CURVATURE-AWARE LORA INITIALIZATION IN FUNCTION SPACE

Anonymous authors
Paper under double-blind review

ABSTRACT

Parameter-efficient fine-tuning methods such as LoRA enable efficient adaptation of large pretrained models, but often lag behind full fine-tuning in both convergence speed and final performance. Recent approaches aim to reduce this gap by aligning LoRA parameter updates with those of full fine-tuning, but such parameter-space alignment only indirectly controls model predictions. Instead, we adopt a function-space perspective and formulate the prediction alignment problem, with the objective of matching the outputs of LoRA fine-tuning to those of full fine-tuning, both after the first training step and at convergence. We show that these two alignment objectives are governed by the curvature of the loss: first-step alignment leads to a curvature-colored gradient, whereas alignment at convergence leads to a Newton-like, curvature-whitened gradient. Building on these characterizations, we introduce Curvature-Guided LoRA (CG-LoRA), with two variants that initialize the adapter column spaces using the dominant singular subspaces of the colored and whitened gradients, respectively. We further derive the initialization spectrum from conditioning considerations, eliminating the need for scale tuning. Across natural language understanding, mathematical reasoning, and code generation benchmarks, CG-LoRA matches or exceeds state-of-the-art non-zero-initialization LoRA variants at a lower initialization cost.

1 INTRODUCTION

Parameter-efficient fine-tuning methods (Han et al., 2024; Houlsby et al., 2019), such as LoRA (Hu et al., 2022), have become a standard way of adapting large pretrained models, as they substantially reduce memory and computational costs compared to full fine-tuning (Biderman et al., 2024). However, a persistent gap remains between LoRA and full fine-tuning, both in convergence speed and in final performance. A recent line of work aims to close this gap with alignment-aware variants of LoRA, typically by matching the first LoRA update to that of full fine-tuning, as LoRA-GA does (Wang et al., 2024), or by modifying the gradient structure accordingly (Wang et al., 2025). While effective, these approaches operate in parameter space and therefore only indirectly control the quantity of primary interest: the model's predictions. This motivates the central question of this work: *what is the appropriate prediction-based notion of alignment between LoRA and full fine-tuning, and how should LoRA be initialized to achieve it?* LoRA-One (Zhang et al., 2025a) takes a first step in this direction. For a single linear or ReLU layer with isotropic Gaussian inputs and a low-rank target, it proves that initializing the factors from the singular subspaces of the one-step full gradient aligns them with the target from the start, and that gradient descent then converges. The guarantee rests on this data model and on a single layer, and the spectrum of the initial factors is set by a tuned scaling constant, with preconditioners added during training to handle ill-conditioning.

Our approach is to formulate the *prediction alignment problem*: how should the factors be chosen so that the predictions of LoRA agree with those of full fine-tuning, either after the first step or at the end of training, for a general network? To avoid restricting the analysis to a specific data model, we work in the lazy regime of fine-tuning, in which the network is linearized at its pretrained weights. This regime has been observed for fine-tuning from pretrained models (Malladi et al., 2023; Afzal et al., 2025) and has already been used to analyze LoRA (Jang et al., 2024); we use it as a constructive device and evaluate what it produces on real models. In this regime, the alignment problem admits a closed-form characterization, governed by an object that parameter-space criteria do not capture: the curvature of the loss with respect to the adapted weights. Aligning predictions at the end of training

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

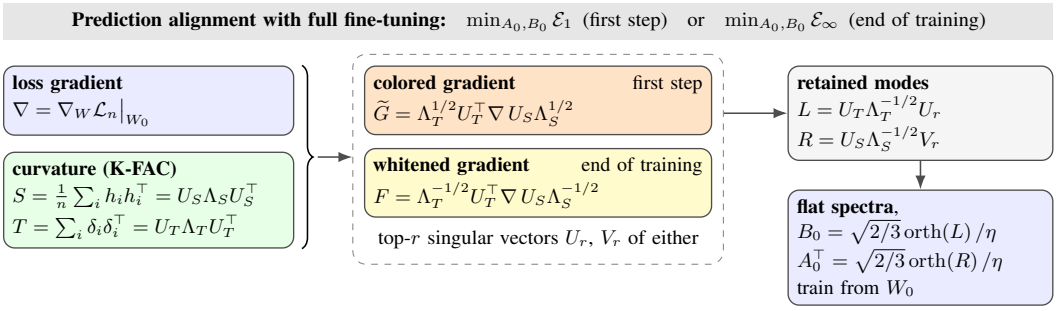


Figure 1: The two proposed initializations. For each training example x_i, $h_i \in \mathbb{R}^{d_{\text{in}}}$ denotes the input to the adapted layer, and $\delta_i \in \mathbb{R}^{d_{\text{out}} \times C}$ contains the derivatives of the C outputs with respect to the layer output. Both rescale the loss gradient $\nabla \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$ at the pretrained weights: coloring it aligns the first-step predictions with those of full fine-tuning (Proposition 5.2), whereas whitening it aligns the limiting predictions with those of full fine-tuning (Theorem 5.1).

requires the LoRA factors to span the leading singular subspaces of a curvature-whitened gradient, corresponding to a Newton-like direction. In contrast, aligning the first optimization step yields a curvature-colored gradient. The factor spectra are then determined by the conditioning of the induced LoRA preconditioner, fixing the initialization magnitude without introducing an additional scale hyperparameter.

Contributions. (a) We formulate LoRA initialization as prediction alignment with full fine-tuning at two horizons: after the first gradient step and at the end of training. We derive both objectives in closed form, using a linearized model of the network for the end-of-training objective (Proposition 4.1, Corollary 4.3).

(b) We show that both alignment problems are governed by the curvature of the loss. End-of-training alignment selects the leading singular subspaces of the gradient whitened by Kronecker-factored curvature (Theorem 5.1), whereas first-step alignment selects those of the gradient colored by the same curvature (Proposition 5.2). In the setting of LoRA-One, these subspaces recover, with high probability, the gradient subspaces underlying LoRA-One (Proposition B.1). We further show that the conditioning of the LoRA preconditioner determines the spectra of the factors, eliminating the need for scale tuning (Proposition 5.3). These results yield two sketched initialization schemes, whitened and colored, that never require forming a matrix as large as the weight matrix itself (Section 5.3).

(c) We evaluate both initializations on T5-base for GLUE against LoRA, rsLoRA, LoRA+, LoRA-Pro, PiSSA, CorDA, LoRA-GA, LoRA-One, LoRAM, EVA and LoRA-DA, and on LLaMA-2-7B for mathematical reasoning and code generation against rsLoRA, PiSSA, LoRA-GA and LoRA-One. We tune each method at its own learning rate and include controls that isolate the effects of subspace selection from those of spectrum selection. Across these benchmarks, CG-LoRA matches or exceeds the strongest baselines while requiring lower computational or memory overhead for initialization.

Notation. For a matrix $\Gamma = (\Gamma_1 \ \cdots \ \Gamma_{d_2}) \in \mathbb{R}^{d_1 \times d_2}$ with columns $\Gamma_j \in \mathbb{R}^{d_1}$, $\mathrm{vec}(\Gamma) := (\Gamma_1^\top, \ldots, \Gamma_{d_2}^\top)^\top \in \mathbb{R}^{d_1 d_2}$ stacks its columns and mat is the inverse operation; for a second matrix Ψ, $\Gamma \otimes \Psi$ is the Kronecker product, the block matrix whose (i,j) block is $\Gamma_{i,j}\Psi$. We write $\|\cdot\|_2$ for the Euclidean norm of a vector and the operator norm of a matrix, and $\|\cdot\|_F$ for the Frobenius norm. $\Gamma^\dagger$ is the Moore–Penrose pseudo-inverse of Γ, $\Pi_\Gamma := \Gamma\Gamma^\dagger$ the orthogonal projection onto its column space $\mathrm{col}(\Gamma)$, and $s_1(\Gamma) \geq s_2(\Gamma) \geq \cdots \geq 0$ its singular values. A thin SVD of Γ is $\Gamma = U\,\mathrm{diag}(s_1(\Gamma), \ldots, s_{\mathrm{rank}\,\Gamma}(\Gamma))V^\top$ with U and V having $\mathrm{rank}\,\Gamma$ orthonormal columns, and we write $\mathrm{orth}(\Gamma) := U$. The invertible and the orthogonal matrices of size r are denoted by $\mathbb{GL}_r$ and $\mathbb{O}_r$. I_d is the identity matrix of size d and $[r] := \{1, \ldots, r\}$.

2 Related work

Low-rank adaptation. On the optimization side, LoRA+ (Hayou et al., 2024b) argues that the two factors should use different learning rates, while rank-stabilized LoRA (Kalajdzievski, 2023) revisits the scaling with rank. Other approaches reduce the number of trainable parameters (Kopiczko

et al., 2024) or the cost of each operation (Chen et al., 2025), whereas DoRA (Liu et al., 2024) and AdaLoRA (Zhang et al., 2023) modify the parameterization or allocate rank adaptively. A complementary line of work asks when a low-rank update can reproduce full fine-tuning. Zeng & Lee (2024) characterize the expressive power of LoRA updates, while, in the lazy-training regime, Jang et al. (2024) show that a perturbed, weight-decayed LoRA loss has no spurious local minima once $r(r+1)/2$ exceeds the number of training outputs.

Initializing the adapters. The original initialization sets the initial LoRA update to zero by initializing one factor at zero and the other randomly (He et al., 2015), so that training begins at the pretrained solution. Subsequent work has shown that this choice can materially affect the optimization dynamics: Hayou et al. (2024a) study which factor should be initialized at zero, Zhu et al. (2024) find that a random, untrained A performs nearly as well as a trained one, Li et al. (2025b) investigate nonzero initializations, and Li et al. (2025a) propose a Nyström construction. PiSSA initializes from the pretrained weights (Meng et al., 2024), EVA from activation statistics (Paischer et al., 2025), and gradient-based methods from the loss gradient (Wang et al., 2024; Zhang et al., 2025a). Related approaches exploit the low-rank subspaces in which gradient descent evolves (Aghajanyan et al., 2021; Gur-Ari et al., 2018), or use Fisher and Kronecker-factored curvature information (Zhang et al., 2026; Feng et al., 2026).

Dynamics of LoRA. The neural tangent kernel describes gradient descent in wide networks through a linearized model (Jacot et al., 2018; Lee et al., 2019), and lazy-training theory characterizes when finite-width networks follow this regime (Chizat et al., 2019), extended to attention architectures (Hron et al., 2020). Malladi et al. (2023); Afzal et al. (2025) provide evidence that this fixed-feature behavior can persist when fine-tuning from a *pretrained* initialization. From this perspective, Jang et al. (2024) and Liu et al. (2025) analyze LoRA's rank requirements and optimization landscape. Analyses beyond the lazy regime remain more restricted, covering settings such as rank-one LoRA or single-neuron targets (Dayi & Chen, 2025; Zhang et al., 2025a; Kim et al., 2025). Finally, LoRA-Pro, Riemannian preconditioning, and LoRA-RITE modify or precondition the LoRA gradient at *every* optimization step (Wang et al., 2025; Zhang & Pilanci, 2024; Yen et al., 2025).

3 Prediction Alignment with Full Fine-Tuning

Setting. A pretrained model f is fine-tuned on $\{(x_i, y_i)\}_{i=1}^n$. We focus on one weight matrix $W \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$ of the model, with pretrained value W_0, and write $f(W; x) \in \mathbb{R}^C$ for the output as a function of the input and of this matrix, all other parameters being frozen. We stack the outputs on the training inputs into $f(W; X) := (f(W; x_1)^\top, \dots, f(W; x_n)^\top)^\top \in \mathbb{R}^{nC}$ and the targets into $Y \in \mathbb{R}^{nC}$, assume that $f(\,\cdot\,; X)$ is differentiable at W_0, and write

$$J := \frac{\partial f(W; X)}{\partial \operatorname{vec}(W)}\Big|_{W_0} \in \mathbb{R}^{nC \times d_{\text{out}} d_{\text{in}}}, \qquad Z := Y - f(W_0; X)$$

for the Jacobian of the outputs at the pretrained weights and the residual. We analyze the squared loss $\mathcal{L}_n(f(W; X)) := \frac{1}{2} \|f(W; X) - Y\|_2^2$, as the kernel analyses of fine-tuning do (Malladi et al., 2023; Jang et al., 2024), and write $\nabla := \nabla_W \mathcal{L}_n(f(W; X))|_{W_0} \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$ for its gradient at the pretrained weights, so that $\operatorname{vec} \nabla = -J^\top Z$ by the chain rule. When several matrices are adapted, each is initialized in the same way, as existing initializations do.

Full fine-tuning and LoRA. Full Fine-Tuning (FFT) updates the matrix without constraint:

$$\min_{\Delta W \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}} \mathcal{L}_n\big(f(W_0 + \Delta W; X)\big). \tag{1}$$

LoRA restricts the update to a product ηBA, with $A \in \mathbb{R}^{r \times d_{\text{in}}}$, $B \in \mathbb{R}^{d_{\text{out}} \times r}$, $r \ll \min(d_{\text{in}}, d_{\text{out}})$ and a fixed scaling $\eta > 0$ (Hu et al., 2022). When the initialization has $B_0 A_0 \neq 0$, the product $\eta B_0 A_0$ is subtracted, from the frozen weights or in the forward pass, so that training starts at W_0 (Wang et al., 2024); the update is then $\eta(BA - B_0 A_0)$. Writing $A = A_0 + \Delta A$ and $B = B_0 + \Delta B$, it equals $\eta(B_0 \Delta A + \Delta B A_0 + \Delta B \Delta A)$. We drop the second-order term $\Delta B \Delta A$ and study

$$\min_{\Delta A \in \mathbb{R}^{r \times d_{\text{in}}},\, \Delta B \in \mathbb{R}^{d_{\text{out}} \times r}} \mathcal{L}_n\big(f(W_0 + \eta(B_0 \Delta A + \Delta B A_0); X)\big), \tag{2}$$

trained from $\Delta A = 0$ and $\Delta B = 0$.

3.1 Alignment at the First Step and at Convergence

First step. Existing initializations align LoRA with full fine-tuning at the first step of training and in parameter space. One gradient step of size ν from the start moves the weights from W_0 to $W_0 - \nu\nabla$ under (1). Under (2), the gradients at $\Delta A = \Delta B = 0$ are $\eta B_0^\top \nabla$ and $\eta \nabla A_0^\top$, so the same step moves the weights to $W_0 - \nu\eta^2(B_0 B_0^\top \nabla + \nabla A_0^\top A_0)$. The two updates differ by νD_1, with $D_1 := \nabla - \eta^2 B_0 B_0^\top \nabla - \eta^2 \nabla A_0^\top A_0 \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$.

LoRA-GA chooses A_0, B_0 to minimize $\|D_1\|_F$ (Wang et al., 2024), while LoRA-One initializes the factors in the leading singular subspaces of the same gradient. LoRA-One further shows that, when the targets are generated by a single linear or ReLU layer applied to isotropic sub-Gaussian inputs (Gaussian for ReLU), gradient descent on LoRA aligns with these subspaces with high probability (Zhang et al., 2025a). Both approaches, however, formulate alignment in parameter space, where every direction of the weight update is weighted equally. Predictions respond to these directions through the network Jacobian and therefore need not weight them equally. Since $f(\,\cdot\,; X)$ is differentiable at W_0, the step changes the predictions by $f(W_0 - \nu\nabla; X) - f(W_0; X) = -\nu J \operatorname{vec}(\nabla) + o(\nu)$ under full fine-tuning, and by $-\nu\eta^2 J \operatorname{vec}(B_0 B_0^\top \nabla + \nabla A_0^\top A_0) + o(\nu)$ under LoRA. The natural next level is to align the first step in function space, that is, to compare the first-order changes of the predictions instead of the updates of the weights: $\mathcal{E}_1 := \|J \operatorname{vec}(D_1)\|_2$.

End of training. The first step is one end of training; the other, which the criteria above do not consider, is the model that fine-tuning actually produces. Gradient descent with step size $\nu > 0$ on (1) generates the weights $W^{(0)} := W_0$ and $W^{(k+1)} := W^{(k)} - \nu \nabla_W \mathcal{L}_n\big(f(W^{(k)}; X)\big)$, for $k \geq 0$, and on (2) it generates in the same way the factor updates $(\Delta A^{(k)}, \Delta B^{(k)})$ from $\Delta A^{(0)} = 0$ and $\Delta B^{(0)} = 0$, hence the weights $W_r^{(k)} := W_0 + \eta(B_0 \Delta A^{(k)} + \Delta B^{(k)} A_0)$. When the two sequences converge, we write $W_\infty := \lim_k W^{(k)}$ and $W_{r,\infty} := \lim_k W_r^{(k)}$ for the two fine-tuned models, and compare their predictions, $\mathcal{E}_\infty := \|f(W_\infty; X) - f(W_{r,\infty}; X)\|_2$. For a nonlinear network, neither the convergence of gradient descent nor the uniqueness of its limit is guaranteed, and the limits have no closed form. In the linearized model of Section 3.2, both limits exist for every step size below an explicit threshold; they do not depend on the step size, and have a closed form; this is the setting in which we study $\mathcal{E}_\infty$.

We seek initializations that align LoRA predictions with those of full fine-tuning after the first training step or at convergence:

$$\text{(first step)} \quad \min_{A_0,\, B_0} \ \mathcal{E}_1 \quad \text{and} \quad \text{(end of training)} \ \min_{A_0,\, B_0} \ \mathcal{E}_\infty. \tag{3}$$

3.2 Alignment in the Linearized Model

The end points W_∞ and $W_{r,\infty}$ of training a nonlinear network have no closed form in general and need not be unique. As in the kernel analyses of wide networks and of fine-tuning (Lee et al., 2019; Malladi et al., 2023; Jang et al., 2024), we compute them in the *linearized model*, in which the network is replaced, for full fine-tuning and for LoRA alike, by its first-order expansion at the pretrained weights,

$$f^{\text{lin}}(W; X) := f(W_0; X) + J \operatorname{vec}(W - W_0). \tag{4}$$

For wide networks trained by gradient descent, the network and its linearization remain uniformly close along training (Lee et al., 2019), and fine-tuning from pretrained models has been reported to behave in the same way (Malladi et al., 2023; Afzal et al., 2025). The linearized model is used only for the end of training: $\mathcal{E}_1$ is defined on the network itself.

In the linearized model, full fine-tuning (1) becomes

$$\min_{\Delta W \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}} \ \mathcal{L}_n\Big(f(W_0; X) + J \operatorname{vec}(\Delta W)\Big). \tag{Full Fine-Tuning}$$

We denote by $\Delta W^\star$ its minimum-norm solution, which is the update found by gradient descent after convergence: for the squared loss, this is a least-squares problem, and gradient descent started at $\Delta W = 0$ with any step size $\nu < 2/s_1^2(J)$ keeps its iterates in the row space of J and converges to $\Delta W^\star$, whatever the value of ν. In the same way, LoRA (2) becomes

$$\min_{\Delta A \in \mathbb{R}^{r \times d_{\text{in}}},\, \Delta B \in \mathbb{R}^{d_{\text{out}} \times r}} \ \mathcal{L}_n\Big(f(W_0; X) + \eta J \operatorname{vec}(B_0 \Delta A + \Delta B A_0)\Big), \tag{LoRA}$$

and we denote by $(\Delta A^\star, \Delta B^\star)$ its minimum-norm solution, to which gradient descent started at $\Delta A = 0$ and $\Delta B = 0$ converges for any step size $\nu < 2/s_1^2(J_r)$, with J_r defined below. With $\theta := (\mathrm{vec}(\Delta A)^\top, \mathrm{vec}(\Delta B)^\top)^\top$, the predictions in (LoRA) are $f(W_0; X) + J_r\theta$, where

$$J_r := \eta J\big(I_{d_{\mathrm{in}}} \otimes B_0 \quad A_0^\top \otimes I_{d_{\mathrm{out}}}\big) \in \mathbb{R}^{nC \times r(d_{\mathrm{in}}+d_{\mathrm{out}})}$$

is the *LoRA Jacobian*. In the linearized model, the end points of training are thus $W_\infty = W_0 + \Delta W^\star$ and $W_{r,\infty} = W_0 + \eta(B_0\Delta A^\star + \Delta B^\star A_0)$, and $\mathcal{E}_\infty$ is the distance between the predictions of the two linearized fine-tuned models, $\mathcal{E}_\infty = \|J\,\mathrm{vec}(\Delta W^\star) - \eta J\,\mathrm{vec}(B_0\Delta A^\star + \Delta B^\star A_0)\|_2$.

4 The Geometry of Prediction Alignment

We now rewrite the two prediction-alignment objectives in a form that exposes the geometry induced by the network Jacobian. The key distinction from parameter-space alignment is that parameter directions are not equally visible in function space: their effect on the predictions is determined by the local geometry of the pretrained model. We first characterize the limiting predictions of full fine-tuning and LoRA in the linearized model, and then use a Kronecker-factored representation of the resulting curvature to obtain tractable expressions for both alignment objectives. Proofs are given in Appendix B.

Proposition 4.1. *For every initialization (A_0, B_0), in the linearized model, $f^{\mathrm{lin}}(W_\infty; X) = f(W_0; X) + \Pi_J Z$ and $f^{\mathrm{lin}}(W_{r,\infty}; X) = f(W_0; X) + \Pi_{J_r} Z$, so that $\mathcal{E}_\infty = \|(\Pi_J - \Pi_{J_r})Z\|_2$. $\mathcal{E}_\infty$ depends on the initialization only through $\mathrm{col}(J_r)$, hence through the column spaces of B_0 and $A_0^\top$ and not through the scaling η nor the singular values of the two factors.*

This proposition shows that, at convergence, prediction alignment is determined by the function-space directions accessible to LoRA. Although the result is derived for the linearized network, evaluating the original nonlinear network at the same parameter solutions W_∞ and $W_{r,\infty}$ changes the prediction discrepancy only by terms quadratic in their distances from W_0; see Proposition C.4 in Appendix C.4.

Curvature. Both deviations $\mathcal{E}_1$ and $\mathcal{E}_\infty$ are norms of vectors in the column space of J, so they are quadratic forms in the Gauss–Newton matrix $M := J^\top J$. For instance $\mathcal{E}_1^2 = \mathrm{vec}(D_1)^\top M\,\mathrm{vec}(D_1)$. Thus, unlike the Frobenius norm in parameter space, the prediction-space objective weights parameter directions according to how strongly they affect the model outputs. Let $h_i \in \mathbb{R}^{d_{\mathrm{in}}}$ and $o_i \in \mathbb{R}^{d_{\mathrm{out}}}$ denote the input and output of the considered linear layer w.r.t. example x_i. Let $\delta_i := (\frac{\partial f(W_0;x_i)}{\partial o_i})^\top \in \mathbb{R}^{d_{\mathrm{out}}\times C}$. Then $M = \sum_i (h_i h_i^\top) \otimes (\delta_i\delta_i^\top)$; it has $(d_{\mathrm{out}}d_{\mathrm{in}})^2$ entries, which is intractable at scale. K-FAC (Martens & Grosse, 2015) replaces it by the Kronecker product of the two factors

$$S := \tfrac{1}{n}\sum_i h_i h_i^\top \in \mathbb{R}^{d_{\mathrm{in}}\times d_{\mathrm{in}}}, \qquad T := \sum_i \delta_i\delta_i^\top \in \mathbb{R}^{d_{\mathrm{out}}\times d_{\mathrm{out}}}, \tag{5}$$

with the reduce or expand variant for weight-shared layers (Eschenhagen et al., 2023). We state our results under the hypothesis $M = S \otimes T$, and the algorithm uses S and T whether or not it holds. When $M = S \otimes T$, both deviations have a closed form in two rescalings of the gradient.

Definition 4.2 (Colored and whitened gradients)**.** Let $S = U_S\Lambda_S U_S^\top$ and $T = U_T\Lambda_T U_T^\top$ be thin SVDs. We define the *colored* and *whitened* gradients as

$$\widetilde{G} := \Lambda_T^{1/2} U_T^\top \nabla U_S \Lambda_S^{1/2}, \qquad F := \Lambda_T^{-1/2} U_T^\top \nabla U_S \Lambda_S^{-1/2}.$$

For $\Gamma \in \{\widetilde{G}, F\}$ with thin SVD $\Gamma = \sum_i s_i u_i v_i^\top$ and index sets $\mathcal{I}, \mathcal{J} \subseteq [\mathrm{rank}(\Gamma)]$ of size r, let $U_\mathcal{J}$ and $V_\mathcal{I}$ collect the u_i, $i \in \mathcal{J}$, and the v_i, $i \in \mathcal{I}$. We define the corresponding *right and left retained modes* of Γ as $R_\mathcal{I}(\Gamma) := U_S\Lambda_S^{-1/2}V_\mathcal{I} \in \mathbb{R}^{d_{\mathrm{in}}\times r}$ and $L_\mathcal{J}(\Gamma) := U_T\Lambda_T^{-1/2}U_\mathcal{J} \in \mathbb{R}^{d_{\mathrm{out}}\times r}$.

Under the Kronecker factorization, the two prediction-alignment objectives now admit particularly simple forms. The first-step discrepancy becomes a curvature-weighted norm of the difference between the LoRA and full-fine-tuning updates, whereas the limiting discrepancy becomes the component of the whitened gradient that lies outside the subspaces accessible to LoRA.

Corollary 4.3. *If $M = S \otimes T$, then*

$$\mathcal{E}_1 = \big\|T^{1/2}D_1S^{1/2}\big\|_F, \qquad \mathcal{E}_\infty = \big\|\big(I - \Pi_{\Lambda_T^{1/2}U_T^\top B_0}\big)F\big(I - \Pi_{\Lambda_S^{1/2}U_S^\top A_0^\top}\big)\big\|_F. \tag{6}$$

This corollary makes the role of curvature explicit. For the first step, the parameter-space criterion $\|D_1\|_F$ is replaced by the curvature-weighted norm $\left\|T^{1/2} D_1 S^{1/2}\right\|_F$, so parameter directions are weighted according to their local effect on the model predictions. After expanding D_1, the two objectives in (3) become low-rank approximation problems governed by the colored and whitened gradients, respectively. At the first step, directions of the gradient that are strongly expressed in function space are preferred. At convergence, every direction is normalized by its effect on the predictions.

5 Curvature-Guided LoRA Initialization

We now use (6), the reduced expressions of $\mathcal{E}_1$ and $\mathcal{E}_\infty$, to construct the LoRA initialization. The initialization has two components: the column spaces of B_0 and $A_0^\top$, which determine the directions accessible to LoRA, and the singular values of the two factors, which determine their scaling within those subspaces.

5.1 Choosing the Factor Subspaces

End-of-training objective. From (6), B_0 and $A_0^\top$ remove from the whitened gradient F one subspace on the left and one on the right. Because these are orthogonal projections, minimizing $\mathcal{E}_\infty$ reduces to a low-rank approximation problem, and the Eckart–Young theorem provides the exact solution.

Theorem 5.1 (End of training)**.** *Assume $M = S \otimes T$ and $r \le \operatorname{rank} F$.*

(i) $\mathcal{E}_\infty \ge \left(\sum_{i>2r} s_i^2(F)\right)^{1/2}$ *for every initialization.*

(ii) *If $A_0^\top = R_{\mathcal{I}}(F) Q_A$ and $B_0 = L_{\mathcal{J}}(F) Q_B$ with $Q_A, Q_B \in \mathbb{GL}_r$, then $\mathcal{E}_\infty = \left(\sum_{i \notin \mathcal{I} \cup \mathcal{J}} s_i^2(F)\right)^{1/2}$; the bound of (i) is attained whenever $\mathcal{I} \cup \mathcal{J} \supseteq [\min(2r, \operatorname{rank} F)]$.*

The theorem gives the optimal value of $\mathcal{E}_\infty$ and subspaces that attain it. A rank-r LoRA initialization can remove up to $2r$ singular directions of the whitened gradient: r through the left factor and r through the right factor. Thus, prediction alignment at convergence is governed by the dominant singular directions of F, mapped back through the curvature.

First-step objective. For $\mathcal{E}_1$, the curvature factors make the corresponding projections oblique rather than orthogonal, so the Eckart–Young argument yields a lower bound but no longer an exact solution. The colored gradient nevertheless provides a natural choice of subspaces.

Proposition 5.2 (First step)**.** *Assume $M = S \otimes T$ and $r \le \operatorname{rank} \widetilde{G}$, and write $\tilde{s}_i := s_i(\widetilde{G})$.*

(i) $\mathcal{E}_1 \ge \left(\sum_{i>2r} \tilde{s}_i^2\right)^{1/2}$ *for every initialization.*

(ii) *If $A_0^\top = \frac{1}{\sqrt{2\eta}} \operatorname{orth}(R_{[r]}(\widetilde{G}))\, Q_A$ and $B_0 = \frac{1}{\sqrt{2\eta}} \operatorname{orth}(L_{[r]}(\widetilde{G}))\, Q_B$ with $Q_A, Q_B \in \mathbb{O}_r$, then*

$$\mathcal{E}_1 \le \left(1 + \tfrac{1}{4}\beta_T^2 + \tfrac{1}{4}\beta_S^2\right)^{1/2} \left(\sum_{i>r} \tilde{s}_i^2\right)^{1/2},$$

where $\beta_T, \beta_S \ge 0$ measure the leakage through the curvature, defined in (9) *(Appendix B.2).*

(iii) *If S and T are multiples of the identity, then $\widetilde{G} \propto \nabla$ and the choice of (ii) gives $\mathcal{E}_1 = \left(\sum_{i>r} \tilde{s}_i^2\right)^{1/2}$.*

LoRA-One (Zhang et al., 2025a) is analyzed for a single linear or ReLU layer with Gaussian inputs and a rank-r target. In this setting, our constructions recover its gradient subspaces up to an explicit bound on the resulting subspace angle (Proposition B.1, Appendix B.2).

5.2 Choosing the Factor Spectra

Neither Theorem 5.1 nor Proposition 5.2 determines the spectra of the factors. The first step of Section 3.1 moves the weights by $-\nu\nabla$ under full fine-tuning and by $-\nu \operatorname{mat}(P \operatorname{vec} \nabla)$ under LoRA, where, with $H := \eta(I_{d_{\text{in}}} \otimes B_0 \;\; A_0^\top \otimes I_{d_{\text{out}}})$ so that $J_r = JH$,

$$P := HH^\top = \eta^2 \left(I_{d_{\text{in}}} \otimes B_0 B_0^\top + A_0^\top A_0 \otimes I_{d_{\text{out}}}\right). \tag{7}$$

Algorithm 1 CG-LoRA, whitened (CG-LoRA-W) and colored (CG-LoRA-C), for one adapted layer

1: **Input:** probe batch, rank r, window $m \geq 2r$, LoRA scaling η; variant *whitened* (W) or *colored* (C).
2: Sketch the top-m singular vectors and values (U_S, Λ_S) of S and (U_T, Λ_T) of T; accumulate $F_{\text{raw}} = U_T^\top \nabla U_S$ from the training loss.
3: Rescale: $\Gamma := F = \Lambda_T^{-1/2} F_{\text{raw}} \Lambda_S^{-1/2}$ (whitened) or $\Gamma := \widetilde{G} = \Lambda_T^{1/2} F_{\text{raw}} \Lambda_S^{1/2}$ (colored).
4: Thin SVD of Γ and its retained modes (Definition 4.2) for $\mathcal{I} = \mathcal{J} = [r]$: $V := \operatorname{orth}(R_{[r]}(\Gamma))$, $U := \operatorname{orth}(L_{[r]}(\Gamma))$.
5: **Output:** $B_0 = \sqrt{2/3}\, U/\eta$ and $A_0^\top = \sqrt{2/3}\, V/\eta$.

A LoRA step is thus a full fine-tuning step preconditioned by P, the *LoRA preconditioner*, and one step leaves the fraction $\|(I-P)u\|_2$ of a direction u uncorrected. We choose the spectra that make this fraction small in the worst case over $\operatorname{col} H$, the directions in which LoRA can move. Write $\eta^2 B_0 B_0^\top = U \Sigma_B U^\top$ and $\eta^2 A_0^\top A_0 = V \Sigma_A V^\top$ with orthonormal U, V spanning the chosen subspaces, and let P have extreme singular values $\lambda_{\max} \geq \lambda_{\min} > 0$ on $\operatorname{col} H$, with condition number $\kappa_0 := \lambda_{\max}/\lambda_{\min}$.

Proposition 5.3 (The flat point)**.** *For every initialization with the chosen subspaces,*

$$\max\left\{ \|(I-P)u\|_2 : \; u \in \operatorname{col} H, \; \|u\|_2 = 1 \right\} \;\geq\; \frac{\kappa_0 - 1}{\kappa_0 + 1} \;\geq\; \frac{1}{3},$$

and the value $\frac{1}{3}$ is attained if and only if $\Sigma_B = \Sigma_A = \frac{2}{3} I$, which we call the flat point*.*

The first inequality is the standard contraction bound for preconditioned Richardson iteration: after optimally scaling the preconditioner so that $\lambda_{\min} + \lambda_{\max} = 2$, the worst-case error contraction factor is $(\kappa_0 - 1)/(\kappa_0 + 1)$ (Saad, 2003). The second inequality is specific to the LoRA preconditioner (7). Because its two Kronecker terms overlap whenever both factors are nonzero, its condition number on $\operatorname{col}(H)$ satisfies $\kappa_0 \geq 2$; equality holds precisely when the two factor spectra are equal and flat. Consequently, the smallest attainable worst-case discrepancy is $1/3$, achieved at $\Sigma_A = \Sigma_B = \frac{2}{3} I$. This choice is also consistent with the broader role of balancing in low-rank factorization. Balanced factors, satisfying $B_0^\top B_0 = A_0 A_0^\top$, are preserved by gradient flow (Du et al., 2018), while poor conditioning of the factors can slow gradient-based optimization of low-rank models. Scaled gradient descent (Tong et al., 2021) and Riemannian preconditioned LoRA (Zhang & Pilanci, 2024) address this issue by continually correcting the optimization geometry during training. In contrast, the flat point provides a balanced, optimally conditioned choice at initialization, fixing the factor scales without an additional tuning parameter.

Algorithm 1 therefore runs CG-LoRA-C at the flat point rather than at the scale $1/(\sqrt{2}\eta)$ of Proposition 5.2(ii), which keeps its subspaces and trades the first step for conditioning: each retained mode is corrected by $\frac{2}{3}$ on each side instead of $\frac{1}{2}$.

5.3 Algorithms

Combining the subspace and spectrum choices gives two variants of our algorithm, Curvature-Guided LoRA (Figure 1, Algorithm 1). CG-LoRA-W uses the retained modes of the whitened gradient F and therefore targets prediction alignment at convergence. CG-LoRA-C uses those of the colored gradient $\widetilde{G}$ and targets prediction alignment after the first gradient step. Both use the paired modes $\mathcal{I} = \mathcal{J} = [r]$, which leave the tail beyond r rather than $2r$ but train better (Appendix F.4), and scale the factors to the flat point.

The implementation never forms the full curvature matrix. For each adapted layer, it first sketches the leading singular subspaces of S and T using randomized range finding (Halko et al., 2011). The products required by the sketch are accumulated through forward and backward hooks, with random-sign probes used on the output side (Hutchinson, 1989) (Algorithm 2, Appendix E). It then forms only the small projected gradient $F_{\text{raw}} \in \mathbb{R}^{m \times m}$, from which F and $\widetilde{G}$ follow by diagonal rescaling. This $m \times m$ matrix is decomposed, its retained modes are mapped back to the original layer coordinates and orthonormalized, and the factors are scaled to the flat point. All passes are shared across the adapted layers. The method therefore requires $O\big(m(d_{\text{in}} + d_{\text{out}})\big)$ additional memory

per layer and decomposes only $m \times m$ matrices, rather than storing a full $d_{\text{out}} \times d_{\text{in}}$ gradient or decomposing the pretrained weight matrix. We refer the reader to Table 17, Appendix F.7.

6 EXPERIMENTS

We evaluate the two variants of CG-LoRA on natural language understanding and generation, and measure their initialization cost. We compare our method to the following baselines. LoRA and rsLoRA serve as zero-initialized controls, so that our rows differ from rsLoRA only in the initialization. PiSSA, CorDA, LoRA-GA, LoRA-One, and LoRAM are comparable weight-, data-, or gradient-dependent initializations of the same adapter. LoRA+ and LoRA-Pro provide reference points for methods that instead modify its training. Appendix D.3 details how each is run and why other LoRA variants are not compared, and reports results for two further initializations, EVA and LoRA-DA. Every arm selects its own learning rate by one fixed rule on a held-out split of the training set, never on the validation set. Both of our initializations are built on a 32-example probe from the gradient of the training loss. Additional controlled experiments isolating the effects of both our proposed choices of the subspace and the flat point scaling, as well as robustness analyses and rank sweeps are reported in Appendices E and F.

6.1 NATURAL LANGUAGE UNDERSTANDING

We evaluate T5-base (Raffel et al., 2020) on five GLUE tasks (Wang et al., 2018): CoLA (Matthews correlation coefficient), MRPC, SST-2, and QNLI (accuracy), and STS-B (Pearson correlation). Following Raffel et al. (2020), all tasks are formulated in a text-to-text format with label words as targets; for STS-B, the score is rounded to the nearest 0.2 and represented as a string. We adapt the 72 query and value projection matrices with rank $r = 8$ and LoRA hyperparameter $\alpha = 2r$ ($\eta = \alpha/\sqrt{r}$ under rsLoRA), train for one epoch, and report results over ten seeds for CoLA and MRPC and five seeds for the remaining tasks. We also evaluate RoBERTa-base (Liu et al., 2019) in Appendix F.3, whose randomly initialized classification head makes fine-tuning unstable and CG-LoRA-W is the most robust of the gradient-based initializations.

Table 1: T5-base results after one epoch, with the learning rate selected separately for each method. We report mean and standard deviation over ten seeds for CoLA and MRPC and five seeds for the remaining tasks; the metric is MCC for CoLA, Pearson correlation for STS-B, and accuracy otherwise. η denotes the LoRA scaling ($\alpha = 2r = 16$). "Spectra" specifies the initialization scale of the factors, where s_i denotes singular values, γ the stability constant used by the respective authors, 16 the constant used for LoRA-GA, $d = d_{\text{out}}$, C_{CorDA} the input covariance used by CorDA and $\beta(W_0)$ the scale of LoRAM (Appendix D.3). Bold: best adapter per column, full fine-tuning excluded.

	η	spectra	CoLA	MRPC	SST-2	QNLI	STS-B
FFT, all weights	–	–	46.4 ± 2.1	86.6 ± 1.4	94.4 ± 0.4	93.1 ± 0.1	88.4 ± 0.3
FFT, query/value	–	–	47.4 ± 2.2	86.2 ± 0.6	94.5 ± 0.3	93.0 ± 0.0	88.1 ± 0.3
LoRA	α/r	$B_0 = 0$	44.3 ± 5.5	81.8 ± 0.8	93.9 ± 0.2	$\mathbf{93.1} \pm 0.1$	88.4 ± 0.4
rsLoRA	$\alpha/\sqrt{r}$	$B_0 = 0$	48.6 ± 2.4	84.4 ± 1.2	94.1 ± 0.2	$\mathbf{93.1} \pm 0.1$	88.3 ± 0.2
LoRA+	α/r	$B_0 = 0$	45.3 ± 2.3	84.1 ± 3.0	93.9 ± 0.2	92.9 ± 0.2	87.8 ± 0.5
LoRA-Pro	$\alpha/\sqrt{r}$	$B_0 = 0$	42.4 ± 1.8	84.9 ± 0.9	93.9 ± 0.1	93.0 ± 0.1	87.9 ± 0.2
PiSSA	$\alpha/\sqrt{r}$	$\sqrt{s_i(W_0)/\eta}$	44.5 ± 2.6	85.1 ± 0.8	93.9 ± 0.1	92.9 ± 0.1	$\mathbf{88.7} \pm 0.2$
CorDA	$\alpha/\sqrt{r}$	$\sqrt{s_i(W_0 C_{\text{CorDA}})/\eta}$	42.4 ± 3.5	81.2 ± 1.1	93.7 ± 0.1	92.9 ± 0.1	87.6 ± 0.5
LoRA-GA	$\alpha/\sqrt{r}$	$d^{1/4}/\sqrt{\gamma}$	44.0 ± 2.6	84.6 ± 0.9	94.1 ± 0.2	92.9 ± 0.1	88.4 ± 0.2
LoRA-One	$\alpha/\sqrt{r}$	$\sqrt{s_i(\nabla)/(s_1(\nabla)\gamma)}$	47.3 ± 5.3	84.8 ± 1.1	93.8 ± 0.1	93.0 ± 0.1	88.5 ± 0.3
LoRAM ($\alpha = r$)	$\alpha/\sqrt{r}$	$\beta(W_0)$, sine bases	44.8 ± 2.6	83.8 ± 0.9	$\mathbf{94.2} \pm 0.2$	92.9 ± 0.2	87.8 ± 0.4
CG-LoRA-W (ours)	$\alpha/\sqrt{r}$	flat, $\sqrt{2/3}/\eta$	$\mathbf{49.8} \pm 2.7$	84.6 ± 0.8	94.1 ± 0.3	$\mathbf{93.1} \pm 0.2$	88.5 ± 0.8
CG-LoRA-C (ours)	$\alpha/\sqrt{r}$	flat, $\sqrt{2/3}/\eta$	49.4 ± 2.5	$\mathbf{85.6} \pm 1.2$	93.9 ± 0.2	92.9 ± 0.2	$\mathbf{88.7} \pm 0.5$

CG-LoRA-W is the best adapter on CoLA and CG-LoRA-C on MRPC, and both match the best adapters on the three saturated tasks (Table 1). On CoLA the two variants reach 49.8 and 49.4, ahead of every baseline and of full fine-tuning (46.4); over the ten seeds the paired difference of CG-LoRA-W to LoRA-One is $+2.5$ ($p = 0.07$, paired t-test) and to rsLoRA $+1.2$ ($p = 0.2$), and at least $+4.5$ to every other adapter ($p \leq 0.03$); its worst seed (45.6) is above that of every baseline. On

MRPC, CG-LoRA-C is the best adapter (85.6), above rsLoRA, LoRA-One and LoRAM ($p \leq 0.04$) and 1.0 below full fine-tuning. On SST-2, QNLI and STS-B the adapters lie within about a point of each other, and the better of our two variants is within 0.2 of the best one. Our constructions thus match or exceed every baseline on the five tasks, with no tuned constant, and at a small initialization cost (Table 3).

6.2 Natural language generation

We evaluate LLaMA-2-7B (Touvron et al., 2023), adapting all 224 linear layers. The model is trained for one epoch on the first 100k examples from MetaMathQA (Yu et al., 2024) and evaluated on GSM8K (Cobbe et al., 2021). We also train on CodeFeedback (Zheng et al., 2024) and evaluate on HumanEval (Chen et al., 2021). Following LoRA-GA and LoRA-One, we report results over three seeds.

Table 2: LLaMA-2-7B, all linear layers, $r = 8$, $\alpha = 16$, one epoch, own learning rate per arm (Appendix D.1). Mean and standard deviation over three seeds, and worst seed. Bold: best adapter.

	GSM8K		HumanEval	
	mean	worst	mean	worst
rsLoRA	59.0 ± 0.4	58.6	27.2 ± 0.9	26.2
PiSSA	58.3 ± 0.7	57.5	25.2 ± 1.4	24.4
LoRA-GA	57.1 ± 1.4	55.6	23.0 ± 1.9	21.3
LoRA-One	58.9 ± 0.8	57.9	$\mathbf{29.5} \pm 3.1$	26.2
CG-LoRA-W	58.7 ± 0.9	57.7	28.7 ± 1.1	**28.0**
CG-LoRA-C	$\mathbf{59.4} \pm 0.6$	**59.0**	28.9 ± 1.3	27.4

Table 3: Measured cost of the initialization on one NVIDIA GH200: wall-clock time (seconds), peak GPU memory above the loaded model and CPU host memory (GiB), see Appendix F.7. Bold: fastest initialization.

	T5-base		LLaMA-2-7B		
	time	memory	time	memory	host
LoRA training	–	1.26	–	13.35	0
PiSSA	2.6	0.01	239	1.32	0
LoRA-GA	**0.6**	1.21	49	2.83	12.6
LoRA-One	6.0	1.21	85	2.76	12.6
CG-LoRA-W	2.3	0.66	**36**	2.83	0
CG-LoRA-C	2.4	0.66	**36**	2.83	0

With three seeds, Table 2 reports the mean and standard deviation and the worst seed of each arm. On GSM8K the colored initialization is the best arm on both counts, 59.4 with a worst seed of 59.0, ahead of all variants. On HumanEval our two constructions are the most reliable arms: their worst seeds, 28.0 (whitened) and 27.4 (colored), are above the worst seed of every baseline, with less than half of LoRA-One's spread. LoRA-One's higher mean (29.5 ± 3.1 against 28.9 ± 1.3 and 28.7 ± 1.1) rests on a single seed (32.3, its other two being 26.2 and 29.9). Our constructions thus match the baselines on both benchmarks, at an initialization cost 1.4 to 6.6 times lower (Table 3).

Table 3 reports the measured cost. On LLaMA-2-7B our initializations take 36 seconds, against 85 for LoRA-One, 49 for LoRA-GA and 239 for PiSSA, that is, 1.4 to 6.6 times less: they never form a $d_{\text{out}} \times d_{\text{in}}$ gradient or decompose a weight, only $m \times m$ matrices with $m = 16$. Their peak memory stays well below that of LoRA training itself on both models, so the initialization fits wherever training does. On T5-base every initialization takes a few seconds. On LLaMA-2-7B the GPU peak is that of a backward pass over the probe for every gradient-based initialization, ours included; LoRA-GA and LoRA-One additionally hold the gradient of every parameter in host memory while they run.

7 Conclusion

In this work, we introduced the prediction alignment problem for parameter-efficient fine-tuning, shifting the focus from aligning parameter updates to aligning model outputs. We showed that this objective naturally leads to a curvature-aware, second-order formulation, and proposed CG-LoRA, an efficient initialization method based on curvature-aware gradients that avoids explicit construction of large curvature matrices. Although our theory is developed in linearized settings, the resulting design principle, which goes beyond first-order gradient matching, remains applicable, competitive and computationally efficient in practice. Several directions remain open. It would be valuable to apply CG-LoRA to LoRA variants other than the original one (Hu et al., 2022), and to turn the proposed low-rank criteria into a rule for allocating ranks to all adapted layers under a fixed budget.

AI USE STATEMENT

In this work, we used generative AI tools for the following tasks: design or provide feedback on research methodology or experiments, implement methods. We have not used generative AI tools for the following tasks: generate synthetic data sets, help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses, assist with translation, clean and reformat dataset, support qualitative and thematic data analysis, interpret results.

Additionally, we used generative AI tools for the following tasks: summarize or analyse existing literature, sourcing/searching for information, identify relevant literature. We have reviewed all AI-assisted work such as reading literature proposed to evaluate relevance, and we read/tested manually any generated code. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

The supplementary material contains our code. Each theoretical result states its assumptions, and Appendices B and C prove them.

REFERENCES

Zahra Rahimi Afzal, Tara Esmaeilbeig, Mojtaba Soltanalian, and Mesrob I. Ohannessian. Linearization explains fine-tuning in large language models. In *Advances in Neural Information Processing Systems*, 2025.

Armen Aghajanyan, Sonal Gupta, and Luke Zettlemoyer. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2021.

Dan Biderman, Jacob Portes, Jose Javier Gonzalez Ortiz, et al. LoRA learns less and forgets less. *Transactions on Machine Learning Research*, 2024.

Guanduo Chen, Yutong He, Yipeng Hu, Kun Yuan, and Binhang Yuan. CE-LoRA: Computation-Efficient LoRA Fine-Tuning for Language Models. *arXiv preprint arXiv:2502.01378*, 2025.

Mark Chen, Jerry Tworek, Heewoo Jun, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Lénaïc Chizat, Edouard Oyallon, and Francis Bach. On lazy training in differentiable programming. In *Advances in Neural Information Processing Systems*, 2019.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Arif Kerem Dayi and Sitan Chen. Low-rank fine-tuning lies between lazy training and feature learning. In *Proceedings of the Thirty Eighth Conference on Learning Theory*, 2025.

Simon S. Du, Wei Hu, and Jason D. Lee. Algorithmic regularization in learning deep homogeneous models: Layers are automatically balanced. In *Advances in Neural Information Processing Systems*, 2018.

Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1936.

Heinz Werner Engl, Martin Hanke, and Andreas Neubauer. *Regularization of Inverse Problems*. Kluwer, 1996.

Runa Eschenhagen, Alexander Immer, Richard E. Turner, Frank Schneider, and Philipp Hennig. Kronecker-factored approximate curvature for modern neural network architectures. In *Advances in Neural Information Processing Systems*, 2023.

Zhi-Quan Feng, Ying-Jia Lin, and Hung-Yu Kao. Learning in the Fisher subspace: A guided initialization for LoRA fine-tuning. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026.

Guy Gur-Ari, Daniel A Roberts, and Ethan Dyer. Gradient descent happens in a tiny subspace. *arXiv preprint arXiv:1812.04754*, 2018.

Nathan Halko, Per-Gunnar Martinsson, and Joel A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 2011.

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.

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

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

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In *Proceedings of the IEEE International Conference on Computer Vision*, 2015.

Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 1963.

Roger A. Horn and Charles R. Johnson. *Topics in Matrix Analysis*. Cambridge University Press, 1991.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, et al. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*, 2019.

Jiri Hron, Yasaman Bahri, Jascha Sohl-Dickstein, and Roman Novak. Infinite attention: NNGP and NTK for deep attention networks. In *Proceedings of the 37th International Conference on Machine Learning*, 2020.

Edward J. Hu, Yelong Shen, Phillip Wallis, et al. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Michael F. Hutchinson. A stochastic estimator of the trace of the influence matrix for Laplacian smoothing splines. *Communications in Statistics - Simulation and Computation*, 1989.

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

Uijeong Jang, Jason D. Lee, and Ernest K. Ryu. LoRA training in the NTK regime has no spurious local minima. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

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

Junsu Kim, Jaeyeon Kim, and Ernest K. Ryu. LoRA training provably converges to a low-rank global minimum or it fails loudly (But it probably won't fail). In *Proceedings of the 42nd International Conference on Machine Learning*, 2025.

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

Ananya Kumar, Aditi Raghunathan, Robbie Jones, Tengyu Ma, and Percy Liang. Fine-tuning can distort pretrained features and underperform out-of-distribution. In *International Conference on Learning Representations*, 2022.

Cheolhyoung Lee, Kyunghyun Cho, and Wanmo Kang. Mixout: Effective regularization to finetune large-scale pretrained language models. In *International Conference on Learning Representations*, 2020.

Jaehoon Lee, Lechao Xiao, Samuel S. Schoenholz, et al. Wide neural networks of any depth evolve as linear models under gradient descent. In *Advances in Neural Information Processing Systems*, 2019.

Bingcong Li, Liang Zhang, Aryan Mokhtari, and Niao He. On the crucial role of initialization for matrix factorization. In *International Conference on Learning Representations*, 2025a.

Shiwei Li, Xiandi Luo, Xing Tang, et al. Beyond zero initialization: Investigating the impact of non-zero initialization on LoRA fine-tuning dynamics. In *Proceedings of the 42nd International Conference on Machine Learning*, 2025b.

Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, et al. DoRA: Weight-decomposed low-rank adaptation. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

Xu-Hui Liu, Yali Du, Jun Wang, and Yang Yu. On the optimization landscape of low rank adaptation methods for large language models. In *International Conference on Learning Representations*, 2025.

Yinhan Liu, Myle Ott, Naman Goyal, et al. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

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

James Martens and Roger Grosse. Optimizing neural networks with Kronecker-factored approximate curvature. In *Proceedings of the 32nd International Conference on Machine Learning*, 2015.

Fanxu Meng, Zhaohui Wang, and Muhan Zhang. PiSSA: Principal singular values and singular vectors adaptation of large language models. In *Advances in Neural Information Processing Systems*, 2024.

Leon Mirsky. Symmetric gauge functions and unitarily invariant norms. *The Quarterly Journal of Mathematics*, 1960.

Marius Mosbach, Maksym Andriushchenko, and Dietrich Klakow. On the stability of fine-tuning BERT: Misconceptions, explanations, and strong baselines. In *International Conference on Learning Representations*, 2021.

Fabian Paischer, Lukas Hauzenberger, Thomas Schmied, Benedikt Alkin, Marc Peter Deisenroth, and Sepp Hochreiter. Parameter efficient fine-tuning via explained variance adaptation. In *Advances in Neural Information Processing Systems*, 2025.

Colin Raffel, Noam Shazeer, Adam Roberts, et al. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 2020.

Yousef Saad. *Iterative Methods for Sparse Linear Systems*. SIAM, 2 edition, 2003.

Akiyoshi Tomihari and Issei Sato. Understanding linear probing then fine-tuning language models from NTK perspective. In *Advances in Neural Information Processing Systems*, 2024.

Tian Tong, Cong Ma, and Yuejie Chi. Accelerating ill-conditioned low-rank matrix estimation via scaled gradient descent. *Journal of Machine Learning Research*, 2021.

Hugo Touvron, Louis Martin, Kevin Stone, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Roman Vershynin. Introduction to the non-asymptotic analysis of random matrices. In *Compressed Sensing: Theory and Applications*. Cambridge University Press, 2012.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, 2018.

Shaowen Wang, Linxi Yu, and Jian Li. LoRA-GA: Low-rank adaptation with gradient approximation. In *Advances in Neural Information Processing Systems*, 2024.

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

Yibo Yang, Xiaojie Li, Zhongzhu Zhou, et al. CorDA: Context-oriented decomposition adaptation of large language models for task-aware parameter-efficient fine-tuning. In *Advances in Neural Information Processing Systems*, 2024.

Jui-Nan Yen, Si Si, Zhao Meng, et al. LoRA done RITE: Robust invariant transformation equilibration for LoRA optimization. In *International Conference on Learning Representations*, 2025.

Longhui Yu, Weisen Jiang, Han Shi, et al. MetaMath: Bootstrap your own mathematical questions for large language models. In *International Conference on Learning Representations*, 2024.

Yuchen Zeng and Kangwook Lee. The expressive power of low-rank adaptation. In *International Conference on Learning Representations*, 2024.

Fangzhao Zhang and Mert Pilanci. Riemannian preconditioned LoRA for fine-tuning foundation models. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, et al. Adaptive budget allocation for parameter-efficient fine-tuning. In *International Conference on Learning Representations*, 2023.

Qingyue Zhang, Chang Chu, Tianren Peng, Qi Li, Xiangyang Luo, Zhihao Jiang, and Shao-Lun Huang. LoRA-DA: Data-aware initialization for low-rank adaptation via asymptotic analysis. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026.

Tianyi Zhang, Felix Wu, Arzoo Katiyar, Kilian Q. Weinberger, and Yoav Artzi. Revisiting few-sample BERT fine-tuning. In *International Conference on Learning Representations*, 2021.

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

Zicheng Zhang, Haoran Li, Yifeng Zhang, et al. The primacy of magnitude in low-rank adaptation. In *Advances in Neural Information Processing Systems*, 2025b.

Tianyu Zheng, Ge Zhang, Tianhao Shen, et al. OpenCodeInterpreter: Integrating code generation with execution and refinement. In *Findings of the Association for Computational Linguistics: ACL 2024*, 2024.

Jiacheng Zhu, Kristjan Greenewald, Kimia Nadjahi, et al. Asymmetry in low-rank adapters of foundation models. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

A NOTATION

All statements and proofs are for one adapted weight matrix, as in the main text. Table 4 lists the symbols.

Table 4: Notation, for one adapted layer with d_{in} inputs and d_{out} outputs, n training inputs and C outputs per input.

symbol	meaning
$\Gamma^\dagger$, $\mathrm{col}(\Gamma)$, Π_Γ	pseudo-inverse of a matrix Γ, its column space, and the orthogonal projector $\Gamma\Gamma^\dagger$ onto it
$s_1(\Gamma) \geq s_2(\Gamma) \geq \cdots$	singular values
vec, mat, $\otimes$	column stacking, its inverse, Kronecker product
W_0; $f(W; X)$, Y, Z	pretrained weight ($d_{\text{out}} \times d_{\text{in}}$); outputs, targets and residual $Z = Y - f(W_0; X)$, all in $\mathbb{R}^{nC}$
J, $M = J^\top J$	Jacobian of the outputs at W_0 ($nC \times d_{\text{out}} d_{\text{in}}$); Gauss–Newton matrix
∇	gradient of the loss at W_0 ($d_{\text{out}} \times d_{\text{in}}$), $\mathrm{vec}\,\nabla = -J^\top Z$
A_0, B_0, η, r	initial LoRA factors ($r \times d_{\text{in}}$ and $d_{\text{out}} \times r$), scaling, rank
H, $J_r = JH$, $P = HH^\top$	LoRA map $\eta(I \otimes B_0 \;\; A_0^\top \otimes I)$, LoRA Jacobian, LoRA preconditioner (7)
D_1; $\mathcal{E}_1$, $\mathcal{E}_\infty$	first-step deviation of the weights; deviations of the predictions (Section 3.1)
S, T	Kronecker factors (5), with thin SVDs $U_S \Lambda_S U_S^\top$ and $U_T \Lambda_T U_T^\top$ of ranks k_S and k_T
$\widetilde{G}$, F	colored and whitened gradients ($k_T \times k_S$, Definition 4.2)
$R_{\mathcal{I}}$, $L_{\mathcal{J}}$	retained right and left modes (Definition 4.2)
κ_0	condition number of P on $\mathrm{col}\, H$

B PROOFS OF THE MAIN RESULTS

This appendix proves the statements of Sections 4 and 5, in the order of the main text. Throughout, $S = U_S \Lambda_S U_S^\top$ and $T = U_T \Lambda_T U_T^\top$ are the thin SVD of Definition 4.2, $k_S := \operatorname{rank} S$, $k_T := \operatorname{rank} T$, and we use freely the identities $\operatorname{vec}(\Gamma_1 \Psi \Gamma_2) = (\Gamma_2^\top \otimes \Gamma_1)\operatorname{vec}(\Psi)$, $(\Gamma_1 \otimes \Gamma_2)(\Gamma_3 \otimes \Gamma_4) = \Gamma_1\Gamma_3 \otimes \Gamma_2\Gamma_4$ and $\|\operatorname{vec}(\Psi)\|_2 = \|\Psi\|_F$ (Appendix C.1). The auxiliary results of Appendix C are cited where they are used.

B.1 SECTION 4

Proof of Proposition 4.1. By Lemma C.1 with $\Gamma = J$, gradient descent on (Full Fine-Tuning) converges for every $\nu < 2/s_1^2(J)$ to $\operatorname{vec}(\Delta W^\star) = J^\dagger Z$, and the predictions converge to $f^{\mathrm{lin}}(W_\infty; X) = f(W_0; X) + \Pi_J Z$. With $\Gamma = J_r$, gradient descent on (LoRA) converges for every $\nu < 2/s_1^2(J_r)$ to $\theta^\star = J_r^\dagger Z$, and $f^{\mathrm{lin}}(W_{r,\infty}; X) = f(W_0; X) + J_r\theta^\star = f(W_0; X) + \Pi_{J_r} Z$. Subtracting gives $\mathcal{E}_\infty = \|(\Pi_J - \Pi_{J_r})Z\|_2$.

Furthermore $\operatorname{col}(J_r) = \{J \operatorname{vec}(B_0 X + Y A_0) : X \in \mathbb{R}^{r \times d_{\mathrm{in}}},\, Y \in \mathbb{R}^{d_{\mathrm{out}} \times r}\}$ does not involve $\eta > 0$, and $\{B_0 X\}$ and $\{Y A_0\}$ are the matrices whose columns lie in $\operatorname{col}(B_0)$ and whose rows lie in $\operatorname{col}(A_0^\top)$, so $\operatorname{col}(J_r)$, hence $\mathcal{E}_\infty$, depends on the factors only through these two column spaces. □

Proof of Corollary 4.3. First step. By definition of $\mathcal{E}_1$, $\mathcal{E}_1^2 = \operatorname{vec}(D_1)^\top J^\top J \operatorname{vec}(D_1) = \operatorname{vec}(D_1)^\top (S \otimes T)\operatorname{vec}(D_1)$. Since $S \otimes T = (S^{1/2} \otimes T^{1/2})(S^{1/2} \otimes T^{1/2})$,

$$\mathcal{E}_1^2 = \left\| (S^{1/2} \otimes T^{1/2}) \operatorname{vec}(D_1) \right\|_2^2 = \left\| \operatorname{vec}\left(T^{1/2} D_1 S^{1/2}\right) \right\|_2^2 = \left\| T^{1/2} D_1 S^{1/2} \right\|_F^2,$$

which is the first identity of (6).

End of training. By Proposition 4.1, $\mathcal{E}_\infty = \|(\Pi_J - \Pi_{J_r})Z\|_2$. Let $J = U_J D_J V_J^\top$ be a thin SVD. By Lemma C.2, $\Pi_J - \Pi_{J_r} = U_J(I - \Pi_G)U_J^\top$ with $G = D_J V_J^\top H$, and by Lemma C.3, under $M = S \otimes T$, U_J can be taken so that $J = U_J(\Lambda_S^{1/2} \otimes \Lambda_T^{1/2})(U_S \otimes U_T)^\top$ and

$$I - \Pi_G = (I_{k_S} - \Pi_A) \otimes (I_{k_T} - \Pi_B), \qquad \Pi_A := \Pi_{\Lambda_S^{1/2} U_S^\top A_0^\top}, \quad \Pi_B := \Pi_{\Lambda_T^{1/2} U_T^\top B_0}.$$

From this SVD, $U_J^\top = (\Lambda_S^{-1/2} U_S^\top \otimes \Lambda_T^{-1/2} U_T^\top) J^\top$, so by (15) $U_J^\top Z = -(\Lambda_S^{-1/2} U_S^\top \otimes \Lambda_T^{-1/2} U_T^\top)\operatorname{vec}(\nabla)$. Since U_J has orthonormal columns,

$$\begin{aligned}
\mathcal{E}_\infty &= \left\| U_J(I - \Pi_G)U_J^\top Z \right\|_2 = \left\| (I - \Pi_G)U_J^\top Z \right\|_2 \\
&= \left\| \left[(I - \Pi_A) \otimes (I - \Pi_B)\right] \left[\Lambda_S^{-1/2} U_S^\top \otimes \Lambda_T^{-1/2} U_T^\top\right] \operatorname{vec}(\nabla) \right\|_2 \\
&= \left\| \left[(I - \Pi_A)\Lambda_S^{-1/2} U_S^\top \otimes (I - \Pi_B)\Lambda_T^{-1/2} U_T^\top\right] \operatorname{vec}(\nabla) \right\|_2 \\
&= \left\| \operatorname{vec}\left((I - \Pi_B)\Lambda_T^{-1/2} U_T^\top \nabla U_S \Lambda_S^{-1/2}(I - \Pi_A)\right) \right\|_2 = \|(I - \Pi_B)F(I - \Pi_A)\|_F,
\end{aligned}$$

by the mixed-product rule, then $\operatorname{vec}(\Gamma_1 \Psi \Gamma_2) = (\Gamma_2^\top \otimes \Gamma_1)\operatorname{vec}(\Psi)$ with $\Gamma_1 = (I - \Pi_B)\Lambda_T^{-1/2} U_T^\top$, $\Gamma_2 = U_S \Lambda_S^{-1/2}(I - \Pi_A)$, and the definition of F. This is (6). □

B.2 SECTION 5

Proof of Theorem 5.1. Write $\Pi_A := \Pi_{\Lambda_S^{1/2} U_S^\top A_0^\top}$ and $\Pi_B := \Pi_{\Lambda_T^{1/2} U_T^\top B_0}$, so that (6) reads $\mathcal{E}_\infty = \|(I - \Pi_B)F(I - \Pi_A)\|_F$.

(i) Expanding the product,

$$(I - \Pi_B)F(I - \Pi_A) = F - \big(\Pi_B F + (I - \Pi_B)F\Pi_A\big),$$

and the subtracted matrix has rank at most $2r$, each of its two terms having rank at most r. By the Eckart–Young–Mirsky theorem (Eckart & Young, 1936; Mirsky, 1960), $\mathcal{E}_\infty \geq \big(\sum_{i>2r} s_i^2(F)\big)^{1/2}$.

(ii) Let $A_0^\top = R_{\mathcal{I}}(F)Q_A = U_S\Lambda_S^{-1/2}V_{\mathcal{I}}Q_A$ and $B_0 = L_{\mathcal{J}}(F)Q_B = U_T\Lambda_T^{-1/2}U_{\mathcal{J}}Q_B$ with Q_A, Q_B invertible. Since $U_S^\top U_S = I$, $\Lambda_S^{1/2}U_S^\top A_0^\top = V_{\mathcal{I}}Q_A$, whose column space is $\mathrm{col}(V_{\mathcal{I}})$; hence $\Pi_A = V_{\mathcal{I}}V_{\mathcal{I}}^\top$, and likewise $\Pi_B = U_{\mathcal{J}}U_{\mathcal{J}}^\top$. With $F = \sum_i s_i u_i v_i^\top$, $(I-\Pi_B)u_i = 0$ for $i \in \mathcal{J}$ and u_i otherwise, and $(I-\Pi_A)v_i = 0$ for $i \in \mathcal{I}$ and v_i otherwise, so

$$(I-\Pi_B)F(I-\Pi_A) = \sum_{i \notin \mathcal{I}\cup\mathcal{J}} s_i u_i v_i^\top, \qquad \mathcal{E}_\infty = \Big(\sum_{i\notin\mathcal{I}\cup\mathcal{J}} s_i^2\Big)^{1/2}.$$

When $\mathcal{I}\cup\mathcal{J} \supseteq [\min(2r, \operatorname{rank} F)]$, every index outside $\mathcal{I}\cup\mathcal{J}$ exceeds $2r$ or $\operatorname{rank} F$, and this is the bound of (i).

□

Proof of Proposition 5.2. Since $T^{1/2} = U_T\Lambda_T^{1/2}U_T^\top$, $S^{1/2} = U_S\Lambda_S^{1/2}U_S^\top$ and U_T, U_S have orthonormal columns, for every $\Psi \in \mathbb{R}^{d_{\mathrm{out}}\times d_{\mathrm{in}}}$

$$\left\|T^{1/2}\Psi S^{1/2}\right\|_F = \left\|\Lambda_T^{1/2}U_T^\top\Psi U_S\Lambda_S^{1/2}\right\|_F. \tag{8}$$

(i) $D_1 = \nabla - \eta^2 B_0B_0^\top\nabla - \eta^2\nabla A_0^\top A_0$, so by (6)

$$\mathcal{E}_1 = \left\|T^{1/2}\nabla S^{1/2} - E\right\|_F, \qquad E := T^{1/2}\big(\eta^2 B_0B_0^\top\nabla + \eta^2\nabla A_0^\top A_0\big)S^{1/2},$$

and E has rank at most $2r$, each of its two terms having rank at most r. Since $T^{1/2}\nabla S^{1/2} = U_T\widetilde{G}U_S^\top$ has the singular values $\tilde{s}_i$ of $\widetilde{G}$, the Eckart–Young–Mirsky theorem gives $\mathcal{E}_1 \ge \big(\sum_{i>2r}\tilde{s}_i^2\big)^{1/2}$.

(ii) Let $\widetilde{G} = \sum_i \tilde{s}_i\tilde{u}_i\tilde{v}_i^\top$ be a thin SVD, $\tilde{U} := (\tilde{u}_i)_{i\le r}$, $\tilde{V} := (\tilde{v}_i)_{i\le r}$, $\Sigma_r := \mathrm{diag}(\tilde{s}_1,\dots,\tilde{s}_r)$, $\widetilde{G}_{>r} := \widetilde{G} - \tilde{U}\Sigma_r\tilde{V}^\top$, and let $\tilde{U}_\perp$, $\tilde{V}_\perp$ be orthonormal bases of the orthogonal complements of $\mathrm{col}(\tilde{U})$ and $\mathrm{col}(\tilde{V})$. The leakage constants of the statement are

$$\begin{aligned} \Theta_T &:= (\tilde{U}^\top\Lambda_T^{-1}\tilde{U})^{-1}\tilde{U}^\top\Lambda_T^{-1}, \quad \beta_T := \left\|\Theta_T\tilde{U}_\perp\right\|_2, \\ \Theta_S &:= (\tilde{V}^\top\Lambda_S^{-1}\tilde{V})^{-1}\tilde{V}^\top\Lambda_S^{-1}, \quad \beta_S := \left\|\Theta_S\tilde{V}_\perp\right\|_2; \end{aligned} \tag{9}$$

note that $\Theta_T\tilde{U} = I_r$ and $\Theta_S\tilde{V} = I_r$.

Let $A_0^\top = \frac{1}{\sqrt{2}\eta}\operatorname{orth}(R)\,Q_A$ and $B_0 = \frac{1}{\sqrt{2}\eta}\operatorname{orth}(L)\,Q_B$ with Q_A, Q_B orthogonal, where $R := R_{[r]}(\widetilde{G}) = U_S\Lambda_S^{-1/2}\tilde{V}$ and $L := L_{[r]}(\widetilde{G}) = U_T\Lambda_T^{-1/2}\tilde{U}$. Since Q_A is orthogonal and $\operatorname{orth}(R)\operatorname{orth}(R)^\top = \Pi_R = R(R^\top R)^{-1}R^\top$,

$$\eta^2 A_0^\top A_0 = \tfrac{1}{2}\Pi_R = \tfrac{1}{2}\,U_S\Lambda_S^{-1/2}\tilde{V}K_A^{-1}\tilde{V}^\top\Lambda_S^{-1/2}U_S^\top, \qquad K_A := R^\top R = \tilde{V}^\top\Lambda_S^{-1}\tilde{V},$$

and likewise $\eta^2 B_0B_0^\top = \frac{1}{2}\Pi_L = \frac{1}{2}\,U_T\Lambda_T^{-1/2}\tilde{U}K_B^{-1}\tilde{U}^\top\Lambda_T^{-1/2}U_T^\top$ with $K_B := \tilde{U}^\top\Lambda_T^{-1}\tilde{U}$. Using $U_T^\top U_T = I$, $U_S^\top U_S = I$ and the definition of $\widetilde{G}$,

$$\begin{aligned} \Lambda_T^{1/2}U_T^\top\big(\eta^2 B_0B_0^\top\nabla\big)U_S\Lambda_S^{1/2} &= \tfrac{1}{2}\tilde{U}K_B^{-1}\tilde{U}^\top\Lambda_T^{-1}\widetilde{G} = \tfrac{1}{2}\tilde{U}\Theta_T\widetilde{G}, \\ \Lambda_T^{1/2}U_T^\top\big(\eta^2\nabla A_0^\top A_0\big)U_S\Lambda_S^{1/2} &= \tfrac{1}{2}\widetilde{G}\Lambda_S^{-1}\tilde{V}K_A^{-1}\tilde{V}^\top = \tfrac{1}{2}\widetilde{G}\Theta_S^\top\tilde{V}^\top. \end{aligned}$$

Hence, by (6) and (8),

$$\mathcal{E}_1 = \left\|\widetilde{G} - \tfrac{1}{2}\tilde{U}\Theta_T\widetilde{G} - \tfrac{1}{2}\widetilde{G}\Theta_S^\top\tilde{V}^\top\right\|_F.$$

Insert $\widetilde{G} = \tilde{U}\Sigma_r\tilde{V}^\top + \widetilde{G}_{>r}$: since $\Theta_T\tilde{U} = I$ and $\tilde{V}^\top\Theta_S^\top = I$, $\tilde{U}\Theta_T\widetilde{G} = \tilde{U}\Sigma_r\tilde{V}^\top + \tilde{U}\Theta_T\widetilde{G}_{>r}$ and $\widetilde{G}\Theta_S^\top\tilde{V}^\top = \tilde{U}\Sigma_r\tilde{V}^\top + \widetilde{G}_{>r}\Theta_S^\top\tilde{V}^\top$, so

$$\widetilde{G} - \tfrac{1}{2}\tilde{U}\Theta_T\widetilde{G} - \tfrac{1}{2}\widetilde{G}\Theta_S^\top\tilde{V}^\top = \widetilde{G}_{>r} - \tfrac{1}{2}\tilde{U}\Theta_T\widetilde{G}_{>r} - \tfrac{1}{2}\widetilde{G}_{>r}\Theta_S^\top\tilde{V}^\top.$$

The columns of $\widetilde{G}_{>r}$ lie in $\mathrm{col}(\tilde{U}_\perp)$ and its rows in $\mathrm{col}(\tilde{V}_\perp)$, so the three terms have their columns in $\mathrm{col}(\tilde{U}_\perp)$, $\mathrm{col}(\tilde{U})$, $\mathrm{col}(\tilde{U}_\perp)$ and their rows in $\mathrm{col}(\tilde{V}_\perp)$, $\mathrm{col}(\tilde{V}_\perp)$, $\mathrm{col}(\tilde{V})$ respectively: they are mutually orthogonal for the Frobenius inner product, and, $\tilde{U}$ and $\tilde{V}$ having orthonormal columns,

$$\mathcal{E}_1^2 = \left\|\widetilde{G}_{>r}\right\|_F^2 + \tfrac{1}{4}\left\|\Theta_T\widetilde{G}_{>r}\right\|_F^2 + \tfrac{1}{4}\left\|\widetilde{G}_{>r}\Theta_S^\top\right\|_F^2.$$

Finally $\widetilde{G}_{>r} = \tilde{U}_\perp\tilde{U}_\perp^\top\widetilde{G}_{>r}$ gives $\left\|\Theta_T\widetilde{G}_{>r}\right\|_F \le \left\|\Theta_T\tilde{U}_\perp\right\|_2\left\|\widetilde{G}_{>r}\right\|_F = \beta_T\left\|\widetilde{G}_{>r}\right\|_F$, and symmetrically $\left\|\widetilde{G}_{>r}\Theta_S^\top\right\|_F \le \beta_S\left\|\widetilde{G}_{>r}\right\|_F$; with $\left\|\widetilde{G}_{>r}\right\|_F^2 = \sum_{i>r}\tilde{s}_i^2$ this is the bound.

(iii) Let $S = \sigma I$ and $T = \tau I$ with $\sigma, \tau > 0$. Then $\Lambda_S = \sigma I$, $\Lambda_T = \tau I$, U_S and U_T are square orthogonal, and $\widetilde{G} = (\sigma\tau)^{1/2}\,U_T^\top\nabla U_S$ is a positive multiple of ∇ written in the orthonormal bases U_T, U_S. Moreover $\Theta_T = (\tilde{U}^\top\tilde{U}/\tau)^{-1}\tilde{U}^\top/\tau = \tilde{U}^\top$, so $\beta_T = \left\|\tilde{U}^\top\tilde{U}_\perp\right\|_2 = 0$, and likewise $\beta_S = 0$. In the identity for $\mathcal{E}_1^2$ proved in (ii), the last two terms are then $\frac{1}{4}\left\|\tilde{U}^\top\widetilde{G}_{>r}\right\|_F^2 = 0$ and $\frac{1}{4}\left\|\widetilde{G}_{>r}\tilde{V}\right\|_F^2 = 0$, so $\mathcal{E}_1^2 = \sum_{i>r}\tilde{s}_i^2$. □

LoRA-One (Zhang et al., 2025a) is analyzed in the following model, in which our constructions recover its subspaces.

Proposition B.1 (LoRA-One's setting)**.** *Consider one layer $f(W;x) = g(Wx)$, with g the identity or the ReLU, $n \ge d_{\mathrm{in}}$ independent inputs $x_i \sim \mathcal{N}(0, I_{d_{\mathrm{in}}})$, and targets $y_i = g\big((W_0 + \Delta^\star)x_i\big)$ with $\mathrm{rank}\,\Delta^\star = r$. Let $c := n$ for the identity and $c := n/2$ for the ReLU, where we assume that W_0 has nonzero rows. Write $\epsilon := \|S - I\|_2$ and $\tau := \|T/c - I\|_2$, and let both constructions use $\mathcal{I} = \mathcal{J} = [r]$.*

LoRA-One's subspaces are $\mathrm{col}(U_\nabla)$ and $\mathrm{col}(V_\nabla)$, with U_∇ and V_∇ the top-r left and right singular vectors of ∇, which are unique when $s_r(\nabla) > s_{r+1}(\nabla)$. For matrices Γ and Ψ whose column spaces have the same dimension, $\sin\angle(\Gamma,\Psi) := \|(I - \Pi_\Psi)\Pi_\Gamma\|_2$ is the sine of the largest principal angle between $\mathrm{col}(\Gamma)$ and $\mathrm{col}(\Psi)$. The distance of (A_0, B_0) to LoRA-One's subspaces is

$$\sin\phi := \max\big\{\sin\angle(A_0^\top, V_\nabla),\ \sin\angle(B_0, U_\nabla)\big\}.$$

(i) *$\mathbb{E}\,S = I$ and $\mathbb{E}\,T = cI$; with probability at least $1 - \delta$, $\epsilon \le 2\varrho + \varrho^2$ with $\varrho := \big(\sqrt{d_{\mathrm{in}}} + \sqrt{2\log(4/\delta)}\big)/\sqrt{n}$, and $\tau = 0$ (linear) or $\tau \le \sqrt{2\log(4d_{\mathrm{out}}/\delta)/n}$ (ReLU).*

(ii) *Linear case: the colored construction returns LoRA-One's subspaces exactly, $\mathrm{col}(B_0) = \mathrm{col}(\Delta^\star)$ and $\mathrm{col}(A_0^\top) = \mathrm{col}(S\Delta^{\star\top})$; the whitened one returns $\mathrm{col}(\Delta^\star)$ and $\mathrm{col}(\Delta^{\star\top})$, with $\sin\phi \le \epsilon$.*

(iii) *ReLU case: if $\epsilon, \tau \le \frac{1}{4}$ and $s_r(\nabla) > s_{r+1}(\nabla)$, both constructions satisfy $\sin\phi \le \min\big\{1,\ 2.6\,(\epsilon+\tau)\,s_r(\nabla)/(s_r(\nabla) - s_{r+1}(\nabla))\big\}$.*

In the linear case the colored construction is LoRA-One, and the whitened one differs from it only by the sampling noise $S - I$ of the inputs. The bound of (iii) is attained up to a constant factor (last paragraph of the proof).

Proof of Proposition B.1. Proof of (i). The layer is the whole model, so $h_i = x_i$, $S = \frac{1}{n}\sum_i x_i x_i^\top$ and $\mathbb{E}\,S = I$. In the linear case $\delta_i = I$, so $T = nI$. In the ReLU case $\delta_i = \mathrm{diag}\,\big(\mathbf{1}\{w_k^\top x_i > 0\}\big)_k$, with w_k the rows of W_0, so $T = \sum_i \delta_i$ is diagonal; each diagonal entry has mean $n/2$, because $x \mapsto -x$ preserves the law of x and exchanges the events $\{w_k^\top x > 0\}$ and $\{w_k^\top x < 0\}$, whose union has probability one.

For the concentration of S, let $X \in \mathbb{R}^{n\times d_{\mathrm{in}}}$ have the rows $x_i^\top$, so that $S = X^\top X/n$ and $s_i(S) = s_i(X)^2/n$. The entries of X are independent standard Gaussians, so for every $u \ge 0$, with probability at least $1 - 2e^{-u^2/2}$ (Vershynin, 2012, Corollary 5.35),

$$\sqrt{n} - \sqrt{d_{\mathrm{in}}} - u \ \le\ s_{d_{\mathrm{in}}}(X) \ \le\ s_1(X) \ \le\ \sqrt{n} + \sqrt{d_{\mathrm{in}}} + u.$$

Take $u = \sqrt{2\log(4/\delta)}$, so that $2e^{-u^2/2} = \delta/2$, and divide by $\sqrt{n}$:

$$\left|\frac{s_i(X)}{\sqrt{n}} - 1\right| \le \varrho \qquad \text{for every } i \le d_{\text{in}}.$$

As S is symmetric positive semidefinite, the singular values of $S - I$ are the $|s_i(S) - 1|$, hence

$$\epsilon = \max_{i \le d_{\text{in}}} \left|\frac{s_i(X)^2}{n} - 1\right| = \max_{i \le d_{\text{in}}} \left|\frac{s_i(X)}{\sqrt{n}} - 1\right| \left(\frac{s_i(X)}{\sqrt{n}} + 1\right) \le \varrho\,(2 + \varrho) = 2\varrho + \varrho^2.$$

For the concentration of T: in the linear case $\tau = 0$. In the ReLU case $T/c = \operatorname{diag}(2\hat{p}_k)$ with $\hat{p}_k := \frac{1}{n}\sum_i \mathbf{1}\{w_k^\top x_i > 0\}$. For each k, the indicators $\mathbf{1}\{w_k^\top x_i > 0\}$, $i \le n$, are independent, take values in $\{0, 1\}$ and have mean $\frac{1}{2}$, so Hoeffding's inequality (Hoeffding, 1963) gives, for every $t \ge 0$,

$$\Pr\left(|2\hat{p}_k - 1| \ge t\right) = \Pr\left(|\hat{p}_k - \tfrac{1}{2}| \ge \tfrac{t}{2}\right) \le 2e^{-nt^2/2}.$$

$T/c - I$ is diagonal, so $\tau = \max_k |2\hat{p}_k - 1|$, and a union bound over the d_{out} rows gives $\Pr(\tau \ge t) \le 2d_{\text{out}}e^{-nt^2/2}$. With $t = \sqrt{2\log(4d_{\text{out}}/\delta)/n}$ the right-hand side is $\delta/2$. The events on S and on T hold together with probability at least $1 - \delta$.

Proof of (ii). Here Z has the blocks $y_i - W_0 x_i = \Delta^\star x_i$ and $T = nI$, so

$$\nabla = -\sum_i \Delta^\star x_i x_i^\top = -n\,\Delta^\star S.$$

Since $n \ge d_{\text{in}}$, X has full column rank with probability one. So $S = X^\top X/n$ is invertible almost surely, $\operatorname{rank}\nabla = r$, and LoRA-One's subspaces are

$$\operatorname{col}(\nabla) = \operatorname{col}(\Delta^\star), \qquad \operatorname{col}(\nabla^\top) = \operatorname{col}(S\Delta^{\star\top}).$$

Take a thin SVD $S = U_S \Lambda_S U_S^\top$, with U_S square, and $U_T = I$, $\Lambda_T = nI$ (any orthonormal U_T gives the same retained subspaces). Then

$$\widetilde{G} = \Lambda_T^{1/2} \nabla U_S \Lambda_S^{1/2} = -n^{3/2}\,\Delta^\star U_S \Lambda_S^{3/2}, \qquad F = \Lambda_T^{-1/2} \nabla U_S \Lambda_S^{-1/2} = -n^{1/2}\,\Delta^\star U_S \Lambda_S^{1/2}.$$

Both have rank r, so their top-r singular vectors span their whole column and row spaces. On the left, both column spaces are $\operatorname{col}(\Delta^\star)$, which $L_{[r]} = \Lambda_T^{-1/2} U_{[r]}$ leaves unchanged: $\operatorname{col}(B_0) = \operatorname{col}(\Delta^\star)$. On the right, the row spaces are $\operatorname{col}(\Lambda_S^p U_S^\top \Delta^{\star\top})$ with $p = \frac{3}{2}$ (colored) and $p = \frac{1}{2}$ (whitened), and $R_{[r]} = U_S \Lambda_S^{-1/2} V_{[r]}$ maps them to

$$\operatorname{col}(A_0^\top) = \operatorname{col}\left(U_S \Lambda_S^{p-1/2} U_S^\top \Delta^{\star\top}\right) = \begin{cases} \operatorname{col}(S\Delta^{\star\top}) & \text{colored,} \\ \operatorname{col}(\Delta^{\star\top}) & \text{whitened.} \end{cases}$$

The colored construction thus returns LoRA-One's subspaces. The whitened one returns LoRA-One's left subspace, and on the right $\operatorname{col}(\Delta^{\star\top})$ instead of $\operatorname{col}(S\Delta^{\star\top})$. Since $(I - \Pi_{S\Delta^{\star\top}})\, S\, \Pi_{\Delta^{\star\top}} = (I - \Pi_{S\Delta^{\star\top}})\, S\Delta^{\star\top} (\Delta^{\star\top})^\dagger = 0$,

$$\sin\phi = \|(I - \Pi_{S\Delta^{\star\top}})\Pi_{\Delta^{\star\top}}\|_2 = \|(I - \Pi_{S\Delta^{\star\top}})(I - S)\Pi_{\Delta^{\star\top}}\|_2 \le \|I - S\|_2 = \epsilon.$$

Proof of (iii). If $2(\epsilon + \tau)s_r(\nabla) > s_r(\nabla) - s_{r+1}(\nabla)$, the bound of the proposition is 1 and there is nothing to prove. Otherwise

$$\boxed{(1 - \epsilon - \tau)\, s_r(\nabla) \;\ge\; \tfrac{1}{2}\left(s_r(\nabla) + s_{r+1}(\nabla)\right)} \tag{10}$$

Multiplying T by a positive scalar multiplies F, $\widetilde{G}$ and $L_{[r]}$ by positive scalars, and changes neither $\operatorname{col}(A_0^\top)$ nor $\operatorname{col}(B_0)$. We therefore replace T by T/c, so that $\tau = \|T - I\|_2$. Since $\epsilon, \tau \le \frac{1}{4}$, S and T are invertible, with singular values in $[1 - \epsilon, 1 + \epsilon]$ and $[1 - \tau, 1 + \tau]$, so U_S and U_T are square orthogonal matrices, and

$$\nabla = U_T \Lambda_T^{1/2}\, F\, \Lambda_S^{1/2} U_S^\top = U_T \Lambda_T^{-1/2}\, \widetilde{G}\, \Lambda_S^{-1/2} U_S^\top.$$

Let Γ be F or $\widetilde{G}$, let $U_{[r]}$ and $V_{[r]}$ be its top-r left and right singular vectors, and $\Sigma_r := \mathrm{diag}(s_1(\Gamma), \dots, s_r(\Gamma))$, so that $\Gamma V_{[r]} = U_{[r]}\Sigma_r$ and $\Gamma^\top U_{[r]} = V_{[r]}\Sigma_r$. Write

$$R := R_{[r]}(\Gamma) = U_S \Lambda_S^{-1/2} V_{[r]}, \qquad L := L_{[r]}(\Gamma) = U_T \Lambda_T^{-1/2} U_{[r]};$$

for both constructions, $\mathrm{col}(A_0^\top) = \mathrm{col}(R)$ and $\mathrm{col}(B_0) = \mathrm{col}(L)$. Substituting the factorizations of ∇ gives

$$\boxed{\begin{aligned} \Gamma = F: &\qquad \nabla R = U_T \Lambda_T^{1/2} F V_{[r]} = T L \Sigma_r, &&\qquad \nabla^\top L = U_S \Lambda_S^{1/2} F^\top U_{[r]} = S R \Sigma_r, \\ \Gamma = \widetilde{G}: &\qquad \nabla S R = U_T \Lambda_T^{-1/2} \widetilde{G} V_{[r]} = L \Sigma_r, &&\qquad \nabla^\top T L = U_S \Lambda_S^{-1/2} \widetilde{G}^\top U_{[r]} = R \Sigma_r \end{aligned}} \tag{11}$$

Applying $s_r(\Gamma_1 \Psi \Gamma_2) \le \|\Gamma_1\|_2 \, s_r(\Psi) \, \|\Gamma_2\|_2$ to the two factorizations of ∇ gives $s_r(\nabla) \le \sqrt{(1+\epsilon)(1+\tau)}\, s_r(F)$ and $s_r(\nabla) \le s_r(\widetilde{G})/\sqrt{(1-\epsilon)(1-\tau)}$. As $1/(1+x) \ge 1-x$ and $(1-\epsilon)(1-\tau) \ge 1-\epsilon-\tau \ge (1-\epsilon-\tau)^2$, in both cases $s_r(\Gamma) \ge \sqrt{(1-\epsilon)(1-\tau)}\, s_r(\nabla) \ge (1-\epsilon-\tau)\, s_r(\nabla)$, and (10) gives

$$\boxed{s_r(\Gamma) \;\ge\; \tfrac{1}{2}\big(s_r(\nabla) + s_{r+1}(\nabla)\big) \;>\; s_{r+1}(\nabla)} \tag{12}$$

Since U_S is orthogonal and $V_{[r]}$ has orthonormal columns, the singular values of $R = U_S \Lambda_S^{-1/2} V_{[r]}$ lie in $[(1+\epsilon)^{-1/2}, (1-\epsilon)^{-1/2}]$; as $\epsilon \le \frac{1}{4}$,

$$\|R\|_2 \le \frac{2}{\sqrt{3}}, \qquad \left\|R^\dagger\right\|_2 \le \frac{\sqrt{5}}{2},$$

and the same holds for L with τ.

To measure how much of the retained modes lies outside LoRA-One's subspaces, let

$$\xi := \|(I - \Pi_{V_\nabla}) R\|_2 , \qquad \zeta := \|(I - \Pi_{U_\nabla}) L\|_2 .$$

By the SVD of ∇,

$$(I - \Pi_{U_\nabla})\, \nabla = \nabla\, (I - \Pi_{V_\nabla}), \qquad \|(I - \Pi_{U_\nabla})\, \nabla\|_2 = s_{r+1}(\nabla).$$

For the whitened construction, (11) gives $SR = \nabla^\top L\, \Sigma_r^{-1}$ and $TL = \nabla R\, \Sigma_r^{-1}$. Writing $S = I + (S - I)$ and $T = I + (T - I)$,

$$R = \nabla^\top L\, \Sigma_r^{-1} - (S - I) R, \qquad L = \nabla R\, \Sigma_r^{-1} - (T - I) L.$$

Multiply the first on the left by $I - \Pi_{V_\nabla}$ and the second by $I - \Pi_{U_\nabla}$, and move the projectors across $\nabla^\top$ and ∇ with the relation above and its transpose, $(I - \Pi_{V_\nabla})\nabla^\top = \nabla^\top (I - \Pi_{U_\nabla})$:

$$\begin{aligned} (I - \Pi_{V_\nabla}) R &= \nabla^\top (I - \Pi_{U_\nabla}) L\, \Sigma_r^{-1} - (I - \Pi_{V_\nabla})(S - I) R, \\ (I - \Pi_{U_\nabla}) L &= \nabla (I - \Pi_{V_\nabla}) R\, \Sigma_r^{-1} - (I - \Pi_{U_\nabla})(T - I) L. \end{aligned}$$

We take norms using $\left\|\Sigma_r^{-1}\right\|_2 = 1/s_r(F)$, $\|(S - I)R\|_2 \le 2\epsilon/\sqrt{3}$, $\|(T - I)L\|_2 \le 2\tau/\sqrt{3}$ and

$$\left\|\nabla^\top (I - \Pi_{U_\nabla}) L\right\|_2 \le s_{r+1}(\nabla)\, \zeta, \qquad \|\nabla (I - \Pi_{V_\nabla}) R\|_2 \le s_{r+1}(\nabla)\, \xi,$$

which hold because $\nabla^\top (I - \Pi_{U_\nabla}) L = \big((I - \Pi_{U_\nabla})\nabla\big)^\top (I - \Pi_{U_\nabla}) L$ and $\nabla (I - \Pi_{V_\nabla}) R = \nabla (I - \Pi_{V_\nabla})(I - \Pi_{V_\nabla}) R$. This gives

$$s_r(F)\, \xi \le s_{r+1}(\nabla)\, \zeta + \frac{2\epsilon}{\sqrt{3}}\, s_r(F), \qquad s_r(F)\, \zeta \le s_{r+1}(\nabla)\, \xi + \frac{2\tau}{\sqrt{3}}\, s_r(F).$$

For the colored construction, (11) gives

$$R = \nabla^\top T L\, \Sigma_r^{-1} = \nabla^\top \big(L + (T - I)L\big) \Sigma_r^{-1}, \qquad L = \nabla S R\, \Sigma_r^{-1} = \nabla \big(R + (S - I)R\big) \Sigma_r^{-1},$$

and the same projections give

$$\begin{aligned} (I - \Pi_{V_\nabla}) R &= \nabla^\top (I - \Pi_{U_\nabla}) \big(L + (T - I)L\big) \Sigma_r^{-1}, \\ (I - \Pi_{U_\nabla}) L &= \nabla (I - \Pi_{V_\nabla}) \big(R + (S - I)R\big) \Sigma_r^{-1}, \end{aligned}$$

and taking norms in the same way gives

$$s_r(\widetilde{G})\,\xi \le s_{r+1}(\nabla)\Big(\zeta + \frac{2\tau}{\sqrt{3}}\Big), \qquad s_r(\widetilde{G})\,\zeta \le s_{r+1}(\nabla)\Big(\xi + \frac{2\epsilon}{\sqrt{3}}\Big).$$

Adding the two inequalities of each pair gives

$$\big(s_r(F) - s_{r+1}(\nabla)\big)(\xi+\zeta) \le \frac{2}{\sqrt{3}}(\epsilon+\tau)\,s_r(F) \qquad \text{(whitened)},$$
$$\big(s_r(\widetilde{G}) - s_{r+1}(\nabla)\big)(\xi+\zeta) \le \frac{2}{\sqrt{3}}(\epsilon+\tau)\,s_{r+1}(\nabla) \quad \text{(colored)}.$$

In the colored case, the right-hand side is even smaller than in the whitened one, since $s_{r+1}(\nabla) < s_r(\widetilde{G})$ by (12); bounding it by $\frac{2}{\sqrt{3}}(\epsilon+\tau)\,s_r(\widetilde{G})$ puts both cases in the same form:

$$\boxed{\big(s_r(\Gamma) - s_{r+1}(\nabla)\big)\,(\xi+\zeta) \;\le\; \frac{2}{\sqrt{3}}\,(\epsilon+\tau)\,s_r(\Gamma)} \tag{13}$$

It remains to pass from ξ and ζ to the angles. Since $\mathrm{col}(A_0^\top) = \mathrm{col}(R)$, $\Pi_{A_0^\top} = RR^\dagger$, so

$$\sin\angle(A_0^\top, V_\nabla) = \big\|(I - \Pi_{V_\nabla})RR^\dagger\big\|_2 \le \big\|R^\dagger\big\|_2\,\xi \le \frac{\sqrt{5}}{2}\,\xi,$$

and likewise $\sin\angle(B_0, U_\nabla) \le \frac{\sqrt{5}}{2}\,\zeta$. Hence, by (13),

$$\sin\phi \;\le\; \frac{\sqrt{5}}{2}\,(\xi+\zeta) \;\le\; \sqrt{\frac{5}{3}}\,(\epsilon+\tau)\,\frac{s_r(\Gamma)}{s_r(\Gamma) - s_{r+1}(\nabla)}.$$

The function $x \mapsto x/(x - s_{r+1}(\nabla)) = 1 + s_{r+1}(\nabla)/(x - s_{r+1}(\nabla))$ decreases on $(s_{r+1}(\nabla), \infty)$, and $s_r(\Gamma) \ge \frac{1}{2}(s_r(\nabla) + s_{r+1}(\nabla))$ by (12), so

$$\begin{aligned}\frac{s_r(\Gamma)}{s_r(\Gamma) - s_{r+1}(\nabla)} \;&\le\; \frac{\frac{1}{2}\big(s_r(\nabla) + s_{r+1}(\nabla)\big)}{\frac{1}{2}\big(s_r(\nabla) + s_{r+1}(\nabla)\big) - s_{r+1}(\nabla)} \\ &=\; \frac{s_r(\nabla) + s_{r+1}(\nabla)}{s_r(\nabla) - s_{r+1}(\nabla)} \;\le\; \frac{2\,s_r(\nabla)}{s_r(\nabla) - s_{r+1}(\nabla)}.\end{aligned}$$

Together, $\sin\phi \le 2\sqrt{5/3}\,(\epsilon+\tau)\,s_r(\nabla)/(s_r(\nabla) - s_{r+1}(\nabla))$, and $2\sqrt{5/3} < 2.6$.

This bound is attained up to a constant factor. Take $T = I$ and $S = I + \epsilon\,(v_r v_{r+1}^\top + v_{r+1} v_r^\top)$, with v_r and v_{r+1} the r-th and $(r+1)$-th right singular vectors of ∇. To first order in ϵ, the whitened construction gives

$$\sin\angle(A_0^\top, V_\nabla) = \frac{\epsilon\,s_r^2(\nabla)}{s_r^2(\nabla) - s_{r+1}^2(\nabla)} \;\ge\; \frac{\epsilon\,s_r(\nabla)}{2\,(s_r(\nabla) - s_{r+1}(\nabla))},$$

which is the bound of (iii) divided by 5.2. □

Proof of Proposition 5.3. Step 1: the singular values of P on $\mathrm{col}\,H$. Write $\Sigma_A = \mathrm{diag}(a_1, \dots, a_r)$ and $\Sigma_B = \mathrm{diag}(b_1, \dots, b_r)$ with $a_1 \ge \cdots \ge a_r > 0$ and $b_1 \ge \cdots \ge b_r > 0$, complete the columns of V and U into orthonormal bases $(v_k)_{k \le d_{\mathrm{in}}}$ and $(u_l)_{l \le d_{\mathrm{out}}}$, and set $a_k := 0$ for $k > r$, $b_l := 0$ for $l > r$. Since $(\Gamma_1 \otimes \Gamma_2)(v \otimes u) = \Gamma_1 v \otimes \Gamma_2 u$,

$$P\,(v_k \otimes u_l) = \big(V\Sigma_A V^\top v_k\big) \otimes u_l + v_k \otimes \big(U\Sigma_B U^\top u_l\big) = (a_k + b_l)\,(v_k \otimes u_l),$$

so the $v_k \otimes u_l$ form an orthonormal basis of singular vectors of P, with singular values $a_k + b_l$. As $P = HH^\top$, $\mathrm{col}\,H = \mathrm{col}\,P$ is spanned by the singular vectors with $a_k + b_l > 0$, that is, with $k \le r$ or $l \le r$. Since $r < \min(d_{\mathrm{in}}, d_{\mathrm{out}})$, the pairs $(k,l) = (r, r+1)$ and $(r+1, r)$ occur, and every pair with $k, l \le r$ gives $a_k + b_l \ge a_r + b_r > \min(a_r, b_r)$; hence

$$\lambda_{\max} = a_1 + b_1, \qquad \lambda_{\min} = \min(a_r, b_r), \qquad \kappa_0 = \frac{a_1 + b_1}{\min(a_r, b_r)}.$$

Step 2: the worst uncorrected direction. Expanding a unit $u \in \operatorname{col} H$ in this basis, $\|(I-P)u\|_2^2 = \sum(1-\lambda)^2 c_\lambda^2$ with $\sum c_\lambda^2 = 1$ over the singular values λ of P on $\operatorname{col} H$, which lie in $[\lambda_{\min}, \lambda_{\max}]$. The maximum over u is attained at a singular vector and equals $\max_\lambda |1-\lambda|$, and since $|1-\lambda|$ is convex in λ,

$$\max\left\{ \|(I-P)u\|_2 : \; u \in \operatorname{col} H, \; \|u\|_2 = 1 \right\} = \max\left(|1-\lambda_{\min}|, \; |1-\lambda_{\max}|\right).$$

Step 3: the first inequality. For $0 < a \le b$ and $\sigma > 0$, $\max_{\lambda\in[a,b]} |1-\sigma\lambda| = \max(|1-\sigma a|, |1-\sigma b|)$ is smallest at $\sigma = 2/(a+b)$, where the two terms are equal, and its minimum is $(b-a)/(b+a)$: this is the optimal damping of Richardson's iteration (Saad, 2003, Section 4.1). Taking $\sigma = 1$, $a = \lambda_{\min}$ and $b = \lambda_{\max}$,

$$\max\left(|1-\lambda_{\min}|, |1-\lambda_{\max}|\right) \;\ge\; \frac{\lambda_{\max}-\lambda_{\min}}{\lambda_{\max}+\lambda_{\min}} = \frac{\kappa_0 - 1}{\kappa_0 + 1},$$

with equality if and only if $\lambda_{\min} + \lambda_{\max} = 2$.

Step 4: the second inequality. By Step 1, $\kappa_0 = (a_1+b_1)/\min(a_r, b_r) \ge (a_1+b_1)/\min(a_1, b_1) \ge 2$, with equality throughout if and only if $a_1 = b_1 = a_r = b_r =: \mu$, that is, $\Sigma_A = \Sigma_B = \mu I$. Since $\kappa \mapsto (\kappa-1)/(\kappa+1)$ is increasing, $(\kappa_0-1)/(\kappa_0+1) \ge \frac{1}{3}$, with equality if and only if $\Sigma_A = \Sigma_B = \mu I$ for some $\mu > 0$.

Step 5: the value $\frac{1}{3}$. It is attained if and only if both inequalities are equalities: $\Sigma_A = \Sigma_B = \mu I$ (Step 4) and $\lambda_{\min} + \lambda_{\max} = 2$ (Step 3). Under the first, $\lambda_{\min} = \mu$ and $\lambda_{\max} = 2\mu$ by Step 1, so the second reads $3\mu = 2$. □

C AUXILIARY RESULTS

This appendix collects the supporting material: three identities (Appendix C.1), gradient descent on least squares (Appendix C.2), two lemmas on the LoRA Jacobian (Appendix C.3), and a result relating the linearized endpoint discrepancy to that of the original network (Appendix C.4).

C.1 THREE IDENTITIES

Kronecker products. For matrices of compatible sizes (Horn & Johnson, 1991),

$$\operatorname{vec}(\Gamma_1\Psi\Gamma_2) = (\Gamma_2^\top \otimes \Gamma_1)\operatorname{vec}(\Psi), \qquad (\Gamma_1 \otimes \Gamma_2)(\Gamma_3 \otimes \Gamma_4) = \Gamma_1\Gamma_3 \otimes \Gamma_2\Gamma_4,$$

and, for symmetric positive semidefinite Γ_1, Γ_2, $(\Gamma_1 \otimes \Gamma_2)^{1/2} = \Gamma_1^{1/2} \otimes \Gamma_2^{1/2}$ and $(\Gamma_1 \otimes \Gamma_2)^\dagger = \Gamma_1^\dagger \otimes \Gamma_2^\dagger$.

The LoRA map. With $H = \eta(I_{d_{\text{in}}} \otimes B_0 \;\; A_0^\top \otimes I_{d_{\text{out}}})$ and $\theta = (\operatorname{vec}(\Delta A)^\top, \operatorname{vec}(\Delta B)^\top)^\top$, the first identity gives $H\theta = \operatorname{vec}\big(\eta(B_0\Delta A + \Delta B A_0)\big)$, so $J_r = JH$, and for every $\Psi \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$

$$\operatorname{mat}\big(P\operatorname{vec}(\Psi)\big) = \eta^2\big(B_0B_0^\top\Psi + \Psi A_0^\top A_0\big), \qquad \text{in particular} \qquad \operatorname{vec}(D_1) = (I - P)\operatorname{vec}(\nabla). \tag{14}$$

The gradient. Since $\partial\mathcal{L}_n/\partial f = f - Y$, the chain rule at W_0 gives

$$\operatorname{vec}(\nabla) = J^\top\big(f(W_0; X) - Y\big) = -J^\top Z. \tag{15}$$

C.2 GRADIENT DESCENT ON LEAST SQUARES

Both linearized problems, (Full Fine-Tuning) and (LoRA), are least-squares problems $\min_u \frac{1}{2}\|\Gamma u - Z\|_2^2$, with $\Gamma = J$ and $u = \operatorname{vec}(\Delta W)$ for full fine-tuning, and $\Gamma = J_r$ and $u = \theta$ for LoRA; in both cases the predictions are $f(W_0; X) + \Gamma u$.

Lemma C.1. *Let $\Gamma \in \mathbb{R}^{nC\times q}$, $0 < \nu < 2/s_1^2(\Gamma)$, $u^{(0)} := 0$ and $u^{(k+1)} := u^{(k)} - \nu\,\Gamma^\top(\Gamma u^{(k)} - Z)$. Then $u^{(k)} \to \Gamma^\dagger Z$, the minimizer of $\|\Gamma u - Z\|_2$ of smallest Euclidean norm, and $\Gamma u^{(k)} \to \Pi_\Gamma Z$; neither limit depends on ν.*

Proof. This is the convergence of the Landweber (Richardson) iteration for least squares started at zero; see Engl et al. (1996, Theorem 6.1). □

C.3 TWO LEMMAS ON THE LORA JACOBIAN

Lemma C.2 (The difference of the two projections)**.** *Let $J = U_J D_J V_J^\top$ be a thin SVD and $G := D_J V_J^\top H$. Then $J_r = U_J G$ and*

$$\Pi_J - \Pi_{J_r} = U_J\big(I - \Pi_G\big)U_J^\top.$$

Proof. $J_r = JH = U_J D_J V_J^\top H = U_J G$. Since U_J has orthonormal columns, $(U_J G)^\dagger = G^\dagger U_J^\top$, so $\Pi_{J_r} = J_r J_r^\dagger = U_J G G^\dagger U_J^\top = U_J \Pi_G U_J^\top$, while $\Pi_J = U_J U_J^\top$. □

Lemma C.3 (Kronecker structure of the projection)**.** *Assume $M = S \otimes T$, with the thin SVDs $S = U_S\Lambda_S U_S^\top$ and $T = U_T\Lambda_T U_T^\top$ of Definition 4.2, $k_S = \operatorname{rank} S$ and $k_T = \operatorname{rank} T$. Then J has a thin SVD*

$$J = U_J\big(\Lambda_S^{1/2} \otimes \Lambda_T^{1/2}\big)(U_S \otimes U_T)^\top$$

for some $U_J \in \mathbb{R}^{nC\times k_S k_T}$ with orthonormal columns, and with $G = (\Lambda_S^{1/2} \otimes \Lambda_T^{1/2})(U_S \otimes U_T)^\top H$ as in Lemma C.2,

$$I - \Pi_G = \big(I_{k_S} - \Pi_{\Lambda_S^{1/2}U_S^\top A_0^\top}\big) \otimes \big(I_{k_T} - \Pi_{\Lambda_T^{1/2}U_T^\top B_0}\big).$$

Proof. $J^\top J = M = (U_S \otimes U_T)(\Lambda_S \otimes \Lambda_T)(U_S \otimes U_T)^\top$ is a thin SVD of $J^\top J$ (the Kronecker identities of Appendix C.1), so $U_J := J(U_S \otimes U_T)(\Lambda_S \otimes \Lambda_T)^{-1/2}$ has orthonormal columns and $J = U_J(\Lambda_S \otimes \Lambda_T)^{1/2}(U_S \otimes U_T)^\top$, which is the SVD. For the projection, the mixed-product rule gives

$$G = \eta \begin{pmatrix} \Lambda_S^{1/2} U_S^\top \otimes \Lambda_T^{1/2} U_T^\top B_0 & \Lambda_S^{1/2} U_S^\top A_0^\top \otimes \Lambda_T^{1/2} U_T^\top \end{pmatrix},$$

so, since $\eta > 0$ does not change column spaces and $\mathrm{col}(\Lambda_S^{1/2} U_S^\top) = \mathbb{R}^{k_S}$, $\mathrm{col}(\Lambda_T^{1/2} U_T^\top) = \mathbb{R}^{k_T}$,

$$\mathrm{col}(G) = \mathbb{R}^{k_S} \otimes \mathrm{col}\left(\Lambda_T^{1/2} U_T^\top B_0\right) + \mathrm{col}\left(\Lambda_S^{1/2} U_S^\top A_0^\top\right) \otimes \mathbb{R}^{k_T},$$

where $\mathcal{F} \otimes \mathcal{G}$ denotes the tensor product of two subspaces (the span of the Kronecker products $x \otimes y$, $x \in \mathcal{F}$, $y \in \mathcal{G}$), and we used $\mathrm{col}(\Gamma_1 \otimes \Gamma_2) = \mathrm{col}(\Gamma_1) \otimes \mathrm{col}(\Gamma_2)$. By the elementary facts $(\mathcal{F}+\mathcal{G})^\perp = \mathcal{F}^\perp \cap \mathcal{G}^\perp$, $(\mathbb{R}^k \otimes \mathcal{G})^\perp = \mathbb{R}^k \otimes \mathcal{G}^\perp$, $(\mathcal{F} \otimes \mathbb{R}^k)^\perp = \mathcal{F}^\perp \otimes \mathbb{R}^k$ and $(\mathcal{F}_1 \otimes \mathcal{G}_1) \cap (\mathcal{F}_2 \otimes \mathcal{G}_2) = (\mathcal{F}_1 \cap \mathcal{F}_2) \otimes (\mathcal{G}_1 \cap \mathcal{G}_2)$,

$$\mathrm{col}(G)^\perp = \mathrm{col}\left(\Lambda_S^{1/2} U_S^\top A_0^\top\right)^\perp \otimes \mathrm{col}\left(\Lambda_T^{1/2} U_T^\top B_0\right)^\perp,$$

and the orthogonal projection onto a tensor product of subspaces is the Kronecker product of the two projections, which gives the formula for $I - \Pi_G = \Pi_{\mathrm{col}(G)^\perp}$. □

C.4 The network at the two end points

Proposition 4.1 compares the two fine-tuned models inside the linearized model. The next result evaluates the network itself at the same end points: the deviation is $\mathcal{E}_\infty$ up to a term of second order in the size of the two updates, which is small when fine-tuning stays close to the pretrained weights and vanishes when f is linear in W ($\beta = 0$).

Proposition C.4 (Back to the network)**.** *Let $\Delta_r^\star := \eta(B_0 \Delta A^\star + \Delta B^\star A_0)$, so that $W_\infty = W_0 + \Delta W^\star$ and $W_{r,\infty} = W_0 + \Delta_r^\star$. Assume that $f(\,\cdot\,; X)$ is twice continuously differentiable on the segments from W_0 to W_∞ and from W_0 to $W_{r,\infty}$, with a second derivative bounded there by $\beta \geq 0$: $\left\| D^2 f(W; X)[\Delta, \Delta] \right\|_2 \leq \beta \left\| \Delta \right\|_F^2$ for every $\Delta \in \mathbb{R}^{d_{\mathrm{out}} \times d_{\mathrm{in}}}$. Then*

$$\left| \; \left\| f(W_\infty; X) - f(W_{r,\infty}; X) \right\|_2 - \mathcal{E}_\infty \; \right| \;\leq\; \frac{\beta}{2} \Big(\left\| \Delta W^\star \right\|_F^2 + \left\| \Delta_r^\star \right\|_F^2 \Big).$$

Proof of Proposition C.4. For Δ on one of the two segments, Taylor's formula with integral remainder gives

$$f(W_0 + \Delta; X) = f(W_0; X) + J \,\mathrm{vec}(\Delta) + \mathcal{R}(\Delta), \;\; \mathcal{R}(\Delta) := \int_0^1 (1-t)\, D^2 f(W_0 + t\Delta; X)[\Delta, \Delta] \, dt,$$

and the bound on the second derivative gives $\left\| \mathcal{R}(\Delta) \right\|_2 \leq \frac{\beta}{2} \left\| \Delta \right\|_F^2$. Hence

$$f(W_\infty; X) - f(W_{r,\infty}; X) = J \,\mathrm{vec}(\Delta W^\star) - J \,\mathrm{vec}(\Delta_r^\star) + \mathcal{R}(\Delta W^\star) - \mathcal{R}(\Delta_r^\star).$$

By Proposition 4.1, $J \,\mathrm{vec}(\Delta W^\star) = \Pi_J Z$ and $J \,\mathrm{vec}(\Delta_r^\star) = J_r \theta^\star = \Pi_{J_r} Z$, so the first difference has norm $\mathcal{E}_\infty$, and the reverse triangle inequality gives

$$\left| \; \left\| f(W_\infty; X) - f(W_{r,\infty}; X) \right\|_2 - \mathcal{E}_\infty \; \right| \leq \left\| \mathcal{R}(\Delta W^\star) \right\|_2 + \left\| \mathcal{R}(\Delta_r^\star) \right\|_2 \leq \frac{\beta}{2} \Big(\left\| \Delta W^\star \right\|_F^2 + \left\| \Delta_r^\star \right\|_F^2 \Big).$$

□

D EXPERIMENTAL DETAILS

D.1 PROTOCOL

Models and adapters. T5-base, the 72 query and value projections of all attention blocks (encoder self-attention, decoder self- and cross-attention) adapted, $r = 8$, $\alpha = 2r$, $\eta = \alpha/\sqrt{r}$ (Kalajdzievski, 2023) except where stated, batch size 32, no adapter dropout, fp32. Every data-dependent initialization sees the same probe: the first 32 training examples on T5-base, and 32 examples drawn with the run seed on LLaMA-2-7B, identical across arms within a seed; LoRA-GA on LLaMA-2-7B uses the 128 examples of its release. Initialization-time measurements on LLaMA-2-7B use MetaMathQA with all 224 linear layers adapted. PiSSA uses the exact SVD of every adapted weight, as in `peft`'s default, both for training and in the timings of Table 3; `peft` also offers a randomized SVD (`pissa_niter_`k), which would shorten its initialization.

Learning-rate rule. Every arm selects its learning rate on seed 42 on held-out training data, never on the validation set. On T5-base, each rate of $\{2\times10^{-5}, 5\times10^{-5}, 1\times10^{-4}, 2\times10^{-4}, 5\times10^{-4}, 8\times10^{-4}, 1\times10^{-3}\}$ is trained on 90% of a stratified split of the training set, for one epoch (1,000 steps on SST-2 and QNLI), and the rate with the highest MCC, accuracy or Pearson correlation on the other 10% is chosen. On LLaMA-2-7B the same seven rates are each trained for 1,000 steps, a third of the epoch, on 32,000 training examples with the cosine schedule compressed to those steps, and scored by the per-example loss on 512 held-out training examples; rates at which training diverges are discarded. Because a short run favours large rates, the smallest rate whose loss is within two paired standard errors of the lowest is chosen. The rules were fixed before selection; the selected rates are in Table 5.

Table 5: Selected learning rates, in units of 10^{-4}, chosen on seed 42 by the rule of Appendix D.1 and used for every seed. T5-base on the left, LLaMA-2-7B on the right. –: not run.

	CoLA	MRPC	SST-2	QNLI	STS-B	GSM8K	HumanEval
FFT, all weights	0.5	2	0.5	0.2	0.5	–	–
FFT, query/value	1	2	0.5	0.2	0.5	–	–
LoRA	8	8	5	5	10	–	–
rsLoRA	8	8	2	2	5	5	2
LoRA+	2	5	0.5	1	2	–	–
LoRA-Pro	2	5	1	0.5	1	–	–
PiSSA	5	5	2	1	5	2	1
CorDA	2	2	1	0.2	2	–	–
LoRA-GA	1	2	0.5	0.2	1	0.2	0.2
LoRA-One	10	8	1	5	8	5	5
LoRAM	5	8	2	0.5	5	–	–
CG-LoRA-W (ours)	10	8	5	5	10	5	5
CG-LoRA-C (ours)	10	10	2	2	8	5	5

Training. One epoch, cosine schedule, 3% warm-up, AdamW without weight decay, ten evaluations over the epoch (the last one is reported), seeds 42–46 (42–51 on CoLA and MRPC) controlling data order, dropout and random initializations. Standard deviations are over seeds (sample standard deviation). LLaMA-2-7B: all 224 linear layers of the 32 blocks adapted, $r = 8$, $\alpha = 16$, $\eta = \alpha/\sqrt{r}$, batch size 32 (4 per device, 8 accumulation steps), sequence length 1,024, bf16 base weights with fp32 adapters, the same optimizer and schedule, one epoch of 3,125 steps on the first 100k examples of MetaMathQA or CodeFeedback, seeds 42–44, evaluated once at the end.

The two constructions. We use reduce Kronecker factors (Eschenhagen et al., 2023) over the 32-example probe, a window $m = 2r = 16$, no power iteration on T5-base (one on LLaMA-2-7B), the gradient of the training loss (the model's token cross-entropy), and 16 random-sign probes (2 on LLaMA-2-7B) of the output logits for the output-side factor (Appendix E). They differ only in step 3 of Algorithm 1, and both use the flat point: every singular value of B_0 and of A_0 equals $\sqrt{2/3}/\eta$, so that $\eta^2 B_0 B_0^\top$ and $\eta^2 A_0^\top A_0$ have all their nonzero singular values equal to $\frac{2}{3}$.

D.2 LAYERS WHOSE GRADIENT BLOCK HAS RANK BELOW r

Theorem 5.1(ii) and Proposition 5.2(ii) need r retained modes, that is $r \le \operatorname{rank} F$ or $r \le \operatorname{rank} \widetilde{G}$, but the lower bounds (i) hold without it. On T5 this hypothesis fails on a few decoder layers, for two reasons, and for both constructions alike (Table 6). First, with reduce factors the input factor S of the first decoder block has rank at most the number of distinct decoder inputs in the probe: the input of that block is the embedding of the label sequence shifted right by the start token, which drops its last token, so all one-token labels give the same input; we measure $\operatorname{rank} S \le 2$ on every task, and the whitened and colored blocks, which carry $\Lambda_S^{\pm 1/2}$, have rank at most 2 on the three adapted matrices of that block (self-attention query and value, cross-attention query, the cross-attention value reads the encoder and is unaffected). Second, when every label is a single token the decoder has one position, its self-attention softmax is constant, and the twelve decoder self-attention queries have an identically zero gradient at initialization and throughout training: their block has rank 0 and those adapters never move, whatever their initialization.

On a layer whose block has rank $k < r$ we keep the k retained modes and complete each factor to an orthonormal frame of r columns (the QR factorization of the mode matrix), so that both factors sit at the flat point on every layer. Since $\eta B_0 A_0$ is subtracted in the forward pass (Section 3), the function is unchanged at initialization, and the $r - k$ completed directions are trained like those of a random initialization. We compared this completion with the alternatives at the learning rates of Table 5 over twenty seeds: (1) LoRA's default initialization on the $r - k$ directions, using random rows of A_0 and zero columns of B_0; (2) removing the $r - k$ directions, yielding a rank-k adapter; and (3) not adapting the three affected layers of the first decoder block. None is significantly better. Paired over the seeds, the differences to the completion are within ± 0.5 point on MRPC for both constructions and between -1.3 and -0.3 on CoLA for CG-LoRA-W. We therefore keep the completion, which needs no case distinction and keeps the flat point.

Table 6: T5-base, seed-42 probe of 32 examples, reduce factors, window $m = 16$: adapted layers (of 72) whose whitened or colored block has numerical rank below $r = 8$ (singular values above 10^{-8} of the largest), identical for the two constructions. Label lengths: number of tokens of the label strings of the task.

	label lengths	rank $< r$	of which rank 0	of which rank 1–2 (first decoder block)
CoLA	1	14	12	2 (rank 1)
SST-2	1	14	12	2 (rank 1)
MRPC	1, 5	3	0	3 (rank 2)
QNLI	4, 5	3	0	3 (rank 2)
STS-B	1, 2	3	0	3 (rank 2)

D.3 BASELINES

All baselines are run through the same pipeline as our method (same data, tokenization, trainer, schedule, seeds and learning-rate rule), with $r = 8$, $\alpha = 2r = 16$ and the adapters on the same matrices; only the initialization, the scaling and, for LoRA+ and LoRA-Pro, the optimizer change. Where an author implementation exists, its initialization is either called directly or transcribed and checked against it on random inputs before any run.

Why these baselines. The LoRA literature is large, and most of its variants act at a different level from an initialization: on the parameterization (DoRA, AdaLoRA, VeRA), on the rank and its allocation across layers, on the optimizer or preconditioner during training (LoRA-Pro, LoRA-RITE, Riemannian preconditioning), or on the update rule itself. An initialization only changes the starting point of a fixed adapter, so it composes with these methods rather than competing with them, and a comparison against them would confound the effect of the initialization with that of a different parameterization or optimizer. We therefore compare against the two zero-initialized controls, LoRA and rsLoRA, which isolate the effect of a non-zero initialization; against five representative published weight-, data- or gradient-dependent initializations of the standard adapter, PiSSA, CorDA, LoRA-GA, LoRA-One and LoRAM, at the same rank, scaling and adapted matrices; and, as a reference for the strongest training-time methods on the same adapter, LoRA+ and LoRA-Pro, run from the zero

initialization. Combining our initialization with the other kinds of variants is possible but outside the scope of this paper.

EVA and LoRA-DA. Among the published initializations that Table 1 leaves out, EVA and LoRA-DA are the closest to ours in their ingredients, since both build the adapter from second-order statistics of the layer. EVA (Paischer et al., 2025) sets A_0 to the top-r principal directions of the layer inputs and $B_0 = 0$: it uses the input factor S, but neither the output factor T nor the gradient. LoRA-DA (Zhang et al., 2026) minimizes an asymptotic estimate of the parameter-space error of the fine-tuned adapter, which trades a bias term, the part of a natural-gradient step on the task data that the adapter cannot represent, against a variance term. It uses both Kronecker factors and the gradient, as CG-LoRA does, but applies the inverse of a Kronecker-factored Fisher (with a diagonal output factor) to the gradient, a natural-gradient step in parameter space, whereas aligning the predictions at the end of training calls for the inverse square roots of S and T on each side and a pull-back (Theorem 5.1). Its factors are scaled by a tuned constant γ, and training starts from a shifted point. A third curvature-based criterion (Feng et al., 2026), which selects the directions by how parameter perturbations change the predictions, has no released implementation that we could find, and we did not run it.

We ran EVA and LoRA-DA through the pipeline of Table 1, with the same seeds and learning-rate rule. For EVA, A_0 spans the exact top-r principal directions of the layer inputs over the non-padded tokens of the first 256 training examples, the value to which the authors' incremental estimate converges, with rank r on every layer as for every other arm, so that EVA's redistribution of the ranks across layers is not used. For LoRA-DA, the initialization is computed by the authors' released code in its released configuration ($\gamma = 1024$, $\beta = 0.03$, 256 examples for the gradient and 256 for the Fisher, shifted start, rsLoRA scaling); only the collection of the Kronecker factors, written for decoder-only models, was adapted to the encoder–decoder model. Table 7 reports the results. Both methods fall within the range of the baselines of Table 1. On CoLA, the task that separates the methods, they reach 43.9 and 44.5, about 5 points below both CG-LoRA variants ($p \leq 0.001$ for each of the four paired comparisons over the ten seeds), with worst seeds of 41.8 and 40.6, against 45.6 for CG-LoRA-W. On MRPC, CG-LoRA-C is 1.8 and 1.3 points above them ($p = 0.02$ and 0.009) and CG-LoRA-W within a point of both, and on SST-2, QNLI and STS-B the four initializations are within about half a point of each other. Curvature statistics alone therefore do not produce the gains on CoLA; what the two constructions add is the form that prediction alignment prescribes for them and the flat spectrum.

Table 7: EVA and LoRA-DA on T5-base under the protocol of Table 1: one epoch, own learning rate per cell, rsLoRA scaling, mean and standard deviation over ten seeds for CoLA and MRPC and five seeds for the other tasks; MCC on CoLA, Pearson correlation on STS-B, accuracy otherwise. The CG-LoRA rows repeat Table 1. Selected learning rates: EVA $2 \cdot 10^{-4}$ (10^{-4} on QNLI); LoRA-DA $5 \cdot 10^{-4}$, $8 \cdot 10^{-4}$, $2 \cdot 10^{-4}$, $2 \cdot 10^{-4}$ and $8 \cdot 10^{-4}$ in column order.

	CoLA	MRPC	SST-2	QNLI	STS-B
EVA	43.9 ± 1.6	83.8 ± 1.7	93.7 ± 0.2	93.0 ± 0.1	88.2 ± 0.4
LoRA-DA	44.5 ± 2.4	84.3 ± 0.9	94.1 ± 0.4	93.1 ± 0.1	88.4 ± 0.3
CG-LoRA-W	49.8 ± 2.7	84.6 ± 0.8	94.1 ± 0.3	93.1 ± 0.2	88.5 ± 0.8
CG-LoRA-C	49.4 ± 2.5	85.6 ± 1.2	93.9 ± 0.2	92.9 ± 0.2	88.7 ± 0.5

LoRA and rsLoRA. LoRA's default initialization (Hu et al., 2022): $B_0 = 0$ and A_0 Kaiming-uniform, through the `peft` library. LoRA uses the scaling $\eta = \alpha/r$, rsLoRA (Kalajdzievski, 2023) $\eta = \alpha/\sqrt{r}$. Both are otherwise identical. rsLoRA's scaling is the one used by every other adapter unless stated.

LoRA+. The default initialization and $\eta = \alpha/r$ (Hayou et al., 2024b), trained with two learning rates, $\lambda_B = 16\lambda_A$, the ratio recommended by the authors. The rate selected by the learning-rate rule is λ_A.

LoRA-Pro. The default initialization and rsLoRA scaling, trained with the LoRA-Pro update of Wang et al. (2025) in its "full" form, which at every step replaces the factor gradients by the closed-form solution that makes the update of BA closest to the full-fine-tuning Adam step (their

Theorem 2.1), transcribed from the authors' release and checked against it to 10^{-8} over 40 steps in float64. We run the update as the paper states it, without the gradient clipping of the release: with the released clipping every step is shrunk by a factor 100–300 under the token loss, and the method does not learn on T5-base at any rate of the grid.

PiSSA. The top-r singular triplets of W_0 (Meng et al., 2024), $B_0 = U_r\Sigma_r^{1/2}/\sqrt{\eta}$, $A_0 = \Sigma_r^{1/2}V_r^\top/\sqrt{\eta}$, the residual $W_0 - \eta B_0 A_0$ frozen, through `peft`'s implementation; with the rsLoRA scaling.

CorDA. The knowledge-preserved mode is not applicable to a single downstream task, so we use the instruction-previewed mode of Yang et al. (2024): the covariance C_{CorDA} of the layer inputs is accumulated over 256 training examples (the authors' count), one unpadded example per forward pass, the top-r singular triplets of $W_0 C_{\mathrm{CorDA}}$ are taken, $B_0 = U_r\Sigma_r^{1/2}/\sqrt{\eta}$ and $A_0 = \Sigma_r^{1/2}V_r^\top C_{\mathrm{CorDA}}^{-1}/\sqrt{\eta}$, and the residual is frozen.

LoRA-GA. Wang et al. (2024): $U\Sigma V^\top$ the randomized SVD of the gradient of the training loss on the 32-example probe ($q = 4r$, four power iterations, as released), $B_0 = U_{[r+1:2r]}$, $A_0 = V_{[1:r]}^\top$ (their direction `ArB2r`), both multiplied by the stable scale $d_{\mathrm{out}}^{1/4}/\sqrt{\gamma}$ with $\gamma = 16$, training started at W_0 (their fork subtracts $\eta B_0 A_0$ from the base weights), with the rsLoRA scaling as in the authors' reproduction instructions (`init.weight=stable`, `peft.use_rslora=True`).

LoRA-One. Zhang et al. (2025a): $U\Sigma V^\top$ the randomized SVD of the negative gradient on the same probe ($q = 512$, 16 power iterations, as released), $B_0 = U_r\Sigma_r^{1/2}/\sqrt{s_1(\nabla)\gamma}$, $A_0 = \Sigma_r^{1/2}V_r^\top/\sqrt{s_1(\nabla)\gamma}$, with the authors' per-task γ (128 on CoLA and QNLI, 64 on MRPC, 16 on SST-2; 128, their default, on STS-B), rsLoRA scaling, and the shifted start $W_0 + \eta B_0 A_0$, as released.

LoRAM. Zhang et al. (2025b): deterministic discrete-sine bases of the lowest r frequencies for both factors, scaled by $\beta = (Q\,\mathrm{var}(W_0)/\mathrm{var}(\Phi_n\Phi_m^\top))^{1/4}$ with $Q = \ln r/\ln\min(d_{\mathrm{in}}, d_{\mathrm{out}})$, started at W_0, with the rsLoRA scaling and $\alpha = r$ that its authors recommend, transcribed from their released code.

Full fine-tuning. All weights (FFT, all weights) or only the 72 adapted matrices (FFT, query/value), AdamW with the same schedule and learning-rate rule.

The controlled setting of Section F.1. $f(x) = W_3\sigma(W_2\sigma(W_1 x))$, $\sigma = \tanh$, NTK parameterization, the two hidden matrices adapted, $r = 4$, squared loss, standard Gaussian inputs and targets. Per-layer Jacobians are exact, K and K_r are $n \times n$, and the deviation of the two linearized gradient flows, $f(t) - f_r(t) = (e^{-tK_r} - e^{-tK})Z$ with $K := JJ^\top$ and $K_r := J_r J_r^\top$, is two matrix exponentials; S_j, T_j and the gradient are exact. Gradient descent uses a fixed fraction of $2/\lambda_{\max}(K)$ as step.

Table 10 uses $n = 2000$ and width 64, so that $\sum_j \mathrm{rank}\, H_j = 864 < \mathrm{rank}\, K = n$: the adapters cannot reach every direction of the training outputs, and the end-of-training deviation is not zero for every initialization. Against the nonlinear network the closed form is within 4% at the two shortest of the eight horizons we evaluate and of order one at the longest.

E THE SKETCH

Algorithm 1 needs, for every adapted layer, the top-m singular vectors and values of S_j and T_j and the $m \times m$ block $U_T^\top \nabla_j U_S$. Algorithm 2 obtains them without forming a $d \times d$ matrix. It follows the randomized range finder with q power iterations and a Rayleigh–Ritz step (Halko et al., 2011); the layer index is dropped.

Signals. For example i of the probe and token ω, the hooks of the layer see its input $h_{i,\omega} \in \mathbb{R}^{d_{\text{in}}}$ in the forward pass and, in a backward pass, the derivative of a scalar with respect to its pre-activation. With reduce factors (Eschenhagen et al., 2023), $\bar{h}_i := \sum_\omega h_{i,\omega}$ over the unmasked tokens, $S = \sum_i \bar{h}_i \bar{h}_i^\top$ and $T = \sum_i \bar{\delta}_i \bar{\delta}_i^\top$, where $\bar{\delta}_i \in \mathbb{R}^{d_{\text{out}} \times C}$ collects the derivatives of the C outputs of example i, summed over its tokens. The normalization of S and T does not change the retained subspaces.

Random output probes. T involves one derivative per output, and the outputs of a language model are the vocabulary logits at every decoder position of the padded batch (every position on LLaMA-2-7B), while the gradient is that of the training loss, over the labelled positions only; excluding the padded positions from T changes the retained output modes, and the accuracy only within the spread of the learning-rate selection (CoLA -0.1 and $+1.3$, MRPC $+0.5$ and -2.9 for the whitened and colored constructions, five seeds, each cell at its own rate), so C is in the tens of thousands. A backward pass of $z^\top f$ for a vector z of output weights gives $\bar{\delta}_i z_i$ at every layer at once, and for random signs, z_i with independent Rademacher entries, $\mathbb{E}[z_i z_i^\top] = I$ gives

$$\mathbb{E}\Big[\sum_i (\bar{\delta}_i z_i)(\bar{\delta}_i z_i)^\top\Big] = \sum_i \bar{\delta}_i \bar{\delta}_i^\top = T,$$

the estimator of Hutchinson (1989) applied to a matrix. We average ℓ independent probes per pass, so each pass over the probe costs one forward and ℓ backward passes, with fresh probes in every pass. With few outputs, the C canonical vectors give T exactly with C backward passes.

The gradient block. A last forward and backward pass of the training loss gives, per token, the loss derivative $\mu_{i,\omega}$ at the pre-activation, so that $\nabla = \sum_{i,\omega} \mu_{i,\omega} h_{i,\omega}^\top$ and

$$U_T^\top \nabla U_S = \sum_{i,\omega} (U_T^\top \mu_{i,\omega})(U_S^\top h_{i,\omega})^\top \in \mathbb{R}^{m \times m}$$

is accumulated exactly from projected signals, token by token.

Algorithm 2 Sketch of one layer

Input: probe batch, window m, power iterations q, output probes ℓ.
Draw $\Omega_S \in \mathbb{R}^{d_{\text{in}} \times m}$ and $\Omega_T \in \mathbb{R}^{d_{\text{out}} \times m}$ with i.i.d. $\mathcal{N}(0,1)$ entries.
for $k = 1, \dots, q+1$ **do**
  One pass: $Y_S \leftarrow \sum_i \bar{h}_i (\bar{h}_i^\top \Omega_S)$ and $Y_T \leftarrow \frac{1}{\ell} \sum_{p=1}^{\ell} \sum_i (\bar{\delta}_i z_{i,p}) \big((\bar{\delta}_i z_{i,p})^\top \Omega_T\big)$
  $\Omega_S \leftarrow \operatorname{orth}(Y_S)$, $\Omega_T \leftarrow \operatorname{orth}(Y_T)$ $\quad O(dm^2)$
end for
One pass: $B_S \leftarrow \sum_i (\Omega_S^\top \bar{h}_i)(\Omega_S^\top \bar{h}_i)^\top$ and $B_T \leftarrow \frac{1}{\ell} \sum_{p,i} (\Omega_T^\top \bar{\delta}_i z_{i,p})(\Omega_T^\top \bar{\delta}_i z_{i,p})^\top$
$B_S = V_S \Lambda_S V_S^\top$, $B_T = V_T \Lambda_T V_T^\top$; $U_S \leftarrow \Omega_S V_S$, $U_T \leftarrow \Omega_T V_T$ $\quad O(dm^2 + m^3)$
One pass of the training loss: $F_{\text{raw}} \leftarrow \sum_{i,\omega} (U_T^\top \mu_{i,\omega})(U_S^\top h_{i,\omega})^\top$ $\quad O(Ndm)$
Output: $U_S, \Lambda_S, U_T, \Lambda_T, F_{\text{raw}}$.

Cost and settings. Each pass over the probe costs $O\big((d_{\text{in}} + \ell d_{\text{out}}) nm\big)$ in the hooks, on top of one forward and ℓ backward passes shared by all layers. T5-base: $m = 16$, $\ell = 16$, $q = 0$. LLaMA-2-7B: $m = 16$, $\ell = 2$, $q = 1$. Table 8 varies q and ℓ around the default $q = 0$, $\ell = 16$: the whitened construction stays within 1.0 point of it on CoLA and 1.5 on MRPC, the colored one within 0.4 on MRPC, while on CoLA it moves by $+2.1$ at $q = 0$, $\ell = 256$ and -2.9 at $q = 2$ (standard deviation 5.2); $\ell = 16$ against $\ell = 256$ at the same q changes a cell by at most 2.1 points while dividing the backward passes of the sketch by 16.

Table 8: Effect of the sketch on training, T5-base, one epoch, five seeds; MCC on CoLA, accuracy on MRPC. Every row uses the learning rate selected at $q = 1$, $\ell = 256$ ($8 \cdot 10^{-4}$ on CoLA for both constructions, 10^{-3} and $5 \cdot 10^{-4}$ on MRPC), so that only the sketch changes; Table 1 instead uses $q = 0$, $\ell = 16$ with its own learning rate.

		whitened		colored	
q	ℓ	CoLA	MRPC	CoLA	MRPC
0	16	49.7 ± 1.4	85.7 ± 0.5	49.4 ± 1.6	83.4 ± 1.7
0	256	48.8 ± 2.9	85.3 ± 1.2	51.5 ± 1.4	83.3 ± 0.8
1	16	49.1 ± 2.1	85.0 ± 0.9	49.2 ± 2.3	83.3 ± 1.4
1	256	50.1 ± 1.3	85.3 ± 0.8	49.8 ± 2.7	83.1 ± 1.1
2	256	48.8 ± 2.2	84.2 ± 1.0	46.5 ± 5.2	83.1 ± 2.5

The window. Theorem 5.1 and Proposition 5.2 are stated for the exact factors. Algorithm 1 applies them within the top-m eigenspaces of S and T, so that the rescaling never divides by their near-null directions, the role damping plays in K-FAC (Martens & Grosse, 2015). Table 9 widens the window towards the exact factors, each cell at its own learning rate. Widening the window does not help: on CoLA both constructions lose 2 to 3.4 points, and on MRPC the whitened one changes by at most 0.5 point while the colored one loses about 2. Truncating to the top $m = 2r$ directions thus acts as a regularizer, as damping does, rather than as an approximation that more directions would refine.

Table 9: Window of the sketch, T5-base, one epoch, seeds 42–46, own learning rate per cell; MCC on CoLA, accuracy on MRPC; $q = 0$, $\ell = 16$, reduce factors. The window $m = 16 = 2r$ is the setting of Table 1, here over its first five seeds.

construction	m	CoLA	MRPC
whitened	16	49.9 ± 2.2	84.6 ± 0.8
	32	47.8 ± 3.4	84.9 ± 1.1
	64	46.7 ± 1.8	85.1 ± 1.5
colored	16	50.0 ± 1.6	85.8 ± 1.2
	32	46.6 ± 3.6	84.0 ± 1.6
	64	47.4 ± 1.5	83.9 ± 1.2

F ADDITIONAL TABLES AND FIGURES

F.1 WHERE THE MODEL IS EXACT

In a wide three-layer $\tanh$ network with the squared loss, two adapted matrices and everything exact, the closed form of the linearized dynamics (Appendix D.3) reproduces gradient descent on the linearized model to a relative error of $7 \cdot 10^{-5}$. The whitened construction with disjoint index sets attains the bound of Theorem 5.1(i) on both layers (Table 10) justifying the theoretical aspect. The closed form tracks the nonlinear network to 4% over the two shortest horizons and departs from it afterwards (Appendix D).

Table 10: Endpoint in the controlled network (setting in Appendix D.3). The whitened construction with disjoint index sets attains the bound of Theorem 5.1(i) on both layers. Bold: the bound, and the values attaining it.

initialization	$\mathcal{E}_\infty$ (layer 1) / bound	$\mathcal{E}_\infty$ (layer 2) / bound	$\lim_t \|f - f_r\|_2$
whitened	43.538 / $\mathbf{43.538}$	66.100 / $\mathbf{66.100}$	$\mathbf{29.90}$
colored	50.550 / 43.538	75.139 / 66.100	33.89
LoRA	61.467 / 43.538	81.753 / 66.100	40.91
LoRA-GA	49.765 / 43.538	74.977 / 66.100	33.80
LoRA-One	55.284 / 43.538	77.084 / 66.100	34.46

F.2 THE FLAT POINT ON OTHER INITIALIZATIONS

The flat point can be applied to the subspaces of the other gradient-based initializations, keeping everything else of the published method. Table 11 does so on CoLA, the task that separates the methods, with the configuration of Table 1 (own learning rate per cell, seeds 42–51). LoRA-One keeps its subspaces (the top singular pairs of the gradient on the same 32 examples) and its shifted start, and LoRA-GA its disjoint subspaces, with the singular values of both factors set to $\sqrt{2/3}/\eta$ instead of their own spectrum rule. The flat point improves both. On LoRA-GA's subspaces it adds 5.0 points ($p < 0.001$, paired t-test), on LoRA-One's 0.7 ($p = 0.6$) while halving the spread (2.4 against 5.3) and raising the worst seed from 39.0 to 45.0; both are then within 0.8 and 1.8 points of CG-LoRA-W ($p = 0.56$ and 0.08), which keeps the best mean and worst seed.

Table 11: The flat point on other initializations, CoLA (MCC), T5-base, one epoch, own learning rate per cell, mean and standard deviation over seeds 42–51, worst seed in parentheses. Own spectra: the published spectrum rule (LoRA-One: $\sqrt{s_i/(s_1\gamma)}$ with its shifted start; LoRA-GA: $d^{1/4}/\sqrt{\gamma}$); flat point: $\sqrt{2/3}/\eta$ on every retained mode, everything else unchanged.

subspaces	own spectra	flat point
whitened gradient (CG-LoRA-W)	–	$\mathbf{49.8} \pm 2.7$ (45.6)
colored gradient (CG-LoRA-C)	–	49.4 ± 2.5 (43.3)
LoRA-One	47.3 ± 5.3 (39.0)	48.0 ± 2.4 (45.0)
LoRA-GA	44.0 ± 2.6 (38.9)	49.0 ± 3.0 (44.4)

This table shows two interesting comparisons.

- At the flat point, what distinguishes CG-LoRA-W/CG-LoRA-C from LoRA-GA and LoRA-One is the choice of the subspace and we isolate the gains of the whitened/colored subspaces.
- Both LoRA-GA and LoRA-One improve slightly on the difficult CoLA task when using the flat point, especially in terms of robustness to the worst seed. This highlights exactly the interest of the flat point. Even though LoRA-GA's spectrum is already flat, its global scaling is $\eta^2\sqrt{d}/\gamma \approx 55$ instead of $\frac{2}{3}$ and a simple learning rate tuning cannot fix it exactly as the importance of the scaling also appears in the dropped bilinear term $\Delta B \Delta A$ which is no longer negligible.

F.3 RoBERTa-base

RoBERTa-base (Liu et al., 2019) is fine-tuned through a randomly initialized classification head, and on the small GLUE tasks a fraction of its runs end at the majority class, depending on the seed and the learning rate (Mosbach et al., 2021; Zhang et al., 2021; Lee et al., 2020). We therefore run the non-zero initializations of Table 1 on CoLA and MRPC over twenty seeds (42–61), with the protocol of Appendix D.1, in two settings: with the head first trained alone on the frozen `[CLS]` features (LP-FT, Kumar et al., 2022; Tomihari & Sato, 2024; ten epochs, one probed head per seed shared by all arms), and without a probe, the head then being trained jointly with the adapters. A seed collapses when it ends with an MCC of at most 5 on CoLA or at the majority class (68.4) on MRPC.

With the probe (Table 12), the collapses concentrate at the top of the learning-rate grid, which the metric rule selects on CoLA for LoRA-One and both our constructions: there LoRA-One collapses on 11 seeds of 20, CG-LoRA-C on 6 and CG-LoRA-W on 1, whereas PiSSA, CorDA, LoRA-GA and LoRAM, at lower rates, never do. CG-LoRA-W has the highest median on CoLA (50.2) and, on MRPC, never collapses and has the smallest spread (1.3). This is consistent with whitening: dividing the gradient by the curvature gives the directions of highest curvature, where a large learning rate first becomes unstable, the smallest weight in the initial factors, whereas the colored construction multiplies by the curvature and LoRA-One takes the raw gradient. The zero-initialized adapters, not shown, do not collapse on CoLA except LoRA+ and LoRA-Pro, and rsLoRA has there the highest mean (49.9).

Without the probe (Table 13) collapses are widespread: on CoLA every initialization but PiSSA and CorDA collapses on some seeds, LoRA-GA, at the lowest rate of the grid, on 13, and on MRPC LoRA-One collapses on 9. CG-LoRA-W is again the most robust gradient-based initialization, with the best median on CoLA (48.9) and, on MRPC, the only one besides CorDA that never collapses, with the highest worst seed (78.4). The probe raises the rate selected on CoLA for LoRA-One and the colored construction to 10^{-3}, and with it their collapses (3 to 11 and 2 to 6), whereas those of the whitened construction fall from 3 to 1.

Table 12: RoBERTa-base with a linear probe of the head (LP-FT), non-zero initializations, one epoch, own learning rate per arm, twenty seeds; MCC on CoLA, accuracy on MRPC. Mean ± standard deviation, median, lowest seed and number of collapsed seeds (MCC ≤ 5; accuracy 68.4, the majority class). Rate: selected learning rate, in units of 10^{-4}.

	CoLA					MRPC				
	rate	mean ± sd	median	min	coll.	rate	mean ± sd	median	min	coll.
PiSSA	5	48.2 ± 1.4	48.3	**45.5**	0	10	83.1 ± 3.5	83.8	70.6	0
CorDA	5	$\mathbf{49.7} \pm 2.0$	49.8	45.0	0	8	$\mathbf{83.7} \pm 2.0$	**84.3**	77.7	0
LoRA-GA	0.5	45.9 ± 2.3	45.9	40.9	0	1	77.6 ± 5.1	80.1	68.4	2
LoRA-One	10	22.4 ± 25.5	0.0	0.0	11	8	70.8 ± 4.3	68.4	68.4	13
LoRAM ($\alpha = r$)	1	44.9 ± 2.2	44.9	42.2	0	5	76.5 ± 6.2	79.5	68.4	5
CG-LoRA-W (ours)	10	48.1 ± 11.5	**50.2**	0.0	1	10	82.0 ± 1.3	82.1	**78.9**	0
CG-LoRA-C (ours)	10	33.7 ± 22.7	46.9	0.0	6	10	78.4 ± 7.1	80.9	68.4	4

F.4 Index sets

Theorem 5.1(ii) is attained by any index sets with $\mathcal{I} \cup \mathcal{J} = [2r]$, and both constructions use the paired choice instead. Table 14 compares four choices for the whitened construction: paired, $\mathcal{I} = \mathcal{J} = [r]$; half-overlapping, $\mathcal{I} = [r]$ and $\mathcal{J} = [r/2] \cup \{r+1, \ldots, 3r/2\}$; disjoint, $\mathcal{I} = [r]$ and $\mathcal{J} = \{r+1, \ldots, 2r\}$, the optimum of the end-of-training criterion; and shifted, $\mathcal{I} = \mathcal{J} = \{r+1, \ldots, 2r\}$, which discards the top r modes. On CoLA the ordering follows how much of the top of the spectrum is kept: paired 50.1, half-overlapping 48.3, disjoint 47.5 and shifted 44.6, so the optimum of the end-of-training criterion is not the best index set. On MRPC all four are within 0.2 points.

Table 13: RoBERTa-base without a linear probe (the classification head, seeded by the run seed, is trained jointly with the adapters), non-zero initializations, otherwise as Table 12: one epoch, own learning rate per arm, twenty seeds; MCC on CoLA, accuracy on MRPC; mean $\pm$ standard deviation, median, lowest seed, number of collapsed seeds; rate in units of 10^{-4}.

	CoLA					MRPC				
	rate	mean $\pm$ sd	median	min	coll.	rate	mean $\pm$ sd	median	min	coll.
PiSSA	5	48.2 ± 2.6	48.3	41.9	0	8	79.9 ± 6.0	82.6	68.4	2
CorDA	2	$\mathbf{48.5} \pm 2.0$	48.3	**44.4**	0	5	$\mathbf{84.2} \pm 2.8$	**84.2**	76.7	0
LoRA-GA	0.2	7.1 ± 11.3	0.0	0.0	13	1	77.0 ± 5.8	78.9	68.4	2
LoRA-One	5	40.4 ± 17.6	46.4	0.0	3	8	73.7 ± 6.1	70.6	68.4	9
LoRAM ($\alpha = r$)	2	34.2 ± 19.0	44.1	0.0	4	2	76.8 ± 4.3	77.5	68.4	2
CG-LoRA-W (ours)	8	42.2 ± 18.4	**48.9**	0.0	3	8	82.4 ± 2.3	82.6	**78.4**	0
CG-LoRA-C (ours)	5	40.0 ± 15.4	46.9	0.0	2	8	79.2 ± 5.8	80.9	68.4	2

Table 14: Index sets of the whitened construction, T5-base, one epoch, own learning rate per arm, five seeds, mean and standard deviation; MCC on CoLA, accuracy on MRPC. Our constructions use the sketch setting of the first runs, $q = 1$ on CoLA and MRPC and $\ell = 256$ output probes, and differ from Table 1 in that setting only (Table 8).

index sets	CoLA	MRPC
paired	50.1 ± 1.3	85.3 ± 0.8
half-overlapping	48.3 ± 1.4	85.2 ± 1.4
disjoint	47.5 ± 3.0	85.1 ± 0.7
shifted	44.6 ± 1.7	85.3 ± 1.2

F.5 Rank

Table 15 varies the rank r alone, with $\alpha = 2r$, the rsLoRA scaling, the window $m = 2r$ and the flat point at every rank, so that only the number of retained modes changes. Each cell selects its own learning rate. The whitened construction is above rsLoRA on CoLA at every rank, by 2.0, 2.1, 6.1 and 1.6 points from $r = 4$ to $r = 32$, and within one point of it on MRPC except at $r = 32$, where it is 1.7 points higher. The colored construction is above rsLoRA on CoLA at $r = 8$ and $r = 16$, and on MRPC at $r = 8$ and $r = 32$.

Table 15: Rank, T5-base, one epoch, own learning rate per cell, five seeds, mean and standard deviation; MCC on CoLA, accuracy on MRPC. The row $r = 8$ is the first five seeds of Table 1.

	CoLA			MRPC		
r	rsLoRA	whitened	colored	rsLoRA	whitened	colored
4	46.4 ± 2.2	48.4 ± 1.6	43.2 ± 2.0	82.5 ± 1.1	83.3 ± 1.6	82.0 ± 2.1
8	47.8 ± 2.3	49.9 ± 2.2	50.0 ± 1.6	84.1 ± 1.4	84.6 ± 0.8	85.8 ± 1.2
16	43.8 ± 1.7	49.9 ± 4.4	44.6 ± 2.1	85.0 ± 1.1	84.2 ± 0.6	84.5 ± 1.2
32	48.8 ± 1.3	50.4 ± 2.0	48.1 ± 1.7	84.2 ± 0.9	85.9 ± 0.4	85.9 ± 0.9

F.6 Kronecker factors

Both constructions use the reduce variant of K-FAC for weight-shared layers (Eschenhagen et al., 2023), which sums the inputs and the output derivatives over the tokens of an example before forming S_j and T_j. Table 16 compares it with the expand variant, which forms them from every token separately, with everything else fixed.

Table 16: Reduce against expand Kronecker factors, T5-base, one epoch, own learning rate per cell, five seeds, mean and standard deviation; MCC on CoLA, accuracy on MRPC. The first two rows use the sketch setting of the first runs, $q = 1$ and $\ell = 256$, and differ from Table 1 in that setting only (Table 8); the last row is the colored construction with the expand factors at the default setting $q = 0$, $\ell = 16$, which is the setting of Table 1 and so is directly comparable with the first five seeds of its CG-LoRA-C row (50.0 ± 1.6 and 85.8 ± 1.2).

	whitened		colored	
K-FAC	CoLA	MRPC	CoLA	MRPC
reduce	50.1 ± 1.3	85.3 ± 0.8	49.8 ± 2.7	83.1 ± 1.1
expand	47.9 ± 1.1	82.8 ± 0.9	50.4 ± 2.0	84.2 ± 1.0
expand, $q = 0, \ell = 16$	–	–	46.2 ± 1.9	85.2 ± 0.7

F.7 Cost: operation counts and the measurements of Table 3

Table 17 gives the operation counts of each initialization. Table 3 was measured on one NVIDIA GH200 with the released cost script. Time runs from the loaded model to the factors: each T5-base arm is built once untimed and then five times, the median being reported, and each LLaMA-2-7B time is the median of three invocations, every arm on the same 32-example probe. GPU memory is the peak allocated during the call above the memory held before it by the loaded model; host memory is the size of the CPU tensors an arm holds at its peak, the bf16 gradient of every parameter for LoRA-GA and LoRA-One and nothing for the others. The LoRA training reference is 20 rsLoRA steps at batch size 32 on T5-base and 3 AdamW steps at batch size 4 on the 12 longest examples under 512 tokens on LLaMA-2-7B, under bf16 autocast.

Table 17: Cost of the initialization for one adapted $d \times d$ layer, rank r, window $m = 2r$, a probe of N tokens, q power iterations and ℓ output probes. Passes: forward (F) and backward (B) passes of the whole model over the probe, shared by all layers. The FLOPs and memory are those added per layer on top of the passes. LoRA-One's randomized SVD uses $k = 512$ and 16 iterations.

initialization	passes	FLOPs per layer	memory per layer
LoRA, rsLoRA	–	$O(rd)$	–
LoRAM	–	$O(d^2)$	–
PiSSA	–	$O(d^3)$	$O(d^2)$
LoRA-GA	F + B	$O(Nd^2 + rd^2)$	$O(d^2)$
LoRA-One	F + B	$O(Nd^2 + 16kd^2)$	$O(d^2)$
whitened, colored	$(q+3)$F + $\big((q+2)\ell + 1\big)$B	$O\big((q+3)\ell Ndm + dm^2 + m^3\big)$	$O(dm)$