SPECTRA: SPECTRAL STABLE LEARNING RATE PREDICTION FOR LOW-RANK ADAPTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

We propose **SPECTRA**, a gradient-spectral framework for predicting stable learning rates for LoRA fine-tuning. Although LoRA substantially reduces the cost of full-parameter SFT, its performance remains highly sensitive to learning-rate selection. Existing studies offer valuable scaling laws, heuristics, and empirical insights, but they do not directly predict an appropriate learning rate for a specific task and LoRA configuration, leaving practitioners reliant on costly hyperparameter sweeps. **Methodologically**, SPECTRA uses a calibrated log-linear predictor to map the initialization-time gradient spectral decay β and LoRA rank to a stable, near-optimal learning rate. **Theoretically**, we derive the local stability threshold for rank-constrained optimization, identify the most fragile module as the shared-rate bottleneck, and establish Adam's gradient-scale invariance in the idealized limit. Under a local monotonic relationship between spectral decay and curvature, faster spectral decay reduces the stability threshold. **Experimentally**, SPECTRA achieves an **88.5%** success rate, outperforming all sweep-free baselines by over **9.7** points and recovering **89%** of the coverage of an exhaustive per-dataset grid search at **4%** of its time cost.

1 INTRODUCTION

We propose **SPECTRA**, a gradient-spectral framework for predicting stable learning rates for LoRA fine-tuning. Although LoRA substantially reduces the cost of full-parameter supervised fine-tuning (SFT), its performance remains highly sensitive to learning-rate selection; existing studies provide useful heuristics, scaling laws, and empirical observations, but still rely on expensive learning-rate sweeps rather than directly predicting an appropriate value. SPECTRA uses teacher-forced gradients from a small calibration set to characterize the gradient spectrum and predict near-optimal learning rates across datasets and LoRA configurations without per-setting hyperparameter search.

With the rapid development of foundation models (Agarwal et al., 2025; Yang et al., 2025; Gao et al., 2025), task-specific post-training (Guo et al., 2025; Li et al., 2025b; Agarwal et al., 2024; Dai et al., 2024) has become increasingly important but also computationally expensive, as full-parameter updates require substantial memory, compute, and storage. Parameter-efficient fine-tuning (Ding et al., 2023) alleviates these costs by updating only a small subset of parameters, among which low-rank adaptation (LoRA) (Hu et al., 2021) has emerged as one of the most widely adopted approaches because of its simplicity, scalability, and strong empirical performance. However, LoRA often remains substantially weaker than full-parameter SFT, with its performance highly sensitive to learning-rate selection. Prior studies have characterized LoRA optimization through scaling laws, optimization dynamics, and empirical learning-rate trends (Lee et al., 2026; Chen et al., 2026; Hayou et al., 2024), yet these findings provide primarily qualitative guidance rather than a principled rule that directly predicts an appropriate learning rate for a given training setting. As a result, practitioners still rely on costly learning-rate sweeps across models and datasets to identify stable, high-performing configurations (Table 1). This repeated search substantially increases tuning cost and weakens the computational efficiency that motivates parameter-efficient fine-tuning.

To address the costly learning-rate search required for LoRA fine-tuning, we propose **SPECTRA**, a gradient-spectral framework that predicts stable, near-optimal learning rates using a small calibration set. SPECTRA characterizes the concentration of task-relevant gradient energy in the leading singular directions using the spectral-decay feature β. This spectral structure provides a strong predictive signal for learning-rate selection, replacing repeated per-configuration sweeps with lightweight model-level calibration.

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.

Table 1: **Learning-rate selection in existing LoRA methods.** Despite advances in parameter-efficient adaptation, learning rates are still commonly selected through explicit grid search or manually specified heuristics. No listed method directly predicts a near-optimal learning rate from the model and data before fine-tuning (Cui et al., 2026; Li et al., 2025a; Paischer et al., 2024; Zhang et al., 2023a).

Method	LR selection	# LR candidates	Joint search	Reported search burden
AdaLoRA	Grid search	8	No	8 LR settings per task/method
AIRA	Grid search	3	No	3 LR settings $\times$ 3 seeds
IGU-LoRA	Grid search	4	No	Best LR selected from 4 candidates
EVA (GLUE)	Grid search	3	LR $\times$ rank	3 LRs $\times$ 4 ranks $=$ 12 configurations/task/method
EVA (VTAB-1K)	Grid search	5	LR $\times$ rank	5 LRs $\times$ 4 ranks $\times$ 3 seeds $\times$ 19 datasets $=$ **1,140** tuning runs
SPECTRA (Ours)	**Spectral prediction**	-	**No**	**One-time sweep per model; reused across datasets/configurations**

Theoretically, we connect the spectral predictor to the stability of constrained LoRA optimization, showing that the admissible learning rate is governed by the most fragile module and that Adam removes dependence on global gradient magnitude. We further show that the convergence-optimal learning rate lies within a factor of two of the stability threshold, providing a principled basis for using spectral information to directly localize high-performing learning rates.

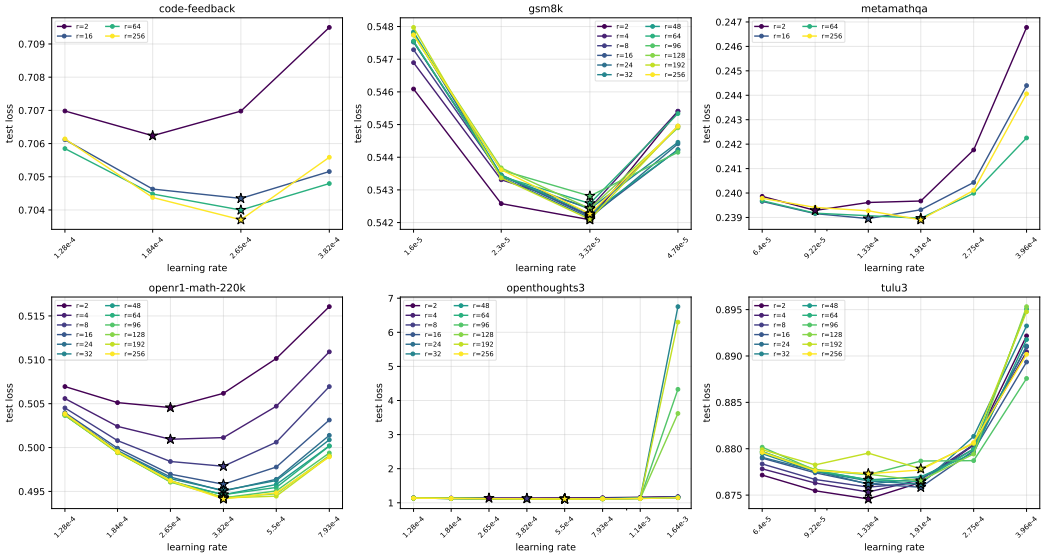

Figure 1: **LoRA performance is highly sensitive to learning-rate selection.** Test loss of Qwen3-1.7B-Base across six fine-tuning datasets and multiple LoRA ranks as the learning rate varies. Stars denote the empirically optimal learning rate for each rank. The optimal learning rate varies substantially across datasets and configurations, while overly large learning rates can sharply degrade performance or cause instability, highlighting the need for costly per-setting learning-rate sweeps.

Contributions. We present **SPECTRA**, a gradient-spectral framework for predicting stable, near-optimal LoRA learning rates for a given dataset and rank. Our main contributions are:

- We formulate LoRA learning-rate selection as an optimization problem coupled to module capacity: rank determines the task-relevant update subspace, while the learning rate governs whether optimization within that subspace is stable and efficient.
- Methodologically, SPECTRA extracts teacher-forced gradients from a small calibration set and characterizes their singular-value spectra using the spectral-decay feature β. Based on this spectral signal, SPECTRA predicts stable, near-optimal LoRA learning rates after lightweight model-level calibration, eliminating repeated per-configuration learning-rate sweeps.
- Theoretically, we derive the stability threshold of constrained LoRA optimization and show that a shared learning rate is governed by the most fragile module. We further establish Adam's invariance to global gradient scaling and connect residual spectral-energy decay to optimization stability, providing the theoretical basis for our spectral learning-rate predictor.
- Experimentally, SPECTRA achieves an **88.5%** success rate, outperforming all sweep-free baselines by over **9.7** points and recovering **89%** of the coverage of an exhaustive per-dataset grid search at **4%** of its cost (**4 vs. 90 hours**).

Related Work. Existing LoRA optimization methods primarily modify optimizer dynamics or derive configuration-level scaling laws, but do not predict an absolute, task-specific learning rate from the data. In contrast, SPECTRA leverages the initialization-time gradient spectrum with lightweight model-level calibration to directly predict stable, near-optimal learning rates across datasets and ranks; detailed comparisons are provided in Appendix A.

2 Sensitivity of LoRA to Learning-Rate Selection

In this section, we quantify the sensitivity of LoRA fine-tuning to learning-rate selection. Across datasets and LoRA configurations, test loss varies non-monotonically with learning rate, with each setting exhibiting a narrow high-performing regime and a distinct optimum. As shown in Figure 1, Qwen3-1.7B-Base exhibits substantial variation in the optimal learning rate across tasks and ranks, spanning more than an order of magnitude. Modestly exceeding the optimum can sharply increase test loss, particularly on OpenThoughts3; moreover, under poorly chosen learning rates, higher-rank LoRA can perform even worse than lower-rank configurations, despite its greater representational capacity. These results highlight the strong interaction between rank and learning rate and show that learning rates transfer poorly across settings, making repeated sweeps a substantial source of tuning cost. Additional analyses are provided in Appendix G.

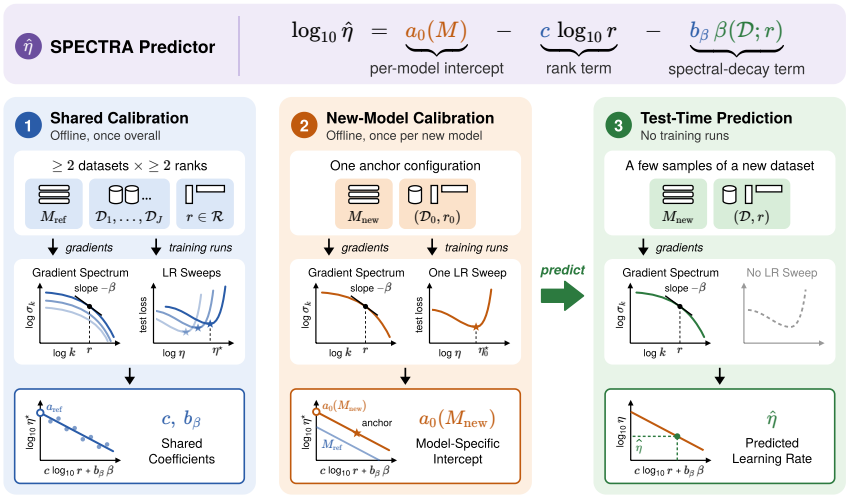

Figure 2: **Overview of the** SPECTRA **pipeline.** SPECTRA first learns shared coefficients from calibration sweeps on a reference model, then estimates a model-specific intercept using a single anchor configuration. For a target rank, it combines these calibrated parameters with initialization-time gradient statistics to predict a stable learning rate without an exhaustive sweep.

3 SPECTRA

Notation and Problem Setup. Let $\ell \in \{1, \ldots, L\}$ index the LoRA target modules, with frozen pre-trained weight $W_0^{(\ell)} \in \mathbb{R}^{m_\ell \times n_\ell}$. LoRA parameterizes the trainable update as $\Delta W^{(\ell)} = \frac{\alpha}{r_\ell} B^{(\ell)} A^{(\ell)}$, where $B^{(\ell)} \in \mathbb{R}^{m_\ell \times r_\ell}, A^{(\ell)} \in \mathbb{R}^{r_\ell \times n_\ell}$, such that $\operatorname{rank}\left(\Delta W^{(\ell)}\right) \le r_\ell$. We consider the teacher-forced SFT objective

$$\mathcal{L}_{\text{SFT}}(\theta) = \frac{1}{|\mathcal{D}|} \sum_{(x,y)\in\mathcal{D}} \sum_t -\log p_\theta\left(y_t \mid x, y_{<t}\right).$$

For an example x from a small calibration set $\mathcal{D}_0$, define the initialization-time teacher-forced gradient

$$G^{(\ell)}(x) := \frac{\partial \mathcal{L}_{\text{SFT}}(x;\theta_0)}{\partial W^{(\ell)}}.$$

Let $\sigma_1^{(\ell)}(x) \ge \cdots \ge \sigma_{N_\ell}^{(\ell)}(x) \ge 0, N_\ell := \min(m_\ell, n_\ell)$, denote the singular values of $G^{(\ell)}(x)$.

To characterize the spectral shape, we directly measure the decay of the singular values with spectral index. We define the local spectral-decay feature as the negative log–log slope

$$\beta_{\ell,\text{sample}}(x;k) := -\frac{\Delta \log \sigma_k^{(\ell)}(x)}{\Delta \log k}.$$

Equivalently, if the spectrum locally follows $\sigma_k^{(\ell)}(x) \propto k^{-\beta}$, then β corresponds to the singular-value decay exponent. A larger $\beta_{\ell,\text{sample}}(x;k)$ therefore indicates a more rapidly decaying gradient spectrum. We average the feature over calibration examples:

$$\beta_\ell(\mathcal{D}_0;k) := \frac{1}{|\mathcal{D}_0|}\sum_{x\in\mathcal{D}_0}\beta_{\ell,\text{sample}}(x;k),$$

and aggregate across the L LoRA target modules:

$$\beta(\mathcal{D}_0;k) := \frac{1}{L}\sum_{\ell=1}^{L}\beta_\ell(\mathcal{D}_0;k).$$

In practice, SPECTRA computes the singular-value spectrum over the full spectral evaluation range $1 \le k \le K_{\max}$, where $K_{\max} = 256$ in our experiments, and estimates the log–log slope over this range. The resulting slope is evaluated at the target LoRA rank r, yielding the spectral-decay feature $\beta(\mathcal{D}_0;r)$. Additional notation is provided in Appendix B.

Algorithm 1 SPECTRA: Calibration and Learning-Rate Prediction

Require: Reference model M_{ref}; calibration datasets $\{\mathcal{D}_j\}_{j=1}^{J}$; candidate ranks $\mathcal{R}$
Ensure: Calibrated predictor $\{a_0(M), c, b_\beta\}$ and predicted learning rate $\widehat{\eta}$

for each dataset $\mathcal{D}_j$ and rank $r \in \mathcal{R}$ **do**
  Compute the initialization-time gradient singular-value spectra over the full spectral evaluation range
  Compute the spectral-decay feature $\beta(\mathcal{D}_j;r)$
  Obtain the optimal learning rate $\eta_j^\star(r)$ from the calibration sweep
end for
Fit c and b_β by linear regression:

$$\log_{10}\eta_j^\star(r) = a_{\text{ref}} - c\log_{10} r - b_\beta\beta(\mathcal{D}_j;r)$$

Model-specific calibration:
For a target model M, choose one anchor configuration $(\mathcal{D}_0, r_0)$
Compute $\beta(\mathcal{D}_0;r_0)$ and obtain the anchor learning rate $\eta_0^\star$

$$a_0(M) \leftarrow \log_{10}\eta_0^\star + c\log_{10} r_0 + b_\beta\beta(\mathcal{D}_0;r_0)$$

Prediction:
For a target rank r and dataset $\mathcal{D}$, compute $\beta(\mathcal{D};r)$

$$\log_{10}\widehat{\eta} \leftarrow a_0(M) - c\log_{10} r - b_\beta\beta(\mathcal{D};r)$$

return $\widehat{\eta}$

SPECTRA. To predict a stable learning rate, SPECTRA adopts a two-stage calibration procedure, as summarized in Algorithm 1. First, using a reference model evaluated across multiple datasets and LoRA ranks, we fit the shared coefficients c and b_β, which capture the explicit rank dependence and gradient spectral decay, respectively. For a new model M, we then calibrate the model-specific intercept $a_0(M)$ using a single anchor configuration with a known stable learning rate.

Given a target rank r and dataset $\mathcal{D}$, SPECTRA computes the initialization-time singular-value spectrum and extracts its log–log decay slope at r. The predicted learning rate is

$$\log_{10}\widehat{\eta} = a_0(M) - c\log_{10} r - b_\beta\beta(\mathcal{D};r).$$

4 Theoretical Analysis

In this section, we analyze SPECTRA through the local teacher-forced SFT dynamics of a spectral LoRA surrogate. The analysis characterizes the stability structure underlying the learning-rate predictor rather than the exact nonlinear dynamics of the factors (A, B), and proceeds in three steps. First, we derive the stability boundary of Adam-preconditioned SFT dynamics restricted to the surrogate subspace, which we call the weight-space boundary $\eta_W^\star$, and establish its dependence

on rank, cross-module bottlenecks, and gradient scale. Second, we transfer $\eta_W^\star$ to the learning rate that the optimizer applies to (A, B) through a single scaling exponent. Third, we linearize the resulting boundary in $(\log_{10} r, \beta)$ to obtain the predictor used by SPECTRA. Formal assumptions and complete proofs are provided in Appendices C and D.

Surrogate and Preconditioned Dynamics. For each target module, let $\mathcal{S}_r := \mathrm{span}\{u_i v_i^\top\}_{i \le r}$ be spanned by the leading r singular directions of the initialization-time gradient (Assumption D.3). $\mathcal{S}_r$ is a gradient-aligned surrogate for the rank-r update space rather than the tangent space of standard LoRA at initialization, and all statements below refer to this surrogate. We treat Adam in the quasi-static regime of Assumption D.6: around θ_0, $\mathcal{L}_{\mathrm{SFT}}$ is represented by its quadratic model with Hessian H, and the Adam preconditioner is frozen at $D := \mathrm{diag}(\sqrt{\hat{v}}) + \epsilon I$, computed from the unprojected gradient and hence independent of r. In whitened coordinates $\tilde{e} = D^{1/2} e$, let $\widetilde{\mathcal{S}}_r := D^{1/2}\mathcal{S}_r$ with orthogonal projector $\widetilde{P}_r$. We define the preconditioned restricted Hessian and the restricted update map

$$\widetilde{H}_r := \widetilde{P}_r\, D^{-1/2} H D^{-1/2}\, \widetilde{P}_r\big|_{\widetilde{\mathcal{S}}_r}, \qquad T_r(\theta) := \theta - \eta\, D^{-1/2} \widetilde{P}_r D^{-1/2}\, \nabla \mathcal{L}_{\mathrm{SFT}}(\theta),$$

whose increments lie in $\mathcal{S}_r$. For $D = I$, T_r reduces to projected gradient descent and $\widetilde{H}_r$ to $H_r := P_r H P_r|_{\mathcal{S}_r}$.

Lemma 4.1 (Local restricted perturbation dynamics)**.** Under Assumption D.2 and for fixed $D \succ 0$, for $e \in \mathcal{S}_r$ with $\tilde{e} = D^{1/2} e$ sufficiently small,

$$D^{1/2}\big(T_r(\theta_0 + e) - T_r(\theta_0)\big) = \big(I - \eta \widetilde{H}_r\big)\tilde{e} + \mathcal{O}\big(\|\tilde{e}\|^2\big).$$

Under Assumption D.6, the whitened difference $\tilde{e}_t$ between any two trajectories of T_r started in $\theta_0 + \mathcal{S}_r$ evolves exactly as $\tilde{e}_{t+1} = (I - \eta \widetilde{H}_r)\tilde{e}_t$, without requiring θ_0 to be stationary. Consequently, for every eigenpair (μ_j, w_j) of $\widetilde{H}_r$, $w_j^\top \tilde{e}_t = (1 - \eta\mu_j)^t\, w_j^\top \tilde{e}_0$.

Proof. See Section C.1 for a detailed proof. □

Lemma 4.2 (Scalar convergence criterion)**.** The recursion $x_{t+1} = \rho x_t$ satisfies $x_t \to 0$ for every x_0 if and only if $|\rho| < 1$. For $\rho = 1 - \eta\lambda$ with $\lambda > 0$, this is equivalent to $0 < \eta < 2/\lambda$.

Proof. See Section C.2 for a detailed proof. □

Proposition 4.1 (Weight-space stability boundary)**.** Under Assumptions D.2 and D.6, the restricted dynamics are contractive in every eigendirection of $\widetilde{H}_r$ if and only if

$$0 < \eta < \eta_W^\star(r), \qquad \eta_W^\star(r) := \frac{2}{\lambda_{\max}(\widetilde{H}_r)}.$$

Proof. See Section C.3 for a detailed proof. □

Proposition 4.2 (Rank monotonicity in weight space)**.** Under Assumptions D.2, D.3 and D.6,

$$r_1 \le r_2 \implies \eta_W^\star(r_1) \ge \eta_W^\star(r_2).$$

Proof. See Section C.4 for a detailed proof. □

From Weight Space to the Optimizer Learning Rate. The optimizer updates (A, B), which enter the model as $s(r)BA$ with $s(r) = \alpha(r)/r = s_0 r^{-\gamma}$; constant α, rsLoRA, and $\alpha \propto r$ correspond to $\gamma = 1, \frac{1}{2}, 0$, respectively. Because the objective depends on (A, B) only through $s(r)BA$ and Adam is invariant to gradient scale, a change of s can be compensated by rescaling the factors and the learning rate. Lemma E.1 shows that this compensation is exact in two limiting regimes: the learning rate scales as s^{-1} when the update is carried by B alone, as immediately after the initialization $B_0 = 0$, and as $s^{-1/2}$ when both factors are updated in balance. We interpolate between the two regimes with a single effective exponent.

Assumption 4.1 (Scaling transfer)**.** There exist $p \in [\frac{1}{2}, 1]$ and $\kappa(M) > 0$, independent of r and β, such that the optimizer-level stability boundary satisfies $\eta_{\mathrm{spec}}^\star(r, \beta) = \kappa(M)\, s(r)^{-p}\, \eta_W^\star(r, \beta)$.

Corollary 4.1 (Decomposition and bound of the rank coefficient). Under Assumption 4.1, locally in $\log_{10} r$ and at fixed β,

$$c = c_W + q_\alpha, \qquad c_W := -\frac{\partial \log_{10} \eta_W^\star}{\partial \log_{10} r} \geq 0, \qquad q_\alpha := -p\gamma \leq 0.$$

Hence $c \geq -p\gamma \geq -\gamma$. In the balanced regime ($p = \frac{1}{2}$), $q_\alpha = -\frac{1}{2}, -\frac{1}{4}, 0$ for standard LoRA, rsLoRA, and constant scaling; the sign of c is fixed by theory only for $\gamma = 0$. For two conventions evaluated on the same $(M, \mathcal{D}, r)$, $\Delta c := c(\gamma) - c(1) = p(1-\gamma)$, which identifies p without reference to β (Appendix L).

Proof. See Section C.5 for a detailed proof. □

Bottleneck Stability Across LoRA Modules. Most LoRA SFT implementations use a shared optimizer learning rate across target modules. The resulting local stability is therefore constrained by the module with the largest preconditioned restricted curvature.

Proposition 4.3 (Bottleneck stability across LoRA modules). Suppose L LoRA modules share a common learning rate η, and let $\eta_{W,\ell}^\star := 2/\lambda_{\max}\big(\widetilde{H}_{r_\ell}^{(\ell)}\big)$. Under Assumption D.4, the joint restricted dynamics are stable if and only if

$$0 < \eta < \min_\ell \eta_{W,\ell}^\star = \frac{2}{\max_\ell \lambda_{\max}\big(\widetilde{H}_{r_\ell}^{(\ell)}\big)}.$$

Proof. See Section C.6 for a detailed proof. □

Proposition 4.4 (Optimal local contraction rate). Let $0 < \mu_{\min} \leq \mu_j \leq \mu_{\max}$ denote the eigenvalues of $\widetilde{H}_r$, let $\rho(\eta) := \max_j |1 - \eta\mu_j|$, and let $\omega_r := \mu_{\min}/\mu_{\max} \in (0, 1]$. Without momentum, ρ is minimized at $\eta_{\text{opt}}^{\text{lin}} = 2/(\mu_{\max} + \mu_{\min})$, and

$$\log_{10} \eta_W^\star(r) - \log_{10} \eta_{\text{opt}}^{\text{lin}} = \log_{10}(1 + \omega_r) \in (0, \log_{10} 2].$$

Consequently, replacing this offset by the constant $\frac{1}{2}\log_{10} 2$ incurs an error of at most $\frac{1}{2}\log_{10} 2 \approx 0.15$ dex, uniformly in $(\mathcal{D}, r)$.

Proof. See Section C.7 for a detailed proof. □

Adam Eliminates Global Gradient-Scale Dependence. The gradient spectrum encodes both global scale and directional structure. We show that, under idealized Adam, neither the boundary targeted by SPECTRA nor its spectral input depends on global scale.

Proposition 4.5 (Global scale invariance). Consider Adam updates

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t, \quad v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^{\odot 2}, \quad \Delta\theta_t = -\eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}.$$

In the limit $\epsilon \to 0$: (i) rescaling the gradient sequence by $g_t \mapsto k g_t$, $k > 0$, leaves every Adam update unchanged; and (ii) rescaling the objective by $\mathcal{L}_{\text{SFT}} \mapsto k\,\mathcal{L}_{\text{SFT}}$ leaves $\widetilde{H}_r$, $\eta_W^\star(r)$, and $\beta_\ell(\mathcal{D}; r)$ unchanged.

Proof. See Section C.8 for a detailed proof. □

Spectral Decay and the Learning-Rate Predictor. We characterize the local shape of the initialization-time gradient spectrum by the spectral-decay exponent

$$\beta_\ell(\mathcal{D}; r) \approx -\frac{\Delta \log \sigma_i^{(\ell)}}{\Delta \log i},$$

estimated over a local spectral window around rank r and averaged over the calibration set (Section E.1). Assumption D.5 posits that, at fixed rank, faster spectral decay increases the preconditioned restricted curvature. Proposition E.1 shows that this holds when the restricted curvature follows the trace-normalized empirical Fisher and the spectrum follows a power law. In that case, $\lambda_{\max}(\widetilde{H}_r)$ is

proportional to the fraction of gradient energy captured by the leading r directions, $1 - R(r)/R(0)$ with $R(k) := \sum_{i>k} \sigma_i^2$, and β is its one-parameter summary.

Proposition 4.6 (Local log-linear learning-rate predictor)**.** Under Assumptions 4.1 to D.3, D.5 and D.6, let (r_0, β_0) be a reference point and $\Delta = (\log_{10} r - \log_{10} r_0,\ \beta - \beta_0)$. The optimizer-level stability boundary admits the first-order approximation

$$\log_{10} \eta^\star_{\mathrm{spec}}(r, \beta) = a_0(M) - c \log_{10} r - b_\beta\, \beta(\mathcal{D}; r) + \mathcal{O}\big(\|\Delta\|^2\big),$$

where $c = c_W + q_\alpha \geq -p\gamma$ and $b_\beta > 0$.

Proof. See Section C.9 for a detailed proof. □

Corollary 4.2 (Global learning-rate predictor)**.** Let all LoRA modules share a common rank $r_\ell \equiv r$, and define the module-averaged decay exponent $\bar{\beta}(\mathcal{D}; r) := \frac{1}{L} \sum_{\ell=1}^{L} \beta_\ell(\mathcal{D}; r)$. Under Propositions 4.3 and 4.6 and Assumption D.7, the shared optimizer-level stability boundary $\widehat{\eta}^\star$ satisfies

$$\log_{10} \widehat{\eta}^\star = a_0(M) - c \log_{10} r - b_\beta\, \bar{\beta}(\mathcal{D}; r) + \mathcal{O}\big(\varepsilon_\Delta + \|\Delta\|^2\big),$$

where ε_Δ bounds the $(\mathcal{D}, r)$-dependence of the gap between the bottleneck module and the module average. By Proposition 4.4, the same form targets the contraction-optimal rate with an additional error of at most $\frac{1}{2} \log_{10} 2 \approx 0.15$ dex.

Proof. See Section C.10 for a detailed proof. □

Additional remarks and theoretical results are provided in Appendix E.

5 Experimental Studies

We conduct a series of experiments to evaluate SPECTRA in learning-rate prediction across diverse model families and datasets. Each evaluation is conducted three times with different random seeds, and we report the average and standard deviation for each metric. We further conduct supplementary experiments to quantify the computational overhead of SPECTRA and examine the influence of different LoRA scaling factors α on model performance.

Models. In our experiments, we use Llama 2 (Touvron et al., 2023), Llama 3.1 (Grattafiori et al., 2024), Qwen3 (Yang et al., 2025), and Gemma 3 (Team et al., 2025) as backbone models for LoRA fine-tuning. Specifically, we evaluate Llama 2 7B, Llama 3.1 8B, Qwen3 1.7B, Qwen3 8B, and Gemma 3 1B, and fine-tune each base model with SFT-LoRA using the learning rate predicted by SPECTRA. For consistency with each model family, we adopt the chat template provided by its corresponding instruction-tuned checkpoint as the training template.

Datasets. Following the setup in Lee et al. (2026); Schulman & Thinking Machines Lab (2025), we use OpenThought (Guha et al., 2026), OpenR1 (Hugging Face, 2025), Code-feedback (Zheng et al., 2024), metamathqa (Yu et al., 2024), Tulu3 (Lambert et al., 2024) as the training corpus. All datasets are designed for instruction-following tasks, including mathematical question answering and code generation.

Metrics. We use test loss as the primary performance metric, evaluated with the same objective as training on the held-out test set. To quantify learning-rate prediction accuracy, we additionally report the prediction success rate, defined as the fraction of configurations for which the predicted learning rate lies within $[1/2.4, 2.4]\times$ the empirically optimal learning rate. This tolerance is centered on the commonly used $2\times$ multiplicative criterion and expanded to $2.4\times$ to account for the finite resolution and uncertainty of the learning-rate sweep. Details of the coarse-to-fine sweep and the resulting error tolerance are provided in the experimental **setup** in Section 5.

Baselines. We compare SPECTRA against three learning-rate selection strategies for LoRA SFT: (i) **default learning rate**, where we use 1×10^{-4}, 2×10^{-4}, 4×10^{-4} following the commonly adopted LoRA setting recommended by Schulman & Thinking Machines Lab (2025); (ii) **Grid search**, which selects one learning rate for each model–dataset pair and applies it uniformly across all LoRA ranks. We evaluate 24 predefined learning rates from 2.0×10^{-6} to 1.1×10^{-3} and select the value yielding the lowest validation loss. See Section H.3 for details. (iii) **TM formula** (Schulman & Thinking Machines Lab, 2025), which predicts the learning rate from model hidden

Table 2: **Learning-rate selection success rate across model families.** We compare SPECTRA with fixed learning rates, the TM formula, and grid search across five models and five datasets. Values report Hit@2.4×, i.e., the percentage of configurations for which the selected learning rate lies within a factor of 2.4 of the empirical optimum. Variances are omitted as they are consistently $\leq 0.2\%$. Best results are shown in **bold**, and second-best results are underlined. Overall, SPECTRA achieves the second-best average success rate while requiring substantially less search time than grid search. Across five models and five datasets, SPECTRA achieves an **88.5%** success rate while reducing learning-rate search time from **90** to **4 hours**. TM formula results were obtained from a Thinking Machines Lab sweep, for which the runtime was unavailable.

Method	Gemma	Llama2	Llama3.1	Qwen3-1.7B	Qwen3-8B	Average	Time Cost
1×10^{-4}	35.0	30.0	50.0	38.6	45.5	39.8	0
2×10^{-4}	65.0	70.0	**100.0**	79.5	79.5	78.8	0
4×10^{-4}	80.0	95.0	54.5	77.3	61.4	73.6	0
TM Formula	65.0	75.0	72.7	61.4	52.3	65.3	-
Grid	**95.0**	**100.0**	**100.0**	**100.0**	**100.0**	**99.0**	90h
SPECTRA	89.1	83.4	92.6	98.2	79.3	88.5	4h

size and model family using an empirical scaling law fitted from learning-rate sweeps over 14 Llama and Qwen models on Tulu3 (Lambert et al., 2024). Specifically, Tinker models the learning rate as $\eta = M_{\text{LoRA}} \left(\frac{2000}{d_{\text{hidden}}}\right)^{p_{\text{model}}+p_{\text{LoRA}}}$, where M_{LoRA} is the LoRA-specific multiplier and p_{model} captures model-family-dependent scaling; the fitted rule uses a LoRA multiplier of 9.8 and $p_{\text{LoRA}} = 0$.

Setup. For each model–dataset pair, we jointly sweep the learning rate and LoRA rank to identify the optimal learning rate at each rank and characterize the rank–loss trade-off. We use LoRA with a fixed scaling parameter $\alpha = 32$ across all experiments. All runs share the same training configuration, and model selection is based on the held-out `final_eval_loss`. We perform the search in two stages. In the *coarse sweep*, we evaluate the endpoint ranks $r \in \{2, 256\}$ using a $\times 2$ geometric learning-rate grid. With a $\times 2$ logarithmic grid, the selected learning rate deviates from the continuous optimum by at most a factor of $\sqrt{2} \approx 1.4$, as shown in Section E.7. We therefore use the coarse optima $\eta_2^\star$ and $\eta_{256}^\star$ to define the refinement range as $[\min(\eta_2^\star, \eta_{256}^\star)/1.4,\ 1.4\max(\eta_2^\star, \eta_{256}^\star)]$, which accounts for this coarse-search uncertainty at both boundaries. In the *fine sweep*, we expand the ranks to $r \in \{2, 16, 64, 256\}$ and refine the search using a denser $\times 1.44$ geometric learning-rate grid. This resolution guarantees a multiplicative error below 1.2×, as shown in Section E.7.

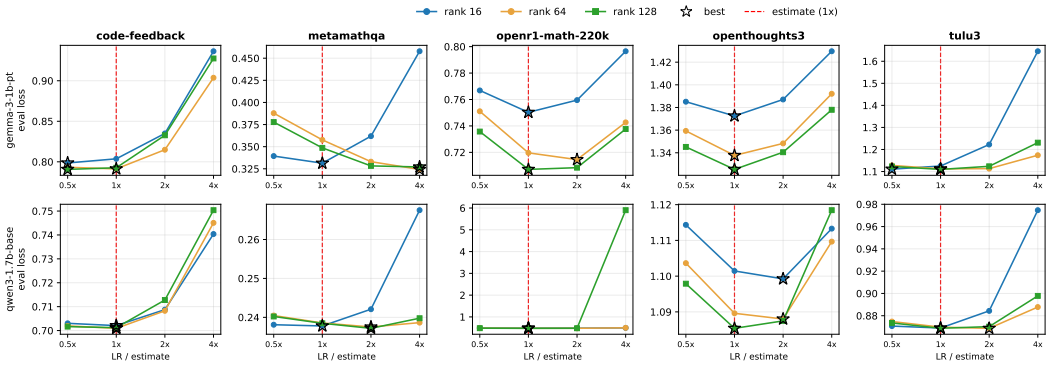

Figure 3: **Empirical validation of** SPECTRA **through real training runs.** For Gemma-3-1B-PT and Qwen3-1.7B-Base across five datasets and LoRA ranks $r \in \{16, 64, 128\}$, we report the held-out evaluation loss obtained using $0.5\times$, $1\times$, $2\times$, and $4\times$ the learning rate predicted by SPECTRA. The red dashed line denotes the prediction ($1\times$), and stars mark the best multiplier for each rank; the predicted rate consistently yields stable, near-optimal performance, whereas larger rates—particularly $4\times$—frequently degrade performance or cause instability.

Results. As shown in Table 2, SPECTRA achieves the highest average learning-rate selection success rate among non-exhaustive methods across five models and five datasets, with **88.5%** of configurations falling within the predefined $2.4\times$ tolerance of the empirical optimum. This exceeds fixed learning rates of 1×10^{-4}, 2×10^{-4}, and 4×10^{-4} by 48.7, 9.7, and 14.9 percentage points,

Table 3: **Learning-rate selection success rates across LoRA scaling rules.** We compare SPECTRA with five baselines on Gemma and Qwen3-1.7B under four scaling rules, $\alpha/r \in \{32/\sqrt{r},\ 32/r,\ 1,\ 32\}$, corresponding to $\alpha(r) \in \{32\sqrt{r},\ 32,\ r,\ 32r\}$. Values report Hit@2.4×, the percentage of configurations for which the selected learning rate lies within a factor of 2.4 of the empirical optimum. Best and second-best results are shown in **bold** and underlined, respectively. Averaged across all eight model–scaling settings, SPECTRA achieves the highest success rate of **84.7%**, outperforming the strongest baseline by **4.7** points; the margin is concentrated in the rank-dependent conventions $\alpha{=}r$ and $\alpha{=}32r$.

α/r	Model	1×10^{-4}	2×10^{-4}	4×10^{-4}	TM Formula	Grid	SPECTRA
$32/\sqrt{r}$	Gemma	65.0	85.0	45.0	35.0	**100.0**	86.2
	Qwen3-1.7B	80.0	70.0	35.0	15.0	**100.0**	93.8
$32/r$	Gemma	35.0	65.0	80.0	65.0	**95.0**	90.0
	Qwen3-1.7B	38.6	79.5	77.3	61.4	**100.0**	**100.0**
1	Gemma	20.0	55.0	70.0	70.0	80.0	**91.2**
	Qwen3-1.7B	40.0	60.0	65.0	60.0	70.0	**97.5**
32	Gemma	40.0	30.0	10.0	5.0	45.0	**63.7**
	Qwen3-1.7B	30.0	20.0	10.0	5.0	50.0	**55.0**
Average (8 settings)		43.6	58.1	49.0	39.5	80.0	**84.7**

respectively, and outperforms the TM formula by 23.2 points. While grid search reaches 99.0% under the constant-α setting, SPECTRA recovers 89% of that coverage with 4 rather than 90 hours of search. Its strong performance across diverse model families indicates that a single fixed learning rate does not reliably capture model-dependent optimization dynamics.

We further validate the learning rates selected by SPECTRA through real training runs in Figure 3. For each model–dataset–rank configuration, we train with $0.5\times$, $1\times$, $2\times$, and $4\times$ the predicted learning rate. Across all evaluated settings, the predicted $1\times$ learning rate yields stable training and remains near the lowest-loss region, whereas larger perturbations, particularly $4\times$, frequently increase evaluation loss substantially or induce optimization instability. These results show that SPECTRA not only selects learning rates close to the empirical optimum, but also places training within a robust stability region without exhaustive learning-rate sweeps.

5.1 Supplementary Experiments

In this section, we conduct additional experiments to examine the influence of our method's performance at different alpha setting. All supplementary experiments consider Gemma and Qwen3-1.7B as representative cases. Additional experiments are presented in Appendix I.

Prediction under Non-Constant α Settings. We vary the LoRA scaling factor α with the rest of the configuration fixed to test whether SPECTRA remains accurate under non-constant-α settings. As shown in Table 3, SPECTRA achieves the highest success rate in four of the six non-constant-α settings and trails only grid search in the remaining two (rsLoRA). Averaged across both models and all four LoRA scaling rules, SPECTRA reaches **84.7%**, outperforming the strongest baseline by **4.7** points. Further results across scaling rules are in Appendices J and K.

Calibration Data Efficiency. We assess the calibration required to estimate (c, b_β) reliably under the cross-model transfer protocol of Table 2. As shown in Table 7, three points per dataset achieve 87.1% Hit@2.4×, close to the 88.5% obtained using all points, whereas random three-point or single-point calibration performs substantially worse. These results indicate that three calibration points per dataset provide a practical accuracy–cost trade-off, while full-grid calibration remains optimal when calibration cost is unconstrained.

6 Discussion and Conclusion

We propose SPECTRA, a gradient-spectral framework for efficient learning-rate selection in LoRA fine-tuning. By combining LoRA rank with the initialization-time spectral-decay feature β, SPECTRA predicts stable, near-optimal learning rates after lightweight model-level calibration; theory characterizes the local stability boundary and its dependence on rank, module bottlenecks, and LoRA scaling. Across five model families and five datasets, SPECTRA achieves an **88.5%** success rate and reduces search time from **90** to **4** hours while preserving fine-tuning performance.

Ethical Statement

This work studies learning-rate selection for LoRA-based supervised fine-tuning and proposes SPECTRA to reduce the cost of hyperparameter search. In accordance with the ICLR Code of Ethics[1], we note that improving the efficiency of model adaptation may also lower the cost of fine-tuning models for unintended applications; moreover, SPECTRA does not address biases or safety degradation inherited from pretrained models or introduced during fine-tuning. Our study focuses on optimization efficiency and does not introduce new datasets, user-facing systems, or safety-critical deployment mechanisms.

Reproducibility

We report results over three random seeds and provide the corresponding means and standard deviations. All model–dataset pairs follow a controlled training configuration, with LoRA-rank sweeps specified in the experimental setup; the calibration procedure for (b, c), including the calibration-point selection strategy, is reported in Section 3. Detailed training hyperparameters, model and dataset configurations, and search ranges are provided in Section H.2, while complete assumptions and proofs supporting the learning-rate predictor are provided in Section 4.

AI Disclosure

In this work, we used generative AI tools to create or edit software code, help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, implement methods, support qualitative and thematic data analysis, design or provide feedback on research methodology or experiments, and grammatical correction. We have not used generative AI tools for propose or refine hypotheses and interpret results. Generate synthetic data sets, clean and reformat dataset, and assist with translation are not applicable to this work. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

References

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations*, volume 2024, pp. 21246–21263, 2024.

Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K Arora, Yu Bai, Bowen Baker, Haiming Bao, et al. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*, 2025.

Nan Chen, Soledad Villar, and Soufiane Hayou. Learning rate scaling across LoRA ranks and transfer to full finetuning. In *Forty-third International Conference on Machine Learning*, 2026.

Jeremy Cohen, Behrooz Ghorbani, Shankar Krishnan, Naman Agarwal, Sourabh Medapati, Michal Badura, Daniel Suo, Zachary Nado, George E. Dahl, and Justin Gilmer. Adaptive gradient methods at the edge of stability. In *NeurIPS 2023 Workshop Heavy Tails in Machine Learning*, 2023.

Xuan Cui, Huiyue Li, Run Zeng, Yunfei Zhao, Jinrui Qian, Wei Duan, Bo Liu, and Zhanpeng Zhou. Igu-lora: Adaptive rank allocation via integrated gradients and uncertainty-aware scoring. In *The Fourteenth International Conference on Learning Representations*, 2026.

Josef Dai, Xuehai Pan, Ruiyang Sun, Jiaming Ji, Xinbo Xu, Mickel Liu, Yizhou Wang, and Yaodong Yang. Safe RLHF: Safe reinforcement learning from human feedback. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024.

Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, et al. Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature machine intelligence*, 5(3):220–235, 2023.

[1] https://iclr.cc/public/CodeOfEthics

Yu Gao, Haoyuan Guo, Tuyen Hoang, Weilin Huang, Lu Jiang, Fangyuan Kong, Huixia Li, Jiashi Li, Liang Li, Xiaojie Li, et al. Seedance 1.0: Exploring the boundaries of video generation models. *arXiv preprint arXiv:2506.09113*, 2025.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Etash Guha, Ryan Marten, Sedrick Keh, Negin Raoof, Georgios Smyrnis, Hritik Bansal, Marianna Nezhurina, Jean Mercat, Trung Vu, Zayne Sprague, et al. Openthoughts: Data recipes for reasoning models. In *International Conference on Learning Representations*, volume 2026, pp. 108059–108130, 2026.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, 2025.

Zeyu Han, Chao Gao, Jinyang Liu, Jeff Zhang, and Sai Qian Zhang. Parameter-efficient fine-tuning for large models: A comprehensive survey. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856.

Soufiane Hayou, Nikhil Ghosh, and Bin Yu. LoRA+: Efficient low rank adaptation of large models. In *Forty-first International Conference on Machine Learning*, 2024.

Ruidan He, Linlin Liu, Hai Ye, Qingyu Tan, Bosheng Ding, Liying Cheng, Jiawei Low, Lidong Bing, and Luo Si. On the effectiveness of adapter-based tuning for pretrained language model adaptation. In *Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing (volume 1: long papers)*, pp. 2208–2222, 2021.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pp. 2790–2799. PMLR, 2019.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations (ICLR)*, 2021.

Hugging Face. Open r1: A fully open reproduction of deepseek-r1, January 2025. URL https://github.com/huggingface/open-r1.

Ke Ji, Jiahao Xu, Tian Liang, Qiuzhi Liu, Zhiwei He, Xiaoyuan Liu, Xingyu Chen, Junying Chen, Benyou Wang, Zhaopeng Tu, et al. The first few tokens are all you need: An efficient and effective unsupervised prefix fine-tuning method for reasoning models. *Advances in Neural Information Processing Systems*, 38:9305–9325, 2026.

Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *European conference on computer vision*, pp. 709–727. Springer, 2022.

Damjan Kalajdzievski. A rank stabilization scaling factor for fine-tuning with lora. *arXiv preprint arXiv:2312.03732*, 2023.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, et al. Tulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024.

Hang Le, Juan Pino, Changhan Wang, Jiatao Gu, Didier Schwab, and Laurent Besacier. Lightweight adapter tuning for multilingual speech translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pp. 817–824, 2021.

Yu-Ang Lee, Ching-Yun Ko, Pin-Yu Chen, and Mi-Yen Yeh. Learning rate matters: Vanilla LoRA may suffice for LLM fine-tuning. In *High-dimensional Learning Dynamics 2026*, 2026.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pp. 3045–3059, 2021.

Lujun Li, Dezhi Li, Cheng Lin, Wei Li, Wei Xue, Sirui Han, and Yike Guo. Aira: Activation-informed low-rank adaptation for large models. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1729–1739, 2025a.

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

Ziniu Li, Congliang Chen, Tian Xu, Zeyu Qin, Jiancong Xiao, Zhi-Quan Luo, and Ruoyu Sun. Preserving diversity in supervised fine-tuning of large language models. In *International Conference on Learning Representations*, volume 2025, pp. 66127–66154, 2025b.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Haozheng Luo, Chenghao Qiu, Maojiang Su, Zhihan Zhou, Zoe Mehta, Guo Ye, Jerry Yao-Chieh Hu, and Han Liu. Fast and low-cost genomic foundation models via outlier removal. In *Forty-second International Conference on Machine Learning*, 2025.

Fabian Paischer, Lukas Hauzenberger, Thomas Schmied, Benedikt Alkin, Marc Peter Deisenroth, and Sepp Hochreiter. One initialization to rule them all: Fine-tuning via explained variance adaptation. In *NeurIPS 2024 Workshop on Fine-Tuning in Modern Machine Learning: Principles and Scalability*, 2024.

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

Zhengxiang Shi and Aldo Lipani. Dept: Decomposed prompt tuning for parameter-efficient fine-tuning. In *International Conference on Learning Representations*, volume 2024, pp. 51862–51882, 2024.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Zhengbo Wang, Jian Liang, Ran He, Zilei Wang, and Tieniu Tan. LoRA-pro: Are low-rank adapters properly optimized? In *The Thirteenth International Conference on Learning Representations*, 2025.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Jui-Nan Yen, Si Si, Zhao Meng, Felix Yu, Sai Surya Duvvuri, Inderjit S Dhillon, Cho-Jui Hsieh, and Sanjiv Kumar. LoRA done RITE: Robust invariant transformation equilibration for LoRA optimization. In *The Thirteenth International Conference on Learning Representations*, 2025.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng YU, Zhengying Liu, Yu Zhang, James Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. Metamath: Bootstrap your own mathematical questions for large language models. In *The Twelfth International Conference on Learning Representations*, 2024.

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

Zhen-Ru Zhang, Chuanqi Tan, Haiyang Xu, Chengyu Wang, Jun Huang, and Songfang Huang. Towards adaptive prefix tuning for parameter-efficient language model fine-tuning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 1239–1248, 2023b.

Tianyu Zheng, Ge Zhang, Tianhao Shen, Xueling Liu, Bill Yuchen Lin, Jie Fu, Wenhu Chen, and Xiang Yue. OpenCodeInterpreter: Integrating code generation with execution and refinement. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 12834–12859, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

Supplementary Material

A Related Work **14**

B Additional Theoretical Notation **15**

C Proofs of Main Text **16**
C.1 Lemma 4.1 . . . 16
C.2 Lemma 4.2 . . . 16
C.3 Proposition 4.1 . . . 17
C.4 Proposition 4.2 . . . 17
C.5 Corollary 4.1 . . . 17
C.6 Proposition 4.3 . . . 18
C.7 Proposition 4.4 . . . 18
C.8 Proposition 4.5 . . . 18
C.9 Proposition 4.6 . . . 19
C.10 Corollary 4.2 . . . 20

D Assumptions **20**

E Additional Theory **22**
E.1 Estimation of the Local Spectral-Decay Exponent . . . 22
E.2 Scaling Transfer in Two Limiting Regimes . . . 23
E.3 Effective LoRA Learning Rate . . . 23
E.4 Stability Boundary with Momentum . . . 24
E.5 From the Stability Boundary to the Training Learning Rate . . . 24
E.6 A Sufficient Condition for Spectral-Decay–Curvature Monotonicity . . . 24
E.7 Learning-Rate Grid Resolution . . . 25

F Model Checkpoints **26**

G Additional Analysis of LoRA Learning-Rate Stability **26**

H Experiment System and Implement Settings **29**
H.1 Computational Resources . . . 29
H.2 Hyperparameters . . . 29
H.3 The grid-search baseline . . . 29

I Additional Experiments **29**
I.1 Single-Point Calibration of a_0 . . . 29
I.2 Calibration Data Efficiency . . . 30
I.3 Ablation of the Predictor Terms . . . 30

J Why $\alpha = 32r$ Breaks the Predictor **30**

K Empirical Validation of SPECTRA Training Stability Across LoRA Alpha Scaling Settings **33**

L A β-Free Test of the Scaling Transfer **34**

M All Results in Low-rank Adaptation **34**

N Sweep Protocol and Full Optimal Learning Rates **36**

A Related Work

In recent years, Large Language Models (LLMs) such as GPT-OSS (Agarwal et al., 2025), DeepSeek-R1 (Guo et al., 2025), and Qwen3 (Yang et al., 2025) have demonstrated strong capabilities across

diverse downstream tasks, but their rapidly growing parameter scales render full fine-tuning prohibitively expensive in both computation and memory. To adapt LLMs efficiently, numerous parameter-efficient fine-tuning (PEFT) (Luo et al., 2025; Han et al., 2024) methods have been proposed, falling into the main paradigms (Ding et al., 2023): addition-based methods, including Adapter Tuning (Le et al., 2021; He et al., 2021; Houlsby et al., 2019), Prefix Tuning (Ji et al., 2026; Zhang et al., 2023b; Li & Liang, 2021), and Prompt Tuning (Shi & Lipani, 2024; Jia et al., 2022; Lester et al., 2021), and reparameterization-based methods, among which Low-Rank Adaptation (LoRA) (Hu et al., 2021) has emerged as a key technique. However, its performance remains highly sensitive to the learning rate, which interacts with adapter rank, initialization, and scaling and is typically selected through costly configuration-specific sweeps.

Existing methods (Lee et al., 2026; Chen et al., 2026) improve LoRA optimization without directly predicting a task-specific learning rate. LoRA+ (Hayou et al., 2024) assigns different learning rates to the two adapter factors, whereas rsLoRA (Kalajdzievski, 2023) modifies rank-dependent scaling. LoRA-Pro (Wang et al., 2025) and LoRA-RITE (Yen et al., 2025) introduce preconditioned or invariant optimization updates. Most closely related, μA (Chen et al., 2026) derives asymptotic learning-rate scaling rules across ranks, initialization schemes, and LoRA scaling factors. These approaches prescribe optimizer modifications or configuration-level scaling laws but do not use task-specific gradient structure to predict an absolute learning rate. In contrast, SPECTRA uses the initialization-time gradient spectrum and lightweight model-level calibration to predict stable, near-optimal learning rates across datasets and LoRA ranks.

B Additional Theoretical Notation

This section introduces notation used only for the local spectral stability analysis. For each LoRA target module ℓ and calibration example $x \in \mathcal{D}_0$, let

$$G^{(\ell)}(x) = U^{(\ell)}(x)\,\Sigma^{(\ell)}(x)\,V^{(\ell)}(x)^\top = \sum_{i=1}^{N_\ell} \sigma_i^{(\ell)}(x)\,u_i^{(\ell)}(x)\,v_i^{(\ell)}(x)^\top$$

denote the singular value decomposition of the initialization-time teacher-forced gradient, with $\sigma_1^{(\ell)}(x) \geq \cdots \geq \sigma_{N_\ell}^{(\ell)}(x) \geq 0$. We define the spectral LoRA surrogate subspace

$$\mathcal{S}_{r_\ell}^{(\ell)}(x) := \operatorname{span}\left\{ u_i^{(\ell)}(x)\,v_i^{(\ell)}(x)^\top : 1 \leq i \leq r_\ell \right\}.$$

The best rank-r_ℓ approximation of $G^{(\ell)}(x)$ in Frobenius norm (Eckart–Young) lies in $\mathcal{S}_{r_\ell}^{(\ell)}(x)$. The subspace itself, however, is neither the set of rank-r_ℓ matrices nor the tangent space of standard LoRA at initialization (Assumption D.3). We use it only as a gradient-aligned surrogate for the stability-relevant update directions; it is not an exact description of the nonlinear dynamics of the factors (A, B).

Let $P_{r_\ell}^{(\ell)}(x)$ denote the orthogonal projector onto $\mathcal{S}_{r_\ell}^{(\ell)}(x)$. We define the initialization-time SFT Hessian with respect to the module weight and its restriction to the surrogate as

$$H^{(\ell)} := \nabla^2_{W^{(\ell)}} \mathcal{L}_{\mathrm{SFT}}(\theta_0), \qquad H_{r_\ell}^{(\ell)}(x) := P_{r_\ell}^{(\ell)}(x)\,H^{(\ell)}\,P_{r_\ell}^{(\ell)}(x)\Big|_{\mathcal{S}_{r_\ell}^{(\ell)}(x)}.$$

Preconditioned quantities. Let $D^{(\ell)} := \operatorname{diag}\big(\sqrt{\hat{v}^{(\ell)}}\big) + \epsilon I$ denote the frozen Adam preconditioner of module ℓ (Assumption D.6). The main-text analysis is carried out in whitened coordinates $\tilde{e} = (D^{(\ell)})^{1/2} e$. We define the whitened surrogate subspace, with orthogonal projector $\widetilde{P}_{r_\ell}^{(\ell)}(x)$, and the preconditioned restricted Hessian

$$\widetilde{\mathcal{S}}_{r_\ell}^{(\ell)}(x) := (D^{(\ell)})^{1/2}\,\mathcal{S}_{r_\ell}^{(\ell)}(x), \qquad \widetilde{H}_{r_\ell}^{(\ell)}(x) := \widetilde{P}_{r_\ell}^{(\ell)}(x)\,(D^{(\ell)})^{-1/2} H^{(\ell)} (D^{(\ell)})^{-1/2}\,\widetilde{P}_{r_\ell}^{(\ell)}(x)\Big|_{\widetilde{\mathcal{S}}_{r_\ell}^{(\ell)}(x)}.$$

For $D^{(\ell)} = I$, these reduce to $\mathcal{S}_{r_\ell}^{(\ell)}(x)$, $P_{r_\ell}^{(\ell)}(x)$, and $H_{r_\ell}^{(\ell)}(x)$.

The analysis is stated for a fixed calibration example x. Because the predictor is affine in β, averaging β over $\mathcal{D}_0$ is equivalent to averaging the resulting boundaries in log space (Section C.10). When the module index ℓ and the example x are unambiguous, we suppress them and write $\mathcal{S}_r$, P_r, H_r, $\widetilde{\mathcal{S}}_r$, $\widetilde{P}_r$, and $\widetilde{H}_r$.

Remark B.1 (Weight-space boundary versus optimizer learning rate)**.** The boundary $\eta_W^\star$ of Proposition 4.1 is defined for the preconditioned dynamics of the effective update ΔW restricted to $\mathcal{S}_r$ and does not depend on the LoRA scaling $s(r) = \alpha(r)/r$. The optimizer instead updates the factors (A, B), which enter the model as $s(r)BA$. The two boundaries are related by Assumption 4.1, $\eta_{\mathrm{spec}}^\star = \kappa(M)\, s^{-p}\, \eta_W^\star$, whose exponent is exact in the two limiting regimes of Lemma E.1: $p = 1$ when the update is carried by B alone, so that an optimizer step of size η displaces ΔW by $\mathcal{O}(s\eta)$, and $p = \frac{1}{2}$ when both factors are updated in balance. Proposition 4.2 is therefore a statement about $\eta_W^\star$ only. The rank dependence of the optimizer learning rate combines it with the parameterization factor $s(r)^{-p} \propto r^{p\gamma}$; under constant α ($\gamma = 1$), this factor grows as r^p and can dominate the weight-space decrease. The increase of the empirical optimum with r at small ranks in Tables 15 and 16 is therefore consistent with Proposition 4.2.

C Proofs of Main Text

C.1 Lemma 4.1

Proof of Lemma 4.1. Work in whitened coordinates $\varphi = D^{1/2}\theta$ and set $\widetilde{\mathcal{L}}(\varphi) := \mathcal{L}_{\mathrm{SFT}}(D^{-1/2}\varphi)$, so that

$$\nabla_\varphi \widetilde{\mathcal{L}} = D^{-1/2}\nabla \mathcal{L}_{\mathrm{SFT}}, \qquad \nabla_\varphi^2 \widetilde{\mathcal{L}}(\varphi_0) = D^{-1/2} H D^{-1/2} =: \widetilde{H}.$$

In these coordinates, the restricted map reads $\widetilde{T}_r(\varphi) = \varphi - \eta \widetilde{P}_r \nabla_\varphi \widetilde{\mathcal{L}}(\varphi)$, and $D^{1/2} T_r(\theta) = \widetilde{T}_r(D^{1/2}\theta)$.

One-step expansion. For $\tilde{e} \in \widetilde{\mathcal{S}}_r$, Assumption D.2 gives $\nabla_\varphi \widetilde{\mathcal{L}}(\varphi_0 + \tilde{e}) = \nabla_\varphi \widetilde{\mathcal{L}}(\varphi_0) + \widetilde{H}\tilde{e} + \mathcal{O}(\|\tilde{e}\|^2)$. Using $\widetilde{P}_r \tilde{e} = \tilde{e}$,

$$\widetilde{T}_r(\varphi_0 + \tilde{e}) - \widetilde{T}_r(\varphi_0) = \tilde{e} - \eta \widetilde{P}_r \widetilde{H} \widetilde{P}_r \tilde{e} + \mathcal{O}\big(\|\tilde{e}\|^2\big) = \big(I - \eta \widetilde{H}_r\big)\tilde{e} + \mathcal{O}\big(\|\tilde{e}\|^2\big).$$

Exact recursion under the quadratic model. Under Assumption D.6, $\nabla_\varphi \widetilde{\mathcal{L}}(\varphi) = \tilde{g}_0 + \widetilde{H}(\varphi - \varphi_0)$ exactly, so $\widetilde{T}_r$ is affine. Let φ_t and φ_t' be two trajectories with $\varphi_0' - \varphi_0 = \tilde{e}_0 \in \widetilde{\mathcal{S}}_r$, and set $\tilde{e}_t := \varphi_t' - \varphi_t$. The drift $-\eta \widetilde{P}_r \tilde{g}_0$ is common to both trajectories and cancels, so

$$\tilde{e}_{t+1} = \tilde{e}_t - \eta \widetilde{P}_r \widetilde{H} \tilde{e}_t.$$

If $\tilde{e}_t \in \widetilde{\mathcal{S}}_r$, then $\widetilde{P}_r \widetilde{H} \tilde{e}_t = \widetilde{H}_r \tilde{e}_t \in \widetilde{\mathcal{S}}_r$; by induction, $\tilde{e}_{t+1} = (I - \eta \widetilde{H}_r)\tilde{e}_t$ for all $t \geq 0$. No stationarity of θ_0 is required.

Since $\widetilde{H}_r$ is symmetric, projecting onto an eigenvector w_j with $\widetilde{H}_r w_j = \mu_j w_j$ gives $w_j^\top \tilde{e}_{t+1} = (1 - \eta\mu_j)\, w_j^\top \tilde{e}_t$, and iterating yields

$$w_j^\top \tilde{e}_t = (1 - \eta\mu_j)^t\, w_j^\top \tilde{e}_0.$$

C.2 Lemma 4.2

Proof of Lemma 4.2. Iterating the scalar recursion gives

$$x_t = \rho^t x_0.$$

Hence

$$|x_t| = |\rho|^t |x_0|,$$

which converges to zero for every finite x_0 if and only if

$$|\rho| < 1.$$

Substituting

$$\rho = 1 - \eta\lambda$$

gives

$$|1 - \eta\lambda| < 1.$$

Equivalently,

$$0 < \eta\lambda < 2.$$

Since $\lambda > 0$,

$$0 < \eta < \frac{2}{\lambda}.$$

C.3 Proposition 4.1

Proof of Proposition 4.1. By Lemma 4.1, the restricted dynamics along every eigenvector w_j of $\widetilde{H}_r$ satisfy

$$w_j^\top \tilde{e}_t = (1 - \eta\mu_j)^t \, w_j^\top \tilde{e}_0.$$

By Assumption D.2, $\widetilde{H}_r \succ 0$ on $\widetilde{\mathcal{S}}_r$, so $\mu_j > 0$. By Lemma 4.2, the j-th coordinate contracts for every initial condition if and only if $0 < \eta < 2/\mu_j$. Requiring this for all j gives

$$0 < \eta < \frac{2}{\max_j \mu_j} = \frac{2}{\lambda_{\max}(\widetilde{H}_r)} = \eta_W^\star(r).$$

C.4 Proposition 4.2

Proof of Proposition 4.2. By the Rayleigh–Ritz characterization,

$$\lambda_{\max}(\widetilde{H}_r) = \max_{\substack{\tilde{e} \in \widetilde{\mathcal{S}}_r \\ \|\tilde{e}\|=1}} \tilde{e}^\top D^{-1/2} H D^{-1/2} \tilde{e}.$$

By Assumption D.3, $\mathcal{S}_{r_1} \subseteq \mathcal{S}_{r_2}$. Since D does not depend on r (Assumption D.6) and $D^{1/2}$ is linear, $\widetilde{\mathcal{S}}_{r_1} \subseteq \widetilde{\mathcal{S}}_{r_2}$. Maximizing over a larger set cannot decrease the maximum, hence

$$\lambda_{\max}(\widetilde{H}_{r_1}) \le \lambda_{\max}(\widetilde{H}_{r_2}).$$

Because $t \mapsto 2/t$ is decreasing on $t > 0$, $\eta_W^\star(r_1) \ge \eta_W^\star(r_2)$.

C.5 Corollary 4.1

Proof of Corollary 4.1. Fix β and let $u = \log_{10} r$. By Assumption 4.1 with $s(r) = s_0 r^{-\gamma}$,

$$\log_{10} \eta_{\text{spec}}^\star = \log_{10} \kappa(M) - p \log_{10} s_0 + p\gamma \, u + \log_{10} \eta_W^\star. \tag{C.1}$$

Differentiating in u at fixed β, for any differentiable local interpolation of $\eta_W^\star$ in u,

$$c = -\frac{\partial \log_{10} \eta_{\text{spec}}^\star}{\partial u} = -\frac{\partial \log_{10} \eta_W^\star}{\partial u} - p\gamma = c_W + q_\alpha.$$

By Proposition 4.2, applied at fixed β within the nested family of Assumption D.5, $\eta_W^\star$ is nonincreasing in r, so $c_W \geq 0$ and $c \geq q_\alpha = -p\gamma \geq -\gamma$, where the last inequality uses $p \leq 1$. Setting $p = \frac{1}{2}$ and $\gamma = 1, \frac{1}{2}, 0$ yields $q_\alpha = -\frac{1}{2}, -\frac{1}{4}, 0$.

For the paired statement, evaluate the same $(M, \mathcal{D}, r)$ under a convention γ and under the reference $\gamma = 1$. The weight-space boundary $\eta_W^\star$ refers to the effective update and is common to both, as are $\kappa(M)$ and p. Subtracting the two instances of Equation (C.1) gives

$$\Delta \log_{10} \eta_{\text{spec}}^\star = \Delta\delta - \Delta c\, u, \qquad \Delta\delta := -p\big(\log_{10} s_0(\gamma) - \log_{10} s_0(1)\big), \qquad \Delta c = p(1-\gamma).$$

C.6 Proposition 4.3

Proof of Proposition 4.3. Because D is diagonal, it is block diagonal across modules, and whitening preserves the block structure of Assumption D.4:

$$\widetilde{H}_{\text{joint}} = \operatorname{diag}\left(\widetilde{H}_{r_1}^{(1)}, \ldots, \widetilde{H}_{r_L}^{(L)}\right).$$

Hence the joint linearized dynamics decouple across modules, $\tilde{e}_{t+1}^{(\ell)} = \big(I - \eta \widetilde{H}_{r_\ell}^{(\ell)}\big)\tilde{e}_t^{(\ell)}$. By Proposition 4.1, module ℓ is stable if and only if $0 < \eta < \eta_{W,\ell}^\star$. Because all modules share the same learning rate, stability must hold simultaneously for every module, which gives

$$0 < \eta < \min_\ell \eta_{W,\ell}^\star = \frac{2}{\max_\ell \lambda_{\max}\big(\widetilde{H}_{r_\ell}^{(\ell)}\big)}.$$

C.7 Proposition 4.4

Proof of Proposition 4.4. For fixed η, $\mu \mapsto |1 - \eta\mu|$ is convex, so its maximum over $[\mu_{\min}, \mu_{\max}]$ is attained at an endpoint:

$$\rho(\eta) = \max\left\{|1 - \eta\mu_{\min}|, |1 - \eta\mu_{\max}|\right\}.$$

The first term decreases and the second increases in η over the relevant range, so the minimax optimum balances them, $1 - \eta\mu_{\min} = -(1 - \eta\mu_{\max})$, which yields

$$\eta_{\text{opt}}^{\text{lin}} = \frac{2}{\mu_{\max} + \mu_{\min}}.$$

With $\eta_W^\star(r) = 2/\mu_{\max}$ and $\omega_r = \mu_{\min}/\mu_{\max} \in (0, 1]$,

$$\frac{\eta_W^\star(r)}{\eta_{\text{opt}}^{\text{lin}}} = \frac{\mu_{\max} + \mu_{\min}}{\mu_{\max}} = 1 + \omega_r \in (1, 2].$$

Taking base-10 logarithms gives $\log_{10} \eta_W^\star(r) - \log_{10} \eta_{\text{opt}}^{\text{lin}} = \log_{10}(1 + \omega_r) \in (0, \log_{10} 2]$. Finally, every $x \in (0, \log_{10} 2]$ satisfies $|x - \frac{1}{2}\log_{10} 2| \leq \frac{1}{2}\log_{10} 2 \approx 0.15$ dex.

C.8 Proposition 4.5

Proof of Proposition 4.5. Part (i). Under the global rescaling $g_t' = kg_t$ with $k > 0$ and zero-initialized moments, induction gives $m_t' = km_t$ and $v_t' = k^2 v_t$, and the same scaling holds after bias correction:

$\hat{m}'_t = k\hat{m}_t$ and $\hat{v}'_t = k^2\hat{v}_t$. Therefore, in the limit $\epsilon \to 0$,

$$\frac{\hat{m}'_t}{\sqrt{\hat{v}'_t}} = \frac{k\hat{m}_t}{\sqrt{k^2\hat{v}_t}} = \frac{\hat{m}_t}{\sqrt{\hat{v}_t}},$$

so $\Delta\theta'_t = \Delta\theta_t$ for every iteration t.

Part (ii). Rescale $\mathcal{L}_{\mathrm{SFT}} \mapsto k\,\mathcal{L}_{\mathrm{SFT}}$. Then $G \mapsto kG$ and $H \mapsto kH$. The singular vectors of G are unchanged, so $\mathcal{S}_r$ is unchanged, while $\sigma_i \mapsto k\sigma_i$ leaves every log–log slope, and hence $\beta_\ell(\mathcal{D}; r)$, unchanged (Section E.1). By part (i), $\hat{v} \mapsto k^2\hat{v}$, so $D \mapsto kD$ in the limit $\epsilon \to 0$. Therefore $\widetilde{\mathcal{S}}_r = D^{1/2}\mathcal{S}_r \mapsto k^{1/2}D^{1/2}\mathcal{S}_r = \widetilde{\mathcal{S}}_r$, $\widetilde{P}_r$ is unchanged, and

$$(kD)^{-1/2}(kH)(kD)^{-1/2} = D^{-1/2}HD^{-1/2}.$$

Hence $\widetilde{H}_r$ and $\eta^\star_W(r) = 2/\lambda_{\max}(\widetilde{H}_r)$ are unchanged. For $D = I$, by contrast, $2/\lambda_{\max}(kH_r) = k^{-1} \cdot 2/\lambda_{\max}(H_r)$, which would contradict part (i).

C.9 Proposition 4.6

Proof of Proposition 4.6. Let $u := \log_{10} r$. We separate the weight-space boundary of the surrogate from the optimizer-level learning rate.

Step 1 (weight-space boundary). For rank r and spectral state β, Proposition 4.1 gives

$$\eta^\star_W(r, \beta) = \frac{2}{\lambda_{\max}\big(\widetilde{H}_r(\beta)\big)},$$

which is well defined because $\widetilde{H}_r(\beta) \succ 0$ (Assumption D.2). Define $F_W(u, \beta) := \log_{10} \eta^\star_W(r, \beta)$, and assume that it admits a twice continuously differentiable local interpolation around a reference state (u_0, β_0).

Step 2 (optimizer-level boundary). By Assumption 4.1,

$$F(u, \beta) := \log_{10} \eta^\star_{\mathrm{spec}}(r, \beta) = \log_{10} \kappa(M) - p \log_{10} s_0 + p\gamma\, u + F_W(u, \beta),$$

which inherits the regularity of F_W.

Step 3 (rank coefficient). By Corollary 4.1, $c := -\partial F/\partial u|_{(u_0,\beta_0)} = c_W + q_\alpha \geq -p\gamma$.

Step 4 (spectral coefficient). Since s does not depend on β, Assumption D.5 gives

$$\frac{\partial F}{\partial \beta} = \frac{\partial F_W}{\partial \beta} = -\frac{1}{\ln 10}\,\frac{1}{\lambda_{\max}\big(\widetilde{H}_r(\beta)\big)}\,\frac{\partial}{\partial \beta}\lambda_{\max}\big(\widetilde{H}_r(\beta)\big) < 0,$$

and we define $b_\beta := -\partial F/\partial \beta|_{(u_0,\beta_0)} > 0$.

Step 5 (local expansion). Let $\Delta = (u - u_0,\ \beta - \beta_0)$. A first-order Taylor expansion of F around (u_0, β_0) yields

$$F(u, \beta) = F(u_0, \beta_0) - c\,(u - u_0) - b_\beta(\beta - \beta_0) + \mathcal{O}\big(\|\Delta\|^2\big).$$

Defining $a_0(M) := F(u_0, \beta_0) + c\,u_0 + b_\beta\beta_0$, which absorbs $\kappa(M)$, s_0, and the momentum factor of Remark E.1, and substituting $u = \log_{10} r$ and $\beta = \beta(\mathcal{D}; r)$ gives

$$\log_{10} \eta^\star_{\mathrm{spec}}(r, \beta) = a_0(M) - c \log_{10} r - b_\beta\,\beta(\mathcal{D}; r) + \mathcal{O}\big(\|\Delta\|^2\big), \qquad c = c_W + q_\alpha \geq -p\gamma, \quad b_\beta > 0.$$

Finally, because $\beta(\mathcal{D}; r)$ itself varies with r, the total rank slope at fixed $\mathcal{D}$ is

$$\frac{\mathrm{d}F}{\mathrm{d}u} = -c - b_\beta\,\frac{\partial \beta(\mathcal{D}; r)}{\partial u},$$

which is the slope visible in Table 15; the partial coefficient c is identified only by the joint fit.

C.10 COROLLARY 4.2

Proof of Corollary 4.2. By Assumption D.7(i), Proposition 4.6 applies to each module with shared coefficients,

$$\log_{10} \eta^{\star}_{\mathrm{spec},\ell} = a_{0,\ell}(M) - c\,u - b_\beta \beta_\ell(\mathcal{D}; r) + \mathcal{O}\big(\|\Delta\|^2\big).$$

Because all modules share $s(r)$, $\kappa(M)$, and p, Assumption 4.1 maps Proposition 4.3 to the optimizer level: the shared boundary is $\min_\ell \eta^{\star}_{\mathrm{spec},\ell} = \kappa(M)\, s(r)^{-p} \min_\ell \eta^{\star}_{W,\ell}$. Taking logarithms and using the definition of the bottleneck gap in Assumption D.7,

$$\min_\ell \log_{10} \eta^{\star}_{\mathrm{spec},\ell} = \bar{a}_0(M) - c\,u - b_\beta \bar{\beta}(\mathcal{D}; r) + \delta(M, \mathcal{D}, r) + \mathcal{O}\big(\|\Delta\|^2\big).$$

By Assumption D.7(ii), $\delta(M, \mathcal{D}, r) = \bar{\delta}(M) + \mathcal{O}(\varepsilon_\Delta)$. Setting $a_0(M) := \bar{a}_0(M) + \bar{\delta}(M)$ gives the stated form. The final statement of the corollary follows from Proposition 4.4.

Aggregation over calibration examples. The analysis is stated for a fixed example x. Since the predictor is affine in β, the log-space average of per-example predictions equals the prediction at the averaged feature, $\frac{1}{|\mathcal{D}_0|} \sum_{x \in \mathcal{D}_0} \log_{10} \widehat{\eta}^{\star}(x) = a_0(M) - c\,u - b_\beta\, \beta(\mathcal{D}_0; r)$.

Choice of $\bar{\beta}$. If the module intercepts coincide, $a_{0,\ell} \equiv \bar{a}_0$, then $\delta = -b_\beta\big(\max_\ell \beta_\ell - \bar{\beta}\big)$, and Assumption D.7(ii) states that the module spread of β is stable across datasets and ranks. The plug-in estimator $\max_\ell \widehat{\beta}_\ell$ of L noisy per-module estimates is biased upward and has higher variance than $\bar{\beta}$, whereas under Assumption D.7 both carry the same $(\mathcal{D}, r)$-dependence up to ε_Δ. Table 4 reports the corresponding ablation.

Table 4: **Module aggregation of the spectral feature.** Hit@2.4× (%), src-mean over the four remaining source models. All modules share rank r; if the module intercepts coincide, $\min_\ell \widehat{\eta}^{\star}_\ell$ is attained at $\max_\ell \beta_\ell$. The module spread of β_ℓ is small ($\max_\ell / \min_\ell \approx$ 1.2–1.6, largest on `k_proj`/`q_proj` and smallest on `down_proj`), so averaging is preferable to either extreme.

Aggregation	Gemma	Llama2	Llama3.1	Qwen3-1.7B	Qwen3-8B	Average
$\bar{\beta} = \mathrm{mean}_\ell\, \beta_\ell$ (ours)	**89.1**	**83.4**	**92.6**	**98.2**	**79.3**	**88.5**
$\max_\ell \beta_\ell$ ($= \min_\ell \widehat{\eta}^{\star}_\ell$)	82.5	83.1	88.4	95.3	75.4	84.9
$\min_\ell \beta_\ell$ ($= \max_\ell \widehat{\eta}^{\star}_\ell$)	66.2	79.1	83.4	93.6	72.3	78.9

D ASSUMPTIONS

We state the assumptions required by the local SFT analysis. Assumptions D.2, D.3 and D.6 characterize the restricted preconditioned dynamics and underlie Lemma 4.1 and Propositions 4.1 and 4.2. Assumption D.4 is used for Proposition 4.3. Assumption D.1 delimits the regime of the scaling transfer (Assumption 4.1 in the main text and Lemma E.1). Assumption D.5 connects the spectral shape to curvature. Assumptions D.7 and D.8 are used only to pass from the per-module theory to the shared predictor of Corollary 4.2 and to its cross-model calibration.

Assumption D.1 (Effective two-factor updates)**.** There exists a warm-up step $t_0 \geq 1$ such that, for all $t \geq t_0$ within the optimization interval used to determine $\eta^{\star}$: (i) both LoRA factors receive numerically effective updates,

$$\|\Delta A_t\|_F > \varepsilon_{\mathrm{num}} \|A_t\|_F, \qquad \|\Delta B_t\|_F > \varepsilon_{\mathrm{num}} \|B_t\|_F,$$

where $\Delta A_t = A_{t+1} - A_t$, $\Delta B_t = B_{t+1} - B_t$, and $\varepsilon_{\mathrm{num}}$ denotes the effective numerical-resolution threshold of the training precision; and (ii) every coordinate of both factors satisfies $\sqrt{\hat{v}_t} \gg \epsilon$, so that the $\epsilon \to 0$ idealization of Proposition 4.5 applies.

Under the standard initialization $B_0 = 0$, $\nabla_A \mathcal{L}_{\mathrm{SFT}} = s\, B_0^\top G = 0$, so neither condition can hold at $t = 0$. The warm-up t_0 covers the steps before B_t becomes nonzero and the gradient of A grows above the ϵ scale.

Assumption D.2 (Local SFT smoothness and restricted curvature)**.** The teacher-forced SFT objective $\mathcal{L}_{\mathrm{SFT}}(\theta)$ is twice continuously differentiable in a neighborhood of the pretrained initialization θ_0, and its Hessian H is locally Lipschitz. Moreover, H is positive definite on $\mathcal{S}_r$, i.e., $e^\top H e > 0$ for every nonzero $e \in \mathcal{S}_r$. Because $D \succ 0$, this is equivalent to $\widetilde{H}_r \succ 0$ on $\widetilde{\mathcal{S}}_r$, since $\tilde{e}^\top \widetilde{H}_r \tilde{e} = e^\top H e$ for $\tilde{e} = D^{1/2} e$.

For teacher-forced token-level cross-entropy, the output-space curvature is positive semidefinite:

$$\nabla_z^2 \ell = \mathrm{diag}(p) - pp^\top \succeq 0.$$

The parameter-space Hessian additionally contains terms induced by the nonlinearity of the network. Assumption D.2 therefore does not require the full SFT Hessian to be globally positive definite; it requires positive curvature only on the surrogate directions.

Assumption D.3 (Rank-r spectral LoRA surrogate)**.** For a fixed calibration example x, let

$$G(x) = U \Sigma V^\top = \sum_{i=1}^{N} \sigma_i u_i v_i^\top$$

denote the SVD of the initialization-time teacher-forced gradient. We use

$$\mathcal{S}_r := \mathrm{span}\left\{u_1 v_1^\top, \ldots, u_r v_r^\top\right\}$$

as a surrogate for the stability-relevant update directions. By construction,

$$r_1 \le r_2 \quad \Longrightarrow \quad \mathcal{S}_{r_1} \subseteq \mathcal{S}_{r_2}.$$

$\mathcal{S}_r$ is an r-dimensional linear subspace. It differs from (a) the set $\{\Delta W : \mathrm{rank}\, \Delta W \le r\}$, which is not a linear subspace and contains the Eckart–Young approximation $\sum_{i \le r} \sigma_i u_i v_i^\top$ as a point rather than as a feasible space, and (b) the tangent space $\{B' A_0 : B' \in \mathbb{R}^{m \times r}\}$ of standard LoRA at $(A_0, B_0 = 0)$, which has dimension mr and whose row space is fixed by the random A_0. $\mathcal{S}_r$ describes an idealized adapter whose update is confined to the leading singular directions of the initialization-time gradient, in the spirit of gradient-aligned initializations. The nesting property holds for this surrogate but not, in general, for independently initialized A_0 at different ranks.

Assumption D.4 (Block-diagonal restricted SFT dynamics)**.** Cross-module SFT curvature coupling is negligible over the surrogate directions. Specifically, after restricting each module ℓ to $\mathcal{S}_{r_\ell}^{(\ell)}$, the joint restricted Hessian is block diagonal:

$$H_{\mathrm{joint}} = \mathrm{diag}\left(H_{r_1}^{(1)}, \ldots, H_{r_L}^{(L)}\right), \qquad H_{r_\ell}^{(\ell)} := P_{r_\ell}^{(\ell)} H^{(\ell)} P_{r_\ell}^{(\ell)}\Big|_{\mathcal{S}_{r_\ell}^{(\ell)}}.$$

Because D is diagonal, the same holds for $\widetilde{H}_{\mathrm{joint}}$.

Assumption D.5 (Local spectral-decay–curvature monotonicity)**.** Around the spectral regime considered by SPECTRA, we model the initialization-time gradient spectra by a smooth local family indexed by the spectral-decay exponent β. For each spectral state β and rank r, let $\widetilde{\mathcal{S}}_r(\beta)$ and $\widetilde{H}_r(\beta)$ denote the corresponding whitened surrogate subspace and preconditioned restricted Hessian. For every fixed β, the surrogate subspaces are nested:

$$r_1 \le r_2 \quad \Longrightarrow \quad \widetilde{\mathcal{S}}_{r_1}(\beta) \subseteq \widetilde{\mathcal{S}}_{r_2}(\beta).$$

For every fixed rank r, the largest restricted curvature varies smoothly with β and is locally increasing:

$$\frac{\partial}{\partial \beta}\lambda_{\max}\big(\widetilde{H}_r(\beta)\big) > 0.$$

Thus, at fixed rank, a more rapidly decaying local gradient spectrum corresponds to larger restricted curvature and hence to a smaller stability boundary. This assumption determines the direction of the spectral correction, while its magnitude is calibrated empirically. Proposition E.1 gives a sufficient condition in terms of the residual gradient energy.

Assumption D.6 (Quasi-static preconditioned dynamics)**.** Over the horizon relevant for stability: (i) $\mathcal{L}_{\mathrm{SFT}}$ is represented by its quadratic model $\mathcal{L}_{\mathrm{SFT}}(\theta_0+e) = \mathcal{L}_{\mathrm{SFT}}(\theta_0)+g_0^\top e+\frac{1}{2}e^\top He$; (ii) the Adam preconditioner is frozen at $D := \mathrm{diag}(\sqrt{\hat{v}_{t_0}}) + \epsilon I \succ 0$, computed from the unprojected gradient and hence independent of r; and (iii) momentum is neglected ($\beta_1 = 0$), except in Remark E.1.

The initialization θ_0 need not be stationary: the nonzero gradient g_0 is precisely the signal that SPECTRA uses. Under (i)–(ii), T_r is affine, so the common drift cancels in the difference of two trajectories and the recursion of Lemma 4.1 is exact. The frozen-preconditioner model is the standard setting for edge-of-stability analyses of adaptive methods (Cohen et al., 2023).

Assumption D.7 (Stable bottleneck offset)**.** (i) All modules share a common rank r, scaling $s(r)$, and transfer constants $(\kappa(M), p)$ of Assumption 4.1, and the per-module predictors share (c, b_β):

$$\log_{10}\eta^\star_{\mathrm{spec},\ell} = a_{0,\ell}(M) - c\log_{10} r - b_\beta \beta_\ell(\mathcal{D}; r) + \mathcal{O}\big(\|\Delta\|^2\big).$$

(ii) Define the bottleneck gap

$$\delta(M, \mathcal{D}, r) := \min_\ell \big[a_{0,\ell}(M) - b_\beta\beta_\ell(\mathcal{D}; r)\big] - \big[\bar{a}_0(M) - b_\beta\bar{\beta}(\mathcal{D}; r)\big], \qquad \bar{a}_0(M) := \frac{1}{L}\sum_{\ell=1}^{L} a_{0,\ell}(M).$$

There exist $\bar{\delta}(M)$ and $\varepsilon_\Delta \geq 0$ such that $|\delta(M, \mathcal{D}, r) - \bar{\delta}(M)| \leq \varepsilon_\Delta$ for all $(\mathcal{D}, r)$ considered.

Assumption D.8 (Model-invariant slopes)**.** For the model family $\mathcal{M}$ considered, the coefficients (c, b_β) of Proposition 4.6 are shared across $M \in \mathcal{M}$ up to an error that is small relative to the evaluation tolerance, whereas the intercept $a_0(M)$ is model-specific.

The local theory does not imply Assumption D.8: c and b_β are derivatives at a model-specific reference point. Appendix J reports that the fitted b_β differs between two models by $1.12\times$ at $\alpha = r$ and by up to $1.50\times$ in the clipping-dominated regime.

E Additional Theory

E.1 Estimation of the Local Spectral-Decay Exponent

For module ℓ and calibration example x, let $\sigma_i^{(\ell)}(x)$ denote the ith singular value of the initialization-time teacher-forced gradient. Over the local spectral window associated with target rank r, we model the singular-value spectrum as

$$\log \sigma_i^{(\ell)}(x) = a_\ell(x; r) - \beta_\ell(x; r)\log i + \varepsilon_i. \tag{E.1}$$

Equivalently, the local spectral-decay exponent can be interpreted as

$$\beta_\ell(x; r) \approx -\frac{\Delta \log \sigma_i^{(\ell)}(x)}{\Delta \log i}.$$

The window must lie within the numerical rank of $G^{(\ell)}(x)$, which is at most N_ℓ and at most the number of supervised tokens of x; slopes are not evaluated beyond it. We compute this local slope

independently for each calibration example and average across the calibration set:

$$\beta_\ell(\mathcal{D}_0; r) := \frac{1}{|\mathcal{D}_0|} \sum_{x \in \mathcal{D}_0} \beta_\ell(x; r).$$

Larger values of $\beta_\ell(\mathcal{D}_0; r)$ therefore correspond to more rapidly decaying local singular-value spectra. Because β depends only on the relative variation of singular values across spectral index, a global rescaling $\sigma_i \mapsto k\sigma_i$, $k > 0$, changes only the intercept in Equation (E.1) and leaves β unchanged. Hence β is compatible with the scale invariance of Proposition 4.5.

E.2 Scaling Transfer in Two Limiting Regimes

Lemma E.1 (Scaling transfer in two limiting regimes)**.** Let the objective depend on the LoRA factors only through $W = W_0 + s\,BA$, and train both factors with Adam without weight decay or gradient clipping, in the $\epsilon \to 0$ limit (Assumption D.1). Fix $k > 0$.
(a) Balanced regime. The runs (A_0, B_0, s, η) and $(k^{-1/2}A_0,\ k^{-1/2}B_0,\ ks,\ k^{-1/2}\eta)$ generate identical effective-weight trajectories $\{W_t\}_{t\geq 0}$.
(b) One-factor regime. If $A_t \equiv A_0$, the runs (A_0, B_0, s, η) and $(A_0,\ k^{-1}B_0,\ ks,\ k^{-1}\eta)$ generate identical trajectories $\{W_t\}_{t\geq 0}$.
Consequently, at fixed weight-space dynamics, the optimizer-level stability boundary satisfies $\eta^\star_{\text{spec}}(ks) = k^{-p}\,\eta^\star_{\text{spec}}(s)$, with $p = \frac{1}{2}$ in (a) and $p = 1$ in (b).

Proof of Lemma E.1. *(a)* Denote the second run by primes and suppose that $A'_\tau = k^{-1/2}A_\tau$ and $B'_\tau = k^{-1/2}B_\tau$ for all $\tau \leq t$; this holds at $t = 0$ by construction. Then $s'B'_\tau A'_\tau = ks \cdot k^{-1}B_\tau A_\tau = sB_\tau A_\tau$, so $W'_\tau = W_\tau$ and the upstream gradient $G_\tau := \partial\mathcal{L}/\partial W|_{W_\tau}$ is common to both runs. By the chain rule,

$$\nabla_{A'}\mathcal{L} = s'B'^{\top}_\tau G_\tau = k^{1/2}\,sB^\top_\tau G_\tau = k^{1/2}\nabla_A\mathcal{L}, \qquad \nabla_{B'}\mathcal{L} = s'G_\tau A'^{\top}_\tau = k^{1/2}\nabla_B\mathcal{L}.$$

Thus the entire gradient history of each factor is rescaled by the same factor $k^{1/2}$, and by Proposition 4.5(i) the normalized Adam directions coincide, $u'^A_t = u^A_t$ and $u'^B_t = u^B_t$. Hence $A'_{t+1} = A'_t - k^{-1/2}\eta\,u^A_t = k^{-1/2}(A_t - \eta u^A_t) = k^{-1/2}A_{t+1}$, and likewise $B'_{t+1} = k^{-1/2}B_{t+1}$, which completes the induction. For $B_0 = 0$, only A_0 is rescaled.

(b) With $A_t \equiv A_0$, suppose that $B'_\tau = k^{-1}B_\tau$ for $\tau \leq t$. Then $s'B'_\tau A_0 = sB_\tau A_0$, so $W'_\tau = W_\tau$, and $\nabla_{B'}\mathcal{L} = s'G_\tau A_0^\top = k\nabla_B\mathcal{L}$. By Proposition 4.5(i), $u'^B_t = u^B_t$, hence $B'_{t+1} = k^{-1}B_t - k^{-1}\eta\,u^B_t = k^{-1}B_{t+1}$.

In both cases the two runs generate the same $\{W_t\}$ and are therefore simultaneously stable or unstable. Mapping $\eta \mapsto k^{-p}\eta$ with the respective p gives $\eta^\star_{\text{spec}}(ks) = k^{-p}\eta^\star_{\text{spec}}(s)$.

Two remarks delimit the lemma. First, decoupled weight decay preserves both statements if the product $\eta\lambda_{\text{wd}}$ is held fixed, whereas gradient clipping breaks Proposition 4.5(i) and hence both statements (Appendix J). Second, standard LoRA keeps A_0 fixed as s varies, so it satisfies (a) only approximately and (b) only during the first steps after $B_0 = 0$. This motivates the effective exponent $p \in [\frac{1}{2}, 1]$ of Assumption 4.1, which Appendix L estimates directly.

E.3 Effective LoRA Learning Rate

The stability analysis is formulated for the effective update, whereas the optimizer operates on the LoRA factors (A, B). We therefore distinguish the weight-space boundary $\eta^\star_W$ of the spectral surrogate from the optimizer-level boundary $\eta^\star_{\text{spec}}$. The full mapping between them depends on the LoRA scaling $s = s_0 r^{-\gamma}$, the factor initialization, and Adam preconditioning, and is not assumed to be exact.

We make only its scaling dependence explicit, through Assumption 4.1, whose exponent is exact in the two limiting regimes of Lemma E.1. The rank coefficient then decomposes as $c = c_W + q_\alpha$ (Corollary 4.1), where $c_W \geq 0$ is the weight-space curvature effect of Proposition 4.2 and $q_\alpha = -p\gamma$

is the parameterization effect. Rank-independent effects of initialization, preconditioning, and momentum, together with $\kappa(M)$ and s_0, are absorbed into the intercept $a_0(M)$, while b_β captures the local dependence on gradient-spectrum decay. The first-order predictor form is generic; theory fixes the sign of b_β (conditional on Assumption D.5) and the lower bound $c \geq -p\gamma$ (conditional on Assumption 4.1), while calibration determines the coefficient values.

This decomposition explains why the optimal learning rates in Table 15 increase with rank at small r despite Proposition 4.2. The main experiments use constant $\alpha = 32$ ($\gamma = 1$), so $q_\alpha = -p \in [-1, -\frac{1}{2}]$ and a negative fitted c is admissible, with $c \geq -1$ in general and $c \geq -\frac{1}{2}$ in the balanced regime. Moreover, because $\beta(\mathcal{D}; r)$ varies with r, the rank slope observed in Table 15 is $-c - b_\beta \, \partial\beta/\partial \log_{10} r$ rather than $-c$.

E.4 Stability Boundary with Momentum

Remark E.1 (Momentum)**.** With exponential-moving-average momentum $m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$ and update $\Delta\theta_t = -\eta D^{-1} m_t$, neglecting bias correction (which is exact asymptotically in t), the restricted dynamics of Lemma 4.1 are stable if and only if

$$0 < \eta < \frac{1 + \beta_1}{1 - \beta_1} \, \eta_W^\star(r) = \frac{2(1 + \beta_1)}{(1 - \beta_1) \, \lambda_{\max}(\widetilde{H}_r)}.$$

For $\beta_1 = 0.9$, the constant is 38 rather than 2 (Cohen et al., 2023). The factor $(1 + \beta_1)/(1 - \beta_1)$ does not depend on (r, β) and is therefore absorbed into $a_0(M)$. Proposition 4.4, in contrast, is stated for $\beta_1 = 0$.

Proof of Remark E.1. Along an eigenvector w_j of $\widetilde{H}_r$, let $y_t := w_j^\top \tilde{e}_t$. In whitened coordinates, the recursion reads $y_{t+1} = y_t - \eta \tilde{m}_t$ with $\tilde{m}_t = \beta_1 \tilde{m}_{t-1} + (1 - \beta_1) \mu_j y_t$. Eliminating $\tilde{m}$ through $\tilde{m}_{t-1} = -(y_t - y_{t-1})/\eta$ gives

$$y_{t+1} = (1 + \beta_1 - h) \, y_t - \beta_1 y_{t-1}, \qquad h := \eta(1 - \beta_1)\mu_j,$$

with characteristic polynomial $z^2 + a_1 z + a_0$, where $a_1 = -(1 + \beta_1 - h)$ and $a_0 = \beta_1$. By the Jury (Schur–Cohn) criterion, both roots lie strictly inside the unit disk if and only if $|a_0| < 1$ and $|a_1| < 1 + a_0$. The first condition holds for $\beta_1 \in [0, 1)$; the second reads $|1 + \beta_1 - h| < 1 + \beta_1$, i.e., $0 < h < 2(1 + \beta_1)$. Requiring this for every j gives the claim.

E.5 From the Stability Boundary to the Training Learning Rate

The predictor targets the contraction-optimal rate rather than the boundary itself. By Proposition 4.4,

$$\log_{10} \eta_{\text{opt}}^{\text{lin}} = \log_{10} \eta_W^\star - \log_{10}(1 + \omega_r), \qquad \log_{10}(1 + \omega_r) \in (0, \log_{10} 2].$$

Writing $\log_{10}(1 + \omega_r) = \frac{1}{2} \log_{10} 2 + \xi_r$ with $|\xi_r| \leq \frac{1}{2} \log_{10} 2$, the constant is absorbed into $a_0(M)$ and ξ_r is a bounded residual. If, in addition, $\omega_r \in [\omega_{\text{lo}}, \omega_{\text{hi}}]$ across the configurations of a model, absorbing the midpoint of $[\log_{10}(1 + \omega_{\text{lo}}), \log_{10}(1 + \omega_{\text{hi}})]$ reduces the residual to half the width of this interval. A further safety factor $\zeta \in (0, 1)$ applied at training time, $\widehat{\eta} = \zeta \, \widehat{\eta}^\star$, likewise shifts only the intercept. The predictor form is therefore preserved:

$$\boxed{\log_{10} \widehat{\eta} = a_0(M) - c \log_{10} r - b_\beta \, \bar{\beta}(\mathcal{D}; r)}.$$

E.6 A Sufficient Condition for Spectral-Decay–Curvature Monotonicity

Proposition E.1 (A sufficient condition for Assumption D.5)**.** Fix a module and a calibration example, and let $R(k) := \sum_{i>k} \sigma_i^2$ denote the residual gradient energy, so that $R(0) = \|G\|_F^2$. Suppose that (i) (trace-normalized Fisher proxy) $\lambda_{\max}(\widetilde{H}_r) = C(M, \ell) \, \lambda_{\max}\big(F|_{\mathcal{S}_r}\big) / \operatorname{tr} F$, where $F := \operatorname{vec}(G) \operatorname{vec}(G)^\top$ is the empirical Fisher of the example and $C(M, \ell) > 0$ does not depend on

(β, r); and (ii) (power-law spectrum) $\sigma_i = \sigma_1 i^{-\beta}$ for $1 \le i \le N$. Then

$$\lambda_{\max}(\widetilde{H}_r) = C(M, \ell)\left(1 - \frac{R(r)}{R(0)}\right),$$

and $\partial_\beta \lambda_{\max}(\widetilde{H}_r) > 0$ for every $r < N$. Hence Assumption D.5 holds.

Proof of Proposition E.1. The matrices $\{u_i v_i^\top\}_{i \le r}$ are orthonormal in the Frobenius inner product, so every $X \in \mathcal{S}_r$ can be written as $X = \sum_{i \le r} c_i u_i v_i^\top$ with $\|X\|_F = \|c\|_2$. Then

$$\operatorname{vec}(X)^\top F \operatorname{vec}(X) = \langle G, X \rangle_F^2 = \Big(\sum_{i \le r} c_i \sigma_i\Big)^2,$$

which, by the Cauchy–Schwarz inequality, is maximized over $\|c\|_2 = 1$ at $c \propto (\sigma_1, \dots, \sigma_r)$. Hence $\lambda_{\max}(F|_{\mathcal{S}_r}) = \sum_{i \le r} \sigma_i^2 = R(0) - R(r)$. Since $\operatorname{tr} F = \|G\|_F^2 = R(0)$, condition (i) yields the displayed identity.

Under (ii), $1 - R(r)/R(0) = \Phi_r(\beta) := Z_r(\beta)/Z_N(\beta)$ with $Z_k(\beta) := \sum_{i \le k} i^{-2\beta}$; the factor σ_1 cancels. Let $\mathbb{E}_k$ denote expectation under $\pi_k(i) \propto i^{-2\beta}$ on $\{1, \dots, k\}$, and $\mathbb{E}_{>r}$ expectation under the same weights restricted to $\{r+1, \dots, N\}$. Since $\partial_\beta \log Z_k = -2\,\mathbb{E}_k[\log i]$ and $\mathbb{E}_N = \Phi_r \mathbb{E}_r + (1 - \Phi_r)\mathbb{E}_{>r}$,

$$\partial_\beta \log \Phi_r = 2\big(\mathbb{E}_N[\log i] - \mathbb{E}_r[\log i]\big) = 2\big(1 - \Phi_r\big)\big(\mathbb{E}_{>r}[\log i] - \mathbb{E}_r[\log i]\big).$$

For $r < N$, $1 - \Phi_r > 0$ and $\mathbb{E}_{>r}[\log i] > \log r \ge \mathbb{E}_r[\log i]$, so $\partial_\beta \log \Phi_r > 0$.

Condition (i) normalizes the Fisher proxy by its trace, so that the proxy inherits the scale invariance of Proposition 4.5(ii). The proposition identifies the captured-energy fraction $1 - R(r)/R(0)$ as the curvature-relevant statistic, of which β is a one-parameter summary under (ii), and thereby connects the spectral-decay feature to the residual-energy curves used in the pipeline overview.

E.7 Learning-Rate Grid Resolution

For each model–dataset–rank configuration, let

$$\eta^\star \in \arg\min_{\eta \in [\eta_{\min}, \eta_{\max}]} \mathcal{L}_{\text{eval}}(\eta)$$

denote the optimal learning rate. We evaluate learning rates on a geometric grid

$$\mathcal{G}_m = \{\eta_{\min} m^j\}_{j \ge 0} \cap [\eta_{\min}, \eta_{\max}], \qquad m > 1,$$

with the interval endpoints included explicitly. Let $\eta_j \le \eta^\star \le \eta_{j+1}$ denote the bracketing grid points and $\rho := \eta_{j+1}/\eta_j \le m$.

Proposition E.2 (Existence of a nearby grid point)**.** For any $\eta^\star \in [\eta_{\min}, \eta_{\max}]$, there exists an evaluated learning rate $\widetilde{\eta} \in \mathcal{G}_m$ such that

$$\frac{1}{\sqrt{m}} \le \frac{\widetilde{\eta}}{\eta^\star} \le \sqrt{m}, \qquad \text{equivalently} \qquad |\log \widetilde{\eta} - \log \eta^\star| \le \frac{1}{2} \log m.$$

Proof. Write $\eta^\star = c\,\eta_j$ with $1 \le c \le \rho$. The multiplicative distances from $\eta^\star$ to the two bracketing points are $\eta^\star/\eta_j = c$ and $\eta_{j+1}/\eta^\star = \rho/c$, so the nearer point has multiplicative error $\min\{c, \rho/c\}$. This is maximized when $c = \rho/c$, i.e., $c = \sqrt{\rho}$. Therefore, the nearer bracketing point $\widetilde{\eta}$ satisfies $\max\{\widetilde{\eta}/\eta^\star, \eta^\star/\widetilde{\eta}\} \le \sqrt{\rho} \le \sqrt{m}$, and taking logarithms yields the equivalent form. □ □

Table 5: Model checkpoints used in our experiments. We fine-tune the base checkpoints with LoRA and use the corresponding instruction-tuned checkpoints only to obtain their chat templates.

Model	Base checkpoint for LoRA SFT	Chat-template source
Llama 2 7B	`meta-llama/Llama-2-7b-hf`	`meta-llama/Llama-2-7b-chat-hf`
Llama 3.1 8B	`meta-llama/Llama-3.1-8B`	`meta-llama/Llama-3.1-8B-Instruct`
Qwen3 1.7B	`Qwen/Qwen3-1.7B-Base`	`Qwen/Qwen3-1.7B`
Qwen3 8B	`Qwen/Qwen3-8B-Base`	`Qwen/Qwen3-8B`
Gemma 3 1B	`google/gemma-3-1b-pt`	`google/gemma-3-1b-it`

Proposition E.2 guarantees that a nearby grid point exists, but the sweep selects the grid point with the lowest evaluation loss, which need not be the nearer one. The following result bounds the selected point.

Proposition E.3 (Resolution of the selected grid point)**.** Let $\widehat{\eta} \in \arg\min_{\eta \in \mathcal{G}_m} \mathcal{L}_{\text{eval}}(\eta)$ denote the selected grid point. (i) If $\mathcal{L}_{\text{eval}}$ is strictly decreasing on $[\eta_{\min}, \eta^\star]$ and strictly increasing on $[\eta^\star, \eta_{\max}]$, then $\widehat{\eta} \in \{\eta_j, \eta_{j+1}\}$ and $\max\{\widehat{\eta}/\eta^\star, \eta^\star/\widehat{\eta}\} \le \rho \le m$. (ii) If, in addition, $\mathcal{L}_{\text{eval}}(\eta) = \phi\big(|\log\eta - \log\eta^\star|\big)$ on $[\eta_j, \eta_{j+1}]$ for some strictly increasing ϕ, e.g., if $\mathcal{L}_{\text{eval}}$ is locally quadratic in $\log\eta$, then $\widehat{\eta}$ is the nearer bracketing point and $\max\{\widehat{\eta}/\eta^\star, \eta^\star/\widehat{\eta}\} \le \sqrt{\rho} \le \sqrt{m}$.

Proof. (i) Among grid points in $[\eta_{\min}, \eta^\star]$, strict monotonicity implies that η_j has the lowest loss; among grid points in $[\eta^\star, \eta_{\max}]$, η_{j+1} does. Hence $\widehat{\eta} \in \{\eta_j, \eta_{j+1}\}$. Both points lie within a factor ρ of $\eta^\star$, since $\eta_j \le \eta^\star \le \eta_{j+1} = \rho\,\eta_j$. (ii) Under the stated form, the bracketing point with the smaller log-distance to $\eta^\star$ has the smaller loss, so $\widehat{\eta}$ is the nearer point, and the bound follows from Proposition E.2. □ □

For the coarse sweep ($m = 2$), the selected learning rate therefore lies within a factor of 2 of the continuous optimum under unimodality, and within $\sqrt{2} \approx 1.41$ under local log-symmetry. For the fine sweep ($m = 1.44$), the corresponding factors are 1.44 and $\sqrt{1.44} = 1.2$. The refinement range $[\min(\eta_2^\star, \eta_{256}^\star)/1.4,\ 1.4\max(\eta_2^\star, \eta_{256}^\star)]$ and the $2.4\times = 2 \times 1.2$ scoring tolerance rely on the log-symmetric case (ii); under unimodality alone, the corresponding values are $[\,\cdot/2,\ 2\cdot\,]$ and $2 \times 1.44 = 2.88\times$. The loss curves in Figure 3 rise more steeply above the optimum than below it, in which case the selected point is biased toward the lower neighbor; case (i) covers this situation. Both statements concern the noiseless loss; evaluation noise across seeds adds to these bounds.

F Model Checkpoints

We conduct all LoRA SFT experiments on the pretrained base checkpoints listed in Table 5. For data formatting, we use the chat template provided by the corresponding instruction-tuned model of each model family; the instruction checkpoints are used only to define the training template and are not used as initialization for LoRA fine-tuning.

G Additional Analysis of LoRA Learning-Rate Stability

To assess whether learning-rate sensitivity generalizes beyond the main-model results, we extend the same sweep analysis to four additional models spanning substantially different architectures and scales: Llama-2-7B, Llama-3.1-8B, Qwen3-8B-Base, and Gemma-3-1B-PT. As shown in Figures 4 to 7, all four models exhibit a consistent pattern: test loss depends non-monotonically on the learning rate, while the empirical optimum varies across datasets and LoRA configurations. In several settings, modestly increasing the learning rate beyond the optimal region causes sharp degradation or instability, particularly on OpenR1-Math-220K and OpenThoughts3. These observations persist from 1B- to 8B-scale models, indicating that learning-rate sensitivity is not specific to a particular architecture or model size. Together with the main results, this analysis reinforces the need for configuration-aware learning-rate prediction rather than relying on a single default value or repeated exhaustive sweeps.

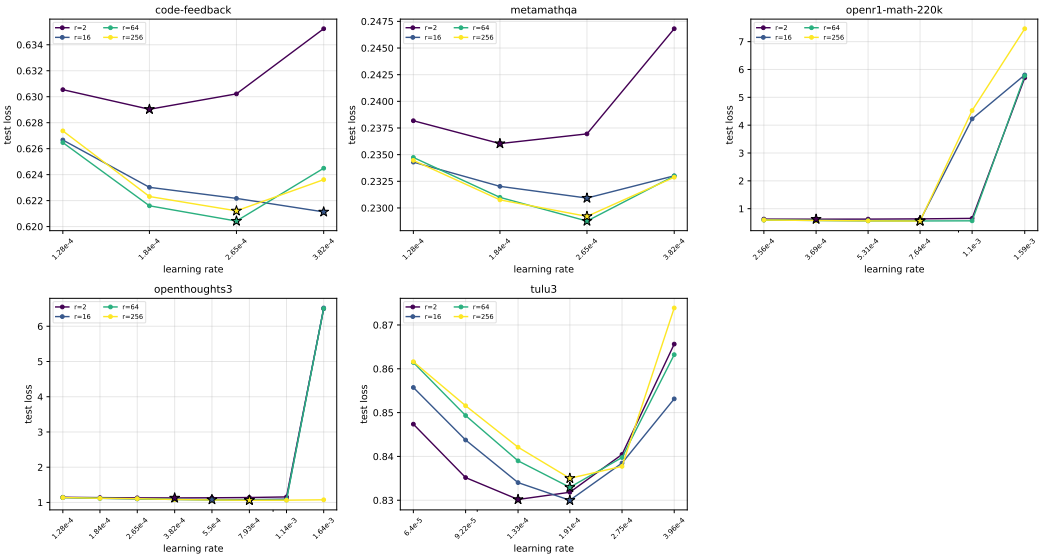

Figure 4: **Learning-rate sensitivity of LoRA on Llama-3.1-8B.**

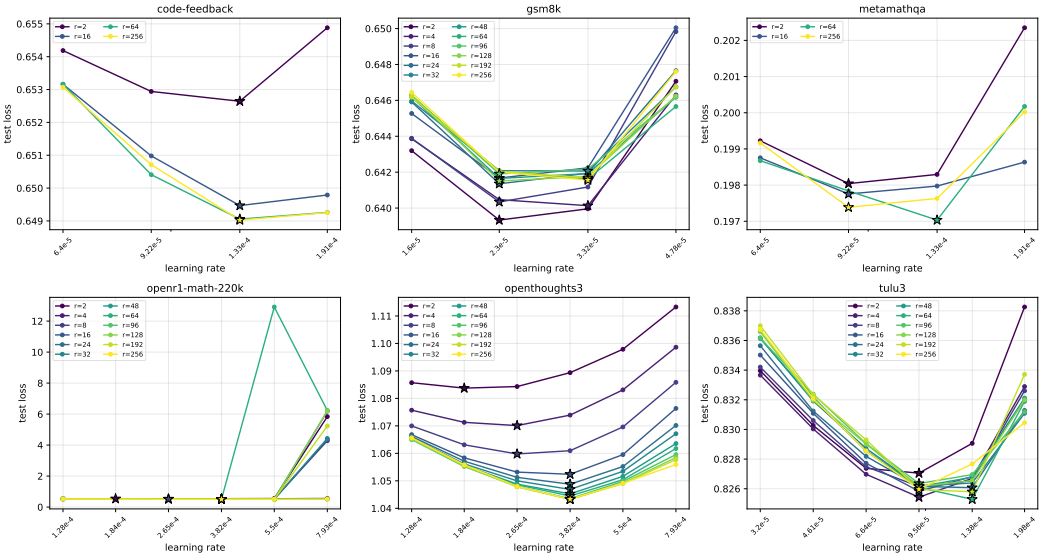

Figure 5: **Learning-rate sensitivity of LoRA on Llama-2-7B.**

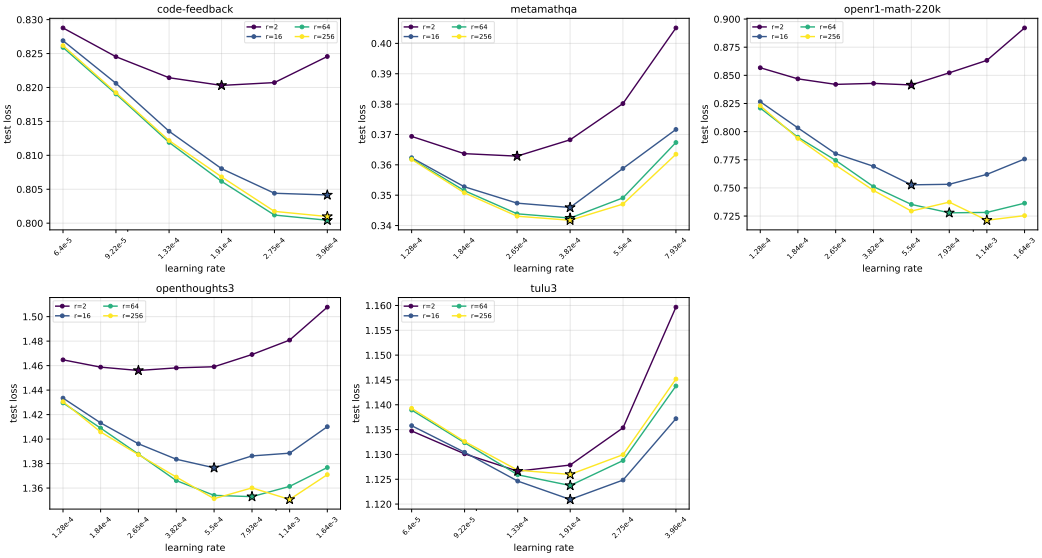

Figure 6: **Learning-rate sensitivity of LoRA on Gemma-3-1B-PT.**

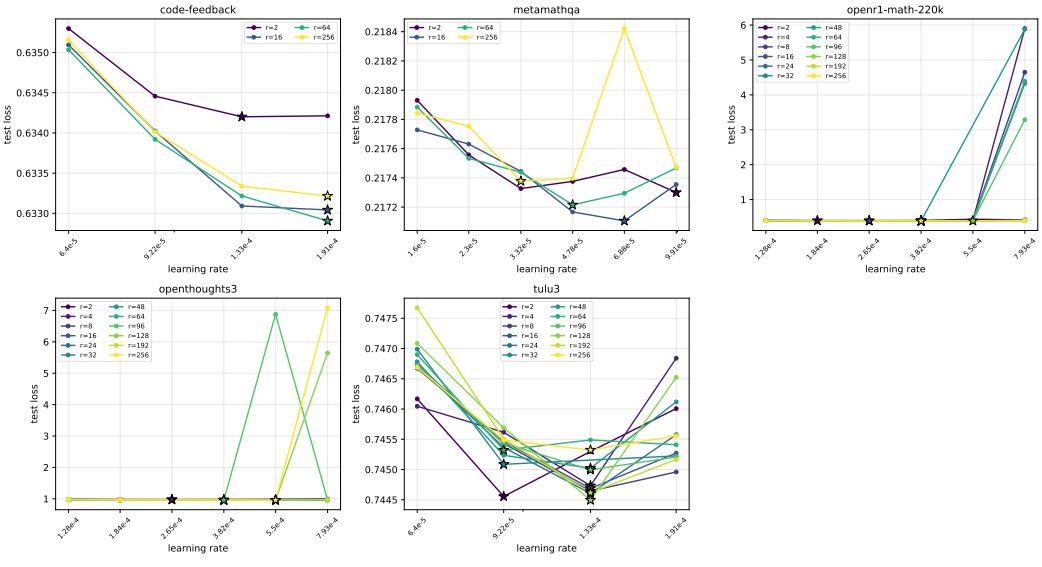

Figure 7: **Learning-rate sensitivity of LoRA on Qwen3-8B-Base.**

H EXPERIMENT SYSTEM AND IMPLEMENT SETTINGS

H.1 COMPUTATIONAL RESOURCES

We perform all experiments using 96 GPUs with 80GB of memory and a 208-core Intel(R) Xeon(R) Platinum 8481C CPU operating at 2.70GHz. Our code is developed in PyTorch and utilizes the Hugging Face Transformer Library for experimental execution. When formatting training examples with each model family's chat template, we use the default system prompt provided by the official source.

H.2 HYPERPARAMETERS

We present the hyperparameters used in the fine-tuning stage for each model. We use **AdamW** (Loshchilov & Hutter, 2019) as the optimizer. Most other hyperparameters are kept consistent across all models and datasets, including a batch size of 32 during deployment and 32 during training. In training, we also use gradient accumulation with 16 steps and set the weight decay to 0.01 for all training runs. In SPECTRA, we set the maximum training steps to 1,000. All supervised fine-tuning training are conducted using mixed precision with bfloat16.

H.3 THE GRID-SEARCH BASELINE

Grid search is the conventional practice we compare against: run a learning-rate sweep on the target model and dataset, then commit to a single learning rate for all ranks. It is the strongest baseline in our comparison, and also the most expensive — it is the search cost that SPECTRA is designed to replace.

Candidate grid. All learning rates are selected from a fixed logarithmic grid $\mathcal{G}$ of 24 points spanning 2.0×10^{-6} to 1.1×10^{-3}, uniformly spaced in $\log_{10}$ with step $\Delta = \frac{1}{23}\log_{10} 550 = 0.119$ decades:

$$\begin{aligned}\mathcal{G} = &\{2.00,\ 2.63,\ 3.45,\ 4.53,\ 5.94,\ 7.78\} \times 10^{-6} \\ &\cup \{1.02,\ 1.34,\ 1.76,\ 2.30,\ 3.02,\ 3.96,\ 5.19,\ 6.81,\ 8.93\} \times 10^{-5} \\ &\cup \{1.17,\ 1.54,\ 2.02,\ 2.64,\ 3.47,\ 4.55,\ 5.96,\ 7.82\} \times 10^{-4} \ \cup\ \{1.03,\ 1.10\} \times 10^{-3}.\end{aligned}$$

The grid brackets the empirical optima of the constant-α convention (2.0×10^{-6} to 1.1×10^{-3}) and of the rank-dependent conventions of Table 9 (8.3×10^{-7} to 1.6×10^{-3}) to within one grid step at either end, and its spacing is finer than the $2.4\times$ (0.380 decade) scoring tolerance by a factor of 3.2, so grid discretisation is never the binding constraint: in every setting reported, the selected point lies strictly inside $\mathcal{G}$.

Selection rule. Grid search selects a *single* learning rate for each $(\text{model}, \text{dataset})$ pair, shared across all ranks. By default, we perform the search at rank $r = 16$, yielding $\eta_{\text{grid}}(M, \mathcal{D})$. This learning rate is then applied to every rank within the same pair and evaluated using Hit@$2.4\times$, following the same scoring protocol as all other methods.

I ADDITIONAL EXPERIMENTS

I.1 SINGLE-POINT CALIBRATION OF a_0

We next examine how much target-model data are required to calibrate the model-specific intercept a_0. We fit (b, c) on each source model and calibrate a_0 on each of the remaining four target models using either a single dataset–rank point or one complete dataset. As shown in Table 6, single-dataset calibration consistently improves Hit@$2.4\times$ over single-point calibration, increasing the average from 83.9% to 88.5%. This gain suggests that averaging across ranks reduces the discretization noise introduced by the learning-rate sweep, while requiring only one dataset for target-model calibration.

Table 6: **Calibration efficiency of a_0.** Hit@2.4× (%) using either a single dataset–rank point or one complete dataset to calibrate a_0 on the target model. (c, b) are fit on the source model; values are averaged over the four target models outside the source. One complete dataset consistently outperforms a single point, since averaging over ranks removes the discretization noise of the sweep grid.

(b, c) source	Single point	Single dataset
Gemma	84.7	**90.4**
Llama2	83.5	**86.6**
Llama-3.1	85.6	**87.8**
Qwen3-1.7B	83.4	**86.9**
Qwen3-8B	82.3	**90.9**
Average	83.9	**88.5**

Table 7: **Calibration efficiency of (c, b_β).** Slopes are fit on subsets of a single source model's clean (dataset, rank) points and transferred to the four remaining models. What matters is dataset coverage rather than raw sample count: three points per dataset already recovers all but 1.4 points of full calibration, whereas three randomly drawn points are far from sufficient.

Calibration subset	Hit@2.4×
Random 3 points	50.9%
Random 6 points	80.0%
1 point / dataset	75.8%
3 points / dataset	87.1%
All points	**88.5%**

I.2 Calibration Data Efficiency

We study how many calibration points are required to reliably estimate (c, b_β) under the cross-model transfer protocol of Table 2: slopes are fit on a subset of one source model's clean (dataset, rank) points and transferred to the four remaining models, averaged over source models and random draws. As shown in Table 7, randomly selecting three points or using only one point per dataset yields substantially lower Hit@2.4× accuracy. Using three points per dataset improves performance to 87.1%, approaching the 88.5% achieved with all points. These results suggest that at least three calibration points per dataset are needed for reliable estimation, while full-grid calibration remains the most accurate when calibration cost is not constrained.

I.3 Ablation of the Predictor Terms

We ablate the two terms of the predictor $\log_{10} \hat{\eta} = a_0(M) - c \log_{10} r - b_\beta \beta$ under the cross-model transfer protocol of Table 2. Table 8 reports src-mean Hit@2.4×. Removing β lowers the average from 88.5% to 54.2%, no better than calibrating a_0 alone (54.1%); keeping β without the rank term retains 63.9%. On the constant-α sweep, 77% of the variance of $\log_{10} \eta^\star$ lies across datasets within a model and 12% across ranks, so the cross-dataset part is what β has to explain. In a pooled in-sample fit over all 172 cells of the five models (one intercept per model, shared c and b_β), the partial residual $\log_{10} \eta^\star - a_0(M) + c \log_{10} r$ is explained by β with $R^2 = 0.73$.

J Why $\alpha = 32r$ Breaks the Predictor

In this section, we investigate why SPECTRA degrades under the aggressive $\alpha = 32r$ scaling. We find that the failure is not caused by the linear rank dependence of α itself. Instead, when $\alpha = xr$, increasing the multiplier x directly increases the absolute adapter scale $\gamma = \alpha/r = x$, which substantially increases gradient clipping and eventually moves training into a clipping-dominated optimization regime. Because the learning-rate scaling modeled by SPECTRA is derived from the underlying, effectively unclipped optimization dynamics, pervasive clipping introduces an additional

Table 8: **Ablation of the predictor terms.** Hit@2.4× (%) under the cross-model transfer protocol of Table 2, src-mean over the four remaining source models. Each variant drops one term of the predictor $a_0(M) - c \log_{10} r - b_\beta \beta$ before fitting the slopes on the source model.

i

Variant	Gemma	Llama2	Llama3.1	Qwen3-1.7B	Qwen3-8B	Average
SPECTRA (all terms)	**89.1**	**83.4**	**92.6**	**98.2**	**79.3**	**88.5**
w/o β	61.3	53.4	46.3	65.6	44.6	54.2
w/o rank term	<u>64.7</u>	<u>67.5</u>	<u>60.2</u>	<u>75.9</u>	<u>51.4</u>	<u>63.9</u>
a_0 only	61.3	58.8	42.0	60.2	48.3	54.1

constraint on the realized update and weakens the correspondence between the predicted and empirical optimal learning rates.

Table 9: **Learning-rate selection success rates under constant α/r scaling.** We compare SPECTRA with five baselines on Gemma and Qwen3-1.7B under three scaling rules of the same form $\alpha = xr$, with $x \in \{1, 8, 32\}$ and therefore $\alpha/r \in \{1, 8, 32\}$. Values report Hit@2.4×, the percentage of configurations for which the selected learning rate lies within a factor of 2.4 of the empirical optimum. Best and second-best results are shown in **bold** and <u>underlined</u>, respectively. Averaged across all six model–scaling settings, SPECTRA achieves the highest success rate of **76.4%**, outperforming the strongest baseline by **17.2 percentage points**; SPECTRA remains the best method at every x (by 19.4, 20.6 and 11.9 points), as a larger scaling factor spreads the optima over a wider range that no single learning rate per dataset can cover.

α/r	Model	1×10^{-4}	2×10^{-4}	4×10^{-4}	TM Formula	Grid	SPECTRA
1	Gemma	20.0	55.0	70.0	70.0	<u>80.0</u>	**91.2**
	Qwen3-1.7B	40.0	60.0	65.0	60.0	<u>70.0</u>	**97.5**
8	Gemma	50.0	50.0	30.0	30.0	<u>60.0</u>	**76.2**
	Qwen3-1.7B	50.0	50.0	30.0	20.0	<u>50.0</u>	**75.0**
32	Gemma	40.0	30.0	10.0	5.0	<u>45.0</u>	**63.7**
	Qwen3-1.7B	30.0	20.0	10.0	5.0	<u>50.0</u>	**55.0**

The failure is controlled by the scale magnitude, not the rank dependence. Table 9 provides a controlled test because $\alpha = r$, $\alpha = 8r$, and $\alpha = 32r$ have exactly the same dependence on rank: in every case α grows linearly with r, so the scale exponent $q = d \log \gamma / d \log r$ is zero throughout. They differ only in the multiplicative constant x, or equivalently in the absolute LoRA scale $\gamma = \alpha/r = x$. Despite this identical rank dependence, the average Hit@2.4× of SPECTRA decreases systematically from 94.4% for $\alpha = r$ to 75.6% for $\alpha = 8r$ and 59.4% for $\alpha = 32r$. The degradation at $\alpha = 32r$ therefore cannot be attributed to an incorrect rank exponent; it emerges progressively as the absolute scaling magnitude increases.

Three measurements track this progression monotonically. First, the empirical optima spread over an increasingly wide range, from 1.23 decades at x=1 to 1.80–1.98 at x=8 and 2.17–2.47 at x=32, so an identical coefficient error translates into a proportionally larger multiplicative miss. Second, the fitted spectral coefficient b_β becomes increasingly model-dependent, differing between the two models by 1.12×, 1.35× and 1.50× respectively, which directly undermines the cross-model transfer that SPECTRA relies on. Third, gradient clipping saturates: the fraction of optimizer steps at which the pre-clip gradient norm exceeds the threshold rises from 3.6% at x=1 to 30.3% at x=8 and 69.1% at x=32, at which point 24 of 40 cells clip at every step. In that regime the update is renormalized at each step and the local stability boundary that SPECTRA predicts is no longer the quantity that limits training. The convention we calibrate on ($\alpha = 32$ fixed) triggers clipping on 1.8% of steps, so the main results of Table 2 are unaffected.

The baselines degrade in the same direction and more steeply: per-dataset grid search falls from 75.0% to 55.0% to 47.5%, and the best fixed learning rate from 67.5% to 50.0% to 35.0%. SPECTRA remains the strongest method at every value of x, leading the strongest baseline by 19.4, 20.6 and 11.9 points; the margin narrows at $x = 32$ because SPECTRA's own accuracy deteriorates in the clipping-dominated regime, but its predicted learning-rate structure remains more informative than any per-dataset constant.

Table 10: **Effect of LoRA scaling on gradient clipping and learning-rate prediction.** We report the average fraction of clipped optimization steps, the number of configurations for which all optimization steps are clipped, and the Hit@2.4× accuracy of SPECTRA on Gemma and Qwen3-1.7B. Larger absolute LoRA scaling $\gamma = \alpha/r$ is associated with substantially more clipping and lower prediction accuracy.

Scaling	$\gamma = \alpha/r$	Avg. Clip (%)	100% Saturated	SPECTRA (Gemma / Qwen)
$\alpha = 32$	$32/r$	1.8	0/36	90.0 / 100.0
$\alpha = r$	1	3.6	0/40	91.2 / 97.5
$\alpha = 128$	$128/r$	20.7	5/40	83.8 / 88.8
$\alpha = 32\sqrt{r}$	$32/\sqrt{r}$	23.5	7/48	86.2 / 93.8
$\alpha = 8r$	8	30.3	8/40	76.2 / 75.0
$\alpha = 32r$	32	69.1	24/40	63.7 / 55.0

Aggressive scaling drives training into a clipping-dominated regime. To identify what changes in the optimization dynamics as x increases, we measure the fraction of optimization steps affected by gradient clipping. The results are summarized in Table 10. As γ increases from 1 to 8 and then 32, the average fraction of clipped steps rises from only 3.6% to 30.3% and finally 69.1%. The degradation in learning-rate prediction follows the same ordering: SPECTRA decreases from $91.2/97.5\%$ on Gemma/Qwen3-1.7B under $\alpha = r$, to $76.2/75.0\%$ under $\alpha = 8r$, and to $63.7/55.0\%$ under $\alpha = 32r$. More strikingly, under $\alpha = 32r$, 24 of the 40 configurations have gradients clipped on every recorded optimization step. Thus, $\alpha = 32r$ is qualitatively different from the weakly clipped regime: for a majority of configurations, clipping is no longer an occasional safeguard but an active component of the optimization rule throughout training.

This behavior follows directly from the LoRA parameterization

$$\Delta W = \frac{\alpha}{r} BA = \gamma BA.$$

Increasing γ increases the scale with which changes in the LoRA factors are mapped into the effective weight update and correspondingly amplifies the gradients propagated to the adapter parameters. When these gradients exceed the clipping threshold, the optimizer no longer applies the raw gradient g_t, but instead operates on a rescaled gradient of the form

$$\widetilde{g}_t = g_t \min\left(1, \frac{\tau_{\text{clip}}}{\|g_t\|}\right).$$

where τ_{clip} denotes the clipping threshold. In the weakly clipped regime, $\widetilde{g}_t \approx g_t$ for most steps, and the realized optimization dynamics remain close to those characterized by the underlying gradient geometry. In the clipping-dominated regime, however, $\|g_t\| > \tau * \text{clip}$ occurs frequently, so the magnitude of the gradient entering the optimizer is explicitly constrained by τ_{clip}. Consequently, the effective training dynamics depend not only on the learning rate and the curvature captured by our predictor, but also on the clipping operation.

This distinction explains why $\alpha = r$ can be predicted accurately while $\alpha = 32r$ degrades despite the two conventions sharing the same rank-scaling exponent. The rank dependence modeled by SPECTRA is still present, but at sufficiently large γ it is partially obscured by a second optimization scale introduced by gradient clipping. The $\alpha = 8r$ experiment provides an important intermediate control: both its clipping rate and its prediction accuracy lie between those of $\alpha = r$ and $\alpha = 32r$. This smooth progression supports a transition driven by increasing scale magnitude rather than a discrete failure associated specifically with the functional form $\alpha \propto r$.

Finally, this failure mode does not affect the constant-$\alpha = 32$ setting used in our main analysis. Under $\alpha = 32$, the effective scale is $\gamma = 32/r$, which decreases with rank, and only 1.8% of optimization steps are clipped on average, with no configuration reaching 100% clipping saturation. The main experiments therefore remain in a weakly clipped regime in which the realized updates closely track the optimization dynamics assumed by SPECTRA. Together, these results identify $\alpha = 32r$ as an out-of-regime stress test: its poor transfer is primarily associated with the unusually large absolute

Table 11: **Learning-rate selection success rates under constant LoRA scaling factors.** We compare SPECTRA with five baselines on Gemma and Qwen3-1.7B under $\alpha \in \{32, 128\}$, corresponding to $\alpha/r \in \{32/r, 128/r\}$. Values report Hit@2.4×, the percentage of configurations for which the selected learning rate lies within a factor of 2.4 of the empirical optimum. Best and second-best results are shown in **bold** and underlined, respectively. Under these constant scaling factors the empirical optimum is nearly rank-invariant, so per-dataset grid search is close to saturated (96.3% on average); SPECTRA recovers 90.7% of that accuracy from a single calibration dataset, while every training-free baseline remains below 75%.

α/r	Model	1×10^{-4}	2×10^{-4}	4×10^{-4}	TM Formula	Grid	SPECTRA
$32/r$	Gemma	35.0	65.0	80.0	65.0	**95.0**	90.0
	Qwen3-1.7B	38.6	79.5	77.3	61.4	**100.0**	**100.0**
$128/r$	Gemma	70.0	80.0	40.0	30.0	**90.0**	83.8
	Qwen3-1.7B	80.0	75.0	35.0	35.0	**100.0**	88.8

adapter scale and the resulting clipping-dominated dynamics, rather than with a failure of the rank dependence captured by the predictor.

K Empirical Validation of SPECTRA Training Stability Across LoRA Alpha Scaling Settings

We further evaluate SPECTRA under the rank-dependent scaling rule $\alpha = 32\sqrt{r}$ using Gemma-3-1B-PT and Qwen3-1.7B-Base across five datasets and ranks $r \in \{16, 64, 128\}$. For each configuration, we train with learning rates equal to $\{0.5, 1, 2, 4\}\times$ the value predicted by SPECTRA. As shown in Figure 8, the predicted learning rate generally lies at or near the low-loss region, whereas larger multipliers frequently increase the evaluation loss, sometimes sharply. These results demonstrate that SPECTRA remains reliable under rank-dependent scaling, consistent with its behavior under constant α.

We further evaluate SPECTRA under constant LoRA scaling, $\alpha \in \{32, 128\}$, on Gemma and Qwen3-1.7B. SPECTRA achieves **90.7%** average Hit@2.4× and is the strongest *sweep-free* method across all four settings, exceeding the best fixed learning rate by **15.8 percentage points** (Table 11). Although grid search reaches 96.3%, SPECTRA recovers **94%** of its coverage using one calibration dataset per model at only **4.5%** of the search cost. When the scaling rule departs from constant α, this ordering reverses: grid search falls to **59.2%**, while SPECTRA leads by **17.2** points (Table 9). These results show that SPECTRA transfers across model families and scaling regimes with only target-specific calibration of a_0.

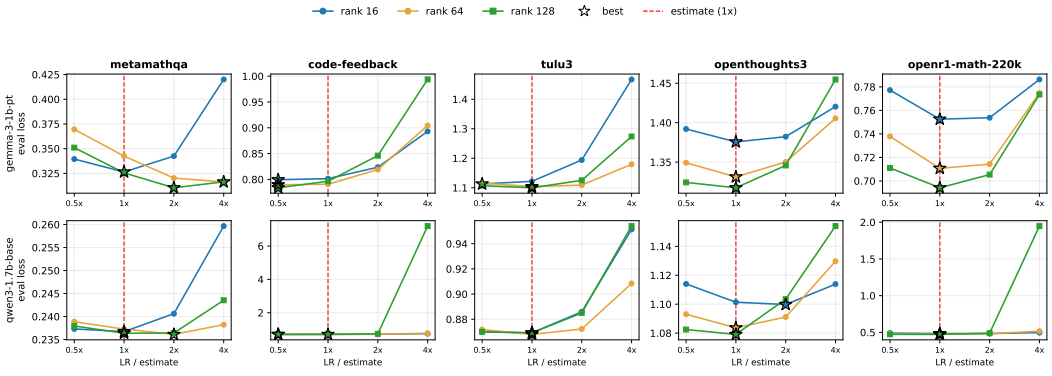

Figure 8: **Real-training validation under rank-dependent scaling $\alpha = 32\sqrt{r}$.** Evaluation loss as a function of the learning rate normalized by the SPECTRA prediction for Gemma-3-1B-PT (top) and Qwen3-1.7B-Base (bottom) across five datasets and ranks $r \in \{16, 64, 128\}$. The red dashed line denotes the predicted learning rate (1×), and stars mark the empirical optimum for each rank among the evaluated multipliers. The SPECTRA prediction generally falls near the low-loss region, while excessively large learning rates often lead to substantial degradation.

L A β-FREE TEST OF THE SCALING TRANSFER

Scaling transfer. We parameterize the LoRA scaling as $s = \alpha/r = s_0 r^{-\gamma}$, so that constant α, rsLoRA ($\alpha \propto \sqrt{r}$), and $\alpha \propto r$ correspond to $\gamma = 1, \frac{1}{2}, 0$. By Assumption 4.1, $\eta^\star_{\text{spec}} = \kappa(M) s^{-p} \eta^\star_W \propto r^{p\gamma}$, so the parameterization contributes $q_\alpha(\gamma) = -p\gamma$ to the rank coefficient and $c(\gamma) = c_W - p\gamma$ (Corollary 4.1). The exponent is exact in the two regimes of Lemma E.1: $p = \frac{1}{2}$ in the balanced two-factor regime studied by Chen et al. (2026) and $p = 1$ when the update is carried by B alone. Under constant α, the parameterization factor grows as r^p; it multiplies $\eta^\star_W$ and does not contradict the weight-space monotonicity of Proposition 4.2, which concerns $\eta^\star_W$ alone.

Paired test. To isolate the dependence on the scaling convention, we write

$$\log_{10} \eta^\star(M, \mathcal{D}, r; \gamma) = a_0(M) + \delta(\gamma) - c(\gamma) \log_{10} r - b_\beta \, \beta(\mathcal{D}; r),$$

where the convention affects only the offset $\delta(\gamma)$ and the rank coefficient $c(\gamma) = c_W - p\gamma$. The intercept $a_0(M)$, the spectral feature $\beta(\mathcal{D}; r)$, and c_W are shared, since β is computed from the gradient of the effective update and c_W refers to weight space. The offset between the boundary and the contraction-optimal rate (Proposition 4.4) depends only on the weight-space spectrum and is shared as well. We evaluate each $(M, \mathcal{D}, r)$ under a target convention and under the constant-$\alpha = 32$ reference ($\gamma = 1$) and take the paired difference

$$\Delta y := \log_{10} \eta^\star_\gamma - \log_{10} \eta^\star_{32} = \Delta\delta - \Delta c \, \log_{10} r, \qquad \Delta c = c(\gamma) - c(1) = p(1 - \gamma).$$

The subtraction eliminates $a_0(M)$, the spectral term, and c_W. An ordinary least-squares fit of Δy against $\log_{10} r$ therefore yields the β-free estimate $\widehat{p} = \widehat{\Delta c}/(1-\gamma)$ for $\gamma < 1$. The constant-$\alpha = 128$ convention ($\gamma = 1$) predicts $\Delta c = 0$ for every p and serves as a null control. Conventions in the clipping-dominated regime ($\alpha \in \{8r, 32r\}$; Appendix J) violate the clip-free premise of Lemma E.1 and are reported separately.

Table 12: **β-free validation of the μA scaling law.** For each model–dataset–rank configuration, we subtract the log-optimal learning rate under the constant-$\alpha = 32$ reference from that under the target scaling convention. The negated slope of this paired difference against $\log_{10} r$ gives the empirical $\Delta c = c(\alpha) - c(32)$ without using any spectral feature.

Scaling rule	γ	α/r	Measured Δc	μA prediction
$\alpha = 32$	1	$32/r$	—	—
$\alpha = 128$	1	$128/r$	-0.09	0.00
$\alpha = 32\sqrt{r}$	0.5	$32/\sqrt{r}$	0.23	0.25
$\alpha = r$	0	1	0.47	0.50

M ALL RESULTS IN LOW-RANK ADAPTATION

Table 13: **Cross-model transfer of (c, b) under constant α.** Rows: target model (ground truth and a_0 calibration). Columns: source model providing (c, b), fit on that model's clean sweeps.

Target \ Source	Gemma	Llama2	Llama3.1	Qwen3-1.7B	Qwen3-8B	src-mean
Gemma	—	83.8	**91.2**	90.0	**91.2**	89.1
Llama2	82.5	—	83.8	**85.0**	82.5	83.4
Llama3.1	94.3	89.8	—	**95.5**	90.9	92.6
Qwen3-1.7B	**100.0**	95.5	98.3	—	98.9	98.2
Qwen3-8B	**84.7**	77.3	77.8	77.3	—	79.3

Table 14: **Constant α: per-model, per-dataset results.** Performance is measured by Hit@$2.4\times$. n denotes the number of ranks evaluated per cell. Best and second-best results in each row are shown in bold and underlined, respectively.

Model	Dataset	n	$1e^{-4}$	$2e^{-4}$	$4e^{-4}$	TM	Grid	SPECTRA
Gemma	OpenR1-Math	4	0.0	0.0	75.0	**100.0**	**100.0**	79.7
	OpenThoughts3	4	0.0	25.0	75.0	**100.0**	**100.0**	98.4
	Tulu3	4	**100.0**	**100.0**	75.0	0.0	**100.0**	76.6
	MetaMathQA	4	50.0	**100.0**	75.0	50.0	75.0	90.6
	Code-Feedback	4	25.0	**100.0**	**100.0**	75.0	**100.0**	**100.0**
Llama2	OpenR1-Math	4	0.0	25.0	**100.0**	25.0	**100.0**	89.1
	OpenThoughts3	4	0.0	25.0	**100.0**	50.0	**100.0**	90.6
	Tulu3	4	**100.0**	**100.0**	75.0	**100.0**	**100.0**	73.4
	MetaMathQA	4	25.0	**100.0**	**100.0**	**100.0**	**100.0**	79.7
	Code-Feedback	4	25.0	**100.0**	**100.0**	**100.0**	**100.0**	84.4
Llama3.1	OpenR1-Math	12	8.3	**100.0**	**100.0**	**100.0**	**100.0**	98.4
	OpenThoughts3	12	8.3	**100.0**	**100.0**	**100.0**	**100.0**	97.9
	Tulu3	12	**100.0**	**100.0**	0.0	25.0	**100.0**	87.5
	MetaMathQA	4	**100.0**	**100.0**	0.0	25.0	**100.0**	81.2
	Code-Feedback	4	**100.0**	**100.0**	0.0	**100.0**	**100.0**	85.9
Qwen3-1.7B	OpenR1-Math	12	0.0	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**
	OpenThoughts3	12	0.0	25.0	**100.0**	**100.0**	**100.0**	97.4
	Tulu3	12	**100.0**	**100.0**	41.7	0.0	**100.0**	97.4
	MetaMathQA	4	**100.0**	**100.0**	25.0	0.0	**100.0**	95.3
	Code-Feedback	4	25.0	**100.0**	**100.0**	75.0	**100.0**	**100.0**
Qwen3-8B	OpenR1-Math	12	8.3	91.7	**100.0**	91.7	**100.0**	78.1
	OpenThoughts3	12	0.0	58.3	**100.0**	**100.0**	**100.0**	73.4
	Tulu3	12	**100.0**	**100.0**	0.0	0.0	**100.0**	89.6
	MetaMathQA	4	75.0	25.0	0.0	0.0	**100.0**	53.1
	Code-Feedback	4	**100.0**	**100.0**	75.0	0.0	**100.0**	95.3

N SWEEP PROTOCOL AND FULL OPTIMAL LEARNING RATES

Table 15: **Fine-grid sweeps under constant α.** Optimal learning rates on the full 12-rank grid for the three models where it was run. Within a dataset the optimum is nearly flat in r; the residual jitter (e.g. Tulu3 alternating between $9.6\text{e}{-5}$ and $1.4\text{e}{-4}$) is one grid step of the sweep and sets the noise floor for any predictor.

Model	Dataset	2	4	8	16	24	32	48	64	96	128	192	256
Llama3.1	OpenR1-Math	1.8e-4	2.6e-4	2.6e-4	3.8e-4	3.8e-4	3.8e-4	3.8e-4	3.8e-4	3.8e-4	3.8e-4	3.8e-4	3.8e-4
	OpenThoughts3	1.8e-4	2.6e-4	2.6e-4	3.8e-4	3.8e-4	3.8e-4	3.8e-4	3.8e-4	3.8e-4	3.8e-4	3.8e-4	3.8e-4
	Tulu3	9.6e-5	9.6e-5	9.6e-5	9.6e-5	1.4e-4	9.6e-5	9.6e-5	1.4e-4	9.6e-5	9.6e-5	1.4e-4	9.6e-5
Qwen3-1.7B	OpenR1-Math	2.6e-4	2.6e-4	3.8e-4	3.8e-4	3.8e-4	3.8e-4	3.8e-4	3.8e-4	3.8e-4	3.8e-4	3.8e-4	3.8e-4
	OpenThoughts3	2.6e-4	3.8e-4	3.8e-4	5.5e-4	5.5e-4	5.5e-4	5.5e-4	5.5e-4	5.5e-4	5.5e-4	5.5e-4	5.5e-4
	Tulu3	1.3e-4	1.3e-4	1.3e-4	1.9e-4	1.3e-4	1.9e-4	1.3e-4	1.9e-4	1.3e-4	1.9e-4	1.9e-4	1.3e-4
Qwen3-8B	OpenR1-Math	1.8e-4	2.6e-4	2.6e-4	3.8e-4	3.8e-4	3.8e-4	3.8e-4	3.8e-4	3.8e-4	5.5e-4	3.8e-4	3.8e-4
	OpenThoughts3	2.6e-4	2.6e-4	3.8e-4	3.8e-4	3.8e-4	3.8e-4	5.5e-4	5.5e-4	3.8e-4	5.5e-4	5.5e-4	5.5e-4
	Tulu3	9.2e-5	1.3e-4	1.3e-4	1.3e-4	1.3e-4	9.2e-5	1.3e-4	9.2e-5	1.3e-4	1.3e-4	1.3e-4	1.3e-4

Table 16: **Swept optimal learning rates under the constant-α convention (α=32).** Five models and five datasets at ranks $\{2, 16, 64, 256\}$; Llama3.1, Qwen3-1.7B and Qwen3-8B were additionally swept on the full 12-rank grid for OpenR1-Math, OpenThoughts3 and Tulu3 (Table 15). Within a dataset the optimum varies by at most one grid step across ranks, whereas across datasets it spans roughly an order of magnitude—the regularity our predictor exploits.

Model	Dataset	r=2	r=16	r=64	r=256
Gemma	OpenR1-Math	5.5e-4	5.5e-4	7.9e-4	1.1e-3
	OpenThoughts3	2.6e-4	5.5e-4	7.9e-4	1.1e-3
	Tulu3	1.3e-4	1.9e-4	1.9e-4	1.9e-4
	MetaMathQA	2.6e-4	3.8e-4	1.9e-4	1.4e-4
	Code-Feedback	1.8e-4	3.8e-4	3.8e-4	3.8e-4
Llama2	OpenR1-Math	3.7e-4	7.6e-4	7.6e-4	7.6e-4
	OpenThoughts3	3.8e-4	5.5e-4	7.9e-4	7.9e-4
	Tulu3	1.3e-4	1.9e-4	1.9e-4	1.9e-4
	MetaMathQA	1.8e-4	2.6e-4	2.6e-4	2.6e-4
	Code-Feedback	1.8e-4	3.8e-4	2.6e-4	2.6e-4
Llama3.1	OpenR1-Math	1.8e-4	3.8e-4	3.8e-4	3.8e-4
	OpenThoughts3	1.8e-4	3.8e-4	3.8e-4	3.8e-4
	Tulu3	9.6e-5	9.6e-5	1.4e-4	9.6e-5
	MetaMathQA	9.2e-5	9.2e-5	1.3e-4	9.2e-5
	Code-Feedback	1.3e-4	1.3e-4	1.3e-4	1.3e-4
Qwen3-1.7B	OpenR1-Math	2.6e-4	3.8e-4	3.8e-4	3.8e-4
	OpenThoughts3	2.6e-4	5.5e-4	5.5e-4	5.5e-4
	Tulu3	1.3e-4	1.9e-4	1.9e-4	1.3e-4
	MetaMathQA	9.2e-5	1.3e-4	1.3e-4	1.9e-4
	Code-Feedback	1.8e-4	2.6e-4	2.6e-4	2.6e-4
Qwen3-8B	OpenR1-Math	1.8e-4	3.8e-4	3.8e-4	3.8e-4
	OpenThoughts3	2.6e-4	3.8e-4	5.5e-4	5.5e-4
	Tulu3	9.2e-5	1.3e-4	9.2e-5	1.3e-4
	MetaMathQA	9.9e-5	6.9e-5	4.8e-5	3.3e-5
	Code-Feedback	1.3e-4	1.9e-4	1.9e-4	1.9e-4