THE BLESSING OF DIMENSIONALITY IN LLM FINE-TUNING: A VARIANCE–CURVATURE PERSPECTIVE

Anonymous authors
Paper under double-blind review

ABSTRACT

Weight-perturbation evolution strategies (ES) can fine-tune billion-parameter language models with surprisingly small populations (e.g., $N \approx 30$), contradicting classical zeroth-order curse-of-dimensionality intuition. A second, seemingly separate phenomenon also appears under fixed hyperparameters, when the fine-tuning reward often rises, peaks, and then degrades, not just with ES but also with GRPO. Both effects admit a common geometric account: fine-tuning landscapes are *low-dimensional in curvature*, with a few stiff directions embedded in a large, nearly flat bulk. The separation of curvature scales produces (i) heterogeneous time scales that yield rise–then–decay under fixed stochasticity, as captured by a minimal quadratic stochastic-ascent model, while the concentration of curvature produces (ii) degenerate improving updates, where many random perturbations share similar components along the stiff directions. For isotropic perturbations, directions without curvature add nothing to the cost of a random perturbation, so the probability of sampling an improving one does not depend on the ambient dimension D, and the scaling question reduces to how the curvature-active dimension k grows with D. Probing the fine-tuning reward landscapes of GSM8K, ARC-C, and WinoGrande with ES across Qwen2.5-Instruct models (0.5B–7B) reveals that reward-improving perturbations remain accessible with small populations at every scale, and direct Hessian-spectrum measurements on the same models yield effective curvature dimensions on the order of tens, growing sub-linearly with D over the tested scales and occupying a vanishing fraction of parameter space. These results reconcile ES scalability with non-monotonic training dynamics under a single geometric account, and they open the door to a broad class of fine-tuning algorithms, beyond policy-gradient RL, that worst-case zeroth-order theory had ruled out as infeasible at scale.

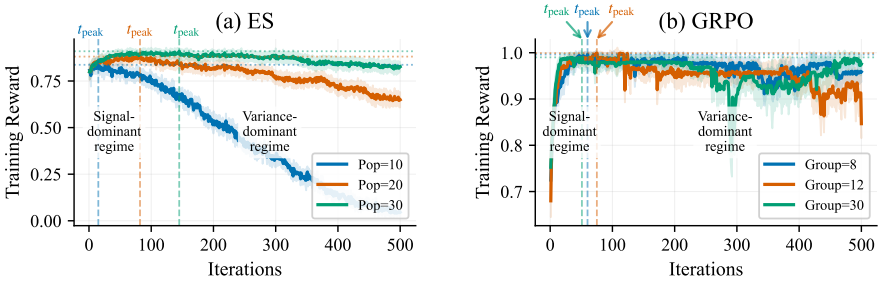

Figure 1: **Training reward rises then decays under fixed stochasticity.** Reward improves rapidly, peaks at t_{peak}, and then degrades in ES and GRPO fine-tuning of `Qwen2.5-1.5B-Instruct` on GSM8K (Cobbe et al., 2021) (both methods trained on the same 100-sample training set; since GRPO does not directly maximize the task reward, training rewards shown are evaluated over the training set at every iteration). The decay is not overfitting or evaluation noise. Section 4 traces it to a curvature–variance interaction, in which early gains from a few stiff directions give way to noise accumulating along the many flat ones.

1 INTRODUCTION

Recent work has shown that large language models (LLMs) can be fine-tuned at scale using simple weight-perturbation evolution strategies (ES; Qiu et al., 2026). This approach achieves performance

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.

competitive with leading reinforcement-learning-based methods such as Proximal Policy Optimization (PPO; Schulman et al., 2017) and Group Relative Policy Optimization (GRPO; Shao et al., 2024). The fact that zeroth-order optimization methods like such can be applied at this scale is surprising because ES estimates an update direction from a finite set of random perturbations whose signal-to-noise ratio typically deteriorates with dimension. Therefore, avoiding a *curse of dimensionality* would seem to require population sizes that grow rapidly with the number of parameters (Duchi et al., 2015). However, ES with populations as small as $N \approx 30$ fine-tunes models with billions of parameters.

At the same time, a second, seemingly independent phenomenon appears. Under fixed hyperparameters, the training reward often improves rapidly from a pretrained checkpoint, reaches a peak, and then *decreases* as training progresses. This appears in both ES and policy-gradient fine-tuning and is strongly modulated by algorithmic stochasticity, which is due to perturbation scale and population size in ES or temperature and group size in GRPO (Fig. 1). Therefore, it likely arises from structural properties of the fine-tuning landscape rather than an artifact of a particular algorithm.

This paper aims to explain (1) why small-population weight-perturbation methods can scale to billion-parameter models, and (2) why stochastic fine-tuning exhibits rise–then–decay behavior under fixed stochasticity. The following hypothesis provides a common explanation.

Central hypothesis: A blessing of dimensionality

LLM fine-tuning landscapes are hypothesized to be ***low-dimensional in curvature***, with local optimization geometry governed by a small set of curvature-active directions. The size of this curvature-active set does ***not*** grow proportionally with the number of model parameters, so small-population weight-perturbation methods can remain effective at billion-parameter scale despite zeroth-order curse-of-dimensionality intuitions.

A unifying geometry–variance framework. Both ES scalability and non-monotonic fine-tuning dynamics are interpreted through a geometry–variance perspective intrinsic to the reward landscape. Locally, curvature is highly anisotropic, with a small number of high-curvature "*stiff*" directions dominating reward improvement, while the vast majority of directions are weakly curved or effectively "*flat*" ("curvature-active" refers to the stiff subspace). Two consequences follow. First, because improvement is governed by the stiff subspace, random-perturbation methods need only access this restricted set of directions rather than estimate a full-dimensional gradient, so effective updates do not require populations that scale with parameter count. Second, the stiff–flat separation induces heterogeneous time scales. Early progress is fast along stiff directions, while fixed noise steadily spreads the iterate along flat directions (*variance accumulation*); once the stiff directions are exhausted, variance accumulation dominates and reward degrades. The mechanism is not a claim about ES specifically but about any stochastic optimizer whose update is a curvature-coupled drift plus persistent noise. ES serves as the *instrument* here because σ and N directly control the injected noise; GRPO is the first-order method on which the same signature is checked.

Contributions. Four contributions build toward a single thesis:

- **A unified account of two puzzles.** ES scalability and rise–then–decay admit a common account in one property of fine-tuning landscapes, a few stiff directions embedded in a large, nearly flat bulk: the concentration of curvature governs which random perturbations reach an improving region, and the separation of curvature scales governs how training evolves once the stiff directions are exhausted. This leads to the blessing-of-dimensionality hypothesis that the number of curvature-active directions does not grow in proportion to parameter count.
- **A mechanism for rise–then–decay.** A tractable model, the dynamical side of this account, shows how stiff–flat time-scale separation under fixed noise is *sufficient* to produce non-monotonic reward trajectories; Appendices C–E discuss anisotropic noise, saddles, and multiple basins.
- **Direct measurement of the curvature-active dimension.** Because directions without curvature add nothing to the cost of a random perturbation, the probability of sampling an improving one does not depend on D (Proposition 5.1), so the hypothesis reduces to how k grows with D. Hessian spectra of the ES-smoothed reward (5 seeds, 0.5B–7B) and of a differentiable surrogate loss (0.5B–3B), plus gradient concentration to 14B (Appendix H), show that the effective curvature dimension is on the order of tens and grows sub-linearly with D.
- **ES-based accessibility probes across scale.** Best-of-N (extreme-value) improvement under random perturbations, the static side, is measured across three tasks and four model sizes, showing that improving perturbations remain accessible with small populations from 0.5B to 7B parameters.

Low-dimensional curvature persists across the model scales studied. As parameter count increases, the curvature-active dimension grows only sub-linearly, so the geometry underlying both phenomena remains confined to a small and shrinking fraction of parameter space: a *blessing of dimensionality*.

2 BACKGROUND

This section reviews four bodies of prior work that motivate the paper's central hypothesis: 1) classical worst-case guarantees for zeroth-order optimization, 2) empirical evidence that fine-tuning is effectively low-dimensional, 3) structural characterizations of the Hessian in overparameterized networks, and 4) stochastic-dynamics accounts of noise-limited training. Each line of work touches one side of the puzzle (either why zeroth-order methods should struggle at scale, or why fine-tuning landscapes might nonetheless be forgiving), but none connects low-dimensional curvature to both ES's empirical scalability and the rise–then–decay dynamics of Figure 1.

Zeroth-order optimization and the curse of dimensionality. Classical zeroth-order (gradient-free) results showed that the number of function evaluations needed to make progress grows quickly with the number of parameters, which has led to skepticism about scaling weight-perturbation methods to overparameterized models (Duchi et al., 2015). Nevertheless, evolution strategies have long been studied as scalable black-box optimizers (Salimans et al., 2017), and Qiu et al. (2026) showed that simple weight-perturbation ES fine-tunes billion-parameter LLMs with $N \approx 30$. It outperformed PPO, GRPO, and Dr.GRPO across seven base LLMs on Countdown, a reasoning task with only sparse outcome rewards, and reached a better reward/KL trade-off than GRPO without reward hacking on a conciseness task. It was also competitive with state-of-the-art RL checkpoints on math benchmarks (AIME2024, Minerva Math, OlympiadBench, AMC, MATH500) and far exceeded the base models on ARC-AGI and Sudoku. This paper addresses that tension.

Blessing of dimensionality via low-dimensional fine-tuning structure. Many fine-tuning problems are effectively low-dimensional despite enormous ambient dimension. Intrinsic-dimension studies showed that downstream objectives could be optimized within a small random subspace, and that this intrinsic dimensionality *decreases* with model size (Li et al., 2018; Aghajanyan et al., 2021). Moreover, the success of low-rank adaptations such as LoRA (Hu et al., 2022) provides complementary evidence that effective fine-tuning updates can be highly structured and low-rank. This evidence motivates a "blessing of dimensionality" viewpoint in which additional parameters need not increase the number of directions that matter for adaptation, and may instead enlarge flat or redundant directions. Concurrent work by Gan & Isola (2026) found that the density of task-improving perturbations near pretrained weights increases with model scale, and that purely random perturbation selection yields competitive performance, directly corroborating the "wheel of fortune" picture developed below (Figure 5).

Hessian spectra with a near-zero bulk and a few outliers. The *Hessian* characterizes local geometry. Its large-magnitude eigenvalues correspond to *stiff* directions and near-zero eigenvalues to *flat* ones. Overparameterized networks repeatedly showed a structured spectrum near trained solutions, a large *bulk* near zero plus a few *outliers* that dominated curvature (Sagun et al., 2017; Ghorbani et al., 2019), and learning dynamics often concentrated in the associated "sharp" subspace (Gur-Ari et al., 2018), similar to as shown in Figure 4. This spectral structure is not claimed as a contribution here. What this paper adds is (i) a link from the static spectrum to *training dynamics*, showing how a stiff–flat separation under fixed stochasticity produces rise–then–decay, and (ii) a *scaling prediction*, that the population needed to access improving perturbations need not grow in proportion to parameter count, which does not follow from spectral structure alone; direct spectral estimates on the models studied here are also reported (Appendix H).

In addition, Song et al. (2025) showed for small networks trained from scratch that confining SGD *updates* to the top-k Hessian eigenspace fails to reduce the loss, so useful signal also lives in the bulk. There is no tension with the present account, which concerns the reward landscape of a *pretrained* model near a good solution, and the rise–then–decay signature itself indicates that late-time dynamics are not confined to the stiff subspace, with the bulk carrying noise rather than signal.

Stochastic optimization dynamics and noise-limited behavior. Constant-step stochastic optimization near minima is often modeled as an Ornstein–Uhlenbeck (OU) process or related SDE, yielding stationary noise floors and curvature-dependent behavior (Mandt et al., 2017; Li et al., 2017); the toy model developed in Section 4 adopts this approximation to make explicit how curvature and noise together produce rise–then–decay.

Non-monotonic training dynamics. Trajectories that improve early and later degrade have been documented in deep RL, where extended on-policy training can exhibit policy collapse or loss of plasticity (Dohare et al., 2023; Moalla et al., 2024), and in supervised learning, where training loss is non-monotonic in "edge of stability" regimes (Arora et al., 2022; Zhu et al., 2023). A reproducible peak–then–decay is observed here across both ES and GRPO fine-tuning, suggesting a mechanism not specific to one optimizer.

Taken together, prior work suggests that zeroth-order methods should need populations that grow with dimension. It also documents non-monotonic training without tying it to a common geometric cause. Neither connects this stiff–flat curvature structure to *both* phenomena; the rest of the paper develops and tests such a connection.

3 Evolution Strategies as a Geometric Probe

Weight-perturbation evolution strategies (ES) are used here primarily as a *geometric probe* of fine-tuning landscapes rather than as an optimizer to be benchmarked. A *fine-tuning landscape* is the map from parameters to a scalar task reward, $\theta \mapsto \mathcal{J}(\theta)$ with $\theta \in \mathbb{R}^D$, where D is the number of model parameters, and $\mathcal{J}$ may be accuracy or another reward evaluated by sampling and decoding. ES is an unusually clean probe of this landscape for two reasons. First, the noise it injects is under direct experimental control. The perturbation scale σ sets how far from θ the landscape is sampled, and the population size N sets how much estimation noise enters each update; both are chosen by the experimenter, independently of the landscape, and the perturbations are isotropic by construction. Sweeping (σ, N) therefore turns ES into a controlled experiment, in which the response of the reward to a known amount of injected noise reveals the curvature of the landscape. Policy-gradient methods such as GRPO also inject noise, but through sampling temperature and group size, and the resulting covariance depends on the policy itself and cannot be set directly (Appendix C). Second, ES requires only reward evaluations, so it applies even when the reward is discrete or non-differentiable.

At iteration t, ES samples perturbations $\varepsilon_k \sim \mathcal{N}(0, I)$, evaluates $r_k = \mathcal{J}(\theta_t + \sigma\varepsilon_k)$, and updates

$$\widehat{g}_t \;=\; \frac{1}{N\sigma}\sum_{k=1}^{N} r_k\,\varepsilon_k, \qquad \theta_{t+1} \;=\; \theta_t + \alpha\,\widehat{g}_t, \tag{1}$$

with the full procedure in Algorithm 1. Equation (1) is treated as a simplified analytical abstraction; in practice all empirical runs use the standard reward-normalized variant

$$\widehat{g}_t = \frac{1}{N\sigma}\sum_{k=1}^{N} \tilde{r}_k\,\varepsilon_k, \quad \tilde{r}_k = \frac{r_k - \bar{r}}{s_r + \epsilon}, \tag{2}$$

Algorithm 1: Weight-Perturbation ES

Input: reward $\mathcal{J}(\theta)$; params θ_0; step size α; perturb. scale σ; pop. size N; iterations T

for $t \leftarrow 0$ **to** $T-1$ **do**
- Sample $\varepsilon_1, \ldots, \varepsilon_N \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, I)$;
- **for** $k \leftarrow 1$ **to** N **do**
  - $r_k \leftarrow \mathcal{J}(\theta_t + \sigma\varepsilon_k)$;
- $\widehat{g}_t \leftarrow \frac{1}{N\sigma}\sum_{k=1}^{N} r_k\,\varepsilon_k$;
- $\theta_{t+1} \leftarrow \theta_t + \alpha\,\widehat{g}_t$;

where $\bar{r}, s_r$ are the within-group mean and standard deviation, $\epsilon > 0$ is a small constant added for numerical stability (avoiding division by zero when $s_r = 0$, e.g. a group with identical rewards), and each candidate is evaluated on a group of prompts. Z-score normalization removes the $\mathcal{O}(\bar{r}^2 D/N)$ zero-order variance term and makes the effective noise state-dependent rather than the literal σ^2/N of the unnormalized idealization (Appendix G).

Gaussian smoothing and coarse-grained geometry. ES can be interpreted as optimizing a Gaussian-smoothed objective,

$$\mathcal{J}_\sigma(\theta) \;\triangleq\; \mathbb{E}_{\varepsilon\sim\mathcal{N}(0,I)}[\mathcal{J}(\theta + \sigma\varepsilon)]\,, \tag{3}$$

where σ controls the smoothing scale. Even when $\mathcal{J}$ is jagged or nondifferentiable, $\mathcal{J}_\sigma$ is differentiable under mild conditions; smoothing suppresses irregularities below scale σ while preserving coarser curvature structure.

A central identity is

$$\nabla\mathcal{J}_\sigma(\theta) \;=\; \frac{1}{\sigma}\,\mathbb{E}_{\varepsilon\sim\mathcal{N}(0,I)}[\mathcal{J}(\theta + \sigma\varepsilon)\,\varepsilon]\,, \tag{4}$$

so $\widehat{g}_t$ is a Monte Carlo estimator of $\nabla\mathcal{J}_\sigma(\theta_t)$.

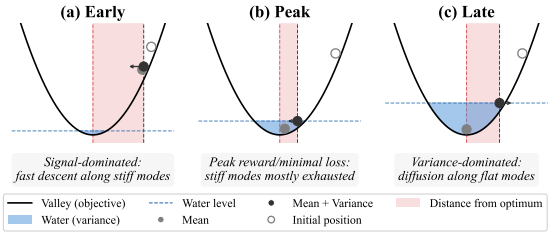

Figure 2: **Rise-then-decay as a rising noise floor: the water-filling analogy. (a)** Fast improvement along stiff directions. **(b)** Stiff directions exhausted; variance limits gains. **(c)** Variance-dominated degradation under fixed stochasticity. Stiff directions saturate quickly while noise steadily fills the flat directions, so the peak marks the point where curvature-driven gains are overtaken by variance accumulation.

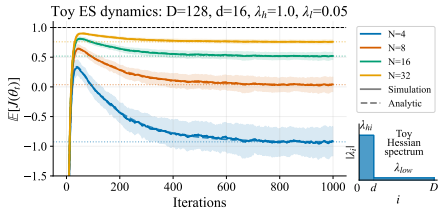

Figure 3: **A minimal spectral heterogeneity reproduces rise-then-decay.** ES on a two-block spectrum (D=128, d=16). Solid: Monte Carlo; dashed: closed-form (Appendix A). Larger N raises the terminal plateau. Two curvature scales plus fixed noise already suffice to produce a peak, isolating heterogeneous curvature as the qualitative driver of rise-then-decay.

Controllable Stochasticity. With a finite population, ES produces a stochastic gradient estimate of ∇J_σ:

$$\widehat{g}_t = \nabla J_\sigma(\theta_t) + \xi_t^{\text{est}}, \qquad \mathbb{E}[\xi_t^{\text{est}} \mid \theta_t] = 0, \tag{5}$$

where ξ_t^{est} is Monte Carlo estimation noise. Its covariance has the standard prefactor

$$\text{Cov}(\xi_t^{\text{est}} \mid \theta_t) \;\approx\; \frac{1}{N\sigma^2}\,\Sigma_{\text{est}}(\theta_t, \sigma), \tag{6}$$

with problem-dependent Σ_{est} that can itself depend on σ (e.g., through reward variability). In the toy model of Section 4, the *effective* stochasticity of parameter updates is summarized by a diffusion scale $\kappa = \sigma^2/N$.

Two kinds of probe. A *dynamical* probe runs ES for many iterations and reads off the reward trajectory (Section 4); a *static* probe draws perturbations around a fixed checkpoint and summarizes the reward changes by the best-of-N improvement, an extreme-value statistic (Section 5).

4 Non-Monotonic Training Dynamics and Low-Dimensional Curvature

Figure 1 shows the phenomenon that motivates this section. During ES fine-tuning on GSM8K under fixed hyperparameters, training reward improves rapidly, peaks, and then degrades toward a lower value. Both the peak time and the depth of the decay vary systematically with population size, which points to a variance-controlled effect rather than overfitting or evaluation noise. GRPO runs on the same training set (Fig. 1b) also decay past a peak, though less starkly, so the phenomenon is not specific to ES; since GRPO's anisotropic noise is not measured here, this is cross-algorithm consistency, not direct validation of the mechanism.

Scope of the model. The generic noisy update $\theta_{t+1} = \theta_t + \alpha(g(\theta_t) + \xi_t)$ with $\mathbb{E}[\xi_t \mid \theta_t] = 0$ is considered, where g is a curvature-coupled drift and ξ_t is persistent noise. ES (Eq. 5) and GRPO are both instances, differing in the covariance of ξ_t, which is approximately isotropic and externally tunable for ES but anisotropic, state-dependent, and nonstationary for GRPO. The isotropic, locally quadratic case is analyzed because it is the minimal instance with closed forms; it shows that heterogeneous curvature time scales plus fixed noise are *sufficient* for rise–then–decay, not that this case literally describes either algorithm. The analysis extends to general noise covariance, including the Fisher-like form of policy-gradient updates such as GRPO, where decay depends on how much of the noise reaches weakly curved directions (Appendix C). It is also noted that further non-convex phenomena such as saddles and/or basin hopping lie outside the model and can add further non-monotonicity (Appendices D–E).

The water-filling analogy. Figure 2 summarizes the intuition in the equivalent *descent* picture. Fast relaxation along stiff directions drives early improvement, while stochasticity accumulates along flat directions like a rising "water level" that limits attainable performance; once the stiff directions saturate, variance accumulation dominates and produces a peak followed by degradation.

Local quadratic stochastic dynamics. Let $\theta^\star$ be a local maximizer and write

$$J(\theta^\star + x) \approx J(\theta^\star) - \tfrac{1}{2} x^\top C x, \qquad C \succeq 0. \tag{7}$$

Without loss of generality, coordinates are shifted so that $\theta^\star = 0$ (equivalently, θ_t below denotes the displacement $\theta_t - \theta^\star$); a constant-step noisy ascent model then takes the form

$$\theta_{t+1} = (I - \alpha C)\theta_t + \alpha \frac{\sigma}{\sqrt{N}} \varepsilon_t, \qquad \varepsilon_t \sim \mathcal{N}(0, I), \tag{8}$$

with effective noise $\kappa = \sigma^2/N$ for the unnormalized abstraction; under the empirical normalized update (2) the same estimator yields a state-dependent noise level rescaled by $1/(s_r + \epsilon)$, so κ is read throughout as a population/perturbation-controlled proxy for finite-sample stochasticity rather than a literal diffusion coefficient. Diagonalizing $C = Q\Lambda Q^\top$ decouples the dynamics into independent modes with contraction factors $a_i = 1 - \alpha\lambda_i$ (Appendix A). High-curvature directions relax quickly, while low-curvature directions relax slowly and are more susceptible to variance accumulation.

A two-block curvature example. To make the time-scale separation concrete, a stylized two-block spectrum is considered, consisting of $d \ll D$ stiff directions with curvature λ_{hi} and many weakly curved directions with $\lambda_{lo} \ll \lambda_{hi}$ (as shown in the inset of Figure 3). Figure 3 shows ES dynamics (both simulation and analytics) on this toy quadratic landscape and demonstrates that simple spectral heterogeneity is sufficient to produce rise–then–decay trajectories, with early gains driven by the stiff subspace and late-time behavior set by variance accumulation in the flat bulk. In the quadratic case the decay does not continue indefinitely. The expected reward settles to a terminal plateau whose height is set by the effective noise $\kappa = \sigma^2/N$, so larger populations plateau higher (Fig. 3).

What the toy model isolates. The toy model isolates two requirements for rise–then–decay in this local regime. The first is a *heterogeneous and low-dimensional curvature spectrum*. With a single dominant time scale the expected trajectory is monotonic, while well-separated time scales can produce a peak. The second is *non-negligible stochasticity*. The late-time plateau is controlled by $\kappa = \sigma^2/N$ through a curvature-weighted functional (Appendix A); its slope also defines a curvature-based effective dimension (Appendix B). Initialization affects how pronounced the peak is, but the mechanism is curvature-dependent relaxation under fixed noise. Rise–then–decay persists under z-score-normalized ES (2) (Figs. 1, 10), so it is not an artifact of the unnormalized model.

What the toy model does not capture. The noise-limited plateau of Fig. 3 is not cleanly visible in Fig. 1 within the training horizon. The plateau is a within-basin idealization that assumes a fixed local quadratic; nonstationary local geometry and drift between nearby basins along flat directions (Appendices D–E) are plausible complementary mechanisms that have *not* been measured here. What can be reported is that the local stable/unstable mix changes along an ES run. On a 1.5B GSM8K trajectory, the negative spectral mass of the smoothed-reward Hessian falls from 23% at the base checkpoint to 4% near the reward peak and reopens to 22% during decay (Appendix H.6). The model also assumes a local maximizer rather than a saddle; since rise–then–decay is a per-eigendirection effect, the mechanism still operates on the stable subset of directions.

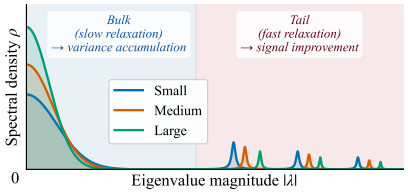

Figure 4: **Schematic curvature structure:** near-zero bulk with a few stiff outliers. Small, medium, and large refer to relative model sizes.

Low-dimensional curvature in fine-tuning landscapes. The variance-controlled rise–then–decay observed in ES (and qualitatively in GRPO) points to a few stiff directions and many flat ones in real fine-tuning, consistent with the "bulk + outliers" picture sketched in Figure 4. In this curvature-based notion of low-dimensionality, what matters is not the ambient parameter count but how many directions are curvature-active and whether that number grows with model size; this is measured directly in Section 5 and found to be on the order of tens and to grow sub-linearly with D. Low-dimensional curvature does not imply a unique improving direction, since many distinct perturbations can make comparable progress by sharing similar components in the curvature-active subspace.

5 Empirical Scalability of ES as a Consequence of Low-Dimensional Curvature

If the curvature-active dimension k grows much more slowly than D, low-dimensional curvature predicts that improvement should remain accessible to a few random perturbations even as D grows.

The key quantity is how likely a single random perturbation is to improve the reward. Generally, ES progresses only if some of its N perturbations do, so this probability sets the population it needs. If curvature is confined to k directions, the other $D-k$ add nothing to the cost of a random perturbation, and adding parameters does not make improvement harder to find. The first two paragraphs below make this precise; the rest test it in two independent ways, by measuring k directly and by measuring how often random perturbations improve the reward.

A north star versus many prize regions. Gradient-based methods, and the classical analysis of zeroth-order optimization, frame each update around a single direction: the gradient. Figure 5(a) shows why recovering this "north star" from random perturbations is hard. The chance that a random direction aligns with a fixed direction to within a threshold, $\langle \varepsilon/\|\varepsilon\|, d\rangle \geq c > 0$, vanishes as $D \to \infty$ (Duchi et al., 2015). This framing overlooks that improving the reward does not require the precise gradient. When curvature is low-dimensional, many distinct directions improve the reward (Fig. 5(b)), and a random perturbation only needs to land in one of them.

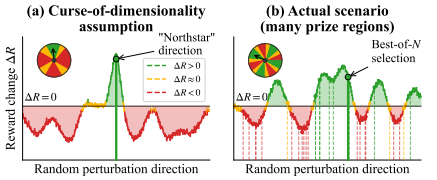

Figure 5: **Optimizing a needle-in-a-haystack vs. "wheel of fortune." (a)** Unique direction: vanishing mass. **(b)** Degenerate set: best-of-N succeeds with fixed population.

Why D drops out. ES never has to estimate the gradient, but the gradient and curvature of the landscape still decide which random perturbations happen to improve the reward. For a perturbation $\sigma\varepsilon$ with $\varepsilon \sim \mathcal{N}(0, I_D)$,

$$\Delta R \;\approx\; \underbrace{\sigma g^\top \varepsilon}_{\text{gain}} \;-\; \underbrace{\tfrac{1}{2}\sigma^2 \varepsilon^\top C \varepsilon}_{\text{curvature penalty}}, \tag{9}$$

with g the reward gradient and C the negative reward Hessian. The gain does not grow with D: $g^\top\varepsilon \sim \mathcal{N}(0, \|g\|^2)$, which is positive for half of all perturbations in any dimension. The penalty is what can grow. A random perturbation has norm about $\sqrt{D}$, and its expected penalty $\frac{1}{2}\sigma^2 \operatorname{tr}(C)$ sums the curvature over every direction it touches. If curvature were spread over all D directions, the penalty would grow in proportion to D, and σ would have to shrink like $1/D$ before any perturbation improves the reward, which is the classical curse. If curvature is confined to k directions, the other $D-k$ add nothing to the penalty, however many there are. They are free, so each improving setting of the k curvature-active components is shared by a $(D-k)$-dimensional family of perturbations; this many-to-one structure is termed *degeneracy*. The probability that a random perturbation improves the reward is its *accessibility*, $p_{\text{imp}} = \Pr(\Delta R > 0)$.

Proposition 5.1 (Accessibility does not depend on D)**.** *Suppose $C = U\Lambda U^\top$ with $U \in \mathbb{R}^{D\times k}$ orthonormal (near-flat eigenvalues treated as zero). Then p_{imp} is determined by $U^\top g$, the norm $\|g_\perp\|$ of the gradient outside the curvature-active subspace, Λ, and σ, and does not depend on D.*

Proof. Let $z = U^\top\varepsilon \sim \mathcal{N}(0, I_k)$ and $w = g_\perp^\top\varepsilon \sim \mathcal{N}(0, \|g_\perp\|^2)$, independent of z. Then $\Delta R \approx \sigma\big((U^\top g)^\top z + w\big) - \frac{1}{2}\sigma^2 z^\top\Lambda z$, whose distribution does not involve D. □

Proposition 5.1 is exact only for rank-k curvature; real spectra are only approximately concentrated, so the effective rank $\hat{k}$ measured below gauges concentration rather than entering the population requirement exactly. The population needed for at least one of N perturbations to improve with confidence $1-\delta$, $N(\delta) = \lceil \log\delta / \log(1-p_{\text{imp}})\rceil$, therefore changes with scale only through the curvature spectrum and the gradient. A curse can return only if curvature spreads to more directions as models grow. This gives two testable predictions: k should grow much more slowly than D, and a fixed population should keep finding improvements over a range of σ that does not close.

The curvature-active dimension grows sub-linearly. The first prediction is checked by measuring curvature directly on the same models, in three ways (Table 2, Appendix H). The main measurement is H_{reward}, the Hessian of the ES-smoothed GSM8K reward, estimated from forward evaluations only via zeroth-order Hessian–vector products and stochastic Lanczos quadrature (5 seeds per model size). Its un-normalized effective rank $\hat{k}$ is

Model	0.5B	1.5B	3B	7B
$\hat{k}$	6.7 ± 1.7	13.3 ± 3.2	13.6 ± 2.5	25.9 ± 3.8

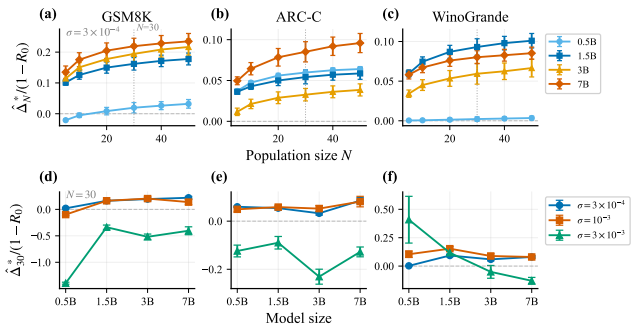

Figure 6: **Fixed-population improvement remains accessible across model scales. Top row (a–c):** Headroom-normalized expected best-of-N improvement $\Delta_N^{\mathrm{rel}}(\sigma) = \mathbb{E}[\max_{i \le N} \Delta R_i]/(1 - R_0)$ as a function of population size N at fixed perturbation scale $\sigma = 3 \times 10^{-4}$ for GSM8K, ARC-C, and WinoGrande across `Qwen2.5-Instruct` (0.5B–7B). Curves saturate by $N \approx 30$–40 without a systematic shift to larger N as model size increases. **Bottom row (d–f):** Headroom-normalized expected best-of-30 improvement versus model size for multiple σ. For each task, a viable range of σ yields positive best-of-30 improvements from 0.5B to 7B, indicating that the improvement-supporting tail of the perturbation-induced ΔR distribution remains accessible at scale. **Error bars** show $\pm 1.96\,\mathrm{SE}$ across $S = 8$ disjoint pseudo-batches of a single fixed candidate pool, reflecting perturbation-sampling variability within that pool (Appendix G).

and a log–log fit gives $\hat{k} \propto D^{0.46}$ ($R^2 = 0.95$). With four sizes on one task, this indicates sub-linear growth rather than a precise exponent: k is not constant, but it stays in the tens while D grows more than tenfold (robustness checks in Appendix H.3). Two further measurements agree qualitatively. H_{loss}, the autodiff Hessian of the token-level cross-entropy loss, has effective rank six to eight orders of magnitude below D at 0.5B–3B (single seed); it disagrees with H_{reward} numerically, but both place curvature-active structure in a vanishing fraction of the space. As auxiliary evidence, the fraction of parameters carrying 90% of the cross-entropy gradient mass stays within 4–11% from 0.5B to 14B. All spectra are taken at base checkpoints, not converged maximizers. A related measure, the minimum random-subspace dimension for 90% of full fine-tuning performance, *decreases* with model size (Aghajanyan et al., 2021), whereas $\hat{k}$ grows sub-linearly.

Best-of-N as an operational proxy for accessibility. The second prediction concerns accessibility itself. Computing p_{imp} would require g and C at each checkpoint, so it is instead measured directly, which also captures the size of the improvement. For each model and task, perturbations $\varepsilon_i \sim \mathcal{N}(0, I)$ are sampled and $\Delta R_i = R(\theta + \sigma\varepsilon_i) - R(\theta)$ is measured. Here $R(\theta) \in [0, 1]$ denotes the mean binary reward (accuracy) of θ on a fixed evaluation pool of prompts, i.e., the empirical estimate of $\mathcal{J}(\theta)$ used in all probes, and $R_0 \triangleq R(\theta)$ is the baseline. Accessibility is summarized by the expected best-of-N improvement

$$\Delta_N^*(\theta, \sigma) = \mathbb{E}\left[\max_{i \le N} \Delta R_i\right], \tag{10}$$

which directly quantifies whether N random draws can reach the improving "prize regions" in Fig. 5(b), even when $\mathbb{E}[\Delta R]$ is small or negative. An empirical curse would appear as loss of improvement at fixed N, or a rightward shift of best-of-N curves with scale. To control for task saturation, headroom-normalized improvements $\Delta_N^*/(1 - R_0)$ are also reported.

Population requirements do not scale with model size. Figures 6(a–c) report the headroom-normalized expected best-of-N improvement as a function of N for GSM8K, ARC-C, and WinoGrande across `Qwen2.5-Instruct` (0.5B–7B). For each task, model, and σ, a pool of $M = 240$ perturbation candidates is constructed, ΔR is evaluated on a fixed prompt set, and $\Delta_N^*(\sigma)$ is estimated by Monte Carlo sampling N distinct candidates from the pool and averaging the maxima (Appendix G). Across all tasks and model sizes, best-of-N improves rapidly at small N and shows clear diminishing returns beyond $N \approx 30$–40, with no systematic rightward shift as model size increases, the most direct empirical signature of a curse. The saturation is not a finite-pool artifact. With $M = 240$ and $N \le 50$, the flattening reflects diminishing returns of expected extrema from the shape of the improvement distribution's upper tail, not pool exhaustion.

Viable perturbation scales persist with model size. Fixing $N = 30$, Figures 6(d–f) show best-of-30 improvement versus model size for several σ. For every task there is a range of sufficiently small σ for which $\Delta_{30}^*(\sigma)$ stays positive from 0.5B to 7B, as the second prediction requires: since the curvature penalty grows only with k, the usable range of σ does not close as models grow. Absolute gains attenuate as headroom shrinks, but improvement persists across a tenfold range of D.

A blessing of dimensionality. The two tests agree: curvature occupies a small, sub-linearly growing set of directions, and a fixed population finds improvements across the same range of scales. Finite populations succeed not by aligning with the gradient but by landing in one of many favorable regions.

6 Discussion and Future Work

Central thesis and practical consequences. These results support a single geometric picture in which fine-tuning landscapes are *low-dimensional in curvature*, *heterogeneous* (widely separated relaxation rates), and *degenerate* (improvement directions form a subspace, not a needle), which together reconcile rise–then–decay (heterogeneity + fixed noise) and ES scalability (degeneracy + small populations). The mechanism makes rise–then–decay diagnostic and intervenable, suggesting **early stopping** near the peak, **noise scheduling** (larger N, smaller σ/temperature over time), and **adaptive step sizes** that shrink once curvature-active progress saturates.

Revisiting algorithm design. Evidence for a low-dimensional, *degenerate* regime reopens a design space often dismissed for LLM fine-tuning. The most immediate opportunity may be long-horizon tasks with sparse rewards. ES scores each perturbation by a single outcome-level reward, so it needs no per-token credit assignment, and its noise is set by σ and N rather than by the length of the trajectory or the rarity of the reward. Policy-gradient methods must instead attribute a sparse terminal reward across many generated tokens, and their gradient estimates grow noisier with the horizon. Low-dimensional curvature explains why the ES side of this trade-off remains workable at scale, consistent with ES outperforming PPO and GRPO on the sparse-reward Countdown task (Qiu et al., 2026). The same picture also favors **population-based perturbation methods**, **structured perturbations** (low-rank or blockwise noise), and **hybrid methods** combining perturbative exploration with gradient updates. The question becomes not "can zeroth-order scale in D?" but "which perturbation distributions best align with the curvature-active directions?"

Future directions. First, *online curvature monitoring*: making the curvature probes of Appendix B cheap enough to run during training, so early stopping and noise scheduling can track curvature rather than held-out reward. Second, *anisotropic and multi-basin dynamics*: connecting Appendices C and E to GRPO's actual noise covariance, estimable with the proxies in Appendix C, to test whether the same mechanism governs its rise–then–decay. Third, *other training regimes*: testing whether supervised fine-tuning and pretraining show the same curvature–variance signature. Finally, *scale and tasks*: extending the reward-Hessian measurements beyond 7B and GSM8K, including long-horizon, sparse-reward tasks, to test the sub-linear growth of k and the algorithmic opportunity above.

Limitations. *Mechanism.* The quadratic model shows heterogeneous curvature plus fixed noise is *sufficient* for rise–then–decay, not the operative mechanism in any given run; it abstracts away anisotropic noise and basin-to-basin drift (App. C–E), its plateau is not cleanly observed in real runs, and basin hopping and eigenvector rotation remain unmeasured beyond (App. H.6). *Theory.* Proposition 5.1 is a reduction, not a guarantee, and assumes exact rank-k curvature; population bounds in terms of effective rank remain open. *Measurements.* Dimensionality k grows sub-linearly (about $D^{0.46}$ on four sizes), not zero; effective-rank estimates shift by 21–43% when the prompt subset is doubled, the surrogate-loss Hessian is single-seed and stops at 3B, gradient concentration is auxiliary, and all three curvature measurements are on GSM8K only. *Probes and algorithms.* A fixed pool of $P{=}320$ prompts upper-bounds the rank of data-dependent curvature operators, probes reach 7B with binary rewards, and only ES and GRPO are tested.

7 Conclusion

This paper set out to explain why small-population weight-perturbation methods scale to billion-parameter fine-tuning, and why fine-tuning reward under fixed stochasticity rises, peaks, and decays. Both admit a common geometric account: a small, sub-linearly growing set of stiff directions embedded in a large, nearly flat bulk. Concentrated curvature keeps improvements accessible, while the gap between stiff and flat curvature scales lets noise accumulate in the flat ones, producing rise–then–decay.

This unification also reopens a design space. Population-based perturbation methods become a viable alternative to policy-gradient RL fine-tuning such as PPO and GRPO, particularly in long-horizon tasks with sparse rewards. Because ES needs only forward evaluations of a reward, it also opens settings that gradient-based fine-tuning handles poorly: non-differentiable or black-box rewards, objectives defined over whole generated outputs, and inference-only training without backpropagation.

The most important implication concerns scale. Because k grows only sub-linearly, the directions that larger models add are nearly free for a random perturbation, suggesting that small populations may remain effective beyond the 7B scale measured here. This is the blessing of dimensionality that classical zeroth-order theory missed.

AI USE STATEMENT

In this work, we used generative AI tools for implementing and executing components of the experimental pipeline, and for interpreting and analyzing experimental results. We have not used generative AI tools for developing the theoretical framework or central hypothesis, formulating the mathematical claims or their proofs, proposing or refining the scientific hypotheses tested in this work, or designing the experimental methodology; these were developed by the authors. Generating synthetic data, cleaning or reformatting datasets, translation, and qualitative/thematic data analysis are not applicable to this work. Additionally, we used generative AI tools for creating and modifying scientific figures, and for drafting and editing portions of the manuscript text. We have reviewed all AI-assisted work. Code, analysis, and figures were checked by the authors before inclusion in this work. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies optimization dynamics of language model fine-tuning using public benchmarks (GSM8K, ARC-C, WinoGrande, Countdown) and openly released pretrained models (Qwen2.5-Instruct). It does not involve human subjects, new data collection, or the release of new models or datasets, and we are not aware of ethical concerns specific to this work beyond those already associated with the cited benchmarks and models.

REPRODUCIBILITY STATEMENT

Model, dataset, and hyperparameter details for all experiments are given in Appendix G (ES reward-probe protocol, models, tasks, perturbation scales, and pool construction) and Appendix G.10 (training hyperparameters for the non-monotonic ES and GRPO runs). The zeroth-order Hessian estimator and its hyperparameters are specified in Appendix H, including seeds and robustness checks (Appendix H.3). The proof of Proposition 5.1 is given in the main text; the toy-model derivations underlying Section 4 are in Appendix A.

REFERENCES

Armen Aghajanyan, Sonal Gupta, and Luke Zettlemoyer. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP 2021), Volume 1: Long Papers*, pp. 7319–7328. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.acl-long.568. URL https://aclanthology.org/2021.acl-long.568/.

Sanjeev Arora, Zhiyuan Li, and Abhishek Panigrahi. Understanding gradient descent on the edge of stability in deep learning. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, volume 162 of *Proceedings of Machine Learning Research*, pp. 948–1024. PMLR, 2022. URL https://arxiv.org/abs/2205.09745.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try ARC, the AI2 reasoning challenge. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2018. URL https://arxiv.org/abs/1803.05457.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *CoRR*, abs/2110.14168, 2021. URL https://arxiv.org/abs/2110.14168.

Shibhansh Dohare, Qingfeng Lan, and A. Rupam Mahmood. Overcoming policy collapse in deep reinforcement learning. In *European Workshop on Reinforcement Learning (EWRL) 2023*, 2023. URL https://openreview.net/forum?id=m9Jfdz4ymO.

John C. Duchi, Michael I. Jordan, Martin J. Wainwright, and Andre Wibisono. Optimal rates for zero-order convex optimization: The power of two function evaluations. *IEEE Transactions on Information Theory*, 61(5):2788–2806, 2015. doi: 10.1109/TIT.2015.2409256. URL https://arxiv.org/abs/1312.2139.

Yulu Gan and Phillip Isola. Neural thickets: Diverse task experts are dense around pretrained weights. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=92oF5bU4cU.

Behrooz Ghorbani, Shankar Krishnan, and Ying Xiao. An investigation into neural net optimization via hessian eigenvalue density. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2232–2241. PMLR, 2019. URL https://arxiv.org/abs/1901.10159.

Guy Gur-Ari, Daniel A. Roberts, and Ethan Dyer. Gradient descent happens in a tiny subspace. *CoRR*, abs/1812.04754, 2018. URL http://arxiv.org/abs/1812.04754.

Peter Hänggi, Peter Talkner, and Michal Borkovec. Reaction-rate theory: fifty years after kramers. *Reviews of Modern Physics*, 62(2):251–341, 1990. doi: 10.1103/RevModPhys.62.251. URL https://link.aps.org/doi/10.1103/RevModPhys.62.251.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022. URL https://arxiv.org/abs/2106.09685.

Michael F. Hutchinson. A stochastic estimator of the trace of the influence matrix for Laplacian smoothing splines. *Communications in Statistics - Simulation and Computation*, 19(2):433–450, 1990. doi: 10.1080/03610919008812866.

Chi Jin, Rong Ge, Praneeth Netrapalli, Sham M. Kakade, and Michael I. Jordan. How to escape saddle points efficiently. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1724–1732. PMLR, 2017. URL https://proceedings.mlr.press/v70/jin17a.html.

Tony Lelièvre, Dorian Le Peutrec, and Boris Nectoux. Eyring-kramers exit rates for the overdamped langevin dynamics: the case with saddle points on the boundary. *Journal de l'École polytechnique — Mathématiques*, 12:881–982, 2025. doi: 10.5802/jep.303. URL https://jep.centre-mersenne.org/articles/10.5802/jep.303/.

Chunyuan Li, Heerad Farkhoor, Rosanne Liu, and Jason Yosinski. Measuring the intrinsic dimension of objective landscapes. In *International Conference on Learning Representations (ICLR)*, 2018. doi: 10.48550/arXiv.1804.08838. URL https://arxiv.org/abs/1804.08838.

Qianxiao Li, Cheng Tai, and Weinan E. Stochastic modified equations and adaptive stochastic gradient algorithms. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 2101–2110. PMLR, 2017. URL https://proceedings.mlr.press/v70/li17f.html.

Stephan Mandt, Matthew D. Hoffman, and David M. Blei. Stochastic gradient descent as approximate bayesian inference. *Journal of Machine Learning Research*, 18(134):1–35, 2017. doi: 10.48550/arXiv.1704.04289. URL https://jmlr.org/papers/v18/17-214.html.

Skander Moalla, Andrea Miele, Daniil Pyatko, Razvan Pascanu, and Caglar Gulcehre. No representation, no trust: connecting representation, collapse, and trust issues in ppo. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA, 2024. Curran Associates Inc. ISBN 9798331314385. URL https://arxiv.org/abs/2405.00662.

Xin Qiu, Yulu Gan, Conor F. Hayes, Qiyao Liang, Yinggan Xu, Roberto Dailey, Elliot Meyerson, Babak Hodjat, and Risto Miikkulainen. Evolution strategies at scale: LLM fine-tuning beyond reinforcement learning. In *Forty-third International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2509.24372. PMLR 306.

Levent Sagun, Utku Evci, V. Ugur Guney, Yann Dauphin, and Leon Bottou. Empirical analysis of the hessian of over-parametrized neural networks, 2017. URL https://arxiv.org/abs/1706.04454.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020. URL https://arxiv.org/abs/1907.10641.

Tim Salimans, Jonathan Ho, Xi Chen, Szymon Sidor, and Ilya Sutskever. Evolution strategies as a scalable alternative to reinforcement learning, 2017. URL https://arxiv.org/abs/1703.03864.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017. URL http://arxiv.org/abs/1707.06347.

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

Minhak Song, Kwangjun Ahn, and Chulhee Yun. Does SGD really happen in tiny subspaces? In *International Conference on Learning Representations (ICLR)*, 2025. URL https://arxiv.org/abs/2405.16002.

Shashanka Ubaru, Jie Chen, and Yousef Saad. Fast estimation of $\mathrm{tr}(f(a))$ via stochastic lanczos quadrature. *SIAM Journal on Matrix Analysis and Applications*, 38(4):1075–1099, 2017. doi: 10.1137/16M1104974.

Zhewei Yao, Amir Gholami, Kurt Keutzer, and Michael W. Mahoney. PyHessian: Neural networks through the lens of the Hessian. In *2020 IEEE International Conference on Big Data (Big Data)*, pp. 581–590, 2020. doi: 10.1109/BigData50022.2020.9378171.

Xingyu Zhu, Zixuan Wang, Xiang Wang, Mo Zhou, and Rong Ge. Understanding edge-of-stability training dynamics with a minimalist example. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=p7EagBsMAEO.

Appendix

Appendix Table of Contents

A. Quadratic OU Toy Model and Variance–Curvature Spectroscopy 13
B. From Toy Slope Laws to Practical Curvature Probes: CLSS vs. SLQ 15
C. Anisotropic Noise and the Lyapunov Equation 17
D. Beyond Quadratic: Saddles and Negative Curvature 19
E. Metastability in a Double-Well: Escape Times and Hopping Criteria 20
F. Additional Rise–Then–Decay Results: Countdown 22
G. Experimental Details: ES Reward-Probe 23
H. Direct Curvature Measurements on Qwen2.5-Instruct 28

A Quadratic OU Toy Model and Variance–Curvature Spectroscopy

This appendix provides analytic details underlying the toy-model mechanism used in Section 4. The main text uses this quadratic Ornstein–Uhlenbeck (OU) analysis to isolate the mechanism by which rise–then–decay dynamics arise generically from (i) heterogeneous curvature scales and (ii) fixed stochasticity, without requiring nonconvex pathologies. It is also shown how the same closed-form expressions yield a "spectroscopy" viewpoint. The terminal plateau varies predictably with the effective noise scale $\kappa = \sigma^2/N$, and the slope encodes a curvature-weighted effective dimension. This slope-fitting idea motivates the CLSS procedure introduced in the next appendix section (Section B).

A.1 From noisy updates to a local quadratic OU model

A generic noisy update abstraction. Consider maximizing an objective $J(\theta)$ with a constant-step noisy update

$$\theta_{t+1} = \theta_t + \alpha\big(g(\theta_t) + \xi_t\big), \tag{11}$$

where $\mathbb{E}[\xi_t \mid \theta_t] = 0$ and $\mathrm{Cov}(\xi_t \mid \theta_t) = \Sigma(\theta_t)$. For ES with isotropic perturbations, a useful local approximation is $\Sigma(\theta_t) \approx (\sigma^2/N)I$ (up to problem-dependent scaling), making the effective noise level explicitly tunable by $\kappa = \sigma^2/N$. Policy-gradient methods (e.g., GRPO) have anisotropic $\Sigma(\theta_t)$; the analysis below extends by replacing $(\sigma^2/N)I$ with a general covariance and solving a Lyapunov equation, but focuses here on the isotropic case to isolate curvature effects.

Why a quadratic model? The rise–then–decay signature in Section 4 indicates a regime where (i) gradients are small and (ii) stochasticity dominates long-time behavior. In this regime, a local Taylor expansion around a critical point $\theta^\star$ is the natural first approximation:

$$J(\theta^\star + x) \approx J(\theta^\star) + \tfrac{1}{2} x^\top H x, \tag{12}$$

where $H = \nabla^2 J(\theta^\star)$. Near a local maximizer, H is negative semidefinite; curvature is reparameterized by $C \triangleq -H \succeq 0$. The quadratic model is not intended as a global description of LLM fine-tuning; rather, it yields exact mode-wise relaxation rates and noise floors, making the origin of non-monotonicity transparent.

A.2 Quadratic toy model: exact dynamics and a criterion for non-monotonicity

Setup. Let $\theta_t \in \mathbb{R}^D$ and define the quadratic reward

$$J(\theta) = 1 - \tfrac{1}{2}\theta^\top C \theta, \tag{13}$$

with maximizer $\theta^\star = 0$. Then $\nabla J(\theta) = -C\theta$. ES-like noisy ascent is modeled by

$$\theta_{t+1} = (I - \alpha C)\theta_t + \alpha \frac{\sigma}{\sqrt{N}}\, \varepsilon_t, \qquad \varepsilon_t \sim \mathcal{N}(0, I_D), \tag{14}$$

and assume $\alpha \in (0, 2/\lambda_{\max}(C))$ for stability on the range of C.

Diagonalization and mode-wise OU/AR(1). Let $C = Q\Lambda Q^\top$ with $\Lambda = \mathrm{diag}(\lambda_1, \dots, \lambda_D)$. In the eigenbasis $x_t = Q^\top \theta_t$, modes decouple:

$$x_{i,t+1} = a_i x_{i,t} + b\,\xi_{i,t}, \qquad a_i = 1 - \alpha\lambda_i, \quad b = \alpha\sigma/\sqrt{N}, \quad \xi_{i,t} \sim \mathcal{N}(0,1). \tag{15}$$

Thus each spectral mode is an independent AR(1)/OU process. For small $\alpha\lambda_i$, the characteristic decay time scales as $\tau_i \approx (\alpha\lambda_i)^{-1}$. Stiff modes ($\lambda_i$ large) relax quickly while flat modes (λ_i small) relax slowly.

Mean/variance dynamics and stationary noise floor. Let $\mu_{i,t} = \mathbb{E}[x_{i,t}]$ and $v_{i,t} = \mathrm{Var}(x_{i,t})$. From (15):

$$\mu_{i,t} = a_i^t \mu_{i,0}, \qquad v_{i,t} = a_i^{2t} v_{i,0} + \frac{b^2}{1 - a_i^2}\big(1 - a_i^{2t}\big). \tag{16}$$

For $\lambda_i > 0$ and $|a_i| < 1$, the stationary variance is

$$v_{i,\infty} = \frac{b^2}{1 - a_i^2} = \frac{\alpha\sigma^2}{N\,\lambda_i(2 - \alpha\lambda_i)}. \tag{17}$$

The expected reward is

$$\mathbb{E}[J(\theta_t)] = 1 - \tfrac{1}{2} \sum_{i:\lambda_i>0} \lambda_i\big(\mu_{i,t}^2 + v_{i,t}\big). \tag{18}$$

At stationarity ($t \to \infty$), $\mu_{i,t} \to 0$ and $v_{i,t} \to v_{i,\infty}$, giving

$$1 - J_\infty = \tfrac{1}{2} \sum_{\lambda_i>0} \lambda_i v_{i,\infty} = \frac{\alpha\sigma^2}{2N} \sum_{\lambda_i>0} \frac{1}{2 - \alpha\lambda_i}. \tag{19}$$

Thus the terminal plateau is controlled by the effective noise $\kappa = \sigma^2/N$ and by a curvature-weighted trace functional of the spectrum.

Mixture-of-exponentials and non-monotonicity. Define amplitudes

$$A_i \triangleq -\tfrac{1}{2}\lambda_i\big(x_{i,0}^2 - v_{i,\infty}\big), \tag{20}$$

assuming deterministic initialization $x_{i,0}$.
Then the deviation from plateau is a mixture of exponentials:

$$\mathbb{E}[J(\theta_t)] - J_\infty = \sum_{\lambda_i>0} A_i a_i^{2t} = \sum_{\lambda_i>0} A_i e^{-\gamma_i t}, \qquad \gamma_i \triangleq -2\log(|1 - \alpha\lambda_i|). \tag{21}$$

Non-monotonicity occurs when the $\{A_i\}$ have mixed signs: fast stiff modes can initially increase reward, while slow flat modes contribute delayed degradation as variance accumulates. A pronounced peak arises when stiff modes begin far from equilibrium ($x_{i,0}^2 \gg v_{i,\infty}$) while flat modes begin near or beyond their noise-limited equilibrium ($x_{i,0}^2 \lesssim v_{i,\infty}$), matching the main-text initialization discussion.

Peak time: a two-timescale characterization. The mixture form (21) implies that peaks arise from competition between modes with different decay rates. Differentiating (treating t as continuous for analysis) gives

$$\frac{d}{dt}\Big(\mathbb{E}[J(\theta_t)] - J_\infty\Big) \;=\; -\sum_{\lambda_i>0} \gamma_i A_i e^{-\gamma_i t}. \tag{22}$$

An interior maximizer $t_{\mathrm{peak}} > 0$ therefore satisfies the implicit condition

$$\sum_{\lambda_i>0} \gamma_i A_i e^{-\gamma_i t_{\mathrm{peak}}} = 0, \tag{23}$$

which is possible only when the amplitudes $\{A_i\}$ have mixed signs.

Closed form in the two-mode case. To make the dependence explicit, consider two distinct eigenvalues $\lambda_{\mathrm{hi}} > \lambda_{\mathrm{lo}} > 0$ with rates $\gamma_{\mathrm{hi}} = -2\log(1 - \alpha\lambda_{\mathrm{hi}})$ and $\gamma_{\mathrm{lo}} = -2\log(1 - \alpha\lambda_{\mathrm{lo}})$, and amplitudes A_{hi} and A_{lo} as in (20). Then

$$\mathbb{E}[J(\theta_t)] - J_\infty \;=\; A_{\mathrm{hi}} e^{-\gamma_{\mathrm{hi}} t} + A_{\mathrm{lo}} e^{-\gamma_{\mathrm{lo}} t}. \tag{24}$$

A rise–then–decay peak *at a positive time* occurs when (i) $A_{\mathrm{hi}} < 0$ and $A_{\mathrm{lo}} > 0$ (equivalently, $x_{\mathrm{hi},0}^2 > v_{\mathrm{hi},\infty}$ and $x_{\mathrm{lo},0}^2 < v_{\mathrm{lo},\infty}$), so the amplitudes have mixed signs and a non-monotonic

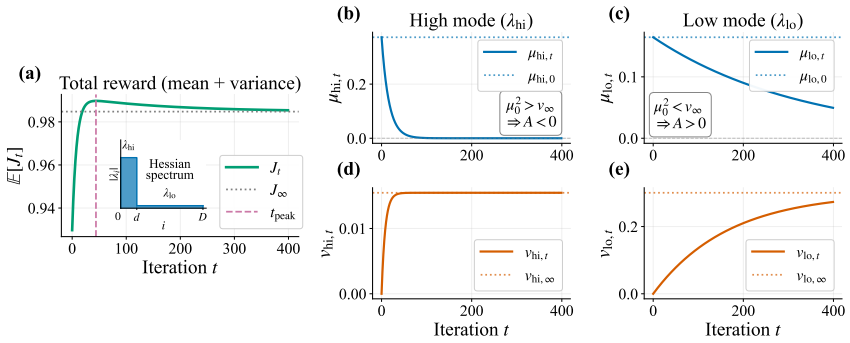

Figure 7: **Mode-wise relaxation rates explain rise–then–decay.** A two-mode illustration of (15)–(21) showing how curvature sets relaxation times. **(a)** Total reward $\mathbb{E}[J_t]$ peaks and then decays toward J_∞ as slow variance continues to accumulate after stiff-mode signal is exhausted (vertical line marks t_{peak}; inset shows the two-block spectrum). **(b,d)** In the stiff mode (λ_{hi}), the mean $\mu_{hi,t}$ decays rapidly and the variance $v_{hi,t}$ saturates quickly. **(c,e)** In the flat mode (λ_{lo}), the mean decays slowly while the variance accumulates over long time, producing delayed degradation. Notice that the flat-mode terminal variance is substantially larger and dominates late-time reward degradation.

trajectory is possible, and (ii) the initial slope $\frac{d}{dt}\mathbb{E}[J(\theta_t)]\big|_{t=0} = -\big(\gamma_{\text{hi}}A_{\text{hi}} + \gamma_{\text{lo}}A_{\text{lo}}\big) > 0$, i.e., $\gamma_{\text{hi}}|A_{\text{hi}}| > \gamma_{\text{lo}}A_{\text{lo}}$, so that reward initially rises rather than decaying monotonically. Condition (ii) also ensures that the argument of the logarithm in the closed-form peak time below is strictly greater than 1, so $t_{\text{peak}} > 0$. Under both conditions the peak time is

$$t_{\text{peak}} \;=\; \frac{1}{\gamma_{\text{hi}} - \gamma_{\text{lo}}} \log\!\left(\frac{\gamma_{\text{hi}}|A_{\text{hi}}|}{\gamma_{\text{lo}}A_{\text{lo}}}\right). \tag{25}$$

This expression shows that larger time-scale separation ($\gamma_{\text{hi}} \gg \gamma_{\text{lo}}$) and larger amplitude ratio ($|A_{\text{hi}}|/A_{\text{lo}}$) yield a more pronounced and later peak.

A.3 Variance–Curvature "Spectroscopy" in the Toy Model

Plateau versus effective noise and curvature-weighted dimension. Equation (19) implies that, in a locally quadratic and stable regime, the terminal gap scales linearly with $\kappa = \sigma^2/N$:

$$1 - J_\infty = \underbrace{\Big(\frac{\alpha}{2}\sum_{\lambda_i>0}\frac{1}{2-\alpha\lambda_i}\Big)}_{\text{curvature functional}} \cdot \kappa. \tag{26}$$

This motivates the curvature-weighted effective dimension

$$d_{\text{eff}}(\alpha) \;\triangleq\; 2\sum_{\lambda_i>0}\frac{1}{2-\alpha\lambda_i}, \qquad \Rightarrow \qquad 1 - J_\infty = \frac{\alpha}{4}\, d_{\text{eff}}(\alpha)\cdot\kappa. \tag{27}$$

For $\alpha\lambda_i \ll 1$, $d_{\text{eff}}(\alpha) \approx \text{rank}(C)$; at finite step size it smoothly reweights directions by curvature through $(2-\alpha\lambda_i)^{-1}$. Thus, in the toy quadratic setting, a slope fit of $(1 - J_\infty)$ versus κ recovers a scalar summary of local curvature complexity.

Slope fitting as "spectroscopy" in the toy setting. Figure 8 illustrates this viewpoint. Varying population size N changes κ and produces different terminal plateaus in the reward trajectory (panel a). Plotting the analytic plateau J_∞ against κ yields an approximately linear dependence whose slope increases with the number of curvature-active directions in a strict rank-d spectrum (panel b). In other words, higher effective dimension implies greater sensitivity of the noise floor to stochasticity. This motivates the checkpointed noise-floor probing method introduced next (Section B).

B From Toy Slope Laws to Practical Curvature Probes: CLSS vs. SLQ

The quadratic OU analysis in Appendix A yields a concrete measurement principle. In a locally stable region, the terminal "noise-floor" gap scales approximately linearly with the effective noise level $\kappa = \sigma^2/N$ (Eq. (26)), and the slope can be mapped to a curvature-weighted effective dimension (Eq. (27)). In the toy setting this relation is exact and enables a simple slope fit to recover d_{eff} (Fig. 8).

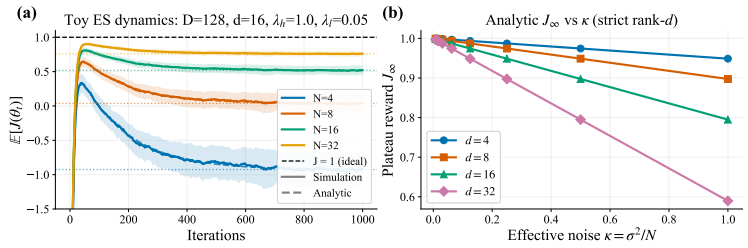

Figure 8: **Toy dynamics and plateau "spectroscopy" via effective noise $\kappa = \sigma^2/N$. (a)** Expected reward trajectories for different populations N (noise levels) on the two-block quadratic landscape, comparing Monte Carlo ES simulations to the analytic OU prediction; larger N reduces effective noise and raises the terminal plateau. **(b)** Analytic terminal plateau J_∞ versus effective noise κ for strict rank-d curvature. The approximately linear dependence predicted by Eq. (26) has a slope that grows with d, showing that plateau-vs-noise measurements recover a curvature-weighted effective dimension in the toy setting.

This motivates a practical way to estimate local geometry for black-box or reward-defined objectives where Hessian-vector products may be unavailable, using *checkpointed* noise-floor probing and slope fitting. This is referred to as *Checkpointed Local Slope Spectroscopy (CLSS)*. CLSS is then contrasted with Stochastic Lanczos Quadrature (SLQ), a standard matvec-based spectrum probe.

B.1 Checkpointed Local Slope Spectroscopy (CLSS)

Given a checkpoint $\theta^\star$, CLSS runs short, local probe trajectories at several controllable noise levels (e.g., varying N at fixed σ in ES), estimates a plateau proxy for each noise level, and fits the small-noise slope of the plateau gap versus $\kappa = \sigma^2/N$. In the quadratic regime, this slope maps to an effective dimension via Eq. (26)–(27). The algorithmic description of CLSS is given in Algorithm 2.

Practical limitations. CLSS is conceptually clean in the quadratic regime but can be difficult to apply reliably in real LLM fine-tuning. It requires (i) a *local* probe regime (trajectories must remain near $\theta^\star$), and (ii) a *measurable* plateau proxy over the probe horizon. In practice, nonconvex drift across regions, long transients, and bounded/truncated rewards (e.g., accuracy in $[0, 1]$) can obscure plateaus and inflate uncertainty. Moreover, effective noise is often anisotropic and state-dependent (particularly for policy-gradient methods), weakening a direct mapping from a scalar slope to a single curvature functional. For these reasons, the main text uses the slope law primarily as a *mechanistic explanation* and relies on more robust operational probes (e.g., best-of-N accessibility) that do not require clean plateau observation.

B.2 Stochastic Lanczos Quadrature (SLQ): matvec-based curvature probing

SLQ estimates trace functionals and (optionally) spectral densities of a symmetric operator A using only matrix–vector products $v \mapsto Av$. It combines Hutchinson trace estimation with Lanczos tridiagonalization and Gaussian quadrature to approximate quantities of the form $\mathrm{tr}(f(A))$ with $\mathcal{O}(sm)$ matvecs, where s is the number of random probes and m the number of Lanczos steps (Hutchinson, 1990; Ubaru et al., 2017). Tooling such as PyHessian implements related workflows for estimating top eigenvalues, trace, and smoothed spectral densities from Hessian–vector products (Yao et al., 2020; Ghorbani et al., 2019). When a reliable differentiable surrogate loss is available and matvecs are stable, SLQ provides rich spectral diagnostics (bulk/outliers, sharp subspaces) that complement the dynamics-based analysis above. The algorithmic description of SLQ is given in Algorithm 3.

B.3 When to use what

- **Use SLQ/PyHessian when matvecs are available.** If you can compute stable matvecs for a symmetric curvature operator tied to a differentiable surrogate (Hessian/GGN/Fisher), SLQ yields detailed spectral structure and top-eigenspace information.
- **Use CLSS when only objective evaluations are available and plateaus are measurable.** CLSS is applicable to black-box rewards and algorithm-native noise control (e.g., varying N in ES), but only to the extent that local probes remain near a checkpoint and produce a reliably measurable plateau.
- **In the LLM setting studied here, favor operational probes.** Because plateau estimation is often unreliable under bounded/truncated rewards and nonstationarity, CLSS is not used as a primary measurement; the main paper instead emphasizes robust probes (best-of-N) and dynamical signatures (rise–then–decay).

Algorithm 2: Checkpointed Local Slope Spectroscopy (CLSS)

Input: checkpoint $\theta^\star$; perturbation scale σ; candidate step sizes $\mathcal{A}$; population sizes $\mathcal{N}$; probe horizon T; tail window w; #seeds R;
locality metric $\mathrm{Loc}(\theta, \theta^\star)$ and threshold τ_{loc}; tail-settling tolerance τ_{stat}; minimum valid seeds $R_{\min}$.
Output: $\widehat{d}_{\mathrm{eff}}(\alpha; \theta^\star)$ with CI, or `FAIL`.

foreach $\alpha \in \mathcal{A}$ **do**
  `// (1) Local probes at fixed` α, σ `and varying population` N `(noise` $\kappa = \sigma^2/N$`)`
  foreach $N \in \mathcal{N}$ **do**
    $\mathcal{V} \leftarrow \emptyset$; `// valid seeds`
    for $r = 1$ **to** R **do**
      Run T probe steps from $\theta^\star$ with (α, σ, N) producing rewards $\{J_t\}_{t=1}^T$ and states $\{\theta_t\}_{t=1}^T$;
      if $\max_{t \le T} \mathrm{Loc}(\theta_t, \theta^\star) > \tau_{\mathrm{loc}}$ **then**
        continue
      $\bar{J}_1 \leftarrow \frac{1}{w} \sum_{t=T-w+1}^{T} J_t$;
      $\bar{J}_0 \leftarrow \frac{1}{w} \sum_{t=T-2w+1}^{T-w} J_t$;
      if $|\bar{J}_1 - \bar{J}_0| > \tau_{\mathrm{stat}}$ **then**
        continue
      $\mathcal{V} \leftarrow \mathcal{V} \cup \{\bar{J}_1\}$;
    if $|\mathcal{V}| < R_{\min}$ **then**
      mark N invalid; **continue**
    $\widehat{J}_\infty(N) \leftarrow \mathrm{mean}(\mathcal{V})$;
  `// (2) Fit small-noise slope of plateau gap vs` $\kappa = \sigma^2/N$
  Let $N_{\max}$ be the largest valid N and set $\widehat{J}_{\mathrm{ref}} \leftarrow \widehat{J}_\infty(N_{\max})$;
  For each valid N, set $g(N) \leftarrow \widehat{J}_{\mathrm{ref}} - \widehat{J}_\infty(N)$;
  Fit $g(N) \approx S_\alpha \cdot (\sigma^2/N) + b$ using the largest few valid N (small-noise regime);
  Output $\widehat{d}_{\mathrm{eff}}(\alpha; \theta^\star) \leftarrow \frac{4}{\alpha} S_\alpha$; `// by Eq. (27)`
  if *fit residuals small and acceptance rates high* **then**
    return $\widehat{d}_{\mathrm{eff}}(\alpha; \theta^\star)$
return `FAIL`.

C Anisotropic Noise and the Lyapunov Equation

The quadratic OU toy model in Appendix A assumed isotropic noise for clarity, which diagonalizes neatly in the eigenbasis of the curvature matrix. In practice, stochastic learning rules inject *anisotropic* and often state-dependent noise. This appendix summarizes the corresponding linearized theory, which shows that the same geometry–variance mechanism persists even when modes do not decouple: curvature still sets relaxation, while the noise covariance determines how variance accumulates across directions.

C.1 Linearized dynamics with anisotropic noise

Near a critical point $\theta^\star$, a general linearized update takes the form

$$x_{t+1} = (I - \alpha H) x_t + \alpha \eta_t, \qquad \mathbb{E}[\eta_t] = 0, \;\; \mathrm{Cov}(\eta_t) = \Sigma, \tag{30}$$

where $x_t = \theta_t - \theta^\star$ and H denotes the local positive-curvature operator that governs contraction of the linearized drift (in the sign convention of Appendix A, $H = C \succeq 0$ near a local maximizer of the reward, equivalently the negative of the reward Hessian). Let $V_t = \mathbb{E}[x_t x_t^\top]$. Then

$$V_{t+1} = (I - \alpha H) V_t (I - \alpha H)^\top + \alpha^2 \Sigma. \tag{31}$$

At stationarity ($V_{t+1} = V_t = V$), this is the *discrete Lyapunov equation*. For small α, expanding yields the continuous-time OU covariance balance

$$HV + VH^\top \approx \alpha \Sigma, \tag{32}$$

Algorithm 3: Stochastic Lanczos Quadrature (SLQ) for $\mathrm{tr}(f(A))$

Input: symmetric matrix/operator A (accessed only via matvec $v \mapsto Av$); scalar function f; #probes s; Lanczos steps m.

Output: estimate $\widehat{\mathrm{tr}(f(A))}$ and (optionally) a standard error.

`// Hutchinson trace estimator:` $\mathrm{tr}(f(A)) = \mathbb{E}[z^\top f(A) z]$ `for` $\mathbb{E}[zz^\top] = I$

for $j = 1$ **to** s **do**

Sample probe $z_j \in \{\pm 1\}^n$ i.i.d. Rademacher (or $z_j \sim \mathcal{N}(0, I)$).

Normalize $q_1 \leftarrow z_j / \|z_j\|_2$.

`// Lanczos tridiagonalization using only matvecs with` A

Run m-step Lanczos on A with start q_1 to obtain tridiagonal $T_j \in \mathbb{R}^{m \times m}$.

`// Gaussian quadrature:` $z^\top f(A) z \approx \|z\|^2 e_1^\top f(T) e_1$

Compute eigendecomposition $T_j = V_j \mathrm{diag}(\theta_j) V_j^\top$.

Set weights $w_{jk} \leftarrow (V_j)_{1k}^2$ for $k = 1, \ldots, m$.

Estimate quadratic form:

$$\widehat{\tau}_j \;\leftarrow\; \|z_j\|_2^2 \sum_{k=1}^{m} w_{jk}\, f(\theta_{jk}). \tag{28}$$

$$\widehat{\mathrm{tr}(f(A))} \;\leftarrow\; \frac{1}{s}\sum_{j=1}^{s} \widehat{\tau}_j, \qquad \widehat{\mathrm{SE}} \;\leftarrow\; \sqrt{\frac{\mathrm{Var}(\{\widehat{\tau}_j\}_{j=1}^{s})}{s}} \;\text{(optional)}. \tag{29}$$

return $\widehat{\mathrm{tr}(f(A))}$ (and $\widehat{\mathrm{SE}}$).

Method	What it needs	What it gives
SLQ / PyHessian	Matvec access $v \mapsto Av$ for a symmetric curvature operator (typically via differentiable surrogate)	Rich spectral diagnostics: top eigenvalues/eigenspace, trace functionals, smoothed spectral density
CLSS (slope probe)	Only objective evaluations under the actual optimizer; controllable noise $\kappa = \sigma^2/N$; a measurable local plateau	Scalar slope-based summary interpretable as curvature-weighted effective dimension in the quadratic regime

Table 1: **High-level comparison.** SLQ provides detailed spectral information when matvecs are available; CLSS is inspired by the toy slope law and can be used with black-box rewards, but requires reliable local plateau estimation.

matching standard OU/SDE approximations of constant-step stochastic optimization (Mandt et al., 2017; Li et al., 2017).

How the plateau depends on anisotropic noise. In the quadratic approximation $J(\theta^\star + x) \approx J(\theta^\star) - \frac{1}{2} x^\top H x$, the expected performance gap depends on the stationary second moment:

$$J(\theta^\star) - \mathbb{E}[J(\theta^\star + x)] \;\approx\; \tfrac{1}{2}\,\mathrm{tr}(HV). \tag{33}$$

Thus, even when H is low-dimensional in curvature (few stiff modes), the *effective noise floor* depends on how Σ injects variance into those modes. Compared to the isotropic case (Appendix A), anisotropy can (i) selectively amplify variance in particular curvature directions, (ii) rotate variance into stiff directions through off-diagonal structure, and (iii) change the apparent "effective dimension" by reweighting modes through Σ rather than through H alone. This mode-reweighting is fundamentally a *discrete-time* effect, arising from the $\alpha^2\Sigma$ term in the discrete Lyapunov equation (31). In the continuous-time limit, taking the trace of (32) and substituting into the expected gap $\frac{1}{2}\mathrm{tr}(HV)$ yields $\frac{\alpha}{4}\mathrm{tr}(\Sigma)$, which depends only on the total variance injected into the stable subspace and is invariant to how that variance is aligned with the curvature modes. The alignment-dependent reweighting therefore vanishes as $\alpha \to 0$ and is a finite-step phenomenon.

Stability and the OU regime. A stationary V exists only when the linear drift is stable, requiring the spectral radius of $(I - \alpha H)$ to be < 1 (in the reward-maximization convention, this corresponds to $H \succeq 0$ on the relevant subspace and α sufficiently small). When this holds, (31) provides a precise statement of the "variance accumulation" mechanism in the presence of anisotropy. The curvature H sets contraction, while Σ sets variance injection.

C.2 POLICY-GRADIENT / GRPO NOISE STRUCTURE

Policy-gradient methods provide a concrete example of anisotropic noise. Let $\mathcal{L}(\theta)$ denote a differentiable surrogate loss used for optimization (e.g., a GRPO/PPO-style objective). A generic stochastic gradient update has the form

$$\theta_{t+1} = \theta_t - \alpha\, \widehat{g}_t, \qquad \widehat{g}_t = \nabla \mathcal{L}(\theta_t) + \zeta_t, \tag{34}$$

where ζ_t is stochastic gradient noise induced by sampling trajectories/tokens and minibatches. Under local linearization around $\theta^\star$, this yields (30) with $H = \nabla^2 \mathcal{L}(\theta^\star)$ and $\Sigma \approx \mathrm{Cov}(\zeta_t)$.

Score-function form and Fisher-like covariance. For a standard policy gradient estimator (for notational simplicity suppressing state conditioning),

$$\widehat{g} = -\frac{1}{B} \sum_{b=1}^{B} \widehat{A}_b\, \nabla_\theta \log \pi_\theta(a_b), \tag{35}$$

where $\widehat{A}_b$ is an advantage-like scalar and B is the batch size. Conditioned on θ, the covariance of $\widehat{g}$ can be decomposed into (i) a Fisher-like term from $\nabla \log \pi$ and (ii) scaling/mixing induced by the random advantages:

$$\Sigma \;\approx\; \mathrm{Cov}(\widehat{g}) \;=\; \frac{1}{B}\Big(\mathbb{E}[\widehat{A}^2\, (\nabla \log \pi)(\nabla \log \pi)^\top] - \mathbb{E}[\widehat{A}\, \nabla \log \pi]\, \mathbb{E}[\widehat{A}\, \nabla \log \pi]^\top\Big). \tag{36}$$

Even when H is low-dimensional in curvature, the effective noise floor is determined by how this anisotropic Σ projects onto curvature-active directions through the Lyapunov equation (32).

Effect of GRPO group normalization. GRPO introduces a *group-relative* normalization that centers (and often scales) advantages within groups. At a high level, centering acts like a control variate that removes components of noise aligned with the within-group mean, while scaling by a within-group standard deviation can change the effective noise magnitude and induce correlations across samples within a group. Consequently, GRPO changes Σ in a structured, data-dependent manner rather than simply reducing variance uniformly. This provides a natural explanation for why non-monotonic reward dynamics can appear in GRPO as well as ES (Fig. 1). Even if the mean update direction improves performance early, anisotropic noise interacting with heterogeneous curvature can drive a variance-dominated late regime.

Empirical proxies for Σ. Recall that $\Sigma = \mathrm{Cov}(\widehat{g})$ here denotes the covariance of the *minibatch-averaged* gradient $\widehat{g} = \frac{1}{B}\sum_{b=1}^{B} \widehat{g}_b$, so its variance scales as $1/B$ relative to the per-sample covariance. An unbiased estimator of Σ from B i.i.d. per-sample gradients $\{\widehat{g}_b\}_{b=1}^{B}$ is therefore

$$\widehat{\Sigma} \;=\; \frac{1}{B(B-1)} \sum_{b=1}^{B} (\widehat{g}_b - \bar{g})(\widehat{g}_b - \bar{g})^\top, \tag{37}$$

where $\bar{g} = \frac{1}{B}\sum_b \widehat{g}_b$ and the additional $1/B$ factor (relative to the standard sample-covariance formula) reflects that this estimates the covariance of the mean rather than of an individual sample. or by low-rank projections (e.g., Hutchinson probing or subspace restrictions) when full matrices are infeasible. Combined with top-eigenspace estimates of H (when available), the Lyapunov relation provides a principled way to predict which directions will accumulate variance and when a variance-dominated regime is expected.

D BEYOND QUADRATIC: SADDLES AND NEGATIVE CURVATURE

The OU analysis in the main text and Appendix A describes *within-basin* dynamics around a locally stable region (a local maximizer for reward, or minimizer for loss). Real fine-tuning landscapes are nonconvex and nonstationary, and two additional effects can complicate plateau-based reasoning and amplify non-monotonicity.

Saddles and negative curvature. OU stationarity requires stable drift. Near strict saddles, negative-curvature directions make the linear drift unstable (e.g., eigenvalues of $(I - \alpha H)$ have magnitude > 1), so a stationary local covariance does not exist. In practice, stochasticity can help escape saddles; perturbed gradient methods escape strict saddles efficiently under suitable conditions (Jin et al., 2017). In such regions, apparent "plateaus" can be transient and slope-based noise-floor estimation can be misleading.

Basin-to-basin transitions and nonstationarity. Even near locally stable regions, finite-step stochastic updates can drift across regions of the landscape, especially along flat directions. This can invalidate single-basin plateau assumptions and produce additional non-monotonicity beyond the OU mechanism. For RL-style fine-tuning, further nonstationarity arises because the effective objective can change with the policy distribution, sampling temperature, KL penalties, or group normalization. Qualitatively, these effects are consistent with the observation that GRPO trajectories can also exhibit peak–then–decay behavior (Fig. 1), even though GRPO optimizes a differentiable surrogate. Both the surrogate geometry and its noise structure can evolve during training.

Implications for these measurements. These considerations motivate two design choices in the main paper. First, rise–then–decay is treated as a *diagnostic* of heterogeneous curvature and variance coupling, rather than as evidence of a literal stationary OU regime throughout training. Second, operational probes (e.g., best-of-N accessibility) that do not require a clean, long-horizon plateau in a single basin are emphasized, and plateau-based slope-fitting ideas are relegated to controlled toy settings and appendix discussion.

E Metastability in a Double-Well: Escape Times and Hopping Criteria

The quadratic OU analysis describes within-basin behavior near a locally stable region. To understand when stochastic optimization transitions between basins ("hopping"), consider the canonical metastable case of overdamped Langevin dynamics in a double-well. The key takeaway is that escape times depend *exponentially* on a barrier-to-noise ratio, so trajectories can appear either (i) effectively trapped or (ii) rapidly delocalized, with a narrow intermediate regime of rare hopping.

E.1 From noisy gradient steps to an overdamped Langevin diffusion

Consider one-dimensional noisy gradient descent on a loss $L(x)$,

$$x_{k+1} \;=\; x_k - \alpha L'(x_k) + \alpha\,\xi_k, \qquad \xi_k \sim \mathcal{N}(0, \sigma^2/N). \tag{38}$$

For small step size α, (38) is the Euler–Maruyama discretization of the overdamped Langevin SDE

$$dx_t \;=\; -L'(x_t)\,dt + \sqrt{2\varepsilon}\,dW_t, \tag{39}$$

with effective noise intensity ε obtained by matching per-step variances. Identifying $dt = \alpha$ and matching $2\varepsilon\,dt \approx \alpha^2(\sigma^2/N)$ yields

$$\varepsilon \;\approx\; \frac{\alpha\sigma^2}{2N} \;=\; \frac{\alpha}{2}\,\kappa, \tag{40}$$

where $\kappa = \sigma^2/N$ is the effective ES noise scale used throughout the paper.

E.2 Eyring–Kramers escape time in 1D

Let x_- be a local minimum of L and let z be the adjacent saddle (in 1D, the local maximum separating the well). Define the barrier height $\Delta L \triangleq L(z) - L(x_-)$. In the small-noise regime $\varepsilon \to 0$, the mean first exit time obeys the Eyring–Kramers law

$$\mathbb{E}[\tau_{\mathrm{esc}}] \;\approx\; \frac{2\pi}{\sqrt{L''(x_-)\,|L''(z)|}}\,\exp\Big(\frac{\Delta L}{\varepsilon}\Big), \tag{41}$$

up to lower-order corrections (Hänggi et al., 1990; Lelièvre et al., 2025). Since one discrete step corresponds to $dt = \alpha$, the expected number of iterations to escape is

$$\mathbb{E}[K_{\mathrm{esc}}] \;\approx\; \frac{1}{\alpha}\,\mathbb{E}[\tau_{\mathrm{esc}}] \;\approx\; \frac{2\pi}{\alpha\sqrt{L''(x_-)\,|L''(z)|}}\,\exp\Big(\frac{\Delta L}{\varepsilon}\Big). \tag{42}$$

Consequently, the probability of at least one hop within T iterations is approximately

$$\Pr(\text{hop by } T) \;\approx\; 1 - \exp\Big(-\frac{T}{\mathbb{E}[K_{\mathrm{esc}}]}\Big) \;\approx\; \frac{T}{\mathbb{E}[K_{\mathrm{esc}}]} \quad \text{when } T \ll \mathbb{E}[K_{\mathrm{esc}}]. \tag{43}$$

E.3 Specialization to the quartic double-well

For the standard quartic double-well

$$L(x) \;=\; \frac{\lambda_{\mathrm{dw}}}{4}\,(x^2 - a^2)^2, \tag{44}$$

the minima are at $x_\pm = \pm a$ and the saddle is at $z = 0$. The barrier height is

$$\Delta L \;=\; L(0) - L(-a) \;=\; \frac{\lambda_{\mathrm{dw}}}{4}\,a^4, \tag{45}$$

and the curvatures are

$$L''(\pm a) = 2\lambda_{\mathrm{dw}} a^2, \qquad L''(0) = -\lambda_{\mathrm{dw}} a^2. \tag{46}$$

Plugging (45)–(46) into (42) yields

$$\mathbb{E}[K_{\mathrm{esc}}] \;\approx\; \frac{2\pi}{\alpha\,\lambda_{\mathrm{dw}} a^2\sqrt{2}}\,\exp\Big(\frac{\Delta L}{\varepsilon}\Big) \;=\; \frac{2\pi}{\alpha\,\lambda_{\mathrm{dw}} a^2\sqrt{2}}\,\exp\Big(\frac{\lambda_{\mathrm{dw}} a^4}{2}\cdot\frac{N}{\alpha\sigma^2}\Big). \tag{47}$$

E.4 A practical hopping criterion and regimes

Escape is controlled by the dimensionless barrier-to-noise ratio

$$\frac{\Delta L}{\varepsilon} \;=\; \frac{\lambda_{\mathrm{dw}} a^4/4}{\alpha\sigma^2/(2N)} \;=\; \frac{\lambda_{\mathrm{dw}} a^4}{2}\cdot\frac{N}{\alpha\sigma^2}. \tag{48}$$

Because $\mathbb{E}[K_{\mathrm{esc}}]$ depends *exponentially* on (48), a coarse threshold for observing at least one hop within T iterations is

$$\frac{\Delta L}{\varepsilon} \;\approx\; \log T, \tag{49}$$

up to logarithmic corrections from the prefactor. This yields three qualitative regimes:

- **Metastable (no hops):** $\Delta L/\varepsilon \gg \log T$.
- **Metastable hopping:** $\Delta L/\varepsilon \approx O(\log T)$.
- **Delocalized:** $\Delta L/\varepsilon \lesssim 1$.

Figure 9 visualizes these three regimes in simulation. As $\Delta L/\varepsilon$ decreases, trajectories transition from remaining confined to one well (top), to occasional barrier crossings over the horizon (middle), and finally to frequent crossings that wash out basin localization (bottom). This sharp qualitative transition over finite horizons reflects the exponential sensitivity of escape times to the barrier-to-noise ratio in Eq. (42).

E.5 Local "water level" within a well

Within one well (e.g., near $x = -a$), L is approximately quadratic with curvature $\kappa_{\mathrm{well}} = L''(-a) = 2\lambda_{\mathrm{dw}} a^2$, yielding an OU approximation with stationary variance

$$\mathrm{Var}(x) \;\approx\; \frac{\varepsilon}{\kappa_{\mathrm{well}}} \;=\; \frac{\alpha\sigma^2}{4N\,\lambda_{\mathrm{dw}} a^2}, \tag{50}$$

making explicit how local curvature suppresses fluctuations within a basin.

Relevance to the main analysis (from within-basin OU to basin hopping). The quadratic OU analysis in Appendix A isolates a *within-basin* mechanism. Near a locally stable region, fixed stochasticity induces a noise-controlled plateau and, with heterogeneous curvature, can produce rise–then–decay without requiring nonconvex pathologies. The double-well analysis in Appendix E complements this picture by characterizing when stochastic learning leaves a basin altogether. Its key message is that basin transitions are controlled by an exponentially sensitive barrier-to-noise ratio (Eq. (48)), yielding three regimes over a finite training horizon T: effectively no hops, rare hops, or delocalization.

In a high-dimensional nonconvex fine-tuning landscape, local neighborhoods can be viewed as collections of basins separated by saddles of varying heights, and different stochastic learning rules induce different effective noise intensities and directions. Basin-hopping need not be ubiquitous, since even modest changes in the effective noise level (through α, σ^2/N, temperature, batch size, or group size) can move training across the hopping threshold (49) for some barriers but not others. Operationally, this provides a second (non-exclusive) source of late-stage non-monotonicity beyond within-basin variance accumulation. If stochasticity is large enough to cross low barriers along weakly

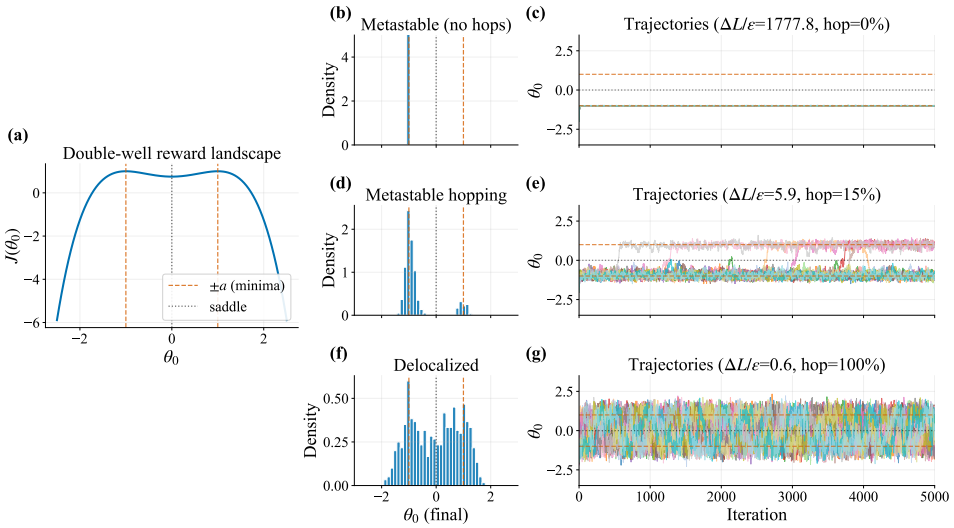

Figure 9: **Metastability and basin hopping in a double-well under stochastic updates.** **(a)** One-dimensional double-well reward landscape $J(\theta_0)$ (with other coordinates set to zero), with minima at $\pm a$ and a saddle near 0. **(b,d,f)** Histograms of θ_0 at the final iteration across runs. **(c,e,g)** Example trajectories of $\theta_0(t)$ across runs. The three rows illustrate the regimes predicted by the barrier-to-noise ratio $\Delta L/\varepsilon$ (Eq. (48)): **top:** metastable confinement (no hops), **middle:** rare basin hopping, and **bottom:** delocalized behavior with frequent transitions. "hop%" denotes the fraction of runs that switch wells at least once under a simple hysteresis-based definition (Appendix E).

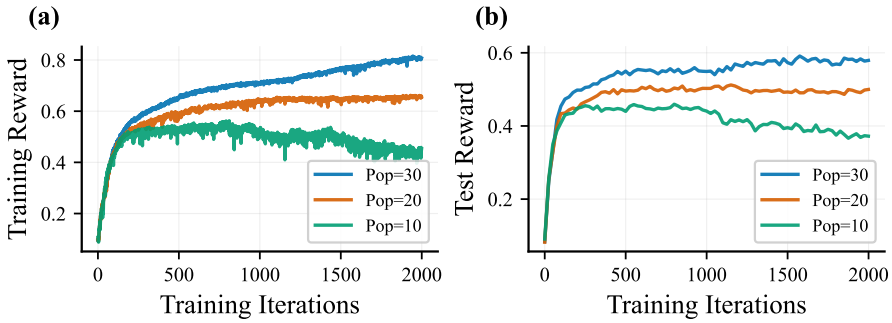

Figure 10: **Rise–then–decay behavior on the Countdown task.** Training (left) and held-out test (right) reward trajectories for ES fine-tuning on the Countdown arithmetic task under fixed hyperparameters, shown for population sizes $N \in \{10, 20, 30\}$. Across all population sizes, reward improves rapidly from the pretrained checkpoint, reaches a peak, and subsequently declines toward a lower value. Larger populations delay the onset and reduce the magnitude of decay but do not eliminate the non-monotonic behavior. These dynamics closely mirror those observed on GSM8K, ARC-C, and WinoGrande, consistent with rise–then–decay arising from stochastic fine-tuning interacting with heterogeneous curvature rather than from a task-specific artifact.

constrained directions, training may drift between nearby basins, leading to additional variability or degradation in reward. Conversely, when $\Delta L/\varepsilon \gg \log T$ for the relevant barriers, training remains effectively confined and the within-basin OU mechanism is the appropriate local description. This perspective motivates treating non-monotonic dynamics as potentially arising from both (i) within-basin variance–curvature effects and (ii) finite-horizon basin-hopping, with their relative importance governed by the same effective noise scale.

F Additional Rise–Then–Decay Results: Countdown

Additional evidence for non-monotonic fine-tuning dynamics is provided on the Countdown task, a structured arithmetic reasoning benchmark distinct from GSM8K, ARC-C, and WinoGrande. Figure 10 shows ES training and test reward trajectories under fixed hyperparameters for population sizes $N \in \{10, 20, 30\}$.

Consistent with the main experiments, all runs exhibit a pronounced rise–then–decay pattern. Reward initially increases as stochastic updates exploit high-curvature (stiff) directions associated with rapid improvement, but later decreases as improvement along these directions saturates and stochastic drift accumulates along weakly curved (flat) directions. Increasing the population size delays the onset of decay and raises peak reward, reflecting reduced sampling variance, but does not qualitatively alter the trajectory shape.

These results reinforce two key conclusions. First, non-monotonic reward trajectories are not tied to a specific task, evaluation protocol, or benchmark family, but recur across tasks under fixed stochasticity. Second, the qualitative dependence on population size is consistent with the geometry–variance framework, in which stochasticity interacts with anisotropic curvature to produce early gains followed by late-time degradation once signal along stiff directions is exhausted. Together with the main results, the Countdown experiments support the hypothesis that rise–then–decay reflects the geometry of fine-tuning landscapes rather than an artifact of a particular dataset or algorithmic detail.

F.1 Countdown task and training setup

Dataset and prompt format. The public `Jiayi-Pan/Countdown-Tasks-3to4` HuggingFace dataset is used, which provides arithmetic-reasoning problems composed of 3–4 input numbers and a target value. Following the dataset's single training split, the last 2,048 examples are reserved as the held-out test set used for the right column of Figure 10, with all preceding examples used for training. Each example is presented to the model with the system message *"You are a helpful assistant. You first think about the reasoning process in your mind and then provide the user with the answer."* and the user message *"Using the numbers {numbers}, create an equation that equals {target}. You can use basic arithmetic operations* $(+, -, *, /)$ *and each number can only be used once. Show your work in* `<think> </think>` *tags. And return the final answer in* `<answer> </answer>` *tags, for example* `<answer> (1 + 2) / 3 </answer>`*."* The assistant response is forced to begin with the prefix *"Let me solve this step by step.\n*`<think>`*"* so that all rollouts share the same reasoning scaffold; reward is binary (1 if the parsed expression equals the target and uses each input exactly once, otherwise 0).

Algorithm and training hyperparameters. The Countdown trajectories in Figure 10 are produced by weight-perturbation ES (Algorithm 1) with baseline-centered rewards, matching the algorithmic setup used for the GSM8K rise–then–decay runs in Figure 1. Training uses `Qwen2.5-0.5B-Instruct` as the base model, learning rate $\alpha = 5 \times 10^{-4}$, perturbation scale $\sigma = 10^{-3}$, population size $N \in \{10, 20, 30\}$ (separate runs per population), and a maximum of $1{,}000$ optimization steps. All other inference settings (`bfloat16` parameters, greedy decoding for reward evaluation, max 512 new tokens) match those described in Appendix G.

G Experimental Details: ES Reward-Probe

This appendix describes the ES reward-probe protocol used to generate the scaling figures in the main text (e.g., Fig. 6) and the related appendix analyses. The goal of these probes is to characterize reward changes under random weight perturbations at controlled noise scale and to estimate extreme-value quantities (best-of-N) in a manner that cleanly separates *perturbation randomness* (algorithmic) from *evaluation-set randomness* (measurement).

G.1 Models and inference

Instruction-tuned models from the Qwen2.5-Instruct family are evaluated:

Model size	HuggingFace model ID
0.5B	`Qwen/Qwen2.5-0.5B-Instruct`
1.5B	`Qwen/Qwen2.5-1.5B-Instruct`
3B	`Qwen/Qwen2.5-3B-Instruct`
7B	`Qwen/Qwen2.5-7B-Instruct`

Models are loaded in `bfloat16`. Inference uses HuggingFace Transformers with vLLM as the generation backend; for consistency, eager attention mode is used (flash attention disabled). Decoding is greedy (temperature 0.0) with a maximum of 512 new tokens. A fixed generation seed (42) is used across candidates to ensure deterministic prompt-to-output mapping given parameters.

G.2 Tasks, datasets, and rewards

Three tasks are probed using the training splits of standard datasets and binary per-prompt rewards:

Task	Dataset	Split	Reward
GSM8K	`gsm8k/main`	train	1 iff final number matches GT
ARC-C	`allenai/ai2_arc` (Challenge)	train	1 iff correct A/B/C/D
WinoGrande	`allenai/winogrande` (`winogrande_xl`)	train	1 iff correct A/B

GSM8K (Cobbe et al., 2021) is prompted in chat style with "Let us think step by step" and the final answer requested after the delimiter ####; reward extraction parses the final number. ARC-C (Clark et al., 2018) uses a multiple-choice prompt listing choices A–D; reward checks the chosen letter. WinoGrande (Sakaguchi et al., 2020) uses a sentence-completion prompt with two options A/B; reward checks the chosen option.

G.3 Evaluation pool

For each task, a fixed evaluation pool of $P = 320$ prompts is constructed by a deterministic random shuffle with `pool_seed=0`. The same pool is used for all model sizes within a task to ensure comparability across scales. All main-text best-of-N and best-of-30 point estimates are computed on this fixed pool (see below), with uncertainty quantified separately.

G.4 Perturbations and candidate generation

For parameters θ, each ES probe candidate applies a single isotropic Gaussian perturbation

$$\theta' = \theta + \sigma\, u, \qquad u \sim \mathcal{N}(0, I), \tag{51}$$

where u matches the shapes of all model parameters. u is not normalized (so $\|u\| \approx \sqrt{d}$ in d parameters). Perturbations are applied in-place and then exactly restored after evaluation; this is critical for numerical fidelity in `bfloat16`. Restoration fidelity is monitored by periodically re-evaluating the baseline reward at θ before and after a perturbation cycle and confirming it remains stable to within measurement noise.

Perturbation scales. Main figures use $\sigma \in \{3 \times 10^{-4}, 10^{-3}, 3 \times 10^{-3}\}$; extended analyses additionally include $\sigma \in \{10^{-4}, 10^{-2}\}$.

Pseudo-batches (for uncertainty). To avoid inflating extreme-value estimates via evaluation-set noise, two sources of randomness are separated: (i) perturbation randomness (the ES sampling process) and (ii) evaluation-set randomness (finite prompt pool). For each (task, model size, σ) condition, a single pool of $M = 240$ candidates is drawn, with perturbation seeds generated from a base seed (1234) via `rng.integers(0,2^31,size=240)`. Point estimates (Section G.6) are computed on this full pool. For uncertainty, the same $M = 240$-candidate pool is randomly partitioned into $S = 8$ disjoint *pseudo-batches* of $M/S = 30$ candidates each, a best-of-N estimate is computed within each pseudo-batch, and $\pm 1.96\times$SE is reported across the $S = 8$ pseudo-batch estimates. This measures perturbation/algorithmic variability within the collected pool rather than evaluation-set noise; it is not equivalent to redrawing S independent M-candidate pools from independent RNG streams, which was not done.

G.5 Reward computation and stored data

For each candidate and prompt, a binary reward $r_i \in \{0, 1\}$ is computed. The mean reward on the pool is

$$R = \frac{1}{P} \sum_{i=1}^{P} r_i, \qquad \Delta R = R_{\text{candidate}} - R_{\text{baseline}}. \tag{52}$$

Per-candidate data is stored as (i) the scalar ΔR on the full pool and (ii) packed bitstrings for the baseline and candidate per-prompt rewards (320 bits) encoded in base64. This enables exact reconstruction and re-aggregation under alternative evaluation procedures (e.g., prompt bootstrap for sensitivity analyses).

G.6 Estimating best-of-N without evaluation-noise inflation

The main paper uses extreme-value statistics because selection is central to ES and because improvement directions are rare. However, best-of-N is sensitive to additional measurement noise. Injecting evaluation-set resampling inside the maximization can inflate the maximum (a "winner's curse"). Best-of-N point estimates are therefore computed on the *full fixed pool*, with pseudo-batches drawn from that same pool used to quantify uncertainty.

Best-of-N point estimate (full pool). Let $\{\Delta R_j\}_{j=1}^{M}$ denote candidate deltas on the full $M = 240$-candidate pool for a given (task, model size, σ) condition. For a given population size N, the point estimate is

$$\widehat{\Delta_N^*}(\theta, \sigma) \;=\; \frac{1}{K} \sum_{k=1}^{K} \max_{j \in \mathcal{I}_k} \Delta R_j, \tag{53}$$

where each $\mathcal{I}_k$ is an independently drawn, uniformly random subset of $\{1, \ldots, M\}$ of size N without replacement, and $K = 2000$ Monte Carlo repetitions are used. This is computationally cheap since it operates only on stored scalar deltas, and it is computed once on the full pool rather than averaged across pseudo-batches.

Uncertainty (pseudo-batch variability). Error bars in the main-text best-of-N curves are obtained separately from the point estimate above. The same $M = 240$-candidate pool is partitioned into $S = 8$ disjoint pseudo-batches of $M/S = 30$ candidates each (Appendix G); a best-of-N estimate is computed within each pseudo-batch by the same Monte Carlo subset-sampling procedure restricted to that pseudo-batch's 30 candidates, and $\pm 1.96 \times \mathrm{SE}$ is reported, where SE is the standard error of the $S = 8$ pseudo-batch estimates $\{\Delta_N^{*,(s)}\}_{s=1}^{S}$. These uncertainty estimates reflect algorithmic variability due to perturbation sampling within the collected pool, which is the relevant uncertainty for population-requirement claims. Because each pseudo-batch contains only $M/S = 30$ candidates, this procedure yields a nonzero error bar only for $N \leq 30$; for $N = 50$ (the one value in $\{5, 10, 20, 30, 50\}$ exceeding the pseudo-batch size), the full-pool point estimate is reported without a pseudo-batch-derived error bar.

Population sizes. For best-of-N curves, $N \in \{5, 10, 20, 30, 50\}$ is evaluated, with $N = 30$ emphasized in the main text.

Evaluation-set uncertainty and extreme-value effects (appendix only). Evaluation-set uncertainty is additionally quantified via bootstrap over prompts (resampling the $P = 320$ prompts with replacement) *without* changing the candidate set. This provides confidence intervals for statistics on the fixed pool. Prompt bootstrap is *not* used to define main-text best-of-N point estimates because doing so introduces extra noise inside the maximization and can inflate maxima, consistent with classical extreme-value sensitivity (often modeled by Gumbel-type behavior for maxima). Prompt bootstrap and subset-size sensitivity studies are used only as robustness checks; the corresponding tables and figures are deferred to the supplementary material due to space constraints, with the bootstrap protocol summarized in Appendix G.6 above.

G.7 Other summary statistics

Also reported are (i) $p(\mathrm{improve}) = \Pr(\Delta R > 0)$ and (ii) $\mathbb{E}[\Delta R]$ across candidates. Unless stated otherwise, these are computed on the full pool with uncertainty obtained by bootstrap over candidates (1000 replicates, resampling candidates with replacement). Figure 11 summarizes baseline performance and average perturbation statistics ($p(\Delta R > 0)$ and $\mathbb{E}[\Delta R]$), highlighting that most perturbations are non-improving even when best-of-N is positive. Figure 12 shows sensitivity of best-of-30 to perturbation scale σ (raw and headroom-normalized), illustrating the existence of an intermediate σ regime where improvements are accessible.

G.8 Headroom normalization

To control for differences in baseline accuracy across model sizes, headroom-normalized improvements are reported:

$$\Delta^{*,\mathrm{rel}} = \frac{\Delta^*}{1 - R_0}, \tag{54}$$

where R_0 is the baseline accuracy on the fixed pool. This measures improvement as a fraction of remaining achievable reward (given the binary accuracy ceiling at 1). The baseline accuracies for different model sizes and different tasks are shown in Figure 11.

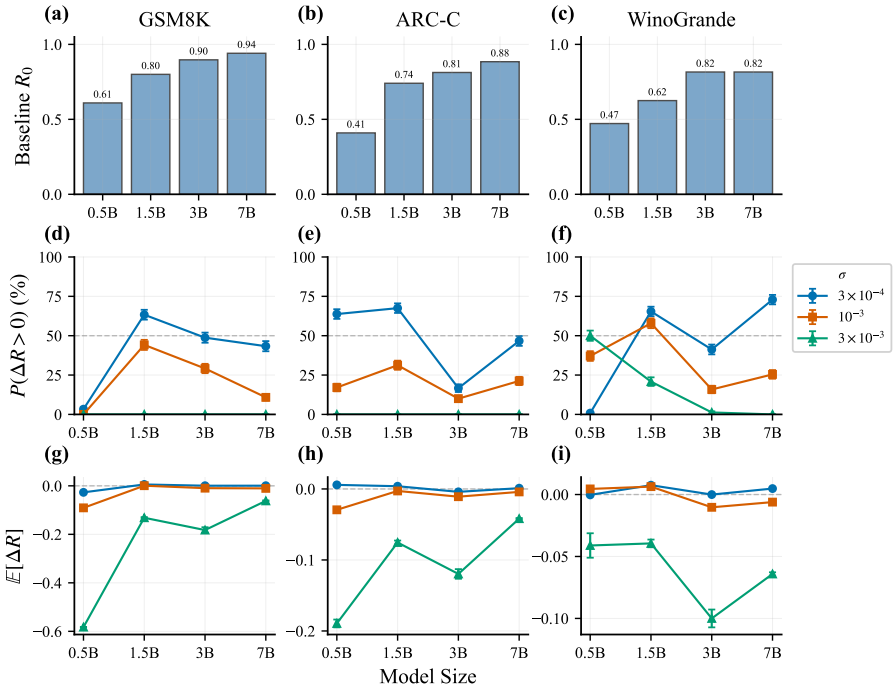

Figure 11: **Baseline performance and average perturbation statistics.** Columns correspond to GSM8K, ARC-C, and WinoGrande; rows show complementary summary statistics computed on the fixed evaluation pool ($P = 320$ prompts). **Top row (a–c):** baseline accuracies R_0 for each model size. **Middle row (d–f):** probability of improvement $p(\Delta R > 0)$ across perturbation scales $\sigma \in \{3 \times 10^{-4}, 10^{-3}, 3 \times 10^{-3}\}$. **Bottom row (g–i):** mean improvement $\mathbb{E}[\Delta R]$ across perturbations at the same σ values. In many regimes the mean improvement is near zero or negative and $p(\Delta R > 0) < \frac{1}{2}$, indicating that most perturbations are non-improving; this motivates extreme-value statistics (best-of-N) in the main text.

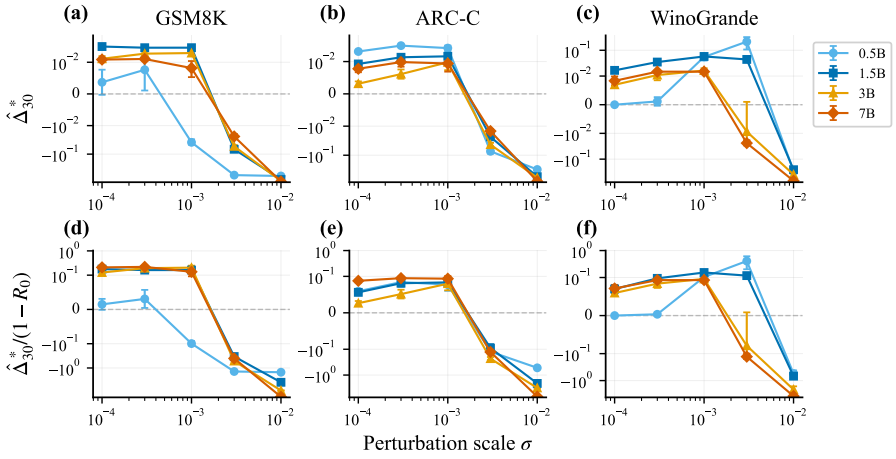

Figure 12: **Perturbation-scale sensitivity of best-of-30 improvements.** Best-of-30 expected improvement $\widehat{\Delta}^*_{30}$ (top row, a–c) and headroom-normalized best-of-30 $\widehat{\Delta}^*_{30}/(1 - R_0)$ (bottom row, d–f) as a function of perturbation scale σ for GSM8K, ARC-C, and WinoGrande. Moderate σ values yield positive best-of-30 improvements across model sizes, while very large σ can produce negative improvements, consistent with a variance-dominated regime.

Headroom-normalized improvement distributions. To complement the best-of-N summaries in the main text, the full distribution of perturbation outcomes is visualized. For each task, model size, and perturbation scale σ, $M = 240$ random weight perturbations are sampled and the headroom-normalized change $\Delta R/(1 - R_0)$ is computed, where $\Delta R = R(\theta + \sigma\varepsilon) - R_0$ and R_0 is the baseline accuracy on the fixed prompt pool. Headroom normalization enables fair comparison across models

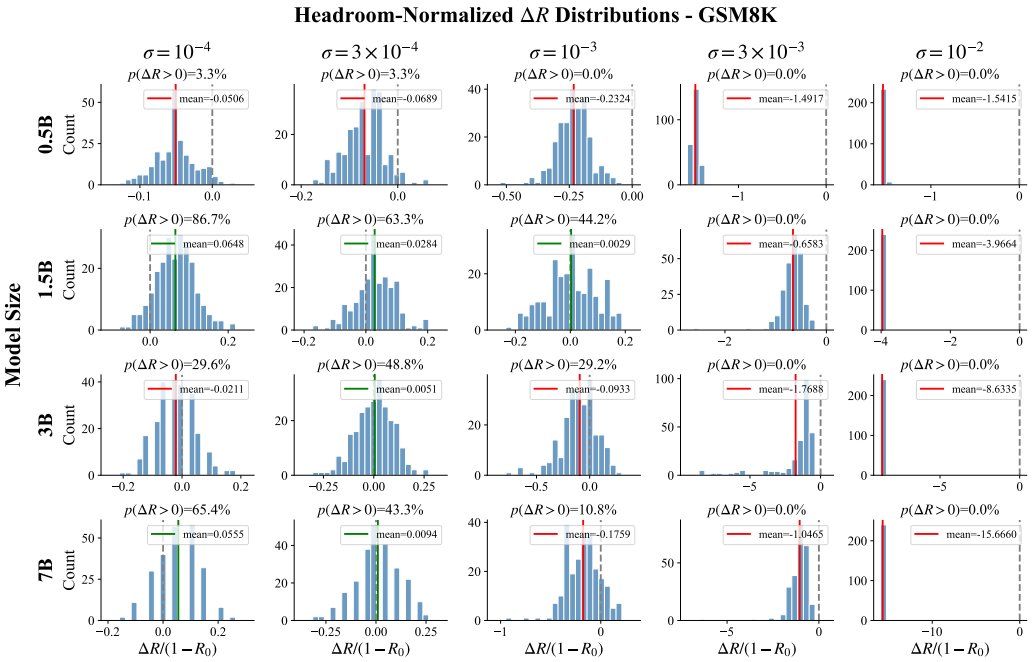

Figure 13: **Headroom-normalized perturbation outcome distributions on GSM8K.** Each panel shows $\Delta R/(1 - R_0)$ for $M = 240$ random perturbations, with the same conventions as Fig. 14. Across model sizes, sufficiently small σ can preserve a nontrivial improving tail, while increasing σ shifts the distribution left and drives $p(\Delta R > 0)$ toward zero, consistent with leaving the local regime where improvements are accessible.

with different baseline accuracies. A value of 0.1 corresponds to capturing 10% of the remaining possible improvement under a binary reward ceiling. Across tasks, a "locality window" in σ is observed. Sufficiently small perturbations can retain a nontrivial improving tail, while larger σ rapidly shifts the distribution negative and drives $p(\Delta R > 0)$ toward zero.

G.9 Auxiliary diagnostics: saturation population and tail quantiles

To complement the best-of-N curves in the main text, improvement accessibility is summarized with two scalar diagnostics. First, the *saturation population* N_{90} is defined as the smallest population size needed to obtain 90% of the expected best-of-N improvement at the largest evaluated population, $N_{\max}$. Second, the 95th percentile $q_{0.95}$ of $\Delta R/(1 - R_0)$ across the candidate pool summarizes the upper tail of the perturbation-induced improvement distribution. Figure 16 shows that both N_{90} and $q_{0.95}$ vary across tasks and perturbation scales, but neither exhibits a systematic increase with model size. These diagnostics provide an additional, compact check that the population required to access improving perturbations does not grow proportionally with the number of model parameters.

G.10 Iterative training runs for Figure 1 (GSM8K rise–then–decay)

The non-monotonic ES and GRPO trajectories in Figure 1 use `Qwen2.5-1.5B-Instruct` as the base model, a frozen training set of 100 GSM8K prompts (with the same prompt format and binary correctness reward as the static probes in Appendix G), `bfloat16` parameters, and a maximum decode length of 512 tokens. Both methods are trained for 1,000 optimization steps under fixed hyperparameters (no learning-rate decay, no early stopping, no KL warm-up), so any non-monotonicity in the trained reward is internal to the optimizer/landscape interaction rather than a hyperparameter schedule.

ES (Algorithm 1, with baseline subtraction). Learning rate $\alpha = 5 \times 10^{-4}$, perturbation scale $\sigma = 10^{-3}$, and population size $N = 30$ are used for the main run; reward differences are baseline-centered against the reward at θ_t before forming the update.

GRPO. Learning rate 5×10^{-6}, group size 20, sampling temperature 0.7, KL coefficient $\beta_{\mathrm{KL}} = 10^{-3}$, PPO clip ratio 0.2, and one PPO epoch per rollout step (one update per batch, 1,000 steps over the 100-prompt frozen training set).

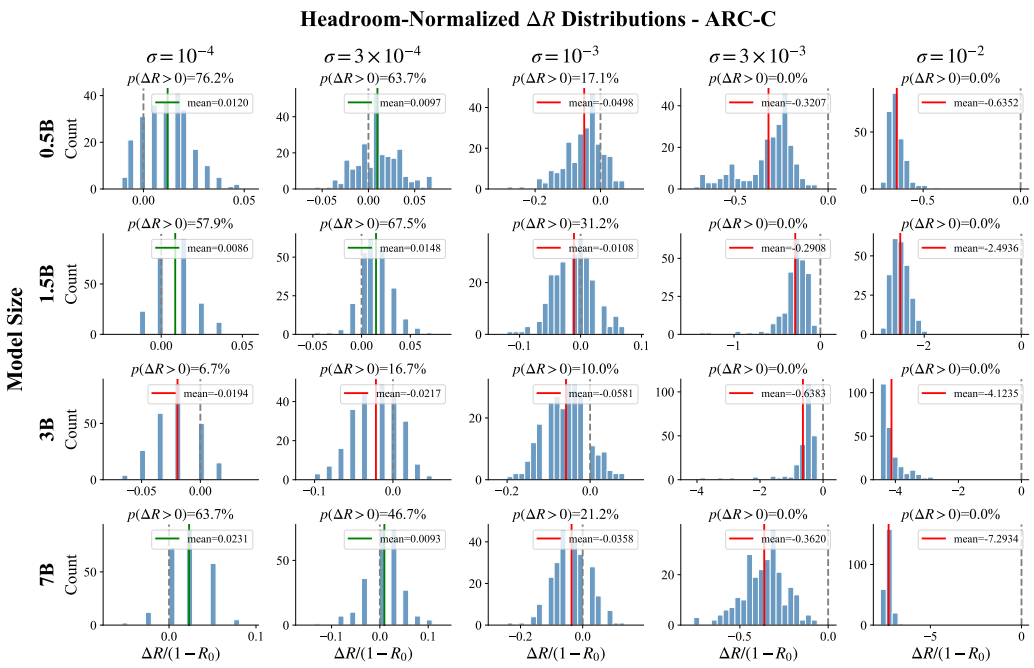

Figure 14: **Headroom-normalized perturbation outcome distributions on ARC-C.** Each panel shows the distribution of $\Delta R/(1 - R_0)$ over $M = 240$ random weight perturbations, where $\Delta R = R_i - R_0$ is the change in accuracy from baseline R_0 and $(1 - R_0)$ is the remaining headroom. Rows correspond to model sizes (0.5B–7B) and columns to perturbation scale σ. The gray dashed vertical line marks $\Delta R = 0$ (no change); the solid vertical line marks the mean (green if positive, red if negative). Panel titles report $p(\Delta R > 0)$, the fraction of perturbations that improve over baseline. Headroom normalization makes the distributions comparable across model sizes and highlights a viable small-σ regime where an improving tail exists versus larger-σ regimes where perturbations predominantly degrade performance.

G.11 Compute Resources

All experiments are evaluation- or rollout-only and require no backpropagation through the base model parameters; compute is therefore characterized primarily by GPU type and parallelism rather than by training-step throughput. The GSM8K iterative ES and GRPO runs in Figure 1 use a single AWS DLAMI node with 8 GPUs running 8 parallel `vLLM` engines, one per candidate evaluation worker; the Countdown ES/GRPO runs in Figure 10 use a single H100 GPU under the SLURM directive `-gres=gpu:h100:1`. The static reward-probe sweeps (Figure 6) require evaluating $M = 240$ candidates on $P = 320$ prompts per (task, model size, σ) cell, which fits comfortably within the same 8-GPU node for models up to 7B in `bfloat16`; per-cell wall-clock was not systematically logged across runs and is therefore reported here only qualitatively.

H Direct Curvature Measurements on Qwen2.5-Instruct

The main-text blessing-of-dimensionality hypothesis is formulated in *reward geometry* terms. It posits that the number of curvature-active directions governing local improvement need not grow proportionally with parameter count. This appendix collects the direct measurements summarized in Table 2: (i) the Hessian spectrum of the ES-smoothed reward J_σ, estimated without autograd; (ii) a differentiable cross-check on the surrogate cross-entropy Hessian; (iii) gradient concentration on the surrogate loss, reported as auxiliary evidence; (iv) robustness checks on the spectral estimates; and (v) a first look at how the spectrum changes along an ES trajectory. All measurements are on GSM8K. Unless stated otherwise, they are taken at the base instruction-tuned checkpoints, which are the starting points of both the static probes and fine-tuning; these are not converged maximizers, and the toy model does not assume initialization at one (a run initialized at the maximizer would show no initial rise).

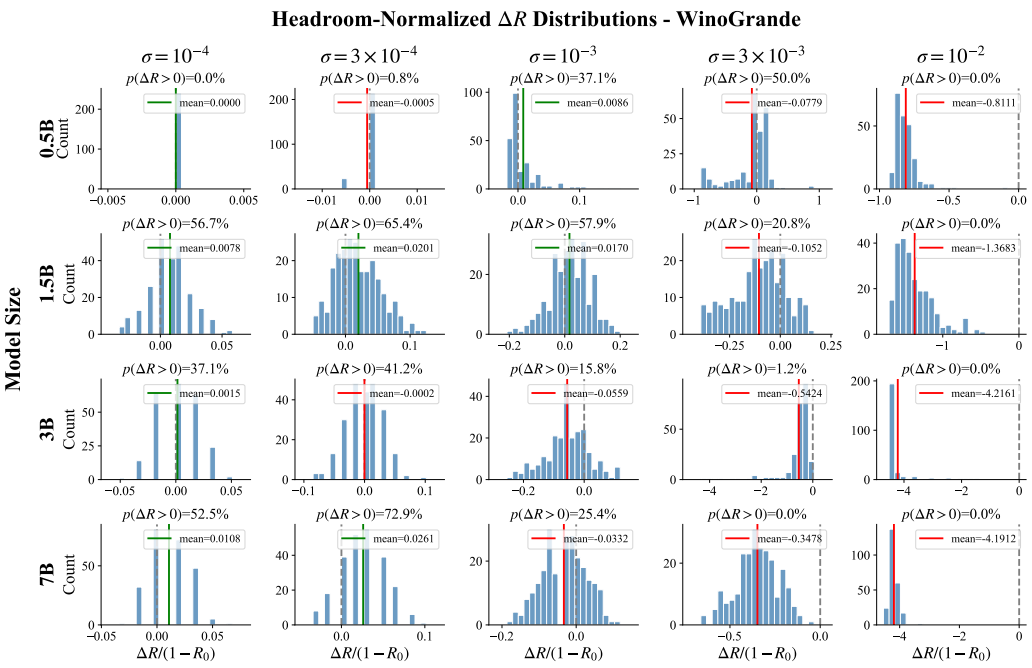

Figure 15: **Headroom-normalized perturbation outcome distributions on WinoGrande.** Each panel shows $\Delta R/(1-R_0)$ for $M = 240$ random perturbations, with the same conventions as Fig. 14. The distributions emphasize that best-of-N improvements arise from the right tail even when the mean is near zero or negative, and that this tail is strongly σ-dependent.

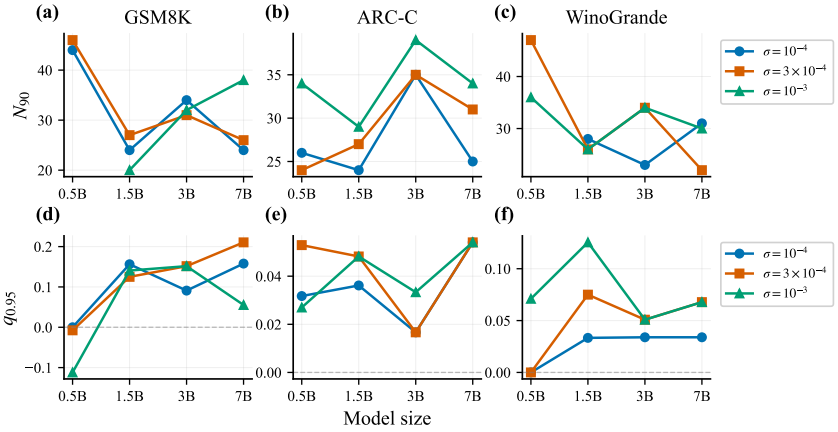

Figure 16: **Auxiliary accessibility diagnostics across model scale. Top row (a–c):** Saturation population N_{90} as a function of model size for GSM8K, ARC-C, and WinoGrande at three perturbation scales σ. Here N_{90} is the smallest population size N such that the expected best-of-N improvement reaches 90% of the value at the largest evaluated population, i.e., $N_{90} \triangleq \min\{N : \Delta^*_N(\sigma) \geq 0.9\,\Delta^*_{N_{\max}}(\sigma)\}$. **Bottom row (d–f):** Upper-tail quantile $q_{0.95}$ of the headroom-normalized improvement distribution $\Delta R/(1-R_0)$ for the same conditions, estimated from the candidate pool of $M = 240$ perturbations. (The dashed gray line marks 0.) While these summaries are task- and σ-dependent and not monotonic in model size, they show no systematic increase in the population required to access the improving tail as model size grows from 0.5B to 7B. This provides an additional check, complementary to Fig. 6, that improvement accessibility does not deteriorate in proportion to ambient parameter count.

H.1 Reward-Hessian spectrum via zeroth-order SLQ

Figure 17 reports Hessian-spectrum-derived summaries of $\nabla^2 J_\sigma(\theta)$ for GSM8K across model sizes. The key observation is that the concentration measures (participation ratio and effective rank, both normalized by D) decrease monotonically with model size, by roughly $4\times$ from 0.5B to 7B, indicating

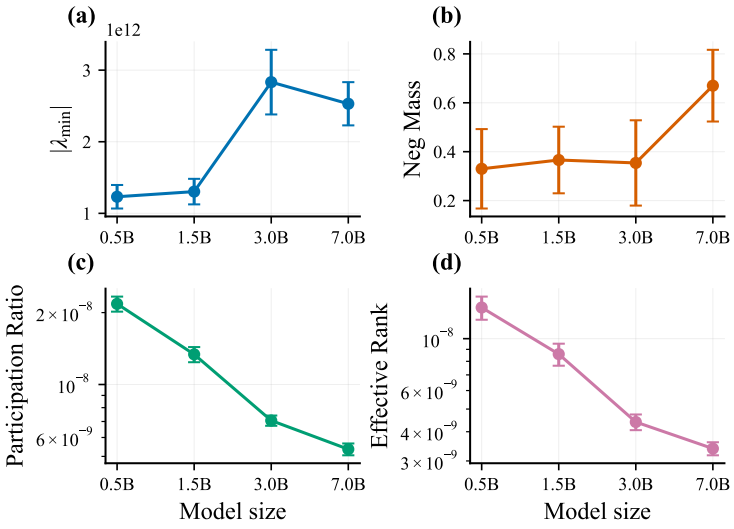

Figure 17: **Reward-based curvature proxy across model scale (GSM8K): concentration of curvature-active structure.** Hessian-spectrum-derived metrics of the *ES-smoothed reward objective* $J_\sigma(\theta) = \mathbb{E}_{\varepsilon\sim\mathcal{N}(0,I)}[J(\theta+\sigma\varepsilon)]$ are estimated using a perturbation-based Hessian–vector product estimator (no autograd) and stochastic Lanczos quadrature (SLQ). Rewards are binary GSM8K training accuracy on a fixed subset of 100 prompts with a fixed prompting and answer-extraction rule. **(a)** magnitude of the most negative eigenvalue $|\lambda_{\min}|$, **(b)** negative spectral mass (total weight on $\lambda < 0$), **(c)** participation ratio, and **(d)** effective rank (both concentration measures; smaller values indicate fewer directions carry most spectral weight). Across Qwen2.5-Instruct model sizes (0.5B–7B), participation ratio and effective rank decrease, indicating that curvature-relevant structure in the reward-defined landscape becomes increasingly *concentrated* rather than expanding in proportion to parameter count. Error bars show mean ± seed variability over 5 random seeds.

that a shrinking *fraction* of directions carries most curvature-relevant structure in the ES-smoothed reward landscape as models scale. At the same time, the magnitude of extreme negative curvature can increase. Together, these findings support a curvature-based blessing-of-dimensionality interpretation in which scaling primarily adds weakly curved directions while the curvature-active structure remains relatively concentrated, consistent with the "bulk + tail" picture used in the main text.

Zeroth-order Hessian–vector products on the ES-smoothed reward. Because the underlying reward J is discrete (binary accuracy) and not amenable to autograd, Hessian–vector products of the Gaussian-smoothed objective J_σ are estimated using only forward evaluations. Starting from the smoothing identity $\nabla^2 J_\sigma(\theta) = \sigma^{-2}\,\mathbb{E}_{\varepsilon\sim\mathcal{N}(0,I)}\big[J(\theta+\sigma\varepsilon)\,(\varepsilon\varepsilon^\top - I)\big]$, this expectation is estimated directly by Monte Carlo rather than via a secondary finite difference along v: for N perturbations $\varepsilon_1,\dots,\varepsilon_N \sim \mathcal{N}(0,I)$, sampled antithetically (in ± pairs), with baseline-subtracted rewards $\tilde{J}_i = J(\theta+\sigma\varepsilon_i) - \bar{J}$,

$$\widehat{H_\sigma v}(\theta;\,\sigma,N) \;=\; \frac{1}{N\sigma^2}\sum_{i=1}^{N}\tilde{J}_i\Big(\varepsilon_i(\varepsilon_i^\top v) - v\Big), \tag{55}$$

which is an unbiased estimator of $\nabla^2 J_\sigma(\theta)\,v$ as $N\to\infty$ (baseline subtraction reduces variance without introducing bias, since $\mathbb{E}[\varepsilon(\varepsilon^\top v) - v] = 0$ under $\varepsilon\sim\mathcal{N}(0,I)$). For the SLQ runs reported in Figure 17, each Lanczos matvec consumes $N = 30$ forward evaluations of J on the fixed prompt subset (matching the ES population size used elsewhere in this paper), at smoothing scale $\sigma = 0.001$, with $m = 10$ Lanczos steps and $s = 100$ Hutchinson probes per model (with full reorthogonalization), aggregated over 5 random seeds ($\{42, 43, 44, 45, 46\}$) on a fixed subset of 100 GSM8K training prompts. Seed variability is reported directly in Figure 17 as the relevant stability check, since the estimator in Eq. (55) has no finite-difference step to separately ablate.

H.2 Un-normalized effective rank and its growth with scale

Figure 17 reports concentration measures normalized by D. The quantity relevant to Proposition 5.1 is the un-normalized effective rank $\hat{k} \triangleq D \times (\text{effective rank}/D)$, the estimate used here of the

curvature-active dimension k. Table 2 collects the 5-seed means, $\hat{k}$, and the two other measurements described below.

Model	D	H_{reward} eff. rank$/D$	$\hat{k}$	H_{loss} eff. rank (+ / −)	params for 90% grad. mass
0.5B	4.9×10^8	$(1.36 \pm 0.34) \times 10^{-8}$	6.7 ± 1.7	251 / 626	5.5%
1.5B	1.5×10^9	$(0.86 \pm 0.21) \times 10^{-8}$	13.3 ± 3.2	19 / 12	5.9%
3B	3.1×10^9	$(0.44 \pm 0.08) \times 10^{-8}$	13.6 ± 2.5	146 / 318	4.2%
7B	7.6×10^9	$(0.34 \pm 0.05) \times 10^{-8}$	25.9 ± 3.8	–	10.1%
14B	1.5×10^{10}	–	–	–	10.8%

Table 2: **Direct curvature measurements on Qwen2.5-Instruct (GSM8K).** $H_{\text{reward}} = \nabla^2 J_\sigma$: effective rank of the ES-smoothed reward Hessian normalized by D, mean $\pm$ s.d. over 5 seeds (Fig. 17); $\hat{k}$ is the same 5-seed statistic un-normalized ($D\times$ the previous column), the estimate used here of the curvature-active dimension. $H_{\text{loss}} = \nabla^2 \mathcal{L}_{\text{CE}}$: effective rank of the positive and negative parts of the autodiff cross-entropy Hessian, single seed (Appendix H.4). Last column: fraction of parameters carrying 90% of the cross-entropy gradient mass (deterministic; auxiliary, Appendix H.5).

A log–log least-squares fit of $\hat{k}$ against D gives $\hat{k} \approx (6.4 \times 10^{-4})\, D^{0.464}$ with $R^2 = 0.947$, i.e., growth close to $\sqrt{D}$. The curvature-active dimension is therefore described as growing *sub-linearly* with D, not as D-free. The monotone decrease of the D-normalized measures in Figure 17 is the flip side of the same fit (roughly $1/\sqrt{D}$) and should not be read as independent evidence of concentration; the quantity that matters for the accessibility argument is the absolute $\hat{k}$. The consequence for the accessibility argument is stated in Section 5. The $D - k$ flat directions add nothing to the curvature penalty, so a curse of dimensionality could reappear only if curvature spreads to more directions as D grows, and the measured growth of k is slow, with $\hat{k}$ in the tens over the tested range. No relation is derived between $N(\delta)$ and k; the best-of-N probes measure the required N directly.

H.3 Robustness checks on the spectral estimates

Two aspects of the SLQ protocol could bias the effective-rank estimates downward: too few Lanczos steps (under-resolution of the spectrum) and too few prompts (a data-dependent operator of rank bounded by the prompt count). Both were checked at the smallest and largest model sizes. *Lanczos depth.* Doubling m from 10 to 20 leaves $\hat{k}$ essentially unchanged, with shifts of $+0.0\%$ at 7B and -2.1% at 0.5B. This matters most at 7B, where $\hat{k} \approx 26$ already exceeds $m = 10$, and indicates no under-resolution. *Prompt subset.* Doubling n_{train} from 100 to 200 moves the participation ratio little (-6.2% at 0.5B, -5.8% at 7B) and the effective rank more (-43% at 0.5B, -21% at 7B). Both *decrease*, i.e., a larger prompt set does not reveal additional curvature-active directions, and the values stay on the same order of magnitude. The effective-rank sensitivity to prompt count is flagged as the main caveat on the absolute values of $\hat{k}$; the qualitative conclusions (tens of directions, sub-linear growth) are unaffected.

H.4 Relating H_{toy}, H_{reward}, and a differentiable surrogate-loss cross-check

This paper uses curvature objects that are related but distinct, made explicit here. $H_{\text{toy}} = -\nabla^2 J(\theta^\star)$ in Appendix A is the local quadratic abstraction that both of the operators below are meant to approximate. $H_{\text{reward}} = \nabla^2 J_\sigma(\theta)$, estimated above via Eq. (55), is the Hessian of the Gaussian-smoothed *reward* objective that ES actually optimizes, obtained purely from forward evaluations since the discrete accuracy reward has no useful autograd path. As an independent cross-check on a fully differentiable surrogate, this appendix additionally estimates $H_{\text{loss}} = \nabla^2 \mathcal{L}_{\text{CE}}(\theta)$, the Hessian of the token-level cross-entropy loss on GSM8K, via autodiff Hessian–vector products and SLQ (100 probes, 30 Lanczos steps, single seed) across Qwen2.5-Instruct 0.5B, 1.5B, and 3B. In every case the effective rank of H_{loss} (positive part: 250.9, 19.0, 146.1; negative part: 625.7, 12.3, 318.0, respectively) is roughly six to eight orders of magnitude smaller than the ambient parameter count (4.9×10^8, 1.5×10^9, 3.1×10^9), corroborating the same low-dimensional-curvature picture obtained from H_{reward} via an entirely different, differentiable, loss-based (rather than reward-based) estimator. This cross-check used a single seed and was not extended to 7B, so its scale-trend is noisier than the 5-seed H_{reward} result above; it is reported as evidence that low-dimensional curvature is not an artifact of the reward-smoothing estimator, not as an independent replication of the monotonic trend in Figure 17.

The two operators disagree in absolute value by one to two orders of magnitude (e.g., $\hat{k} \approx 14$ for H_{reward} versus 146/318 for H_{loss} at 3B). There is no first-principles prediction for this gap, and

the two are not expected to match. H_{reward} is the curvature of a smoothed binary reward estimated zeroth-order, whereas H_{loss} is the curvature of a dense per-token loss estimated by exact autodiff. Participation ratio and effective rank are scale-invariant, so the σ^{-2} prefactor in Eq. (55) does not explain the gap either. Both measurements support the qualitative finding that curvature-active structure occupies a vanishing fraction of the ambient space.

H.5 Gradient concentration on the surrogate loss (auxiliary)

As a third, first-order line of evidence that extends past the 7B ceiling of the spectral measurements, the full cross-entropy gradient g is computed for Qwen2.5-Instruct 0.5B–14B on the first 10 GSM8K training examples (a single deterministic computation per model, so there is no seed axis), and its concentration is summarized by two statistics: $d_g(\infty) \triangleq \|g\|_2^2/\|g\|_\infty^2$, an effective dimension that equals the number of nonzero entries when they are equal in magnitude and is small when a few entries dominate; and the smallest fraction of parameters whose squared entries carry 90% of $\|g\|_2^2$.

Model	D	$d_g(\infty)$	$d_g(\infty)/D$	params for 90% of $\|g\|_2^2$
0.5B	494,032,768	473.1	9.6×10^{-7}	5.52%
1.5B	1,543,714,304	70.0	4.5×10^{-8}	5.87%
3B	3,085,938,688	268.5	8.7×10^{-8}	4.17%
7B	7,615,616,512	1029.3	1.4×10^{-7}	10.07%
14B	14,770,033,664	1069.9	7.2×10^{-8}	10.77%

The fraction of parameters carrying 90% of the gradient mass stays within a narrow 4–11% band across a $30\times$ range of D, and the relative concentration $d_g(\infty)/D$ drops by about $13\times$ from 0.5B to 14B even though $d_g(\infty)$ itself grows. The scale trend is weak ($R^2 = 0.30$ for a log–log fit of $d_g(\infty)$ against D, versus 0.947 for $\hat{k}$); since the computation is deterministic this is not noise but genuine non-monotonicity across model sizes. These statistics are reported as auxiliary evidence only. The accessibility argument of Section 5 needs *many* reachable improving directions, whereas gradient concentration measures how much of the gradient lives in the single sharpest direction; the two are related but not the same quantity.

H.6 Curvature along an ES trajectory

The measurements above are taken at base checkpoints. To see whether the local stable/unstable mix changes over training, as the rise–then–decay mechanism would suggest, the zeroth-order SLQ measurement is repeated (same σ, N, m, s, and prompt subset as Figure 17) at two checkpoints of the 1.5B GSM8K ES run in Figure 1: near the evaluation-reward peak (iteration 200, 74.3% accuracy) and well into the decay (iteration 900, 65.5% accuracy). The negative spectral mass of $\nabla^2 J_\sigma$, i.e., the fraction of spectral weight on $\lambda < 0$ in the reward-maximization convention, moves from 23% at the untrained base checkpoint to 4% near the peak and back to 22% during decay. In the descent picture of Section 4, this is the mix of stabilizing and destabilizing directions contracting as the iterate approaches the reward peak and reopening past it. This is a single trajectory at a single model size and does not measure eigenvector rotation or basin transitions; it is reported as a first consistency check on the within-basin picture, not as a validation of it.