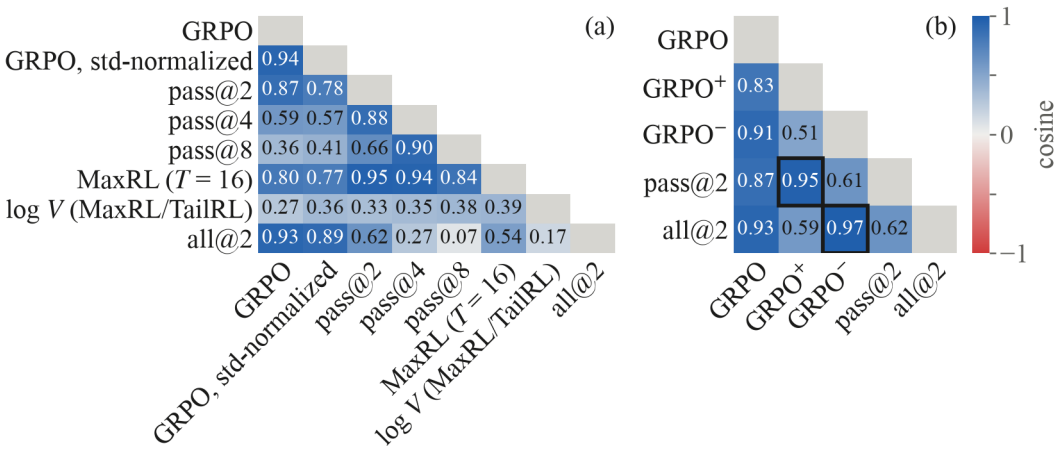


Figure 4: Cosine similarity between estimated objective and target objective. Qwen3-4B, step 200. Left: pairs of objectives. Right: GRPO, its successful-rollout (GRPO$^+$) and failed-rollout (GRPO$^-$) gradients, pass@2 and all@2. Every model and stage: Figure 14, Figure 15.

We also study the effectiveness of non-uniform allocation. In particular, allocating rollouts proportional to the oracle $\sqrt{V(1-V)}$ of each problem (Neyman, 1934; Nguyen et al., 2026; Zeng et al., 2026) improves estimation at lower budgets. In the non-oracle case, we allocate half of our rollout budget uniformly across prompts to estimate $\hat{V}$ per problem, and the second half proportional to $\sqrt{\hat{V}(1-\hat{V})}$ with estimates provided from the pilot rollouts; this resembles the pipeline from Kim et al. (2026) and XRPO (Bamba et al., 2025). We again see faster convergence among lower budgets than uniform sampling, suggesting the merits of low-budget pass-rate estimation for rollout budget allocation.

Takeaways. We can determine the optimal per-prompt rollout budget via Monte-Carlo simulation on our sketches. Using an initial set of rollouts to estimate pass rates and then allocating dynamically outperforms uniform allocation, especially for smaller rollout budgets.

4.4 Baselines and Objectives

In expectation, changing the baseline does not affect the policy gradient, although in practice it may lead to different optimization trajectories and convergence over the course of training (Chung et al., 2021). For various choices of baseline, we measure within-prompt gradient noise as in subsection 4.1. In addition to GRPO, we evaluate fixed-V baselines for various V, as well as the theoretically minimum-variance baseline which is composed of the mean reward weighted by squared per-trajectory norms (Greensmith et al., 2004).

Results are in Table 13; detailed noise measures for each estimator across checkpoints are in Table 14 and Table 15.

Beyond baseline convergence, we measure the degree to which different optimization objectives align with one another. We measure several different optimization objectives: pass@1 (standard REINFORCE); pass@k (Walder & Karkhanis, 2025; Chen et al., 2025; Tang et al., 2025); MaxRL (Tajwar et al., 2026); and TailRL (Ramasubramanian et al., 2026). Details of each are in subsection G.2.

Additionally, we show that the following proposition holds both theoretically and empirically:

Proposition 1. *With binary rewards and GRPO-style group-mean baselines, the gradient over successful trajectories is, up to a scaling factor, an unbiased estimate of the gradient of the* pass@2 *objective. The gradient over unsuccessful trajectories is, up to the same factor, an unbiased estimate of the gradient of the* all@2 *objective, the probability that two independently drawn trajectories are both correct.*

Table 2: Ratio of net displacement to the root sum of step norms, and the fitted growth exponent α of net displacement as a function of optimization step.

Online GRPO	Steps	R	null R [95% range]	R vs. null	α
Qwen3-0.6B	1,600	4.44	4.35 [4.26, 4.43]	+2%	0.53
Qwen3-1.7B	1,600	4.68	4.35 [4.26, 4.43]	+8%	0.62
Qwen3-4B	1,575	4.52	4.34 [4.26, 4.44]	+4%	0.59

Proof. See subsection A.3.

Results. Cross-objective alignment is shown in Figure 4. As expected, the pass@k objectives deviate from the standard pass@1 objective as k increases; pass@2 aligns with the positive gradients of GRPO and all@2 with the negative gradients, as predicted by the above identity. The pass@2 / all@2 identity holds strongly for all models tested, with the exception of Qwen3-0.6B at the beginning of training, for which the set of negative trajectories has a much noisier gradient than the set of positive trajectories.

Takeaways. Objectives that weigh differently among prompts diverge over the course of training. GRPO positives track pass@2 and negatives track all@2, confirming the theory.

5 Sketching Training Trajectories

Here we propose and validate a mechanism for tracking gradient geometry within a given training run, which we term *trajectory tracing*. Sketching is cheap and easily parallelizable, and can therefore run inside training. At each optimizer step we sketch the minibatch gradient, Adam-preconditioned gradient, momentum and Adam update, costing 64 KB per step, or roughly 100MB over a 1,600-step training run. By replaying the sketched updates, with their cached norms, we estimate checkpoint displacement to within 1%. Collecting such statistics at every step allows us to track relative displacement over the whole training trajectory, rather than being limited to a fixed set of checkpoints.

Setup. We trace online GRPO on Qwen3-0.6B, 1.7B and 4B trained on DAPO-Math-17k over 1,600 optimizer steps. If the updates were filtered via momentum and did not share a noise direction, for timesteps t the net displacement would grow as $t^{1/2}$. Further, the ratio R of net displacement to the root sum of squared step norms, $\sqrt{\sum_t \|m_t\|^2}$, would be $\sqrt{(1+\beta_1)/(1-\beta_1)} \approx 4.36$ for $\beta_1 = 0.9$ (subsection A.7).

Results. All three runs exceed the null only slightly: R is 4.44, 4.68 and 4.52, 2–8% above the null and above its 95% range (Table 2), and $\|\Delta_t\|$ grows as $t^{0.53}$ to $t^{0.62}$, after the learning-rate warmup. Over this horizon the directed part of the updates is small next to the noise, as the noise scales of section 4 predict for 32 prompts per step; since it grows as t^2 in $\|\Delta_t\|^2$ and the noise as t, it should dominate over longer horizons, as in reports of near-linear weight dynamics over full runs (Wang et al., 2026). While progress towards the final direction rises monotonically and nearly linearly, a momentum-filtered random walk does as well (subsection A.7). Comparing training paradigms by tracing, with matched data, step sizes and optimizer state, is left to future work.

6 Related Work

Gradient noise and batch size. Ilyas et al. (2020) finds that policy-gradient and value estimates in critic-based continuous control correlate poorly with the true gradient at standard sample sizes, more so later in training; we make the corresponding measurement for full-model gradients in RLVR. The gradient noise scale relates gradient statistics to batch sizes (McCandlish et al., 2018; Zhang et al., 2025a; Merrill et al., 2025); gradient diversity (Yin et al., 2018), stiffness (Fort et al., 2019) and coherent gradients (Chatterjee, 2020) describe agreement between examples. Cheng et al. (2026) find that the compute-optimal number of rollouts per prompt grows with the budget and then saturates, and that the number of prompts per batch mainly affects stability; see also Khatri et al. (2026).

Estimators, baselines and objectives. Baselines are analyzed as control variates (Greensmith et al., 2004), though they also affect learned behavior (Chung et al., 2021). Leave-one-out and group-mean

baselines (Kool et al., 2019; Ahmadian et al., 2024; Shao et al., 2024) are standard in RLVR (Guo et al., 2025; Lambert et al., 2025); DAPO (Yu et al., 2025) and Dr. GRPO (Liu et al., 2025) change GRPO's sampling and normalization, and FADE (Decugis et al., 2026) analyzes the effects of positive and negative gradients in GRPO on policy diversity. Zhou et al. (2026) analyze the GRPO gradient as a U-statistic, decompose its error into prompt-sampling, within-prompt and baseline terms and derive a budget-independent group size, and Tang et al. (2026) derive rules for allocating rollouts between prompts, with a two-stage sampling split (Cochran, 1977); we measure the quantities that these models depend on. Pass@k and best-of-N objectives (Tang et al., 2025; Walder & Karkhanis, 2025; Chen et al., 2025; Chow et al., 2025) assign higher implicit weight to harder problems to avoid trading pass@k for pass@1 (Yue et al., 2025), MaxRL (Tajwar et al., 2026) and TailRL (Ramasubramanian et al., 2026) target the log pass rate and the reward's upper tail; one work finds success training only on failures to preserve pass@k (Zhu et al., 2025b). Thrampoulidis et al. (2026) connect advantage shaping to pass@k policy gradients. We measure the agreement between these objectives.

Rollout allocation. Our variance-optimal sampling derives from Neyman (1934), and matches similar Neyman-sampling-based approaches such as VIP (Nguyen et al., 2026) and CurES (Zeng et al., 2026). GVM (Yao et al., 2025) allocates samples to minimize gradient variance in rejection sampling, XRPO (Bamba et al., 2025) spends half of its rollout budget to estimate problem difficulty, and Knapsack RL (Li et al., 2026) and pilot-based filtering (Kim et al., 2026) allocate by expected learning signal; others select prompts by predicted difficulty (Zheng et al., 2025; Zhang et al., 2025b), and Reinforce-Ada (Xiong et al., 2025) samples adaptively, noting that small pilots miss hard prompts.

Gradient geometry in LLM training. Prior measurements in LLM RL use LoRA gradients (Chen et al., 2026), final-layer gradients (Barakat et al., 2026; Guo et al., 2026), linking pass@k and pass@1 via interference, or task-aggregated gradients (Wu et al., 2026; Cai et al., 2026). In weight space, RLVR updates are sparse (Mukherjee et al., 2025), avoid principal directions (Zhu et al., 2025a) and evolve nearly linearly (Wang et al., 2026); on-policy RL stays closer to the base model than SFT (Shenfeld et al., 2026; Chu et al., 2025); training moves in a small subspace (Gur-Ari et al., 2018; Li et al., 2018).

Sketching and projecting gradients. CountSketch and other random sketching projections preserve inner products in lower-dimensional vector spaces (Charikar et al., 2002; Johnson & Lindenstrauss, 1984; Achlioptas, 2003; Weinberger et al., 2009; Woodruff, 2014). Sketched gradients can also be used for data attribution during training (Pruthi et al., 2020; Park et al., 2023; Xia et al., 2024; Choe et al., 2025; Schioppa, 2024). The empirical neural tangent kernel (eNTK) (Jacot et al., 2018; Mohamadi et al., 2023) measures inner products between gradients, and is often approximated with language-model-head gradients (Malladi et al., 2023).

7 Discussion

We use sketched gradients to measure geometry in large-scale RLVR pipelines. Given the relatively cheap cost, we propose that sketched gradients should be part of the standard set of metrics collected over the course of a run, alongside standard gradient norm statistics. Greater understanding of the optimization dynamics of large models may yield greater insight into their learned behaviors, allowing for more informed training and post-training of models than would be possible by the existing suites of metrics used during training.

Reproducibility Statement

Anonymized code for training, sketching, and analysis is made available in the supplementary material, along with each figure and table and the data underlying them. Experimental settings are detailed in Appendix B.

AI Use Statement

We used large language models (LLM-based coding and writing agents) as follows. *Implementing methods and running experiments:* LLMs implemented the data-collection pipeline (gradient

sketching inside the training framework), the analysis code and the generation of figures and tables, under the authors' direction, and launched and monitored experiments; the authors specified the methods and checked the outputs. *Methodology and interpretation:* LLMs gave feedback on methodology, reviewed drafts, checked claims against the data and assisted in interpreting results; the final interpretations are the authors'. *Mathematical claims and proofs:* the authors formulated the claims; LLMs assisted in writing and checking the appendix derivations (the CountSketch variance, the noise model, the exact form of Proposition 1), which the authors verified. *Writing:* the authors wrote the main text; LLMs suggested edits, including citations, figure captions and corrections, which the authors reviewed and accepted or rewrote. Much of the appendix prose was drafted with LLM assistance and edited by the authors. The illustrative pipeline figure (Figure 1) was generated with LLM assistance under the authors' guidance. The authors take full responsibility for the content.

REFERENCES

Dimitris Achlioptas. Database-friendly random projections: Johnson-Lindenstrauss with binary coins. *Journal of Computer and System Sciences*, 66(4):671–687, 2003. doi: 10.1016/S0022-0000(03)00025-4.

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting REINFORCE-style optimization for learning from human feedback in LLMs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12248–12267, Bangkok, Thailand, 2024. URL https://aclanthology.org/2024.acl-long.662/.

Udbhav Bamba, Minghao Fang, Yifan Yu, Haizhong Zheng, and Fan Lai. XRPO: Pushing the limits of GRPO with targeted exploration and exploitation. *arXiv preprint arXiv:2510.06672*, 2025.

Anas Barakat, Souradip Chakraborty, Khushbu Pahwa, and Amrit Singh Bedi. Why pass@k optimization can degrade pass@1: Prompt interference in LLM post-training, 2026.

Min Cai, Yu Liang, Longzheng Wang, Yan Wang, Yueyang Zhang, Long Xia, Zhiyuan Sun, Xi Ye, and Daiting Shi. Advancing general-purpose reasoning models with modular gradient surgery, 2026.

Moses Charikar, Kevin Chen, and Martin Farach-Colton. Finding frequent items in data streams. In Peter Widmayer, Stephan Eidenbenz, Francisco Triguero, Rafael Morales, Ricardo Conejo, and Matthew Hennessy (eds.), *Automata, Languages and Programming (ICALP 2002)*, volume 2380 of *Lecture Notes in Computer Science*, pp. 693–703. Springer, 2002. doi: 10.1007/3-540-45465-9_59.

Satrajit Chatterjee. Coherent gradients: An approach to understanding generalization in gradient descent-based optimization. In *International Conference on Learning Representations*, 2020.

Yulin Chen, He He, and Chen Zhao. The unlearnability phenomenon in RLVR for language models. In *International Conference on Machine Learning (ICML)*, 2026. arXiv:2605.16787.

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

Zhoujun Cheng, Yutao Xie, Yuxiao Qu, Amrith Setlur, Shibo Hao, Varad Pimpalkhute, Tongtong Liang, Feng Yao, Zhengzhong Liu, Eric Xing, Virginia Smith, Ruslan Salakhutdinov, Zhiting Hu, Taylor Killian, and Aviral Kumar. IsoCompute playbook: Optimally scaling sampling compute for LLM RL. *arXiv preprint arXiv:2603.12151*, 2026.

Sang Keun Choe, Hwijeen Ahn, Juhan Bae, Kewen Zhao, Youngseog Chung, Adithya Pratapa, Willie Neiswanger, Emma Strubell, Teruko Mitamura, Jeff Schneider, Eduard Hovy, Roger Baker Grosse, and Eric P. Xing. What is your data worth to GPT? LLM-scale data valuation with influence functions. In *Advances in Neural Information Processing Systems*, 2025.

Yinlam Chow, Guy Tennenholtz, Izzeddin Gur, Vincent Zhuang, Bo Dai, et al. Inference-aware fine-tuning for best-of-N sampling in large language models. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://openreview.net/forum?id=77gQUdQhE7.

Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V. Le, Sergey Levine, and Yi Ma. SFT memorizes, RL generalizes: A comparative study of foundation model post-training. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 10818–10838, 2025. URL https://proceedings.mlr.press/v267/chu25c.html.

Wesley Chung, Valentin Thomas, Marlos C. Machado, and Nicolas Le Roux. Beyond variance reduction: Understanding the true impact of baselines on policy optimization. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, 2021.

William G. Cochran. *Sampling Techniques*. John Wiley & Sons, 3rd edition, 1977.

Damek Davis and Benjamin Recht. What is the objective of reasoning with reinforcement learning? *arXiv preprint arXiv:2510.13651*, 2025.

Juliette Decugis, Sean O'Brien, Francis Bach, Gabriel Synnaeve, and Taco Cohen. Don't let gains FADE: Breaking down policy gradient weights in RL. *arXiv preprint arXiv:2607.01490*, 2026.

Stanislav Fort, Paweł Krzysztof Nowak, Stanisław Jastrzebski, and Srini Narayanan. Stiffness: A new perspective on generalization in neural networks, 2019.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Evan Greensmith, Peter L. Bartlett, and Jonathan Baxter. Variance reduction techniques for gradient estimates in reinforcement learning. *Journal of Machine Learning Research*, 5:1471–1530, 2004. URL https://www.jmlr.org/papers/v5/greensmith04a.html.

Tyler Griggs, Sumanth Hegde, Eric Tang, Shu Liu, Shiyi Cao, Dacheng Li, Charlie Ruan, Philipp Moritz, Kourosh Hakhamaneshi, Richard Liaw, Akshay Malik, Matei Zaharia, Joseph E. Gonzalez, and Ion Stoica. Evolving SkyRL into a highly-modular RL framework. https://novasky-ai.notion.site/skyrl-v01, 2025. Notion Blog.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, et al. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645(8081):633–638, 2025. doi: 10.1038/s41586-025-09422-z. arXiv:2501.12948.

Peizheng Guo, Jianqi Zhang, Xingyu Zhang, Yun Fan, Jiahuan Zhou, Changwen Zheng, and Wenwen Qiang. GUPO: Gradient uncertainty-aware policy optimization for post-training large language models. *arXiv preprint arXiv:2608.17411*, 2026.

Guy Gur-Ari, Daniel A. Roberts, and Ethan Dyer. Gradient descent happens in a tiny subspace, 2018.

Zhiwei He, Tian Liang, Jiahao Xu, Qiuzhi Liu, Xingyu Chen, Yue Wang, Linfeng Song, Dian Yu, Zhenwen Liang, Wenxuan Wang, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. DeepMath-103K: A large-scale, challenging, decontaminated, and verifiable mathematical dataset for advancing reasoning. In *International Conference on Learning Representations (ICLR)*, 2026.

Andrew Ilyas, Logan Engstrom, Shibani Santurkar, Dimitris Tsipras, Firdaus Janoos, Larry Rudolph, and Aleksander Madry. A closer look at deep policy gradients. In *International Conference on Learning Representations (ICLR)*, 2020. URL https://openreview.net/forum?id=ryxdEkHtPS.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

William B. Johnson and Joram Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. In *Conference in Modern Analysis and Probability (New Haven, Conn., 1982)*, volume 26 of *Contemporary Mathematics*, pp. 189–206. American Mathematical Society, 1984. doi: 10.1090/conm/026/737400.

Devvrit Khatri, Lovish Madaan, Rishabh Tiwari, Rachit Bansal, Sai Surya Duvvuri, Manzil Zaheer, Inderjit S. Dhillon, David Brandfonbrener, and Rishabh Agarwal. The art of scaling reinforcement learning compute for LLMs. In *International Conference on Learning Representations (ICLR)*, 2026.

Woojeong Kim, Ziyi Yang, Jing Nathan Yan, and Jialu Liu. Spend your rollouts where it counts: Rollout allocation for group-based RL post-training. *arXiv preprint arXiv:2605.26606*, 2026.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Wouter Kool, Herke van Hoof, and Max Welling. Buy 4 REINFORCE samples, get a baseline for free! In *ICLR 2019 Workshop on Deep Reinforcement Learning Meets Structured Prediction*, 2019. URL https://mlanthology.org/iclrw/2019/kool2019iclrw-buy/.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, et al. Tulu 3: Pushing frontiers in open language model post-training. In *Conference on Language Modeling (COLM)*, 2025. URL https://openreview.net/forum?id=i1uGbfHHpH.

Chunyuan Li, Heerad Farkhoor, Rosanne Liu, and Jason Yosinski. Measuring the intrinsic dimension of objective landscapes. In *International Conference on Learning Representations*, 2018.

Ziniu Li, Congliang Chen, Tianyun Yang, Tian Ding, Ruoyu Sun, Ge Zhang, Wenhao Huang, and Zhi-Quan Luo. Knapsack RL: Compute-efficient reinforcement learning via heterogeneous rollout allocation. In *International Conference on Machine Learning (ICML)*, 2026. URL https://icml.cc/virtual/2026/poster/60948.

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 (COLM)*, 2025. URL https://openreview.net/forum?id=5PAF7PAY2Y.

Sadhika Malladi, Alexander Wettig, Dingli Yu, Danqi Chen, and Sanjeev Arora. A kernel-based view of language model fine-tuning. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 23610–23641. PMLR, 2023.

Sam McCandlish, Jared Kaplan, Dario Amodei, and OpenAI Dota Team. An empirical model of large-batch training. *arXiv preprint arXiv:1812.06162*, 2018.

William Merrill, Shane Arora, Dirk Groeneveld, and Hannaneh Hajishirzi. Critical batch size revisited: A simple empirical approach to large-batch language model training. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Mohamad Amin Mohamadi, Wonho Bae, and Danica J. Sutherland. A fast, well-founded approximation to the empirical neural tangent kernel. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 25061–25081. PMLR, 2023.

Sagnik Mukherjee, Lifan Yuan, Dilek Hakkani-Tur, and Hao Peng. Reinforcement learning finetunes small subnetworks in large language models. In *Advances in Neural Information Processing Systems 38 (NeurIPS 2025)*, 2025. URL https://neurips.cc/virtual/2025/poster/120324.

Jerzy Neyman. On the two different aspects of the representative method: The method of stratified sampling and the method of purposive selection. *Journal of the Royal Statistical Society*, 97(4): 558–625, 1934. doi: 10.1111/j.2397-2335.1934.tb04184.x.

Hieu Trung Nguyen, Bao Nguyen, Wenao Ma, Yuzhi Zhao, Ruifeng She, and Viet Anh Nguyen. Adaptive rollout allocation for online reinforcement learning with verifiable rewards. In *International Conference on Learning Representations (ICLR)*, 2026.

Sung Min Park, Kristian Georgiev, Andrew Ilyas, Guillaume Leclerc, and Aleksander Madry. TRAK: Attributing model behavior at scale. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 27074–27113. PMLR, 2023.

Garima Pruthi, Frederick Liu, Satyen Kale, and Mukund Sundararajan. Estimating training data influence by tracing gradient descent. In *Advances in Neural Information Processing Systems*, volume 33, pp. 19920–19930, 2020.

Shrinivas Ramasubramanian, Daman Arora, Fahim Tajwar, Guanning Zeng, Qingyang Wu, Zhongzhu Zhou, Chenfeng Xu, Haiwen Feng, Yuda Song, Aarti Singh, Ruslan Salakhutdinov, J. Andrew Bagnell, Jeff Schneider, and Andrea Zanette. Tail-likelihood reinforcement learning. *arXiv preprint arXiv:2609.02987*, 2026.

Andrea Schioppa. Efficient sketches for training data attribution and studying the loss landscape. In *Advances in Neural Information Processing Systems*, 2024. doi: 10.52202/079017-1190.

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

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. *arXiv preprint arXiv:2402.03300*, 2024.

Idan Shenfeld, Jyothish Pari, and Pulkit Agrawal. RL's razor: Why online reinforcement learning forgets less. In *International Conference on Learning Representations (ICLR)*, 2026.

Fahim Tajwar, Guanning Zeng, Yueer Zhou, Yuda Song, Daman Arora, Yiding Jiang, Jeff Schneider, Ruslan Salakhutdinov, Haiwen Feng, and Andrea Zanette. Maximum likelihood reinforcement learning. *arXiv preprint arXiv:2602.02710*, 2026.

Kenton Tang, Yuzhu Chen, and Fengxiang He. Generalisation of RLHF under reward shift and clipped KL regularisation. *arXiv preprint arXiv:2602.21765*, 2026.

Yunhao Tang, Kunhao Zheng, Gabriel Synnaeve, and Rémi Munos. Optimizing language models for inference time objectives using reinforcement learning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 59066–59085, 2025. URL https://proceedings.mlr.press/v267/tang25o.html.

Christos Thrampoulidis, Sadegh Mahdavi, and Wenlong Deng. Advantage shaping as surrogate reward maximization: Unifying pass@K policy gradients. *Transactions on Machine Learning Research (TMLR)*, 2026.

Christian Walder and Deep Tejas Karkhanis. Pass@K policy optimization: Solving harder reinforcement learning problems. In *Advances in Neural Information Processing Systems 38 (NeurIPS 2025)*, pp. 152416–152445, 2025. doi: 10.52202/085713-5096.

Tianle Wang, Jiayu Liu, Zhongyuan Wu, Shenghao Jin, Wei Chen, Hao Xu, and Ning Miao. Linear dynamics in the RLVR training of large language models, 2026.

Kilian Weinberger, Anirban Dasgupta, John Langford, Alex Smola, and Josh Attenberg. Feature hashing for large scale multitask learning. In *Proceedings of the 26th Annual International Conference on Machine Learning (ICML)*, pp. 1113–1120, 2009. doi: 10.1145/1553374.1553516.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 1992. doi: 10.1007/BF00992696.

David P. Woodruff. Sketching as a tool for numerical linear algebra. *Foundations and Trends in Theoretical Computer Science*, 10(1–2):1–157, 2014. doi: 10.1561/0400000060.

Runzhe Wu, Ankur Samanta, Ayush Jain, Scott Fujimoto, Jeongyeol Kwon, Ben Kretzu, Youliang Yu, Kaveh Hassani, Boris Vidolov, and Yonathan Efroni. Imbalanced gradients in RL post-training of multi-task LLMs. In *Findings of the Association for Computational Linguistics: EACL 2026*, 2026.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. LESS: Selecting influential data for targeted instruction tuning. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 54104–54132. PMLR, 2024.

Wei Xiong, Chenlu Ye, Baohao Liao, Hanze Dong, Xinxing Xu, et al. Reinforce-Ada: An adaptive sampling framework under non-linear RL objectives. *arXiv preprint arXiv:2510.04996*, 2025.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Jiarui Yao, Yifan Hao, Hanning Zhang, Hanze Dong, Wei Xiong, Nan Jiang, and Tong Zhang. Optimizing chain-of-thought reasoners via gradient variance minimization in rejection sampling and RL. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Dong Yin, Ashwin Pananjady, Max Lam, Dimitris Papailiopoulos, Kannan Ramchandran, and Peter Bartlett. Gradient diversity: a key ingredient for scalable distributed learning. In *Proceedings of the 21st International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2018.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, et al. DAPO: An open-source LLM reinforcement learning system at scale. In *Advances in Neural Information Processing Systems 38 (NeurIPS 2025)*, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/a4277440d50f1f15d2cb4c14f7e0c0d2-Abstract-Conference.html.

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 38 (NeurIPS 2025)*, 2025.

Yongcheng Zeng, Zexu Sun, Bokai Ji, Erxue Min, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, Haifeng Zhang, Xu Chen, and Jun Wang. CurES: From gradient analysis to efficient curriculum learning for reasoning LLMs. In *International Conference on Learning Representations (ICLR)*, 2026.

Hanlin Zhang, Depen Morwani, Nikhil Vyas, Jingfeng Wu, Difan Zou, Udaya Ghai, Dean Foster, and Sham Kakade. How does critical batch size scale in pre-training? In *International Conference on Learning Representations*, 2025a.

Ruiqi Zhang, Daman Arora, Song Mei, and Andrea Zanette. SPEED-RL: Faster training of reasoning models via online curriculum learning. *arXiv preprint arXiv:2506.09016*, 2025b. Presented at the AI4Math workshop at ICML 2025.

Haizhong Zheng, Yang Zhou, Brian R. Bartoldson, Bhavya Kailkhura, Fan Lai, Jiawei Zhao, and Beidi Chen. Act only when it pays: Efficient reinforcement learning for LLM reasoning via selective rollouts. In *Advances in Neural Information Processing Systems 38 (NeurIPS 2025)*, 2025. URL https://neurips.cc/virtual/2025/poster/115272.

Hongyi Zhou, Kai Ye, Erhan Xu, Jin Zhu, Ying Yang, Shijin Gong, and Chengchun Shi. Demystifying group relative policy optimization: Its policy gradient is a U-statistic. *arXiv preprint arXiv:2603.01162*, 2026.

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, 2025a.

Xinyu Zhu, Mengzhou Xia, Zhepei Wei, Wei-Lin Chen, Danqi Chen, and Yu Meng. The surprising effectiveness of negative reinforcement in LLM reasoning. In *Advances in Neural Information Processing Systems 38 (NeurIPS 2025)*, 2025b. URL https://neurips.cc/virtual/2025/poster/116793.

APPENDIX

A Theory **16**

A.1 Variance of Sketched Inner Products and Cosines 16

A.2 Baselines and Per-Rollout Noise 16

A.3 GRPO Positive and Negative Gradients (Proposition) 17

A.4 Per-Prompt Reliability 17

A.5 Batch Noise and Optimal Allocation 18

A.6 Weighted Targets 19

A.7 Momentum-Filtered Noise 19

B Experimental Details **20**

B.1 Models, Data and Training 20

B.2 Snapshot Pools 20

B.3 Reliability Conventions 20

B.4 Trajectory Runs 21

C Sketch Validation **22**

C.1 Sketch Layout 22

C.2 Fidelity Metrics 22

C.3 Validation Results 22

C.4 Adam-Preconditioned Gradients 24

D Within-Prompt Noise **26**

D.1 Noise Scale and Pass Rate 26

E Cross-Prompt Structure **30**

F Rollout Allocation **34**

G Baselines and Objectives **37**

G.1 Baselines 37

G.2 Objectives 37

G.3 Successful and Failed Rollouts 43

H Trajectory Tracing **45**

A THEORY

A.1 VARIANCE OF SKETCHED INNER PRODUCTS AND COSINES

Let $h : [d_\theta] \to [d_{sketch}]$ be drawn from a pairwise-independent family and $s : [d_\theta] \to \{\pm 1\}$ from a four-wise independent family, and write $\delta_{ij} = \mathbf{1}[h(i) = h(j)]$, so that $\mathbb{E}\,\delta_{ij} = 1/d_{sketch}$ for $i \neq j$.

Inner products. Expanding the sketch,

$$\langle S(a), S(b)\rangle = \sum_{k=1}^{D} \sum_{i,j:\, h(i)=h(j)=k} s(i)s(j)\, a_i b_j = \langle a, b\rangle + \sum_{i\neq j} s(i)s(j)\, \delta_{ij}\, a_i b_j. \tag{1}$$

Since $\mathbb{E}[s(i)s(j)] = 0$ for $i \neq j$, the sketched inner product is unbiased. For $i \neq j$ and $k \neq l$, $\mathbb{E}[s(i)s(j)s(k)s(l)]$ is 1 if $\{k, l\} = \{i, j\}$ and 0 otherwise, so

$$\operatorname{Var}\langle S(a), S(b)\rangle = \frac{1}{D}\sum_{i\neq j}\left(a_i^2 b_j^2 + a_i b_i a_j b_j\right) = \frac{1}{D}\Big(\|a\|^2\|b\|^2 + \langle a, b\rangle^2 - 2\sum_i a_i^2 b_i^2\Big). \tag{2}$$

Cosines. Take unit vectors a, b with $c = \langle a, b\rangle$ (the c_{ij} of §2, for one pair), and write each sketched quantity as its exact value plus a fluctuation:

$$\langle S(a), S(b)\rangle = c + \varepsilon_{ab}, \qquad \|S(a)\|^2 = 1 + \varepsilon_{aa}, \qquad \|S(b)\|^2 = 1 + \varepsilon_{bb}.$$

To first order in the fluctuations,

$$\cos\big(S(a), S(b)\big) = \frac{c + \varepsilon_{ab}}{\sqrt{(1+\varepsilon_{aa})(1+\varepsilon_{bb})}} \approx c + \varepsilon_{ab} - \frac{c}{2}\big(\varepsilon_{aa} + \varepsilon_{bb}\big) = c + \sum_{i\neq j} s(i)s(j)\,\delta_{ij}\, M_{ij},$$

with $M = ab^\top - \frac{c}{2}(aa^\top + bb^\top)$. The variance of the last sum is $\frac{2}{D}\sum_{i\neq j}(M^s_{ij})^2$, where $M^s = \frac{1}{2}(M + M^\top)$. A direct computation gives $\|M^s\|_F^2 = \frac{1}{2}(1-c^2)^2$ and $M^s_{ii} = a_i b_i - \frac{c}{2}(a_i^2 + b_i^2)$, so

$$\operatorname{Var}\cos\big(S(a), S(b)\big) \approx \frac{1}{D}\Big[(1-c^2)^2 - 2\sum_i\Big(a_i b_i - \frac{c}{2}(a_i^2 + b_i^2)\Big)^2\Big] \le \frac{(1-c^2)^2}{D} \le \frac{1}{D}, \tag{3}$$

and the bias of the sketched cosine is of order $1/d_{sketch}$. When each vector's energy is spread over many coordinates the diagonal correction is negligible, and summing equation 3 over pairs predicts the Gram error of a sketch from the exact cosines alone.

A.2 BASELINES AND PER-ROLLOUT NOISE

Score identity. For $\tau \sim \pi_\theta(\cdot \mid q)$,

$$\mathbb{E}[g_\theta(\tau)] = \sum_\tau \pi_\theta(\tau \mid q)\,\nabla_\theta \log \pi_\theta(\tau \mid q) = \nabla_\theta \sum_\tau \pi_\theta(\tau \mid q) = 0.$$

Hence for any baseline b that is conditionally independent of τ_i given q,

$$\mathbb{E}\big[(r_i - b)\, g_\theta(\tau_i)\big] = \mathbb{E}\big[r\, g_\theta(\tau)\big] = \nabla_\theta V(q).$$

Group mean. With $\hat{V} = \frac{1}{n}\sum_j r_j$ and the leave-one-out mean $\hat{V}_{-i} = \frac{1}{n-1}\sum_{j\neq i} r_j$,

$$r_i - \hat{V} = \frac{n-1}{n}\big(r_i - \hat{V}_{-i}\big),$$

so the group-mean advantage is $\frac{n-1}{n}$ times the RLOO advantage (Kool et al., 2019; Ahmadian et al., 2024), and $\mathbb{E}[\hat{g}_q] = \frac{n-1}{n}\nabla_\theta V(q)$.

Control variate. Writing the estimate as $\frac{1}{n}\sum_j r_j\, g_\theta(\tau_j) - b\,\bar{g}$, with $\bar{g}$ the group's mean score ($\mathbb{E}\,\bar{g} = 0$), the baseline is the coefficient of a control variate. The scalar that minimizes the trace of the covariance is

$$b^* = \frac{\sum_i \|g_\theta(\tau_i)\|^2\, r_i}{\sum_i \|g_\theta(\tau_i)\|^2}$$

(Greensmith et al., 2004), the min-variance baseline of Table 13; under $T^+ = T^-$ its expectation is V.

A.3 Proposition 1: GRPO Positive and Negative Gradients

Proposition 1. *With binary rewards and GRPO-style group-mean baselines, the gradient over successful trajectories is, up to a scaling factor, an unbiased estimate of the gradient of the* pass@2 *objective. The gradient over unsuccessful trajectories is, up to the same factor, an unbiased estimate of the gradient of the* all@2 *objective, the probability that two independently drawn trajectories are both correct.*

Proof. First, define the pass@2 and all@2 objectives.

$$J_{pass@2}(\pi) = \mathbb{E}_{q\sim p(q)}\left[1-(1-V^{\pi}(q))^2\right]$$
$$J_{all@2}(\pi) = \mathbb{E}_{q\sim p(q)}\left[V^{\pi}(q)^2\right]$$

As $p(q)$ does not depend on π, differentiating each objective yields:

$$\nabla_\theta J_{pass@2}(\pi_\theta) = 2\mathbb{E}_{q\sim p(q)}\left[(1-V^{\pi_\theta}(q))\,\nabla_\theta V^{\pi_\theta}(q)\right]$$
$$\nabla_\theta J_{all@2}(\pi_\theta) = 2\mathbb{E}_{q\sim p(q)}\left[V^{\pi_\theta}(q)\nabla_\theta V^{\pi_\theta}(q)\right]$$

Note that $\nabla_\theta V^{\pi_\theta}(q)$ is simply the policy gradient on prompt q under policy π_θ. This gradient is invariant to the choice of baseline, and thus can be estimated either from purely positive samples or purely negative samples (by setting the baseline $V=0$ or $V=1$, respectively).

Let $g_+(q)$ be the mean gradient over all positive samples, and $g_-(q)$ be the mean gradient over all negative samples, with the expectation taken with respect to π_θ.

$$g_+(q) = \mathbb{E}_{\substack{\tau_+\sim\pi_\theta(\tau_+|q)\\ r(\tau_+)=1}}\left[\nabla_\theta \log \pi_\theta(\tau_+|q)\right]$$
$$g_-(q) = \mathbb{E}_{\substack{\tau_-\sim\pi_\theta(\tau_-|q)\\ r(\tau_-)=0}}\left[\nabla_\theta \log \pi_\theta(\tau_-|q)\right]$$
$$\nabla_\theta V^{\pi_\theta}(q) = V^{\pi_\theta}(q)g_+(q) = -(1-V^{\pi_\theta}(q))g_-(q)$$

Substituting into the pass@2 and all@2 estimators and omitting the shared factor of 2, we obtain:

$$\nabla_\theta J_{pass@2}(\pi_\theta) = \mathbb{E}_{q\sim p(q)}\left[\left(V^{\pi_\theta}\left(1-V^{\pi_\theta}\right)(q)\right)g_+(q)\right]$$
$$\nabla_\theta J_{all@2}(\pi_\theta) = \mathbb{E}_{q\sim p(q)}\left[-\left(V^{\pi_\theta}\left(1-V^{\pi_\theta}\right)(q)\right)g_-(q)\right]$$

Noting that the GRPO gradient is simply $\hat{V}\left(1-\hat{V}\right)(\hat{g}_+ - \hat{g}_-)$, and that, with $k_+ \sim \mathrm{Bin}(k,V)$ successes, $\mathbb{E}[\hat{V}(1-\hat{V})\,\hat{g}_+] = \frac{k-1}{k}(1-V)\nabla V = \frac{k-1}{2k}\nabla\,\mathrm{pass@2}$ and $\mathbb{E}[-\hat{V}(1-\hat{V})\,\hat{g}_-] = \frac{k-1}{k}V\nabla V = \frac{k-1}{2k}\nabla\,\mathrm{all@2}$ exactly, we see that the successful-trajectory gradients estimate the pass@2 gradient, while the unsuccessful-trajectory gradients estimate the all@2 gradient.

A.4 Per-Prompt Reliability

Write the estimates from n and m disjoint rollouts of prompt q as

$$\hat{g}_q^{(n)} = \bar{g}_q + e_n, \qquad \hat{g}_q^{(m)} = \bar{g}_q + e_m, \qquad \mathbb{E}\|e_n\|^2 = \frac{B_q\,\|\bar{g}_q\|^2}{n},$$

with independent errors that have no preferred alignment with $\bar{g}_q$ or with each other. In high dimension,

$$\left\langle \hat{g}_q^{(n)}, \hat{g}_q^{(m)} \right\rangle \approx \|\bar{g}_q\|^2, \qquad \left\|\hat{g}_q^{(n)}\right\|^2 \approx \|\bar{g}_q\|^2\Big(1+\frac{B_q}{n}\Big),$$

which gives

$$R_q(n;m) \approx \big[(1+B_q/n)(1+B_q/m)\big]^{-1/2}. \tag{4}$$

Three cases matter:

- split halves ($m=n$): $R_q=(1+B_q/n)^{-1}$, which equals $\frac{1}{2}$ at $n=B_q$;
- a much larger reference ($m\to\infty$): $R_q=(1+B_q/n)^{-1/2}$, which equals $\frac{1}{2}$ at $n=B_q/3$;
- the depth-pool reference ($m=2048-n$): the crossing solves $(1+B_q/n)\big(1+B_q/(2048-n)\big)=4$.

A.5 Batch Noise and Optimal Allocation

Batch error. Prompt q receives n_q rollouts. Write $\hat{g}_q=\bar{g}_q+e_q$ with independent errors, $\mathbb{E}\,e_q=0$ and, for the group-mean baseline rescaled by $n_q/(n_q-1)$ to remove its bias,

$$\mathbb{E}\|e_q\|^2 \approx \frac{\sigma_q^2}{n_q-1}.$$

For $\hat{g}=\sum_q \hat{g}_q$ and $g=\sum_q \bar{g}_q$, independence gives

$$\mathbb{E}\|\hat{g}-g\|^2 \approx \sum_q \frac{\sigma_q^2}{n_q-1},$$

and if the error has no preferred alignment with g, $\cos^2(\hat{g},g)\approx \|g\|^2/\big(\|g\|^2+\mathbb{E}\|\hat{g}-g\|^2\big)$, which gives

$$\cos(\hat{g},g) \approx \left(1+\frac{\sum_q \sigma_q^2/(n_q-1)}{\big\|\sum_q \bar{g}_q\big\|^2}\right)^{-1/2}. \tag{5}$$

With equal allocation $n_q=n$, the numerator is $\sum_q \|\bar{g}_q\|^2 B_q/(n-1)=\bar{B}\sum_q\|\bar{g}_q\|^2/(n-1)$, so the relevant average noise scale is the norm-weighted mean $\bar{B}$.

Random prompts. Draw P prompts i.i.d. with n rollouts each, let $\hat{g}_{P,n}=\frac{1}{P}\sum_q \hat{g}_q$, and let $s^2=\mathbb{E}_q\|\bar{g}_q\|^2$ and $g=\mathbb{E}_q\bar{g}_q$, so that $\|g\|^2=\rho\, s^2$. Then

$$\mathbb{E}\|\hat{g}_{P,n}-g\|^2 \approx \frac{s^2}{P}\Big((1-\rho)+\frac{\bar{B}}{n-1}\Big),$$

which gives

$$\cos(\hat{g},g) \approx \left(1+\frac{(1-\rho)+\bar{B}/(n-1)}{\rho P}\right)^{-1/2}; \tag{6}$$

drawing P of N pool prompts without replacement multiplies the $(1-\rho)$ term by $(N-P)/(N-1)$. At a fixed budget $T=nP$ the error is proportional to

$$\frac{1}{T}\Big[(1-\rho)\,n+\bar{B}\,\frac{n}{n-1}\Big], \qquad \text{minimized at} \qquad (n^*-1)^2=\frac{\bar{B}}{1-\rho},$$

independently of T: the optimal subsample size of two-stage sampling (Cochran, 1977), and the form of the group-size result of Zhou et al. (2026).

Neyman allocation. Minimizing $\sum_q \sigma_q^2/m_q$ over $m_q=n_q-1$ subject to $\sum_q m_q=M$ gives, from the Lagrangian,

$$m_q = M\,\frac{\sigma_q}{\sum_{q'}\sigma_{q'}}, \qquad \min\sum_q \frac{\sigma_q^2}{m_q}=\frac{\big(\sum_q \sigma_q\big)^2}{M}.$$

Uniform allocation over N prompts gives $N\sum_q\sigma_q^2/M$, a ratio of $N\sum_q\sigma_q^2/\big(\sum_q\sigma_q\big)^2=1+\mathrm{CV}^2(\sigma_q)$. The Gram matrix enters only $\|g\|^2$, so the optimal allocation does not depend on it; under uniform prompt sampling only its trace and its total $\mathbf{1}^\top K\mathbf{1}$ enter equation 6, through the gradient diversity of Yin et al. (2018). With a cap $n_q\le n_{\max}$ the optimum is $m_q=\min(n_{\max}-1,\lambda\sigma_q)$, with λ set by the budget.

A.6 WEIGHTED TARGETS

For a target $T = \sum_q w_q \hat{g}_q$ over a fixed prompt set, with orthogonal prompt directions and noise independent across prompts and halves, the split-half reliability is

$$R_T \approx \frac{\sum_q u_q}{\sum_q u_q / R_q}, \qquad u_q = w_q^2 \|\bar{g}_q\|^2,$$

the u-weighted harmonic mean of the per-prompt reliabilities. Objectives that put weight on prompts with pass rates near 0 or 1, such as pass@k for large k ($w_q = k(1-V)^{k-1}$) and the log pass rate ($w_q = 1/V$), therefore inherit the low reliability of those prompts.

A.7 MOMENTUM-FILTERED NOISE

Let $g_t = G + \xi_t$ with ξ_t i.i.d., $\mathbb{E}\,\xi_t = 0$, $\mathbb{E}\|\xi_t\|^2 = \sigma^2$, and $m_t = \beta m_{t-1} + (1-\beta) g_t$.

Pure noise ($G = 0$). In the stationary regime,

$$\mathbb{E}\|m_t\|^2 = \sigma^2 \frac{1-\beta}{1+\beta}, \qquad \mathbb{E}\langle m_t, m_{t+k}\rangle = \beta^k\, \mathbb{E}\|m_t\|^2.$$

Each ξ_s enters $\Delta_T = \sum_{t \le T} m_t$ with total weight $1 - \beta^{T-s+1}$, so for $T \gg 1/(1-\beta)$,

$$\mathbb{E}\|\Delta_T\|^2 \approx T\sigma^2, \qquad \frac{\mathbb{E}\|\Delta_T\|^2}{\sum_t \mathbb{E}\|m_t\|^2} \approx \frac{1+\beta}{1-\beta}.$$

The square root of this ratio is the straight-line ratio of section 5. Since $\mathbb{E}\langle m_t, \Delta_T\rangle \approx \sigma^2$, one update's cosine with the displacement is about $\sqrt{(1+\beta)/(1-\beta)}/\sqrt{T}$, against $1/\sqrt{T}$ for one raw gradient. Adam's second-moment estimate changes slowly compared with $1/(1-\beta_1)$ steps, so the same holds for the applied update in preconditioned coordinates.

Projection onto the endpoint. For $t \gg 1/(1-\beta)$, $\mathbb{E}\langle \Delta_t, \Delta_T\rangle \approx t\sigma^2$ and $\mathbb{E}\|\Delta_T\|^2 \approx T\sigma^2$, so the normalized projection $\langle \Delta_t, \Delta_T\rangle / \|\Delta_T\|^2 \approx t/T$. Each step's increment has mean about σ^2 and relative fluctuation about $\sqrt{T(1-\beta)/((1+\beta)\, d_{\text{eff}})}$, where d_{eff} is the effective dimension of the noise. Under the null the trajectory therefore moves monotonically and nearly linearly along its own endpoint direction whenever $d_{\text{eff}} \gg T(1-\beta)/(1+\beta)$ (about 84 for $T = 1{,}600$, $\beta = 0.9$), and the per-step cosine with the endpoint is about $1/\sqrt{T}$. Monotone progress is therefore not evidence of directed motion.

B EXPERIMENTAL DETAILS

B.1 MODELS, DATA AND TRAINING

Table 3: Experimental settings. The probed checkpoints (§3 exact-gradient cells, §4) and the traced runs (§5, path validation) come from different training runs with different recipes.

Item	Setting
Models	Qwen3-0.6B-Base, Qwen3-1.7B-Base, Qwen3-4B-Base and Llama-3.1-8B-Instruct; for Llama, the "base" stage is the Instruct checkpoint before RL
Probed checkpoints (§3 cells, §4)	
Training	GRPO on DeepMath-103K (101,877 prompts after deduplication), 8 rollouts per prompt, clip 0.2/0.2, no length penalty, no dynamic-sampling filter; checkpoint t is $16t$ optimizer steps of 256 sequences (step 200: 3,200 steps).
Checkpoints	base, step 20 (initial rise in reward) and the reward plateau (step 200; step 90 for Llama-3.1-8B-Instruct)
Probe prompts	DAPO-Math-17k rows (breadth 256, depth 16), not in these checkpoints' training set
Probe sampling	temperature 1.0, at most 8,192 response tokens; breadth probes of 256 prompts $\times$ 512 rollouts per generation (8 strided shards of 32 prompts), depth probes of 16 prompts $\times$ 2,048 rollouts (4 shards of 4); the sketched stream is the token-level importance-weighted score gradient, untruncated
Traced runs (§5, path validation)	
Data	DAPO-Math-17k (deduplicated, not English-filtered); math verifier
Algorithm	GRPO without standard-deviation normalization; dual-clip PPO (ϵ 0.2/0.28, $c = 10$) with token-mean loss; token-level truncated importance sampling at 2; no KL
Sampling	temperature 1.0, top-p 1.0, at most 8,192 response and 2,048 prompt tokens
Batches	512 prompts $\times$ 16 rollouts per rollout batch, taken as 16 optimizer steps of 32 prompts
Optimizer	AdamW, learning rate 10^{-6}, 160 warmup steps, $\beta = (0.9, 0.999)$, weight decay 0.1, gradient clipping at 1.0
Run options	Qwen3-0.6B and 1.7B: DAPO overlong penalty (buffer 4,096) with overlong filtering and the dynamic-sampling filter; Qwen3-4B: neither, 3 epochs
Parallelism	FSDP with 8 ranks; rollouts from vLLM
Hardware	Nodes of 8 NVIDIA H100 GPUs; one node per training run

Step units. Checkpoints and validation paths are indexed by training step; exact checkpoint t matches optimizer step $16t - 1$ of the traces. Trajectory statistics are indexed by optimizer step.

B.2 SNAPSHOT POOLS

For each model and checkpoint, we collect a pool of rollouts and sketch each per-sequence gradient. In our largest collection run, we select 256 prompts from DAPO-Math-17k and collect 2048 rollouts for each prompt from each checkpoint. Rollouts are collected from the same harness used during training, from DAPO-Math-17k problems, a different dataset from the checkpoints' training data (DeepMath-103K). For checkpoints beyond the base model, we additionally collect the Adam-preconditioned gradient using the checkpoint's optimizer state. Prompts with uniform reward receive no gradient under GRPO, and are therefore excluded from Gram-matrix analysis.

Unless otherwise mentioned, sketch reliability studies are computed over 16 prompts with 2048 rollouts each; prompt allocation studies are computed over 256 prompts, with 2048 rollouts on Qwen3-4B and 1024 rollouts for all other models.

B.3 RELIABILITY CONVENTIONS

Per-prompt reliability is the expected cosine between the estimate from n of a prompt's rollouts and the estimate from its remaining rollouts, averaged over random draws. Under the *conditional* convention both sets must contain both outcomes; under the *unconditional* convention a zero estimate

counts as cosine 0, so the unconditional value includes the chance that a small group carries no signal. Aggregate targets use *split-half* reliability: one random split of every prompt's rollouts into two halves, shared by all targets. Where the median curve does not reach 0.5 within 512 rollouts, we extrapolate with a sigmoid in $\log n$ whose ceiling is fixed at 1. Against its remaining rollouts, a prompt whose noise scale exceeds half the pool cannot reach 0.5, so extrapolated values are indicative only; the main text quotes $n_{0.5}$ from the noise scale.

B.4 Trajectory Runs

We trace online GRPO on Qwen3-0.6B, 1.7B and 4B, and online and offline positives-only training on Qwen3-4B (positives only: advantage 1 on successful rollouts and 0 on failures; offline: plain SFT on a fixed set of the base model's successful rollouts). At every optimizer step we sketch the batch gradient g, the preconditioned gradient $g/\sqrt{\hat{v}}$, the momentum m and the applied update $\hat{m}/\sqrt{\hat{v}}$, flat (width 4,096, 16 KB each) and per group, with one hash table for all steps, and store their exact per-group norms. Matched comparisons use 1,575–1,600 optimizer steps.

- **GRPO:** Standard online GRPO on Qwen3-0.6B-Base, Qwen3-1.7B-Base and Qwen3-4B-Base
- **4B online/offline:** Qwen3-4B with online rejection-sampling (positives only) and offline SFT on positives only, to determine whether centered gradient coefficients or online samples govern the optimization geometry

C SKETCH VALIDATION

C.1 SKETCH LAYOUT

The per-group sketch applies an independent sketch of width 1,024 to each parameter group and concatenates the results. In the validation cells the groups are the blocks plus the embedding, head and final norm (31 groups at 0.6B, 39 at 4B; 3.2–4.0 $\times 10^4$ numbers per gradient). In the snapshot pools the attention, MLP and normalization of each block are separate groups (87 at 0.6B and 1.7B, 111 at 4B; 0.9–1.1 $\times 10^5$ numbers). A flat sketch of width 4,096 is stored alongside and is used for Table 9, the Llama-3.1-8B pool and all traces. Cosines are formed from sketched norms; exact per-group norms are stored for magnitudes.

C.2 FIDELITY METRICS

To validate sketches, we cache a set of full gradients for a small model and compute its normalized Gram matrix. For a set of gradients $g_1, \cdots, g_k$, the normalized Gram matrix is defined by $G_{ij} = \cos(g_i, g_j)$. Then, for various proposed sketches $S_1, \cdots, S_M$ we construct the approximate Gram matrix $\hat{G}_{ij}^{(m)} = \cos(S_m(g_i), S_m(g_j))$. For convenience, let $c_{ij} = \cos(g_i, g_j)$ and $\hat{c}_{ij} = \cos(S(g_i), S(g_j))$ for some sketch S.

We evaluate $\hat{G}$ candidates in two ways. First, we measure the root mean square of the off-diagonal residual between the Gram matrices (its Frobenius norm over $\sqrt{k(k-1)}$). We refer to this as the **Gram error**. Normalizing by the Frobenius norm of the true Gram matrix gives the relative Gram error J_{Gram}^{rel}. A null predictor that always predicts zero would achieve a relative Gram error of 1.

$$J_{Gram}(S_m) = \frac{\left\| G - \hat{G}^{(m)} \right\|_F}{\sqrt{k(k-1)}} = \sqrt{\frac{1}{k(k-1)} \sum_{i \neq j} \left(c_{ij} - \hat{c}_{ij}\right)^2}$$

$$J_{\mathrm{Gram}}^{\mathrm{rel}}(S_m) = \sqrt{\frac{\sum_{i \neq j} \left(c_{ij} - \hat{c}_{ij}\right)^2}{\sum_{i \neq j} c_{ij}^2}}$$

Next, we can measure how well an optimization trajectory's geometry is preserved by a given sketch. By caching the exact update norms along with their sketches, we can replay the trajectory from the starting parameters to any arbitrary point in the optimization sequence. We may then compute our estimated displacement from the original model parameters at a given checkpoint step, and compare it against the actual cached checkpoint's displacement. We formalize this cost below as the relative error of our sketch-estimated displacement norm against the true displacement norm, and term this metric the **path reconstruction error**. We similarly consider the relative error $J_{path}^{(t),rel}$.

$$\|\theta_t - \theta_0\|^2 = \sum_{i=1}^{t} \sum_{j=1}^{t} \|m_i\| \|m_j\| \cos(m_i, m_j),$$

$$\hat{L}_t(S_m) \triangleq \sqrt{\sum_{i=1}^{t} \sum_{j=1}^{t} \|m_i\| \|m_j\| \cos\left(S_m(m_i), S_m(m_j)\right)},$$

$$J_{\mathrm{path}}^{(t)}(S_m) \triangleq \left| \|\theta_t - \theta_0\| - \hat{L}_t(S_m) \right|; \qquad J_{\mathrm{path}}^{(t),\mathrm{rel}}(S_m) \triangleq \frac{J_{\mathrm{path}}^{(t)}(S_m)}{\|\theta_t - \theta_0\|}.$$

C.3 VALIDATION RESULTS

Stored exactly, the best single block has Gram error 0.18–0.85 depending on the cell (median block 0.22–1.06), against 0.09–0.71 for the all-parameter sketch (Figure 5). The only block that beats the sketch in any cell is block 2 of Qwen3-1.7B at step 200 (0.52 against 0.71). At step 200 the absolute

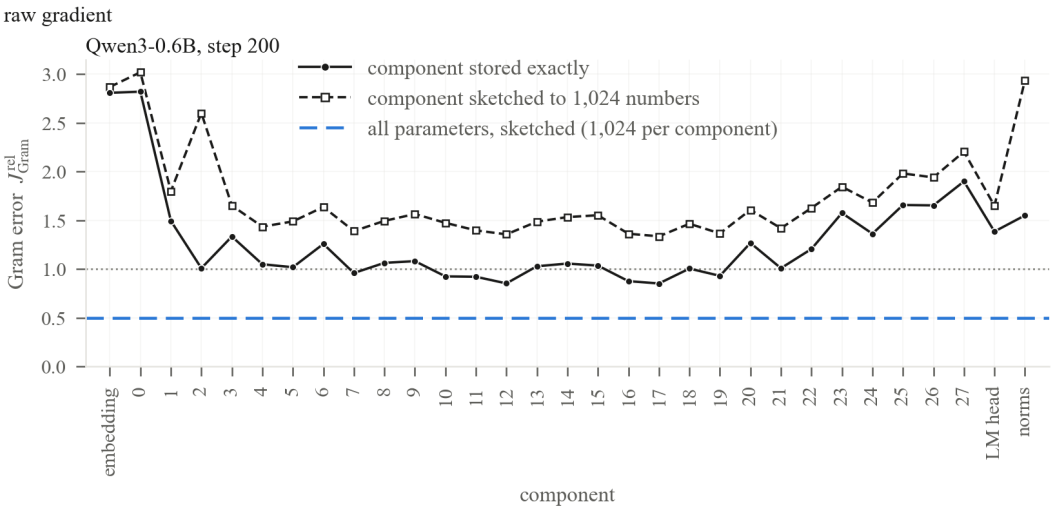


Figure 5: Proxies per-layer for the whole-model gradient geometry.

error of the sketch keeps falling with width while the partial-model proxies stay above the error of predicting zero; relative errors look large because the exact cosines are small (root mean square 0.03–0.06).

Table 4: Path reconstruction error at the final checkpoint, against the Adam-only checkpoint displacement: all parameters at 1,024 buckets per group, and the last block and head at the largest sketch width tested (3,072 and 1,024 numbers). The path sums the applied Adam updates scaled by the learning rate, after gradient clipping.

Run	All parameters	Last block	LM head
Qwen3-0.6B, to step 165	0.83%	2.61%	2.99%
Qwen3-1.7B, to step 250	0.43%	1.01%	25.3%
Qwen3-4B, to step 75	0.97%	1.48%	27.5%

Table 5: Gram error of the all-parameter count-sketch at four widths (buckets per parameter group; the stored vector has 31 groups for Qwen3-0.6B and 1.7B, 39 for Qwen3-4B and 35 for Llama-3.1-8B-Instruct, so 1,024 per group is 31,744, 39,936 and 35,840 numbers per trajectory), against the LM-head and last-block proxies stored exactly (every parameter of the subset: 156M, 311M, 389M, 525M numbers for the LM head and 16M, 50M, 101M, 218M for the last block of Qwen3-0.6B, 1.7B, 4B and Llama-3.1-8B-Instruct). C: exact full-parameter cosine Gram of the cell's 64 trajectories (32 for Llama-3.1-8B-Instruct, whose exact Gram holds every gradient in memory, so its absolute error sums over a quarter as many pairs); $\tilde{C}$: the estimate; both errors over off-diagonal entries. The relative error is the absolute error divided by $\|C\|_F$, which is small at step 200, so values above 1 there mean an error larger than the exact cosines themselves.

		all-parameter sketch buckets per group				stored exactly	
Model	Stage	16	64	256	1,024	lm_head	last block
Relative, $J^{\mathrm{rel}}_{\mathrm{Gram}}$							
Qwen3-0.6B	base	1.47	0.79	0.49	0.39	0.86	1.23
	step 20	0.33	0.19	0.12	0.09	0.37	0.47
	step 200	1.88	1.02	0.64	0.50	1.38	1.90
Qwen3-1.7B	step 200	2.76	1.52	0.92	0.71	1.45	1.50
Qwen3-4B	base	1.30	0.64	0.33	0.20	0.84	0.97
	step 200	1.93	1.01	0.58	0.39	1.40	1.93
Llama-3.1-8B-Instruct	step 90	2.52	1.39	0.74	0.46	1.30	1.83
Absolute: Frobenius norm $\|\tilde{C} - C\|_F$ *of the off-diagonal error; divided by* $\sqrt{64 \cdot 63}$ *it is the root-mean-square* J_{Gram}							
Qwen3-0.6B	base	4.15	2.24	1.39	1.10	2.43	3.48
	step 20	3.82	2.21	1.41	1.02	4.22	5.46
	step 200	4.18	2.28	1.43	1.11	3.08	4.22
Qwen3-1.7B	step 200	7.47	4.12	2.49	1.93	3.91	4.04
Qwen3-4B	base	4.95	2.43	1.27	0.78	3.21	3.69
	step 200	3.98	2.07	1.20	0.80	2.89	3.99
Llama-3.1-8B-Instruct	step 90	2.49	1.37	0.73	0.46	1.28	1.81

C.4 Adam-Preconditioned Gradients

At each checkpoint beyond the base model we also sketch the Adam-preconditioned gradient, $g/(\sqrt{\hat{v}} + \epsilon)$ with the checkpoint's own second-moment estimate, from the same backward pass; base models have no optimizer state of their own. The preconditioned results follow the raw-gradient trends. The all-parameter sketch validates better in preconditioned space, with relative Gram error 0.21 against 0.50 at 1,024 buckets per group, while the LM head and the last block stored exactly stay above predicting zero (Table 6). At 16 rollouts per prompt, the group-mean or the minimum-variance baseline is the most reliable, or within 0.01 of it, in every trained cell. Proposition 1 holds in both metrics: the corrected cosines of GRPO$^+$ with pass@2 and of GRPO$^-$ with all@2 are 0.97–1.03 in preconditioned space, against 0.97–1.02 raw. The gains of Neyman and pilot allocation over uniform agree with the raw values to within 0.06, except the pilot for Qwen3-0.6B at 2,048 rollouts (1.38 against 1.15). The best number of rollouts per prompt agrees within two Monte-Carlo standard errors at 25–32 of the 26–32 budgets per cell, except Qwen3-1.7B at step 20, where the preconditioned optimum moves to 256 rollouts per prompt (96 without the embedding group).

Table 6: Sketch validation in the raw and the Adam-preconditioned metric on the same trajectories (Qwen3-0.6B, step 200, 64 trajectories): relative Gram error $J^{\mathrm{rel}}_{\mathrm{Gram}}$ of the all-parameter sketch at 64, 256 and 1,024 buckets per parameter group, and of the LM head and the last block stored exactly, against each metric's own exact Gram; rms $|\cos|$: root mean square of the exact off-diagonal cosines. The Adam metric uses the checkpoint's own second-moment estimate.

		all parameters, sketched			stored exactly	
Metric	rms $\lvert\cos\rvert$	64	256	1,024	LM head	last block
raw gradient	0.0350	1.01	0.64	**0.50**	1.37	1.91
Adam-preconditioned	0.0326	0.89	0.46	**0.21**	2.18	1.27

D WITHIN-PROMPT NOISE

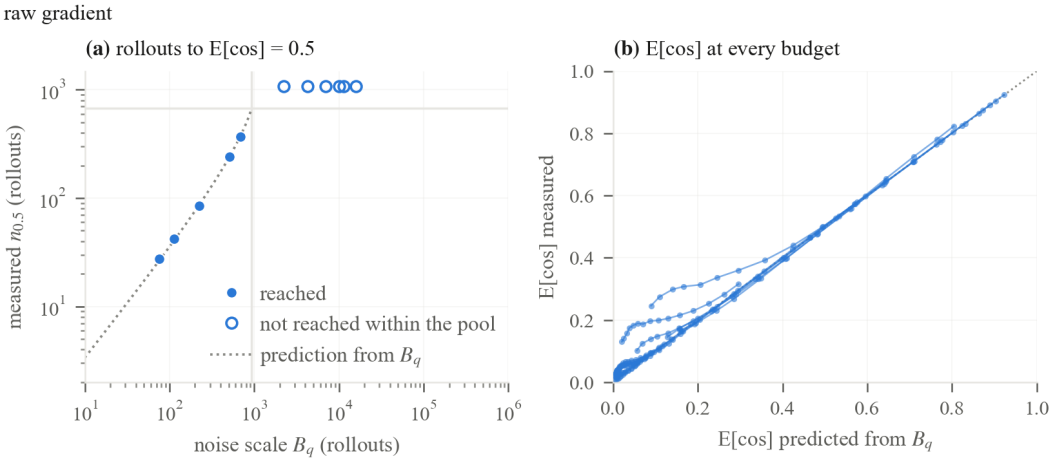


Figure 6: Per-prompt noise scale predicts per-prompt reliability, Qwen3-4B step 20. (a) Rollouts each prompt needs to reach cosine 0.5 with its remaining rollouts, against its noise scale B_q; hollow: not reached within the pool; dotted: the prediction of equation 4. (b) Measured against predicted cosine at every budget, one line per prompt.

D.1 NOISE SCALE AND PASS RATE

Per-prompt noise scales follow $B_q \approx \kappa/(V(1-V))$ with one constant per checkpoint (κ = 57–233; Figure 8). In log space the fit explains 41–76% of the variance across prompts, against a split-half reliability of $\log B_q$ itself of 0.58–0.90, so pass rate captures much of the predictable variation. Weighting by measured gradient norms, the mean noise scale is 1.7–3.5 times below the median prompt's at trained checkpoints and 0.9–2.5 times below it at the base model. At the base model, including the cross-prompt signal lowers the batch-level noise scale by a further factor of 19 to 59 (Table 8).

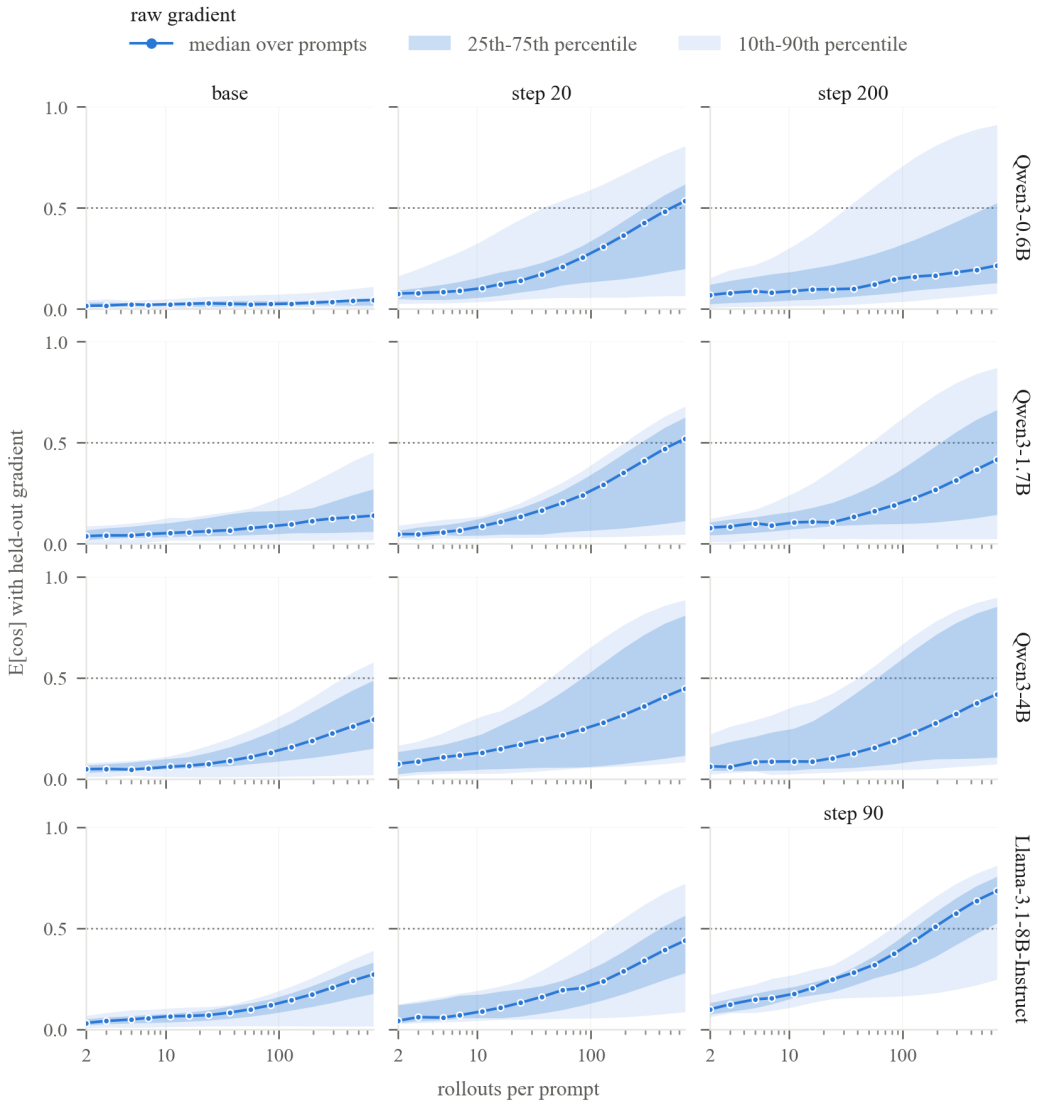


Figure 7: Per-prompt reliability against rollouts per prompt, every model and checkpoint: median over prompts with 25–75th and 10–90th percentile bands.

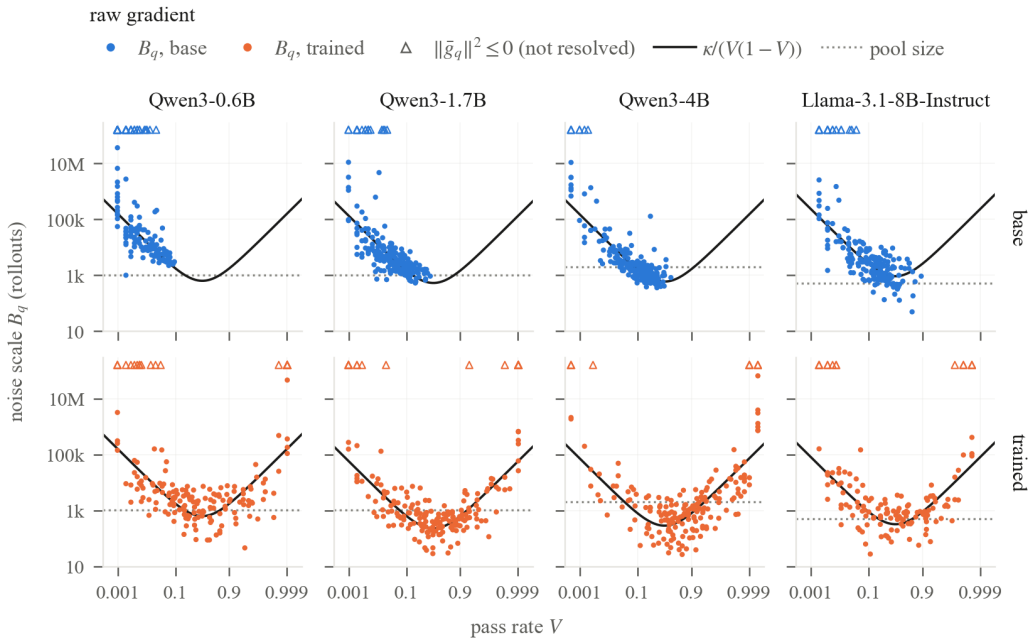


Figure 8: Per-prompt noise scale against pass rate, base model (top) and final checkpoint (bottom), with the fit $\kappa/(V(1-V))$ (line); triangles: prompts whose signal is not resolved ($\|\bar{g}_q\|^2$ estimate ≤ 0); dotted: pool size.

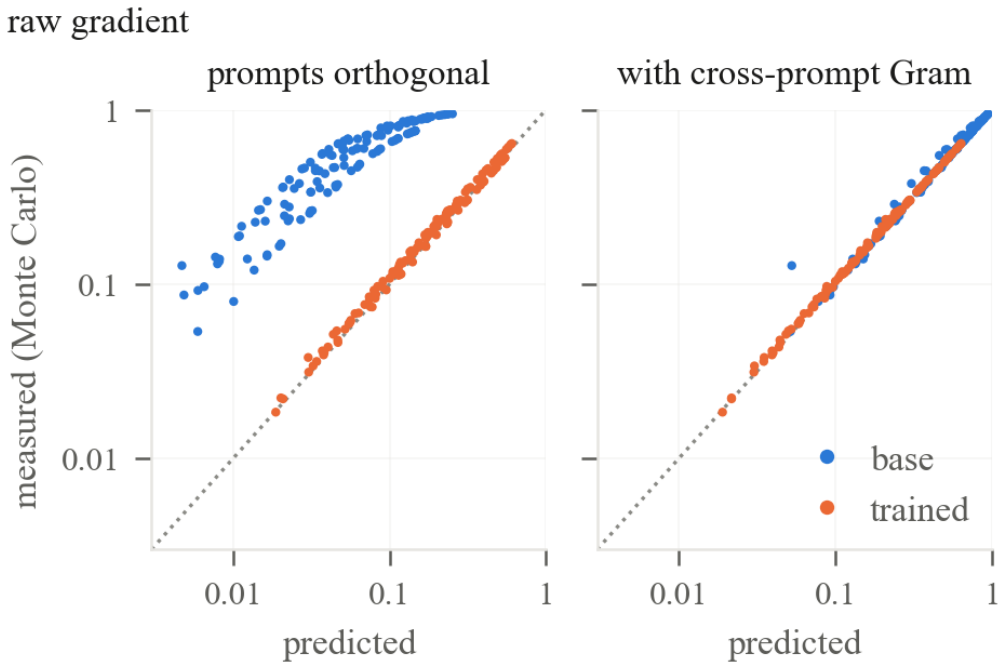


Figure 9: Batch agreement predicted by equation 5 against the Monte-Carlo value, for every checkpoint, allocation rule and budget: treating prompts as orthogonal (left), and with the measured cross-prompt Gram (right). Median relative errors per cell: Table 8.

Table 7: Median-prompt reliability of the per-prompt GRPO gradient estimate: $\mathrm{E}[\cos]$ between the estimate from b rollouts and the estimate from the prompt's remaining rollouts, median over prompts with both correct and incorrect rollouts (16 prompts, 2,048 rollouts each). The traced runs train with 16 rollouts per prompt; the probed checkpoints were trained with 8.

		rollouts per prompt								
Model	Stage	2	4	8	16	32	64	128	256	512
Conditional on both halves carrying signal										
Qwen3-0.6B	base	0.01	0.02	0.02	0.02	0.02	0.02	0.03	0.03	0.04
	step 20	0.06	0.07	0.09	0.12	0.16	0.22	0.30	0.40	0.50
	step 200	0.03	0.05	0.06	0.07	0.09	0.13	0.17	0.18	0.21
Qwen3-1.7B	base	0.06	0.06	0.05	0.06	0.07	0.08	0.10	0.13	0.14
	step 20	0.06	0.06	0.07	0.11	0.15	0.22	0.30	0.39	0.49
	step 200	0.07	0.07	0.08	0.10	0.13	0.18	0.22	0.30	0.39
Qwen3-4B	base	0.06	0.06	0.07	0.07	0.08	0.12	0.16	0.21	0.27
	step 20	0.10	0.10	0.12	0.15	0.18	0.22	0.28	0.34	0.42
	step 200	0.07	0.07	0.10	0.10	0.11	0.16	0.22	0.30	0.39
Llama-3.1-8B	base	0.04	0.05	0.06	0.07	0.08	0.11	0.15	0.20	0.25
	step 20	0.05	0.05	0.07	0.10	0.14	0.20	0.24	0.32	0.41
	step 90	0.13	0.14	0.15	0.20	0.27	0.34	0.44	0.55	0.66
Unconditional (a zero estimate counts as cosine 0)										
Qwen3-0.6B	base	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.01
	step 20	0.02	0.03	0.05	0.09	0.16	0.22	0.30	0.40	0.50
	step 200	0.00	0.01	0.01	0.03	0.05	0.07	0.11	0.15	0.20
Qwen3-1.7B	base	0.00	0.00	0.01	0.02	0.03	0.04	0.06	0.08	0.11
	step 20	0.01	0.03	0.05	0.10	0.14	0.21	0.29	0.38	0.48
	step 200	0.00	0.01	0.02	0.04	0.08	0.14	0.21	0.29	0.38
Qwen3-4B	base	0.01	0.02	0.03	0.05	0.07	0.10	0.15	0.20	0.25
	step 20	0.00	0.01	0.01	0.02	0.04	0.08	0.14	0.21	0.29
	step 200	0.01	0.04	0.06	0.09	0.11	0.16	0.22	0.30	0.39
Llama-3.1-8B	base	0.00	0.01	0.02	0.03	0.05	0.10	0.15	0.20	0.25
	step 20	0.01	0.02	0.04	0.06	0.10	0.14	0.19	0.26	0.35
	step 90	0.02	0.06	0.11	0.15	0.22	0.30	0.39	0.50	0.61

Table 8: Noise scales in rollouts: median per-prompt B_q, norm-weighted mean $\bar{B}$, and the batch-level noise scale including the cross-prompt signal; the pass-rate fit constant κ and its R^2 for $\log B_q$; and the median relative error (%) of the predicted batch agreement in Figure 9, from pass rate alone, from $\bar{B}$ with prompts treated as orthogonal, and with the cross-prompt signal.

			noise scale			pass-rate fit		error of prediction (%)		
Model	Stage	prompts	median B_q	$\bar{B}$	with Gram	κ	R^2	pass rate	orthogonal	Gram
Qwen3-0.6B	base	170	13,898	6,194	166	159	0.50	91.4	93.9	1.0
	step 200	180	2,494	886	817	158	0.41	8.3	6.6	0.9
Qwen3-1.7B	base	214	3,608	1,594	39	133	0.55	84.5	88.7	0.3
	step 200	196	718	279	304	57	0.62	3.0	3.3	1.7
Qwen3-4B	base	225	1,526	1,647	28	149	0.76	82.2	88.6	4.1
	step 200	191	1,236	356	308	72	0.60	14.7	9.7	1.8
Llama-3.1-8B	base	202	2,572	1,047	56	233	0.60	88.5	86.4	0.4
	step 90	139	1,011	587	560	81	0.47	5.3	8.9	5.0

E CROSS-PROMPT STRUCTURE

The spread of the cross-prompt cosines at step 200 is not sketch noise: widening the sketch from 4,096 to 113,664 shrinks its standard deviation at Qwen3-4B only from 0.033 to 0.031, as removing the sketch variance $1/D$ predicts. This orthogonality is not per-prompt estimation noise either: at trained checkpoints, the summed gradients of two disjoint halves of the prompts agree only at chance level, even when only prompts with reliable gradients are used (Table 10).

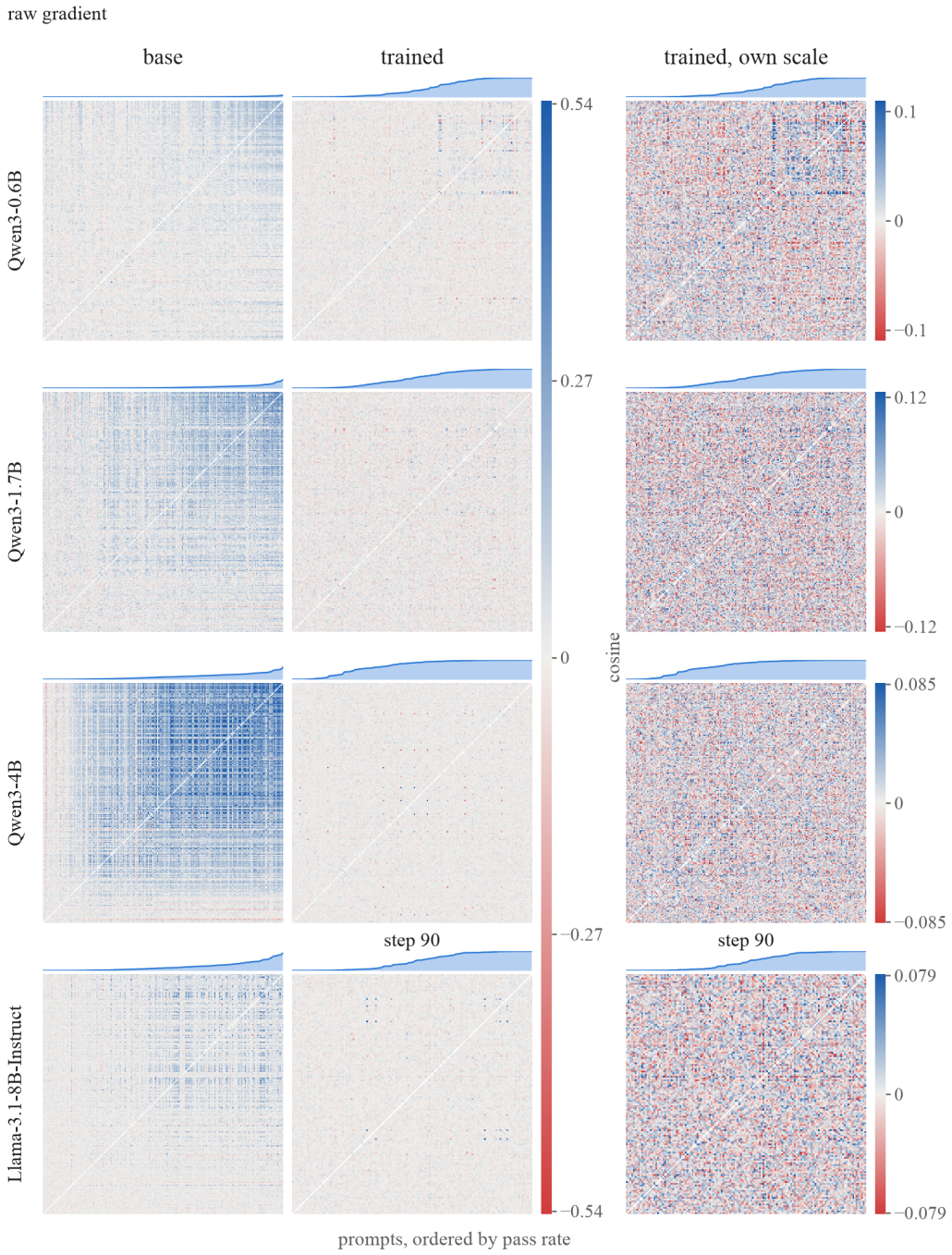


Figure 10: Cosines between per-prompt gradients for every model at the base model and the final checkpoint, one color scale for the left two columns, prompts ordered by pass rate (top strips). Right: the trained checkpoint again on its own color scale (its 99th percentile of $|\cos|$, 0.08–0.13, against 0.54 for the shared scale). At that scale the trained Grams show no structure beyond a small block of mutually aligned prompts in Qwen3-0.6B.

Table 9: Cross-prompt coherence: cosine between the GRPO gradients of different prompts, each estimated from its full pool of rollouts. Prompts: prompts with both correct and incorrect rollouts, of all prompts. Flat sketch of width 4,096, whose noise floor $1/\sqrt{4096} = 0.016$ is the typical $|\cos|$ between unrelated gradients.

Model	Stage	prompts	rollouts / prompt	mean cos	sd	pairs > 0
Qwen3-0.6B	base	170 / 256	1,024	0.0240	0.044	70%
	step 200	180 / 256	1,024	0.0006	0.040	50%
Qwen3-1.7B	base	214 / 256	1,024	0.0607	0.080	78%
	step 200	196 / 256	1,024	−0.0001	0.046	50%
Qwen3-4B	base	225 / 256	2,048	0.1362	0.140	85%
	step 200	191 / 256	2,048	0.0008	0.033	51%
Llama-3.1-8B	base	202 / 256	512	0.0227	0.055	66%
	step 90	139 / 256	512	0.0002	0.030	50%

Table 10: Cross-prompt agreement in aggregate. The prompts of a checkpoint are split at random into two halves and the per-prompt gradients of each half are summed; r is the cosine between the two sums, averaged over 2,000 splits. Each per-prompt gradient uses the prompt's full pool of rollouts (breadth collection; flat sketch of width 4,096). n: prompts with both outcomes. Chance: 97.5% quantile of r when each prompt's gradient is given a random sign. Reliable prompts: prompts whose within-prompt split-half cosine (between the estimates from two random halves of the prompt's rollouts, averaged over 32 splits) is at least 0.25; – where fewer than four prompts qualify. Same count: r on random subsets of all prompts of the reliable set's size.

		all prompts			reliable prompts (split-half ≥ 0.25)			
Model	Stage	n	r	chance	n	r	chance	same count
Per-prompt directions (unit vectors)								
Qwen3-0.6B	base	170	0.68	0.10	1	–	–	–
	step 200	180	0.05	0.08	55	−0.01	0.11	0.01
Qwen3-1.7B	base	214	0.88	0.19	43	0.80	0.35	0.58
	step 200	196	−0.01	0.09	125	−0.03	0.10	0.00
Qwen3-4B	base	225	0.95	0.36	162	0.96	0.41	0.93
	step 200	191	0.07	0.07	131	0.06	0.07	0.05
Llama-3.1-8B	base	202	0.70	0.12	35	0.70	0.31	0.28
	step 90	139	0.02	0.06	51	0.02	0.09	0.01
Per-prompt GRPO gradients (the batch gradient)								
Qwen3-0.6B	base	170	0.74	0.21	1	–	–	–
	step 200	180	0.02	0.10	55	0.03	0.13	0.01
Qwen3-1.7B	base	214	0.91	0.33	43	0.83	0.45	0.64
	step 200	196	−0.03	0.10	125	−0.04	0.10	−0.02
Qwen3-4B	base	225	0.95	0.42	162	0.96	0.48	0.94
	step 200	191	0.06	0.09	131	0.06	0.09	0.04
Llama-3.1-8B	base	202	0.76	0.19	35	0.68	0.35	0.34
	step 90	139	0.01	0.06	51	0.04	0.08	0.00

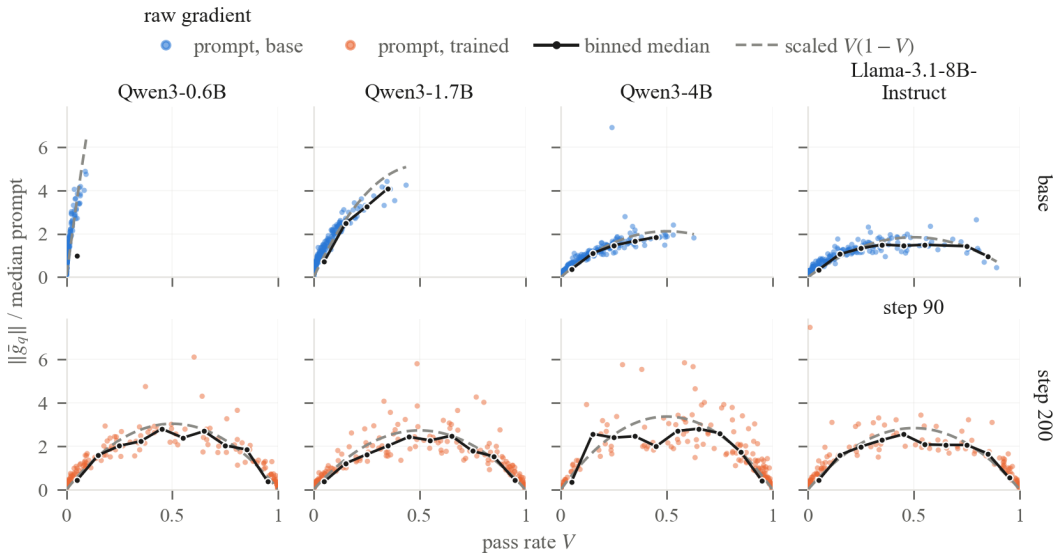


Figure 11: Per-prompt gradient norm, relative to the median prompt, against pass rate, with binned medians and a scaled $V(1-V)$.

F ROLLOUT ALLOCATION

Beyond calculating gradient variance, the set of sketched gradients allows for the simulation of various sampling approaches. For example, for a given budget of rollouts and a randomly drawn set of problems, we can sweep through proposed allocations of rollouts per prompt and see which allocations best approximate the high-budget held-out target.

Table 11: Best rollouts per prompt: median [range] over budgets with an interior optimum. Final: step 200 (Llama-3.1-8B: step 90).

Model	Base	Final
Qwen3-0.6B	28 [16, 128]	16 [8, 32]
Qwen3-1.7B	24 [16, 24]	12 [8, 16]
Qwen3-4B	16 [12, 24]	14 [12, 32]
Llama-3.1-8B	16 [12, 24]	16 [4, 32]

Score norms of successful and failed rollouts: Table 12.

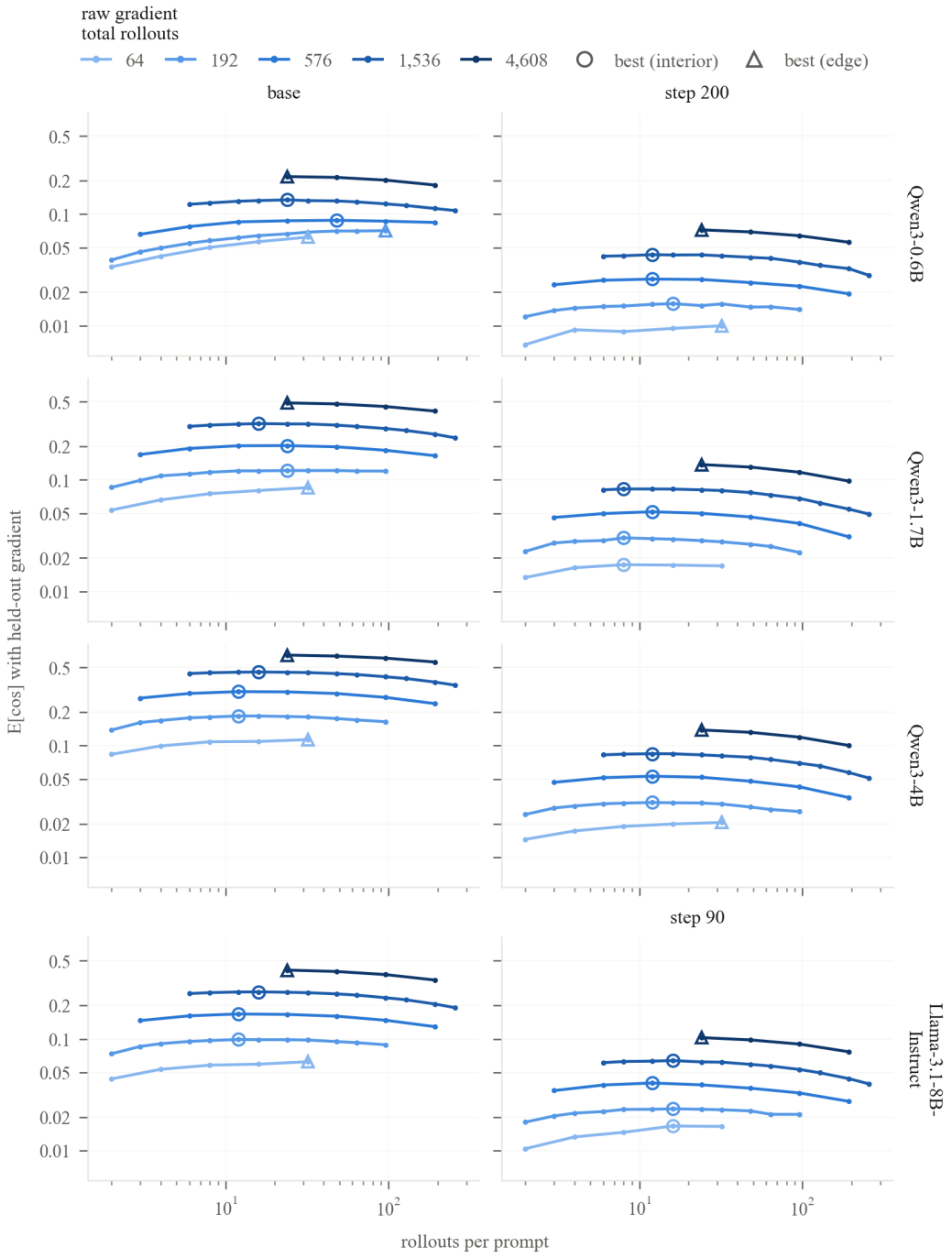


Figure 12: Agreement against rollouts per prompt at fixed budgets, every model at the base model and the final checkpoint.

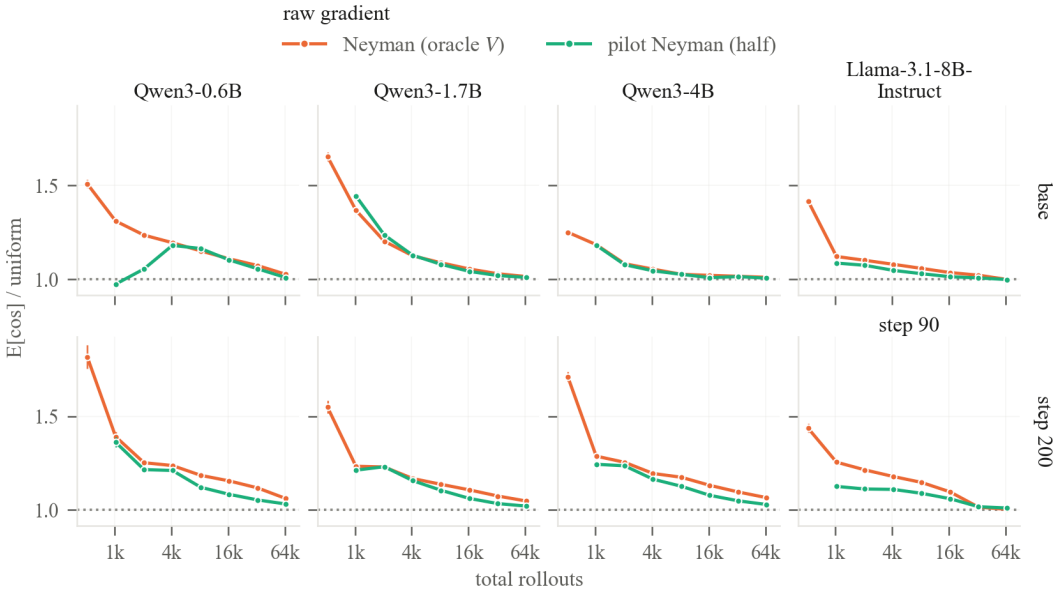


Figure 13: Gain in expected cosine over uniform allocation for Neyman allocation with oracle pass rates and for the two-stage pilot rule (half the budget uniform, the rest by the pilot $\hat{V}$).

Table 12: Do successful and failed rollouts have the same score norm? $T^{\pm} = \mathrm{E}[\|g_\theta(\tau)\|^2 \mid r = 1/0]$ per prompt, from the exact per-rollout norms (breadth cells, 512 rollouts per prompt, prompts with at least 5 of each outcome): median and interquartile range over prompts, the share of prompts with $T^+ > T^-$, the median ratio of norms per response token, and the interquartile ratio of $T_q = (1-V)T^+ + VT^-$.

Model	Stage	prompts	T^+/T^- [IQR]	$T^+ > T^-$	per token	IQR ratio of T_q
Qwen3-0.6B	base	64	0.52 [0.39, 0.68]	8%	1.01	1.74
	step 200	134	0.91 [0.78, 1.03]	30%	0.97	2.28
Qwen3-1.7B	base	155	0.60 [0.50, 0.70]	2%	0.71	1.46
	step 200	155	0.91 [0.80, 1.01]	26%	0.92	1.58
Qwen3-4B	base	192	0.81 [0.68, 0.92]	12%	0.42	1.41
	step 200	140	0.84 [0.69, 1.03]	26%	1.00	2.06
Llama-3.1-8B-Instruct	base	171	0.88 [0.71, 1.02]	28%	0.95	1.63
	step 90	109	0.95 [0.77, 1.08]	38%	1.01	1.67

G Baselines and Objectives

G.1 Baselines

Table 13 gives per-prompt reliability by baseline at Qwen3-4B step 20, Table 14 the number of rollouts at which each baseline reaches cosine 0.5 on every checkpoint, and Table 15 the reliability at the training group size of 16 on every checkpoint. The reliabilities of the group mean and the min-variance baseline agree to within 0.01 in Table 13 and Table 15. At 16 rollouts the group mean is more reliable than every fixed baseline at every trained checkpoint; at the base models, where most pass rates are near 0, the fixed baseline $V = 0$ does as well or slightly better, as the per-rollout noise under a fixed baseline b, $V(1-V) + (V-b)^2$ in units of the squared score norm, predicts.

G.2 Objectives

All objectives we compare are functions of each prompt's pass rate V, so their gradients are reweightings $\sum_q w_q \nabla V_q$ of the same per-prompt directions (section 2): pass@k, $1-(1-V)^k$, has $w = k(1-V)^{k-1}$; all@2, V^2, has $w = 2V$; the success and failure parts of GRPO have $w \propto 1-V$ and $w \propto V$ (Proposition 1); and GRPO with standard-deviation normalization divides each prompt's estimate by the group's reward standard deviation, which turns the objective into an arcsine transform of V (Davis & Recht, 2025). MaxRL and TailRL are defined as follows.

- **MaxRL:** Tajwar et al. (2026) maximize the log pass rate through its expansion $\log V = -\sum_{k\geq 1}(1-V)^k/k$. Truncating at T terms gives the objective $\sum_{k=1}^{T} \frac{1}{k}\,\mathrm{pass@}k$, whose gradient is $\frac{1-(1-V)^T}{V}\,\nabla V$: each prompt is weighted by about $1/V$ once T is large, which favors hard prompts. We use $T = 16$. This is the expected update of the authors' estimator with groups of T rollouts, which divides the group-centred advantage by the group's mean reward: when a group has at least one success, the estimate is the mean score of its successful rollouts minus the mean score of all rollouts, with expectation $\Pr[\text{at least one success}]\,\mathbb{E}[g_\theta(\tau) \mid r=1] = \frac{1-(1-V)^T}{V}\nabla_\theta V$.
- **TailRL:** Ramasubramanian et al. (2026) maximize, for a randomly drawn reward threshold, the log-probability that a rollout's reward exceeds it; its gradient is a mixture of best-of-k gradients that weights rare high-reward rollouts. With binary rewards every threshold in $(0, 1]$ defines the same event $r = 1$, so the objective is $\log V$, the $T \to \infty$ limit of MaxRL. We report this shared limit as $\log p$; its gradient $\nabla V/V$ weights each prompt by $1/V$.

At the plateau, GRPO has cosine 0.68–0.80 with MaxRL ($T = 16$), 0.20–0.42 with the log pass rate and 0.78–0.94 with its standard-deviation-normalized variant; at the base model all three are at 0.65 or above (Table 17). The log pass rate, which weights prompts by $1/V$, is the least reliable objective in every cell (0.02–0.13 at the plateau), standard-deviation normalization lowers the reliability of GRPO at every plateau checkpoint, and all@2 is within 0.01 of GRPO or more reliable at every plateau checkpoint (Table 16). At the plateau pass@8 and all@2 are orthogonal (-0.06 to 0.07; Table 17), and at every plateau checkpoint pass@k reliability falls with k (0.76, 0.68 and 0.54 for $k = 2, 4, 8$ at Qwen3-4B), because the pass@k weight concentrates on the few prompts with low pass rates, which are the noisiest (subsection A.6).

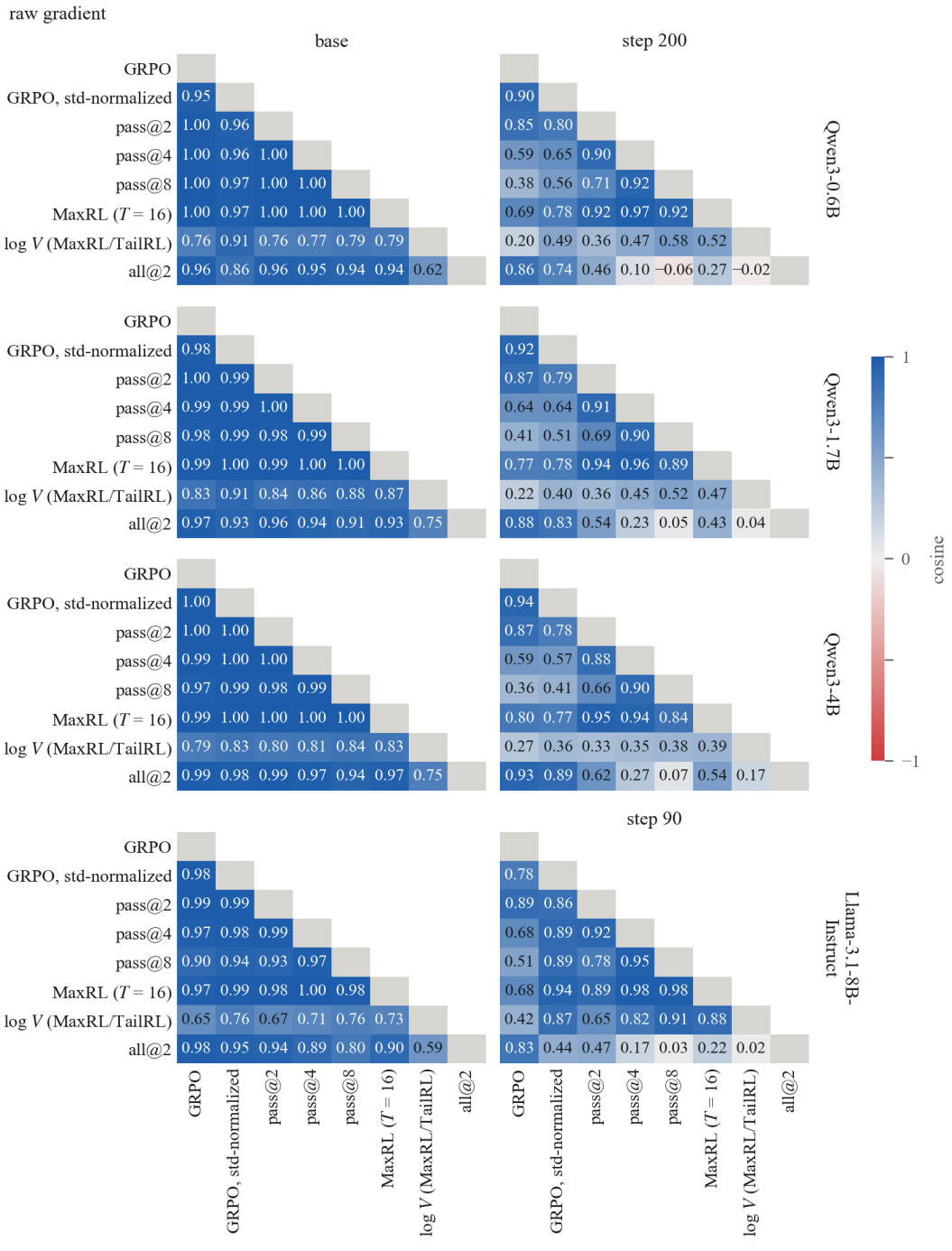


Figure 14: Cosines between the aggregate gradients of the objectives, every model at the base model and the final checkpoint. All objectives agree at the base model; at the plateau pass@8 and all@2 are orthogonal.

Table 13: Per-prompt reliability of policy-gradient estimators that differ in their baseline, Qwen3-4B step 20: $\mathrm{E}[\cos]$ between the estimate from b of a prompt's rollouts and the GRPO estimate from its remaining rollouts, median over prompts with both outcomes (13 prompts, 2,048 rollouts each). All estimators have the same expected gradient up to scale, except the min-variance baseline, whose weights include each rollout's own score norm, so it holds only approximately. Min-variance: the reward mean weighted by squared per-rollout score norms; fixed b: a constant baseline.

	rollouts per prompt								
Estimator	2	4	8	16	32	64	128	256	512
Conditional: both halves contain both outcomes									
group mean (GRPO)	0.10	0.10	0.12	0.15	0.18	0.22	0.28	0.34	0.42
min-variance	0.10	0.10	0.12	0.15	0.18	0.22	0.28	0.34	0.42
fixed b=0 (successes only)	0.03	0.05	0.05	0.06	0.08	0.12	0.16	0.22	0.29
fixed b=0.25	0.08	0.08	0.08	0.09	0.10	0.11	0.14	0.18	0.24
fixed b=0.5	0.10	0.09	0.07	0.06	0.08	0.10	0.14	0.19	0.25
fixed b=0.75	0.07	0.07	0.04	0.05	0.06	0.08	0.10	0.14	0.19
fixed b=1 (failures only)	0.05	0.06	0.07	0.07	0.08	0.09	0.11	0.15	0.20
Unconditional: a zero estimate counts as cosine 0									
group mean (GRPO)	0.00	0.01	0.01	0.02	0.04	0.08	0.14	0.21	0.29
min-variance	0.01	0.01	0.01	0.02	0.04	0.08	0.14	0.21	0.29
fixed b=0 (successes only)	0.00	0.01	0.01	0.03	0.04	0.08	0.14	0.21	0.28
fixed b=0.25	0.01	0.01	0.02	0.02	0.04	0.05	0.07	0.09	0.11
fixed b=0.5	0.01	0.01	0.02	0.02	0.03	0.05	0.06	0.08	0.10
fixed b=0.75	0.01	0.01	0.01	0.02	0.03	0.04	0.05	0.07	0.08
fixed b=1 (failures only)	0.01	0.01	0.01	0.02	0.03	0.06	0.09	0.13	0.17

Table 14: Rollouts per prompt at which the median prompt's reliability reaches $\mathrm{E}[\cos] = 0.5$, for each baseline method on every checkpoint (depth collection, 2,048 rollouts per prompt; cosine to the GRPO estimate from the prompt's remaining rollouts). Group baselines: GRPO (group mean) and MV (min-variance: $\|\phi\|^2$-weighted group mean); fixed V: the constant baseline V. † extrapolated past 512 rollouts with a sigmoid in $\log b$ whose ceiling is fixed at 1; $>10^4$ when that extrapolation exceeds 10^4; – not estimable (the median never exceeds 0.05). Bold: the highest measured median reliability at 512 rollouts on that checkpoint (within 0.005).

		group baseline		fixed V				
Model	Stage	GRPO	MV	0	.25	.5	.75	1
Conditional: both halves contain both outcomes								
Qwen3-0.6B	base	–	–	–	–	–	–	–
	step 20	505	**480**	3,100†	1,100†	3,000†	6,200†	$>10^4$
	step 200	$>10^4$	$>10^4$	$>10^4$	$>10^4$	$>10^4$	–	$>10^4$
Qwen3-1.7B	base	$>10^4$	$>10^4$	$>10^4$	$>10^4$	$>10^4$	–	–
	step 20	**560**†	**550**†	1,800†	930†	1,500†	850†	1,300†
	step 200	**1,600**†	**1,600**†	1,500†	1,800†	5,700†	$>10^4$	7,100†
Qwen3-4B	base	**6,300**†	**9,800**†	$>10^4$	$>10^4$	$>10^4$	$>10^4$	$>10^4$
	step 20	**1,300**†	**1,300**†	3,200†	$>10^4$	$>10^4$	$>10^4$	$>10^4$
	step 200	**1,600**†	**1,500**†	1,700†	**1,300**†	2,000†	7,300†	$>10^4$
Llama-3.1-8B-Instruct	base	**7,000**†	**7,800**†	9,400†	$>10^4$	$>10^4$	$>10^4$	$>10^4$
	step 20	**1,100**†	**1,100**†	5,300†	2,800†	8,100†	$>10^4$	3,200†
	step 90	**187**	**185**	970†	465	313	275	281
Unconditional: a zero estimate counts as cosine 0								
Qwen3-0.6B	base	–	–	–	–	–	–	–
	step 20	490†	**488**	1,800†	1,000†	2,400†	4,600†	8,400†
	step 200	**5,200**†	**5,300**†	8,500†	$>10^4$	$>10^4$	–	$>10^4$
Qwen3-1.7B	base	$>10^4$	$>10^4$	$>10^4$	–	–	–	–
	step 20	**560**†	**560**†	3,200†	1,600†	1,600†	1,800†	2,800†
	step 200	**940**†	**930**†	7,100†	$>10^4$	$>10^4$	7,700†	$>10^4$
Qwen3-4B	base	**3,400**†	**3,600**†	5,000†	$>10^4$	$>10^4$	$>10^4$	$>10^4$
	step 20	**1,500**†	**1,500**†	1,700†	$>10^4$	$>10^4$	$>10^4$	5,900†
	step 200	**1,100**†	**1,100**†	1,600†	**1,200**†	1,500†	2,900†	5,400†
Llama-3.1-8B-Instruct	base	**2,600**†	**2,700**†	3,000†	$>10^4$	$>10^4$	$>10^4$	$>10^4$
	step 20	**1,400**†	**1,400**†	4,100†	3,200†	5,200†	$>10^4$	3,800†
	step 90	**253**	**246**	1,800†	860†	327	550†	337

Table 15: Per-prompt reliability of the estimators of Table 13 at 16 rollouts per prompt (the traced runs' group size; the probed checkpoints were trained with 8), on every checkpoint; median over prompts with both outcomes. Fixed b: constant baseline ($b = 0$: successes only, $b = 1$: failures only). Llama-8B-Instruct: Llama-3.1-8B-Instruct.

	Qwen3-0.6B			Qwen3-1.7B			Qwen3-4B			Llama-8B-Instruct		
Estimator	base	step 20	step 200	base	step 20	step 200	base	step 20	step 200	base	step 20	step 90
Conditional: both halves contain both outcomes												
group mean	0.02	0.12	0.07	0.06	0.11	0.10	0.07	0.15	0.10	0.07	0.10	0.20
min-variance	0.03	0.13	0.08	0.07	0.11	0.10	0.08	0.15	0.09	0.07	0.10	0.20
b=0	0.03	0.09	0.06	0.07	0.07	0.08	0.08	0.06	0.08	0.07	0.06	0.08
b=0.25	0.01	0.10	0.04	0.04	0.08	0.08	0.06	0.09	0.09	0.06	0.06	0.12
b=0.5	0.00	0.08	0.03	0.03	0.08	0.05	0.03	0.06	0.07	0.03	0.07	0.14
b=0.75	0.00	0.05	0.02	0.01	0.10	0.05	0.02	0.05	0.07	0.02	0.09	0.19
b=1	0.00	0.04	0.02	0.01	0.08	0.04	0.02	0.07	0.09	0.01	0.07	0.19
Unconditional: a zero estimate counts as cosine 0												
group mean	0.00	0.09	0.03	0.02	0.10	0.04	0.05	0.02	0.09	0.03	0.06	0.15
min-variance	0.00	0.10	0.03	0.02	0.10	0.04	0.05	0.02	0.09	0.03	0.06	0.15
b=0	0.00	0.08	0.03	0.02	0.05	0.01	0.05	0.03	0.08	0.03	0.05	0.06
b=0.25	0.00	0.09	0.01	0.01	0.06	0.03	0.04	0.02	0.09	0.03	0.05	0.09
b=0.5	0.00	0.07	0.01	0.01	0.07	0.03	0.02	0.02	0.07	0.02	0.04	0.13
b=0.75	0.00	0.05	0.01	0.01	0.07	0.03	0.01	0.02	0.05	0.02	0.03	0.11
b=1	0.00	0.04	0.02	0.00	0.06	0.03	0.01	0.02	0.04	0.01	0.02	0.09

Table 16: Split-half reliability of each target's aggregate gradient, every breadth cell (one random split of each prompt's rollouts, shared by all targets). Pool sizes differ between models (512–2,048 rollouts per prompt), so values are comparable within a column.

	Qwen3-0.6B		Qwen3-1.7B		Qwen3-4B		Llama-3.1-8B	
Target	base	step 200	base	step 200	base	step 200	base	step 90
GRPO (group mean)	0.76	0.43	0.92	0.64	0.97	0.78	0.83	0.33
GRPO, std-normalized	0.67	0.32	0.90	0.54	0.97	0.67	0.78	0.13
Fixed $b = 0.5$	0.53	0.18	0.59	0.42	0.94	0.56	0.74	0.11
Fixed b, pooled (batch mean)	0.76	0.20	0.92	0.42	0.97	0.55	0.79	0.12
Successes, group-mean baseline	0.76	0.32	0.92	0.51	0.96	0.65	0.76	0.20
Failures, group-mean baseline	0.20	0.40	0.58	0.56	0.90	0.69	0.69	0.27
pass@2	0.76	0.39	0.92	0.65	0.97	0.76	0.80	0.25
pass@4	0.75	0.38	0.92	0.57	0.97	0.68	0.73	0.17
pass@8	0.75	0.32	0.90	0.44	0.96	0.54	0.60	0.11
all@2	0.72	0.50	0.88	0.64	0.97	0.77	0.82	0.37
MaxRL, $T = 16$	0.75	0.34	0.91	0.54	0.97	0.66	0.73	0.13
$\log p$ (MaxRL/TailRL limit)	0.45	0.11	0.67	0.13	0.74	0.09	0.30	0.02

Table 17: Cosine between the aggregate gradients of pairs of target objectives, on every model and stage with a breadth collection (256 prompts each). GRPO positives / negatives: the positive / negative rollouts' terms of GRPO, each keeping the GRPO baseline.

	Qwen3-0.6B		Qwen3-1.7B		Qwen3-4B		Llama-3.1-8B	
Target pair	base	step 200	base	step 200	base	step 200	base	step 90
GRPO positives vs pass@2	0.99	0.88	0.99	0.92	1.00	0.95	0.94	0.93
GRPO negatives vs all@2	0.35	0.90	0.87	0.94	0.97	0.97	0.85	0.90
GRPO positives vs GRPO negatives	0.22	0.17	0.75	0.37	0.93	0.51	0.58	0.23
GRPO vs GRPO, std-normalized	0.95	0.90	0.98	0.92	1.00	0.94	0.98	0.78
GRPO vs pass@2	1.00	0.85	1.00	0.87	1.00	0.87	0.99	0.89
GRPO vs pass@4	1.00	0.59	0.99	0.64	0.99	0.59	0.97	0.68
GRPO vs pass@8	1.00	0.38	0.98	0.41	0.97	0.36	0.90	0.51
GRPO vs MaxRL, $T{=}16$	1.00	0.69	0.99	0.77	0.99	0.80	0.97	0.68
GRPO vs $\log p$ (MaxRL, TailRL)	0.76	0.20	0.83	0.22	0.79	0.27	0.65	0.42
GRPO vs all@2	0.96	0.86	0.97	0.88	0.99	0.93	0.98	0.83
pass@8 vs $\log p$ (MaxRL, TailRL)	0.79	0.58	0.88	0.52	0.84	0.38	0.76	0.91
pass@8 vs all@2	0.94	−0.06	0.91	0.05	0.94	0.07	0.80	0.03

Table 18: Estimability of each target objective, Qwen3-4B step 200. Split-half cosine: cosine between the target's aggregate gradient on two disjoint halves of every prompt's rollouts (one random split per prompt, shared by all targets). Norm: aggregate gradient norm relative to GRPO.

Target	split-half cosine	norm / GRPO
GRPO	0.776	1.00
Fixed $V{=}0$ (positives only)	0.352	1.33
Fixed $V{=}0.25$	0.475	1.19
Fixed $V{=}0.5$	0.565	1.12
Fixed $V{=}0.75$	0.540	1.13
Fixed $V{=}1$ (negatives only)	0.426	1.23
Fixed V, pooled	0.551	1.12
GRPO positives	0.646	0.49
GRPO negatives	0.695	0.66
pass@2	0.757	0.48
pass@4	0.682	0.24
pass@8	0.535	0.12
all@2	0.769	0.63
GRPO, std-normalized	0.674	2.66
MaxRL, $T{=}16$	0.664	2.79
$\log p$ (MaxRL, TailRL)	0.087	11.90

Table 19: Cosine between the aggregate gradients of the target objectives, Qwen3-4B step 200 (256 prompts, 2,048 rollouts each). Column (j) is the target in row (j); (1) is GRPO.

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
(1) GRPO	–	0.94	0.87	0.59	0.36	0.80	0.27	0.93	0.83	0.91
(2) GRPO, std-normalized		–	0.78	0.57	0.41	0.77	0.36	0.89	0.74	0.87
(3) pass@2			–	0.88	0.66	0.95	0.33	0.62	0.95	0.61
(4) pass@4				–	0.90	0.94	0.35	0.27	0.84	0.27
(5) pass@8					–	0.84	0.38	0.07	0.63	0.08
(6) MaxRL, $T{=}16$						–	0.39	0.54	0.91	0.54
(7) $\log p$ (MaxRL, TailRL)							–	0.17	0.31	0.18
(8) all@2								–	0.59	0.97
(9) GRPO positives									–	0.51
(10) GRPO negatives										–

G.3 Successful and Failed Rollouts

Across disjoint halves of each prompt's rollouts (Table 20), with each target's split-half reliability divided out, the pairs of Proposition 1 have cosines of 0.97–1.02 at every trained checkpoint.

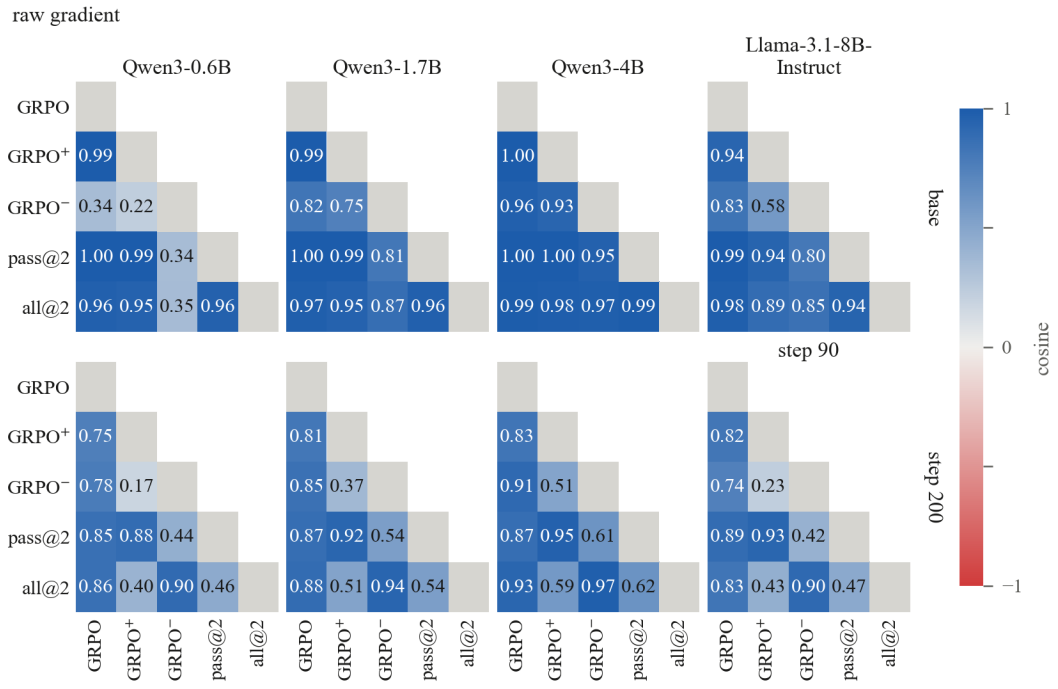


Figure 15: Cosines between GRPO, its positive and negative parts, pass@2 and all@2, every model at the base model and the final checkpoint.

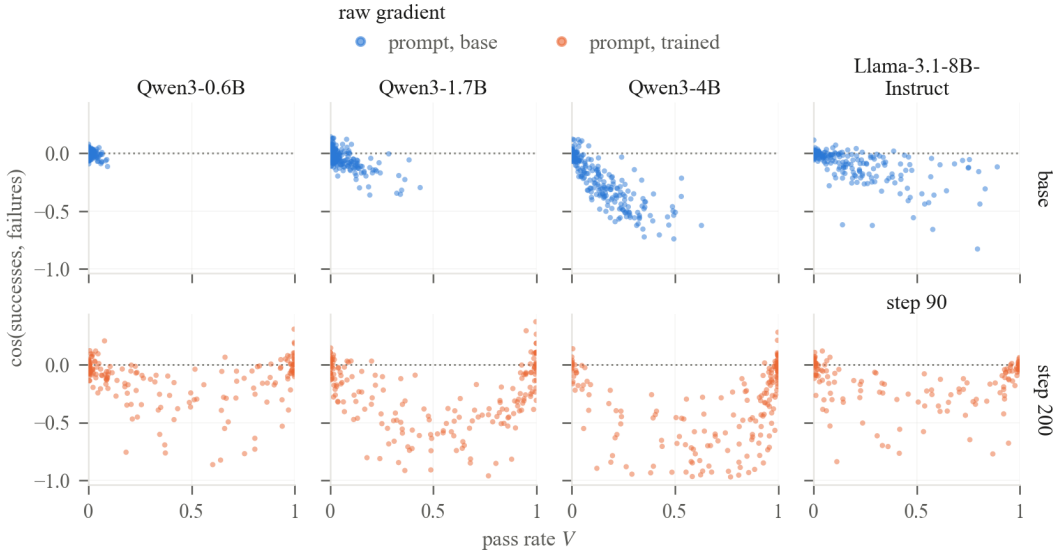


Figure 16: Within-prompt cosine between the mean gradients of successful and failed rollouts against pass rate, every model and checkpoint. The score identity predicts -1; near $V = 0$ or 1 one of the two means rests on few rollouts.

Within a prompt, the score identity $\mathbb{E}[g_\theta(\tau)] = 0$ gives $Vg_+ + (1 - V)g_- = 0$, where $g_\pm$ are the mean score gradients of successful and failed rollouts, so the two means should be exactly anti-

Table 20: Proposition 1 across disjoint halves. Same: cosine between the two targets' aggregate gradients built from the same rollouts (shared sampling noise included). Disatt.: the mean of the two cross-half cosines (target X on one half of every prompt's rollouts, Y on the other) divided by the geometric mean of the targets' split-half reliabilities, which estimates the cosine between their expected gradients; Proposition 1 predicts 1 for the first two pairs. Values above 1 are within the estimate's noise.

		GRPO$^+$, pass@2		GRPO$^-$, all@2		GRPO, pass@8		pass@8, all@2	
Model	Stage	same	disatt.	same	disatt.	same	disatt.	same	disatt.
Qwen3-0.6B	base	0.99	1.00	0.35	0.46	1.00	1.00	0.94	0.98
	step 200	0.88	0.97	0.90	0.98	0.38	0.29	−0.06	−0.18
Qwen3-1.7B	base	0.99	1.00	0.87	0.96	0.98	0.99	0.91	0.94
	step 200	0.92	0.98	0.94	0.98	0.41	0.41	0.05	0.02
Qwen3-4B	base	1.00	1.00	0.97	0.99	0.97	0.98	0.94	0.95
	step 200	0.95	1.00	0.97	1.00	0.36	0.37	0.07	0.08
Llama-3.1-8B-Instruct	base	0.94	0.96	0.85	0.89	0.90	0.97	0.80	0.94
	step 90	0.93	1.02	0.90	1.02	0.51	0.45	0.03	0.11

aligned. Measured across prompts, the cosine between the two means is negative for 60–88% of prompts in every cell, with medians from -0.005 to -0.22 at the base model and -0.07 to -0.27 at the plateau (Figure 16). The medians fall short of -1 because one of the two means rests on few rollouts when the pass rate is near 0 or 1: at Qwen3-4B step 200 the median is -0.66 for the 52 prompts with $0.2 < V < 0.8$.

H TRAJECTORY TRACING

Table 21: Trajectory statistics at matched length (optimizer steps): straight-line ratio R, net displacement over $\sqrt{\sum_t \|m_t\|^2}$, with its finite-length null for momentum-filtered isotropic noise ($\beta_1 = 0.9$; 200 simulations, 95% range); growth exponent α of $\|\Delta_t\|$ (0.5 for a random walk, 1 for a straight line).

Run	Steps	R	null R [95% range]	α
Qwen3-0.6B, online GRPO	1,600	4.44	4.35 [4.26, 4.43]	0.53
Qwen3-1.7B, online GRPO	1,600	4.68	4.35 [4.26, 4.43]	0.62
Qwen3-4B, online GRPO	1,575	4.52	4.34 [4.26, 4.44]	0.59
Qwen3-4B, online positives only	1,575	4.27	4.34 [4.26, 4.44]	0.72
Qwen3-4B, offline positives only	1,575	4.48	4.34 [4.26, 4.44]	0.44

I LIMITATIONS

Our measurements cover math reasoning with binary rewards and the GRPO family of estimators, one training run per model, and Qwen3 base models with one instruction-tuned Llama checkpoint. $n_{0.5}$ beyond the pool size is estimated from the noise model rather than measured.