WHEN CAN COMPRESSION IMPROVE LORA? A BIAS–VARIANCE THEORY OF RESTRICTED ADAPTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Compressed variants of LoRA can sometimes outperform standard LoRA despite using far fewer trainable parameters. We study this effect as restricted estimation: compression reduces estimation variance but introduces task-dependent projection bias. A local quadratic decomposition and an exact sequence model yield an energy–noise condition for when restriction helps. At matched parameter budgets, sparse residual correction is beneficial when its joint recovery exceeds the opportunity cost of displaced compressed capacity, with an additional retained-noise term under adaptive selection. For Uni-LoRA-style parameter sharing, we characterize recovery after selected coordinates are released and the remaining shared parameters are refitted. This motivates Projected LoRA with Sparse Adapter (ProLoSA), which combines a compressed branch with selected residual coordinates. Sequence and Transformer controls test the mechanism and its limits, while downstream experiments on language understanding, reasoning, instruction tuning, and vision show matched-budget gains over compressed LoRA in several settings.

1 INTRODUCTION

Low-Rank Adaptation (LoRA) (Hu et al., 2022) fine-tunes frozen models through trainable low-rank factors, while compressed PEFT methods further restrict the adaptation space through parameter sharing or structured parameterizations (Li et al., 2024; Bałazy et al., 2025; Li et al., 2025). Such restrictions can preserve downstream performance with far fewer trainable parameters and, in some reported settings, even outperform standard LoRA (Bałazy et al., 2025; Li et al., 2025). This raises a statistical question: *when can restricting the adaptation space improve estimation, and what determines whether this advantage emerges or disappears?*

We study compressed LoRA as restricted estimation, connecting classical compressed and projected estimation (Maillard & Munos, 2009; Kabán, 2014; Thanei et al., 2017) with parameter-efficient adaptation (Aghajanyan et al., 2021; Hu et al., 2022). A local quadratic analysis and an exact sequence model characterize when variance reduction outweighs compression-induced projection bias, while a matched-budget extension shows that residual correction is useful only when joint recovery offsets displaced compressed capacity and adaptive retained noise. This motivates ProLoSA, which augments Uni-LoRA-style parameter sharing (Li et al., 2025) with selected LoRA-factor residual coordinates, using warmup SNIP (Lee et al., 2019) as a practical selector. Controlled sequence and Transformer experiments test the mechanism, followed by downstream evaluation on language and vision tasks.

Contributions. (i) We derive local and exact sequence-model conditions for when compressed adaptation can outperform a less restricted estimator. (ii) We characterize matched-budget residual recovery under parameter sharing, including opportunity-cost and adaptive-noise effects. (iii) We instantiate this mechanism in ProLoSA and test both its statistical signatures and downstream behavior across language and vision benchmarks.

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.

1.1 Related Work

Our analysis connects restricted estimation and projection risk (Maillard & Munos, 2009; Kabán, 2014; Thanei et al., 2017) with intrinsic-dimension views of parameter-efficient adaptation (Aghajanyan et al., 2021). Compressed PEFT methods, including Tied-LoRA (Renduchintala et al., 2024), VeRA (Kopiczko et al., 2024), VB-LoRA (Li et al., 2024), FourierFT (Gao et al., 2024), LoRA-XS (Bałazy et al., 2025), and Uni-LoRA (Li et al., 2025), reduce adaptation cost through parameter sharing, compact parameterizations, or structured update spaces. Our focus is instead the statistical tradeoff between a chosen compressed estimator and full LoRA, particularly the task signal excluded by restriction.

Analyses of LoRA expressivity, optimization, and invariance address complementary questions (Zeng & Lee, 2024; Liu et al., 2025; Wang et al., 2025; Yen et al., 2025; Wu et al., 2026). SparseAdapter and RoSA introduce sparse or sparse–low-rank adaptation (He et al., 2022; Nikdan et al., 2024), whereas ProLoSA releases coordinates of a compressed LoRA-factor vector. Further scope and the distinction from weight-space sparse adaptation are discussed in Appendix A.1; RoSA reference runs are reported in Appendix D.8.

2 When Does Compression Help?

The preceding discussion motivates a central question: beyond reducing the number of trainable parameters, when can restricting LoRA to a compressed parameterization actually improve estimation? We study this question through a local bias–variance analysis, first comparing standard LoRA with a Uni-LoRA-style compressed estimator and then using the resulting decomposition to motivate sparse residual correction.

Consider L adapted linear layers. For layer $\ell \in \{1, \dots, L\}$, let $W_\ell \in \mathbb{R}^{m_\ell \times n_\ell}$ denote the frozen pretrained weight. Standard LoRA learns a rank-r update

$$\Delta W_\ell = B_\ell A_\ell, \qquad A_\ell \in \mathbb{R}^{r \times n_\ell}, \quad B_\ell \in \mathbb{R}^{m_\ell \times r}.$$

Collecting all trainable LoRA-factor entries across the adapted layers gives

$$\theta = \mathrm{vec}\big(\{A_\ell, B_\ell\}_{\ell=1}^{L}\big) \in \mathbb{R}^D, \qquad D = \sum_{\ell=1}^{L} r(m_\ell + n_\ell).$$

Thus, standard LoRA directly estimates the full D-dimensional factor vector θ. Following the subspace formulation of Uni-LoRA (Li et al., 2025), a compressed parameterization instead trains $z \in \mathbb{R}^d$, with $d \ll D$, and reconstructs the full LoRA factors as

$$\theta = Pz, \qquad P \in \mathbb{R}^{D \times d},$$

where P is a fixed reconstruction matrix. Hence the compressed estimator is restricted to the subspace $\mathrm{col}(P) \subseteq \mathbb{R}^D$.

2.1 Restricted estimation and local risk

Let $R(\theta) := \mathbb{E}_{(x,y)\sim\mathcal{D}}[\ell(f_\theta(x), y)]$ denote the population risk of the adapted model, where $\mathcal{D}$ is the data distribution and ℓ is the task loss. Define the full-space and restricted population optima as

$$\theta^\star = \arg\min_{\theta \in \mathbb{R}^D} R(\theta), \qquad z_U^\star = \arg\min_{z \in \mathbb{R}^d} R(Pz), \qquad \theta_U^\star = P z_U^\star.$$

Here the subscript L denotes standard LoRA and U denotes the Uni-LoRA-style compressed estimator. Around the full-space optimum, we use the local quadratic approximation

$$R(\theta) - R(\theta^\star) \approx \tfrac{1}{2}\|\theta - \theta^\star\|_H^2, \qquad H = \nabla^2 R(\theta^\star) \succeq 0, \tag{1}$$

where $\|v\|_H^2 := v^\top H v$ and H is the Hessian of the population risk at $\theta^\star$.

Let $\hat{\theta}_L \in \mathbb{R}^D$ denote the estimator produced by standard LoRA and let $\hat{z}_U \in \mathbb{R}^d$ denote the estimator in the compressed coordinates, with $\hat{\theta}_U = P\hat{z}_U$. For the local analysis, assume that the estimators are centered at their respective population optima,

$$\mathbb{E}[\hat{\theta}_L] = \theta^\star, \qquad \mathbb{E}[\hat{z}_U] = z_U^\star,$$

and define their covariance matrices as

$$\Sigma_L = \mathrm{Cov}(\hat{\theta}_L) \in \mathbb{R}^{D\times D}, \qquad \Sigma_z = \mathrm{Cov}(\hat{z}_U) \in \mathbb{R}^{d\times d}.$$

The corresponding expected excess risks, measured relative to the full-space optimum, are

$$\mathcal{E}_L := \mathbb{E}\Big[R(\hat{\theta}_L) - R(\theta^\star)\Big], \qquad \mathcal{E}_U := \mathbb{E}\Big[R(\hat{\theta}_U) - R(\theta^\star)\Big].$$

Under the local quadratic approximation,

$$\begin{aligned} \mathcal{E}_L &\approx \tfrac{1}{2}\,\mathrm{tr}(H\Sigma_L), \\ \mathcal{E}_U &\approx \underbrace{\tfrac{1}{2}\|\theta_U^\star - \theta^\star\|_H^2}_{B_{\mathrm{proj}}} + \underbrace{\tfrac{1}{2}\,\mathrm{tr}(HP\Sigma_z P^\top)}_{V_U}, \end{aligned} \tag{2}$$

where B_{proj} is the bias introduced by restricting the estimator to $\mathrm{col}(P)$, and V_U is the estimation-variance contribution of the compressed estimator.

Therefore, compression has lower expected excess risk than standard LoRA, $\mathcal{E}_U < \mathcal{E}_L$, whenever

$$B_{\mathrm{proj}} < \mathcal{E}_L - V_U. \tag{3}$$

Thus, the benefit of compression is governed by a projection-bias/variance tradeoff rather than by parameter count alone. Appendix A.2 gives the full derivation and extends the comparison to nonzero estimation and optimization bias.

2.2 An exact energy–noise condition

To make the general condition in Eq. (3) explicit, we specialize it to an isotropic Gaussian sequence model. This setting isolates the two quantities in the preceding decomposition: task energy excluded by compression and estimation noise removed by restricting the parameter space.

Fix a target $\theta^\star \in \mathbb{R}^D$ and a reconstruction matrix $P \in \mathbb{R}^{D\times d}$ with orthonormal columns, $P^\top P = I_d$, where $0 < d < D$. We take the population risk to be

$$R(\theta) = \tfrac{1}{2}\|\theta - \theta^\star\|_2^2, \tag{4}$$

so that $R(\theta^\star) = 0$ and the Hessian in Eq. (1) is exactly $H = I_D$. We observe

$$y = \theta^\star + \sigma_{\mathrm{eff}}\epsilon, \qquad \epsilon \sim \mathcal{N}(0, I_D),$$

where σ_{eff}^2 is the per-coordinate observation-noise variance. The full-space estimator and its compressed counterpart are

$$\hat{\theta}_L = y, \qquad \hat{\theta}_U = PP^\top y,$$

respectively. Accordingly, $\mathcal{E}_L = \mathbb{E}_\epsilon[R(\hat{\theta}_L)]$ and $\mathcal{E}_U = \mathbb{E}_\epsilon[R(\hat{\theta}_U)]$.

Proposition 1 (Fixed-task and ensemble energy–noise conditions) *Define the component of the target excluded by the compressed subspace as*

$$q = (I_D - PP^\top)\theta^\star.$$

For the fixed target $\theta^\star$, expectation over observation noise gives

$$\mathcal{E}_L = \tfrac{1}{2}D\sigma_{\mathrm{eff}}^2, \qquad \mathcal{E}_U = \tfrac{1}{2}\|q\|_2^2 + \tfrac{1}{2}d\sigma_{\mathrm{eff}}^2. \tag{5}$$

Hence, compression has lower expected excess risk, $\mathcal{E}_U < \mathcal{E}_L$, if and only if

$$v_\perp := \frac{\|q\|_2^2}{D-d} < \sigma_{\mathrm{eff}}^2, \tag{6}$$

where $v_\perp$ is the excluded target energy per discarded dimension.

For interpretation, suppose additionally that $\theta^\star$ is drawn independently of P with $\mathbb{E}[\theta^\star] = 0$ and $\mathrm{Cov}(\theta^\star) = vI_D$. Then

$$\mathbb{E}_{\theta^\star}\|q\|_2^2 = (D-d)v,$$

and averaging over both target generation and observation noise reduces the condition to

$$v < \sigma_{\mathrm{eff}}^2.$$

Equation (6) is the exact sequence-model counterpart of Eq. (3): compression helps when the task energy discarded per removed dimension is smaller than the corresponding noise variance. The fixed-task condition is the relevant one for repeated training or dataset realizations of a single task, because such experiments average over estimation noise while holding $\theta^\star$ fixed. The ensemble condition is included only to give a population-level interpretation of the same threshold.

The proof is given in Appendix A.3. Both conditions are relative to the chosen parameterization and noise model. Because residual energies in LoRA factor space can change under gauge transformations that leave the induced weight update unchanged, the neural bias–variance diagnostics later in the paper are instead evaluated in prediction space.

2.3 Residual recovery under a fixed budget

Compression removes signal as well as noise. A residual branch can recover selected directions, but under a fixed parameter budget it must replace part of the compressed representation. The relevant comparison is therefore the joint recovery obtained after refitting both branches, rather than the energy captured by selected coordinates alone.

Proposition 2 (Fixed-task matched-budget correction) *Fix $\theta^\star$, and let $P_B \in \mathbb{R}^{D\times B}$ and $P \in \mathbb{R}^{D\times d}$ have orthonormal columns, where $d = B - K$. For a fixed support $I \subseteq [D]$ with $|I| = K$, let $E_I = [e_i]_{i\in I} \in \mathbb{R}^{D\times K}$ collect the corresponding standard basis vectors, and assume $\mathrm{rank}([P, E_I]) = B$. For any matrix M, let $\Pi_M = MM^\dagger$ denote the orthogonal projector onto $\mathrm{col}(M)$, where $\dagger$ is the Moore–Penrose pseudoinverse.*

Observe $y = \theta^\star + \sigma_{\mathrm{eff}}\epsilon$ with $\epsilon \sim \mathcal{N}(0, I_D)$, and define the two least-squares estimators

$$\hat\theta_{U,B} = \Pi_{P_B} y = P_B P_B^\top y, \qquad \hat\theta_{\mathrm{hyb}} = \Pi_{[P,E_I]} y.$$

Let

$$q = (I - PP^\top)\theta^\star, \qquad A_I = (I - PP^\top)E_I,$$

and define the recovered residual energy and the task-dependent opportunity term as

$$\mathcal{E}_I = \|\Pi_{A_I} q\|^2, \qquad C_{\mathrm{task}} = \|q\|^2 - \|(I - P_B P_B^\top)\theta^\star\|^2. \tag{7}$$

Finally, define the expected excess risks

$$\mathcal{E}_{U,B} = \mathbb{E}_\epsilon\left[\frac{1}{2}\|\hat\theta_{U,B} - \theta^\star\|^2\right], \qquad \mathcal{E}_{\mathrm{hyb}} = \mathbb{E}_\epsilon\left[\frac{1}{2}\|\hat\theta_{\mathrm{hyb}} - \theta^\star\|^2\right].$$

Then

$$\mathcal{E}_{U,B} - \mathcal{E}_{\mathrm{hyb}} = \frac{1}{2}\big(\mathcal{E}_I - C_{\mathrm{task}}\big). \tag{8}$$

Hence the hybrid improves on the budget-B reference exactly when its residual recovery exceeds the task-dependent opportunity term.

Both fitted subspaces have rank B, so their observation-noise contributions are identical and equal to $B\sigma_{\mathrm{eff}}^2/2$; these terms therefore cancel in Eq. (8). A proof, including this decomposition, is given in Appendix A.4. When $\mathrm{col}(P) \subseteq \mathrm{col}(P_B)$, $C_{\mathrm{task}} \geq 0$; for unrelated subspaces it may have either sign. For an independent isotropic target with $\mathbb{E}[\theta^\star\theta^{\star\top}] = vI_D$, $\mathbb{E}[C_{\mathrm{task}}] = Kv$, recovering the ensemble criterion in Appendix A.4. If the support is selected from the same noisy observation, the fixed-support cancellation no longer holds exactly: the gain becomes $\frac{1}{2}(\mathbb{E}\mathcal{E}_{\hat I} - C_{\mathrm{task}}) - N_{\mathrm{adapt}}$, where N_{adapt} is the adaptive-selection noise correction defined in Eq. (29); it measures the departure from the fixed-support noise term and can have either sign. Thus practical performance depends jointly on residual recovery, effective rank, and retained noise.

Releasing shared coordinates. To connect Proposition 2 to the normalized one-hot sharing used by Uni-LoRA and ProLoSA, we specialize to the groups induced by P. Let G_j be a group of m_j full-space coordinates that share one compressed parameter. Since the corresponding column of P is normalized, $q \perp \mathrm{col}(P)$ implies $\sum_{i\in G_j} q_i = 0$. For $I_j = I \cap G_j$ and $k_j = |I_j| < m_j$, the exact recovery contributed by group j is

$$\mathcal{E}_{I,j} = \sum_{i\in I_j} q_i^2 + \frac{\left(\sum_{i\in I_j} q_i\right)^2}{m_j - k_j}, \qquad \mathcal{E}_I = \sum_j \mathcal{E}_{I,j}. \tag{9}$$

The first term is the energy of the released coordinates themselves, whereas the second is the additional recovery obtained by refitting the compressed parameter shared by the remaining coordinates. If an entire group is released, its residual energy is fully recoverable, but one released direction is redundant with the original shared column; Appendix A.5 handles this case through the effective-rank correction and also gives the curvature-weighted extension.

The same idea extends beyond the Euclidean sequence model. Under the local quadratic approximation, define $\Delta(I)$ as the decrease in the local objective obtained by releasing support I and jointly refitting the compressed branch. After profiling out the compressed variables,

$$\Delta(I) = \frac{1}{2} g_I^\top M_I^\dagger g_I, \qquad I^\star = \arg\max_{|I|=K} g_I^\top M_I^\dagger g_I,$$

where g_I is the profiled residual gradient on the released coordinates and M_I is their corresponding Schur-complement curvature; their explicit forms are given in Appendix A.5. Exact support search is combinatorial. Residualized Gradient (RG) and Recovery–Fisher (RF) are coordinate-wise approximations to this recovery objective, whereas warmup SNIP is a separate trajectory-based heuristic whose finite-data effectiveness is evaluated empirically.

3 ProLoSA: Sparse Residual Correction

Section 2 shows that compression can improve estimation when variance reduction outweighs the projection bias from excluded task-relevant directions. Under a fixed budget, residual correction is beneficial when its recovered signal offsets the compressed capacity it displaces and, under adaptive selection, any retained-noise cost. Motivated by these results, ProLoSA augments a Uni-LoRA-style compressed parameterization with a small set of factor-space residual coordinates, preserving most of the variance reduction from compression while selectively restoring poorly represented directions.

Concretely, ProLoSA splits a fixed trainable-parameter budget $B = d + K$ between the compressed vector z and K sparse residual coefficients α:

$$\theta = Pz + R\alpha, \qquad R = [e_{i_1}, \ldots, e_{i_K}], \tag{10}$$

so allocating more parameters to the residual branch necessarily reduces the compressed dimension, $d = B - K$.

We instantiate P using Uni-LoRA's normalized one-hot reconstruction (Li et al., 2025): each full LoRA-factor coordinate is assigned to one shared latent parameter, and the columns of P are normalized. The compressed branch therefore ties groups of factor coordinates to shared values, while $R\alpha$ releases K selected coordinates from these constraints. After activation, z and α are optimized jointly, allowing both the released coordinates and the remaining shared values to refit, as characterized by Eq. (9). The reconstructed θ is unpacked back into the ordinary rank-r factors $\{A_\ell, B_\ell\}$, so the final update remains standard LoRA and can be merged into the frozen backbone at inference. Unlike RoSA, ProLoSA introduces sparsity in LoRA-factor rather than weight space; Appendix A.1 discusses this distinction.

The ideal recovery objective above does not directly yield a practical selector because exact support search is combinatorial. We therefore use a two-stage procedure based on warmup SNIP sensitivity (Lee et al., 2019). First, only the compressed vector z is trained. After a score-free warmup, a fixed scoring window accumulates

$$s_i = \sum_{t \in \mathcal{T}_{\text{score}}} \left| \theta_i^{(t)} g_i^{(t)} \right|, \qquad g^{(t)} = \nabla_{\theta^{(t)}} \mathcal{L}^{(t)}. \tag{11}$$

We then select the top-K coordinates, set $\alpha = 0$, fix P and R, and jointly optimize z and α for the remaining training steps. Zero initialization preserves the current prediction when the residual branch is activated. Warmup SNIP is an inexpensive practical proxy rather than an optimizer of the joint-recovery objective; its relation to recovery and its empirical limitations are examined in Sections 4–5 and Appendix A.5. Appendix B gives the full algorithm and experiment-specific schedules.

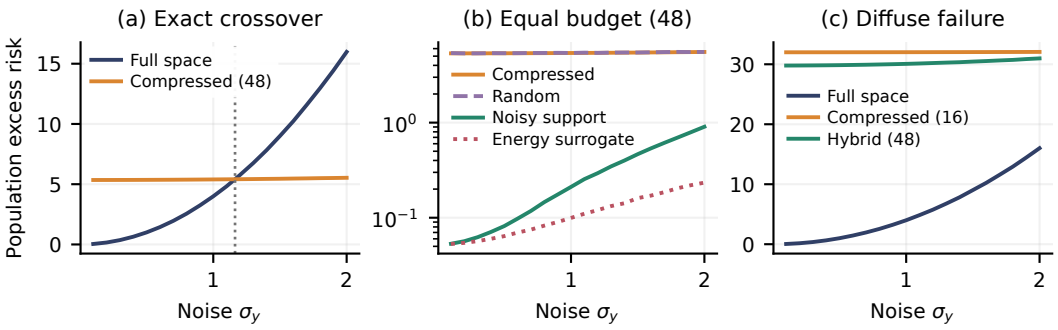

Figure 1: **Sequence-model tests.** (a) Full-space and compressed risks cross near the fixed-task prediction $\sigma_y^\star \approx 1.16$. (b) At budget $B = 48$, the energy surrogate and noisy-residual support improve on compression, while random support nearly ties it. (c) A dense target remains bias dominated and sparse correction recovers little. Targets and projections are fixed across 200 noise trials; full decompositions are in Appendix C.1.

4 Testing the Mechanism

We test the mechanism through exact sequence-model predictions, direct predictive bias–variance measurements under nonlinear LoRA-factor training, and Transformer controls of geometric recovery and practical support selection. Additional controls in Appendix E examine adaptive selection, residual geometry, and local-approximation error. The downstream benchmarks in Section 5 serve a separate practical purpose.

4.1 Projection-blind sequence estimation

We instantiate the sequence model from Section 2, $y = \theta^\star + \sigma_{\text{eff}}\epsilon$, with $\epsilon \sim \mathcal{N}(0, I_D)$ and $\sigma_{\text{eff}} = \sigma_y/\sqrt{n_{\text{eff}}}$. The target $\theta^\star$ is drawn independently of the compression projection, and both are fixed while only observation noise is resampled over 200 trials, preventing the compressed subspace from being chosen to match the target. We use a spike-and-heavy-tail target with concentrated residual energy and a dense Gaussian target as a diffuse, bias-dominated failure regime.

With $D = 4096$, $n_{\text{eff}} = 512$, and matched budget $B = 48$, we compare budget-48 compression with $d = 16$, $K = 32$ hybrids using random, noisy-residual, or target-informed energy-surrogate supports. Full target generation and estimation details are given in Appendix C.1.

Figure 1 confirms the fixed-task energy–noise crossover. At matched budget, random support performs similarly to compression, whereas the energy surrogate recovers substantially more signal; noisy-residual selection also changes the retained noise. The diffuse case remains bias dominated. Appendix E.3 shows that this recoverability gap persists under matched residual energy, while Appendix E.5 shows that, across 30 independently generated target/projection pairs, random support yields little average gain but target-informed supports remain beneficial over a broad range of allocations.

4.2 Predictive bias–variance tradeoffs in LoRA-factor training

We next measure predictive bias and variance under actual rank-4 LoRA-factor training. A fixed Transformer teacher has concentrated spike–slab weight updates calibrated to 10% of base-output variance. LoRA trains all 4608 factor entries; Uni-LoRA and ProLoSA use the same budget of 576, with ProLoSA allocating $d = 512$ and $K = 64$. A validation-selected configuration is fixed across all noise levels, and 20 paired datasets of size 256 are trained for 1000 steps. Full details of the teacher construction, configuration selection, and predictive bias–variance estimation are provided in Appendix C.4.

Figure 2 shows a noise-driven reversal: LoRA has the lowest risk at noise multiples 0, 0.5, and 1, whereas both compressed methods achieve lower risk at 2 and 4 as their variance reduction outweighs the added squared bias. At matched budget, ProLoSA reduces Uni-LoRA's risk by 10.4–

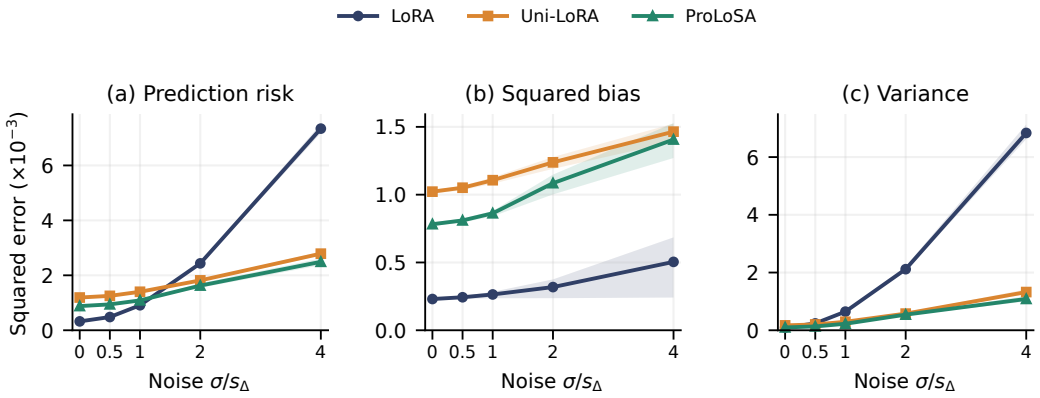

Figure 2: **Predictive bias–variance tradeoffs with trained LoRA factors.** Clean prediction risk, finite-replicate-corrected squared bias, and variance over 20 paired training datasets. Shading gives pointwise 95% paired-bootstrap intervals. Noise is normalized by teacher-displacement RMS, and $\widehat{R} = \widehat{B}^2 + \widehat{V}$.

26.1% across all five noise levels. Through noise multiple 2, it lowers both squared bias and variance; at 4, the improvement is primarily variance driven. The ProLoSA–Uni-LoRA gap also persists under extended training and a large-sample noiseless control (Appendix C.4).

These results support a predictive bias–variance mechanism under the tested protocol, but the measured quantities also reflect finite-sample fitting, optimization, random design, and adaptive selection and therefore should not be interpreted as pure projection bias. Moreover, concentration is imposed in weight space whereas ProLoSA releases factor coordinates, motivating the following controls on whether practical selectors realize available geometric recovery.

4.3 Geometric recoverability versus practical selector realizability

The sequence model separates two questions: whether useful residual directions exist and whether a finite-data selector can identify them. We test this distinction with concentrated and diffuse fixed Transformer teachers, both calibrated to 10% of the base-output variance. At budget 128, compression is compared with $d = K = 64$ hybrids using random, gradient, SNIP, or target-informed energy supports. For each support, we measure exact Euclidean residual recovery and paired prediction-MSE gain; full experimental details are given in Appendix E.1.

Figure 3 shows that recoverable residual directions can exist even when a practical selector fails to exploit them. The target-informed energy surrogate improves over budget-matched compression throughout the tested grid, whereas SNIP (Lee et al., 2019) is condition dependent and random support provides no comparable systematic recovery. Thus selector failure need not imply the absence of useful residual structure; it can instead reflect imperfect support estimation under finite data and nonlinear optimization. This distinction also motivates the recovery-oriented RG and RF approximations in Appendix A.5, although closer alignment with the local geometric objective does not by itself guarantee better finite-data selection.

Adaptive selection creates a second statistical bottleneck: realized gain depends on recovered signal, displaced compressed capacity, and retained noise. Independent-selection controls show that cleaner support estimation can help, but the benefit shrinks when total data cost is held fixed. Thus, geometric recoverability alone does not guarantee realized improvement. Full controls and diagnostics are reported in Appendix E.

5 Downstream Performance and Ablations

The preceding analysis and controlled experiments characterize when compression can improve estimation and how sparse residual correction can recover task-relevant directions excluded by the

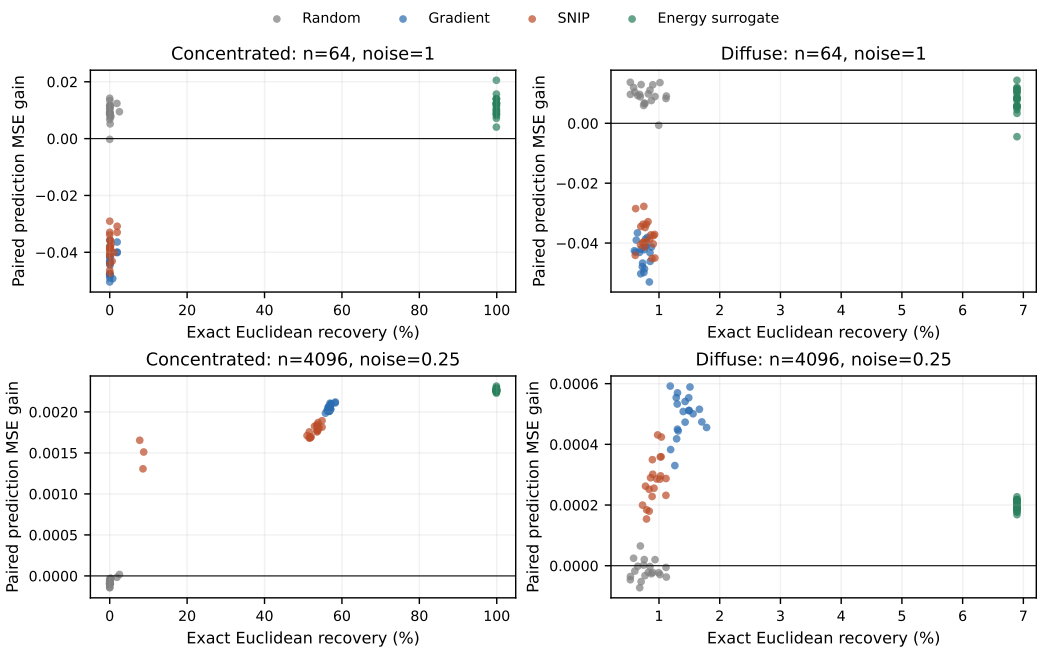

Figure 3: **Geometric recoverability versus selector realizability.** Across 20 paired seeds per condition, target-informed energy support reveals recoverable residual structure, whereas warmup SNIP realizes positive prediction-MSE gains only conditionally and random support shows no comparable systematic recovery. Hybrids use a 256-step pilot followed by a fresh 744-step fit; the energy rule is a surrogate, not the joint-recovery optimum.

compressed subspace. We now ask whether these mechanisms translate into practical gains on standard downstream adaptation tasks. We evaluate ProLoSA across language understanding, mathematical reasoning, instruction tuning, and vision classification, comparing it with standard LoRA and compressed PEFT baselines under the corresponding parameter budgets. We further use targeted ablations to examine how residual support selection and budget allocation affect performance. Table 1 summarizes the main downstream results, while Appendix D provides task-level scores, aggregation details, and result provenance.

Natural Language Understanding. We evaluate ProLoSA on six tasks from the GLUE benchmark (Wang et al., 2018), with RoBERTa-base and RoBERTa-large backbones. Table 1 shows that, at the same 0.023M parameter budget, ProLoSA achieves average GLUE scores of $85.5/88.7$ on RoBERTa-base/large, compared with $85.3/88.3$ for Uni-LoRA. The published baseline results are taken from Uni-LoRA (Li et al., 2025), while the ProLoSA results are ours. To control for run-to-run variation on RoBERTa-large, we further rerun Uni-LoRA and ProLoSA using the same fixed splits and five matched seeds. Across the six tasks, ProLoSA obtains 88.734 versus 88.222 for Uni-LoRA, a paired improvement of $+0.511$ points with a 95% confidence interval of $[0.042, 0.981]$. The task-level gains are heterogeneous, with full paired results reported in Appendix D.1.

Mathematical reasoning. We fine-tune Llama-3.1-8B on MetaMathQA (Yu et al., 2023) and evaluate on GSM8K (Cobbe et al., 2021) and MATH500 (Hendrycks et al., 2021; Lightman et al., 2023). Across seeds 42–44, ProLoSA achieves $76.17/27.33$ with 2.62M trainable parameters, one quarter of rank-4 LoRA's 10.49M. The differences from LoRA and Uni-LoRA are modest relative to the three-seed variability and are therefore treated as descriptive. Appendix D.2 gives the full results and validation-only selection protocol.

Instruction tuning. We train LoRA, Uni-LoRA, and ProLoSA on 50K SmolTalk examples (Ben Allal et al., 2025) with three seeds. At the same 2.62M effective-parameter budget as Uni-

Table 1: **Downstream performance across language tasks.** Parameters are trainable/effective parameter counts under each method's convention. For GLUE, published results for LoRA, VeRA, VB-LoRA and Uni-LoRA are taken from Uni-LoRA (Li et al., 2025), while ProLoSA results are ours. Best results within each backbone and evaluation setting are shown in bold. Full evaluation protocols, aggregation details, and additional baseline provenance are provided in the appendix.

Method	RoBERTa-base		RoBERTa-large		Llama-3.1-8B				
	Params.	GLUE	Params.	GLUE	Params.	*Mathematical reasoning*		*Instruction tuning*	
						GSM8K	MATH500	IFEval	IFBench
LoRA	0.295M	**86.6**	0.786M	87.8	10.49M	76.04	26.40	46.64	16.89
VeRA	0.043M	85.2	0.061M	87.8			–		
VB-LoRA	0.075M	85.4	0.162M	88.2			–		
Uni-LoRA	0.023M	85.3	0.023M	88.3	2.62M	76.12	26.87	46.64	15.78
ProLoSA	0.023M	85.5	0.023M	**88.7**	2.62M	**76.17**	**27.33**	**46.89**	**18.11**

Table 2: **Matched vision transfer on eight datasets.** Average top-1 accuracy (%) over five matched seeds using the same reconstructed splits. Δ denotes ProLoSA minus Uni-LoRA with paired 95% confidence intervals. Both methods use 72K/144K parameters for ViT-Base/Large. Full task-level results and published-baseline comparisons are in Appendix D.7.

Backbone	Uni-LoRA	ProLoSA	Δ (95% CI)
ViT-Base	85.191	**86.385**	$+1.194\ [0.899, 1.489]$
ViT-Large	87.782	**88.134**	$+0.352\ [0.193, 0.511]$

LoRA, ProLoSA attains a similar mean IFEval score (Zhou et al., 2023) and a higher mean IFBench score (Pyatkin et al., 2025); full results and settings are given in Appendix D.5.

Vision transfer. We evaluate ProLoSA on ViT-Base and ViT-Large across eight image-classification datasets using five matched seeds. Under the same reconstructed splits and parameter budgets, ProLoSA improves the paired eight-task average over Uni-LoRA on both backbones (Table 2). Full task-level results, uncertainty, and published-baseline comparisons are reported in Appendix D.7.

Ablations. Under the equal budget $d + K = 23{,}040$, warmup SNIP achieves the highest mean among the tested support rules on both CoLA and MRPC, whereas random support does not improve over compression and the sparse-dominant allocation performs substantially worse. Appendix D.6 provides the full budget-allocation, support-selection, and warmup ablations. We further evaluate the recovery-oriented RG and RF selectors in Appendix D.9; under the tested protocol, neither improves over the task-specific SNIP configuration. Appendix D.8 reports RoSA separately as an independent-budget reference, since its parameter budget and sparse-update space differ from those of ProLoSA.

6 Discussion and Conclusion

Compressed adaptation can improve estimation when variance reduction outweighs the task signal excluded by restriction. At matched budgets, sparse residual correction is beneficial only when its joint recovery exceeds the opportunity cost of displaced compressed capacity and retained estimation noise; under parameter sharing, joint refitting further makes coordinate-wise capture insufficient. Controlled sequence and Transformer experiments support these mechanisms while highlighting the effects of finite-data selection and optimization, including a noise-driven risk reversal in LoRA-factor training. ProLoSA instantiates this view by combining a compressed branch with selected residual coordinates: the theory defines an ideal recovery objective, while practical gains depend on whether useful supports can be identified reliably. Overall, compressed adaptation is governed by *compression geometry, estimation noise, and support realizability*; improving finite-data estimators of joint recovery remains an important direction for future work.

Reproducibility Statement.

Derivations and proofs are provided in Appendix A, and the ProLoSA parameterization and training procedure are detailed in Appendix B. Appendices C–E document the controlled mechanism experiments, downstream settings, and additional validation experiments. Appendix C.4 further documents the fixed weight-space spike–slab teacher, validation-only configuration selection, paired prediction decomposition, and extended-training controls. The fixed-teacher experiments retain source code, targets, supports, per-run predictions, and random seeds; the saved predictions support direct recomputation of the reported decomposition components.

Ethics Statement

This work studies statistical mechanisms of parameter-efficient fine-tuning using public models and benchmarks. It involves no human subjects, private data, or personally identifiable information. We do not foresee ethical concerns beyond those generally associated with fine-tuning large language models.

The Use of Large Language Models

Large language models assisted the exposition, exploratory mathematical derivations, and experiment-code development. The authors take responsibility for verifying all research claims, theoretical results, experimental descriptions, and empirical findings, and for the entire content of this paper.

References

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

Alan Ansell, Ivan Vulić, Hannah Sterz, Anna Korhonen, and Edoardo M. Ponti. Scaling sparse fine-tuning to large language models, 2024.

Klaudia Bałazy, Mohammadreza Banaei, Karl Aberer, and Jacek Tabor. LoRA-XS: Low-rank adaptation with extremely small number of parameters. In *28th European Conference on Artificial Intelligence*. IOS Press, 2025. doi: 10.3233/faia251185.

Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Guilherme Penedo, Lewis Tunstall, Andrés Marafioti, Hynek Kydlíček, Agustín Piqueres Lajarín, Vaibhav Srivastav, Joshua Lochner, Caleb Fahlgren, Xuan-Son Nguyen, Clémentine Fourrier, Ben Burtenshaw, Hugo Larcher, Haojun Zhao, Cyril Zakka, Mathieu Morlon, Colin Raffel, Leandro von Werra, and Thomas Wolf. Smollm2: When smol goes big – data-centric training of a small language model. *arXiv preprint arXiv:2502.02737*, 2025.

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

Ziqi Gao, Qichao Wang, Aochuan Chen, Zijing Liu, Bingzhe Wu, Liang Chen, and Jia Li. Parameter-efficient fine-tuning with discrete Fourier transform. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 14884–14901. PMLR, 2024.

Babak Hassibi and David G. Stork. Second order derivatives for network pruning: Optimal brain surgeon. In *Advances in Neural Information Processing Systems*, volume 5, pp. 164–171, 1993.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2 edition, 2009.

Shwai He, Liang Ding, Daize Dong, Jeremy Zhang, and Dacheng Tao. SparseAdapter: An easy approach for improving the parameter-efficiency of adapters. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 2184–2190, 2022. doi: 10.18653/v1/2022.findings-emnlp.160.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, 2021.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. doi: 10.48550/arXiv.2106.09685. URL https://openreview.net/forum?id=nZeVKeeFYf9.

William James and Charles Stein. Estimation with quadratic loss. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pp. 361–379, Berkeley, CA, 1961. University of California Press.

Ata Kabán. New bounds on compressive linear least squares regression. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*, volume 33 of *Proceedings of Machine Learning Research*, pp. 448–456. PMLR, 2014.

Soroush Abbasi Koohpayegani, K L Navaneet, Parsa Nooralinejad, Soheil Kolouri, and Hamed Pirsiavash. NOLA: Compressing LoRA using linear combination of random basis. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=TjfXcDgvzk.

Dawid Jan Kopiczko, Tijmen Blankevoort, and Yuki M. Asano. VeRA: Vector-based random matrix adaptation. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=NjNfLdxr3A.

Namhoon Lee, Thalaiyasingam Ajanthan, and Philip H. S. Torr. Snip: Single-shot network pruning based on connection sensitivity. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=B1VZqjAcYX.

Kaiyang Li, Shaobo Han, Qing Su, Wei Li, Zhipeng Cai, and Shihao Ji. Uni-LoRA: One vector is all you need. In *The 39th Conference on Neural Information Processing Systems*, 2025.

Yang Li, Shaobo Han, and Shihao Ji. VB-LoRA: Extreme parameter efficient fine-tuning with vector banks. *Advances in Neural Information Processing Systems*, 37:16724–16751, 2024. doi: 10.52202/079017-0533.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. *arXiv preprint arXiv:2305.20050*, 2023.

Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. DoRA: Weight-decomposed low-rank adaptation. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*. PMLR, 2024. doi: 10.48550/arXiv.2402.09353. URL https://proceedings.mlr.press/v235/liu24bn.html.

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. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/c34262c35aa5f8c1a091822cbb2020c2-Abstract-Conference.html.

Odalric-Ambrym Maillard and Rémi Munos. Compressed least-squares regression. In *Advances in Neural Information Processing Systems*, volume 22, 2009.

Mahdi Nikdan, Soroush Tabesh, Elvir Crnčević, and Dan Alistarh. RoSA: Accurate parameter-efficient fine-tuning via robust adaptation. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*. PMLR, 2024.

Bowen Ping, Shuo Wang, Hanqing Wang, Xu Han, Yuzhuang Xu, Yukun Yan, Yun Chen, Baobao Chang, Zhiyuan Liu, and Maosong Sun. Delta-CoMe: Training-free delta-compression with mixed-precision for large language models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 31056–31077, 2024.

Valentina Pyatkin, Saumya Malik, Victoria Graf, Hamish Ivison, Shengyi Huang, Pradeep Dasigi, Nathan Lambert, and Hannaneh Hajishirzi. Generalizing verifiable instruction following. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Adithya Renduchintala, Tugrul Konuk, and Oleksii Kuchaiev. Tied-LoRA: Enhancing parameter efficiency of LoRA with weight tying. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 8694–8705, 2024. doi: 10.18653/v1/2024.naacl-long.481. URL https://aclanthology.org/2024.naacl-long.481.

Weixi Song, Zuchao Li, Lefei Zhang, Hai Zhao, and Bo Du. Sparse is enough in fine-tuning pre-trained large language models. In *Proceedings of the 41st International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=10hu2D3hAg.

Gian-Andrea Thanei, Christina Heinze, and Nicolai Meinshausen. Random projections for large-scale regression. In *Big and Complex Data Analysis: Methodologies and Applications*, pp. 51–68. Springer, 2017.

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*, pp. 353–355, 2018. doi: 10.18653/v1/w18-5446.

Sheng Wang, Boyang Xue, Jiacheng Ye, Jiyue Jiang, Liheng Chen, Lingpeng Kong, and Chuan Wu. PRoLoRA: Partial rotation empowers more parameter-efficient LoRA. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pp. 2829–2841, 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. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/ea184f920a0f0f8d8030aa1bd7ac9fd4-Abstract-Conference.html.

Zirui Wang, Tingfeng Lan, Zhaoyuan Su, Juncheng Yang, and Yue Cheng. ZipLLM: Efficient LLM storage via Model-Aware synergistic data deduplication and compression. In *23rd USENIX Symposium on Networked Systems Design and Implementation (NSDI 26)*, pp. 2371–2387, Renton, WA, May 2026. USENIX Association. ISBN 978-1-939133-54-0. URL https://www.usenix.org/conference/nsdi26/presentation/wang-zirui.

Yize Wu, Ke Gao, Ling Li, and Yanjun Wu. Stable-LoRA: Stabilizing feature learning of low-rank adaptation. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/2962d47082d07cd6e28272ab471e0526-Abstract-Conference.html.

Jui-Nan Yen, Si Si, Zhao Meng, Felix Yu, Venkata Sai Surya Subramanyam Duvvuri, Inderjit Dhillon, Cho-Jui Hsieh, and Sanjiv Kumar. LoRA done RITE: Robust invariant transformation equilibration for LoRA optimization. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/bcbc0f660d2dde42f9d1d0ecb14a6f9a-Abstract-Conference.html.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T. Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. Metamath: Bootstrap your own mathematical questions for large language models. *arXiv preprint arXiv:2309.12284*, 2023.

Yuchen Zeng and Kangwook Lee. The expressive power of low-rank adaptation. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=likXVjmh3E.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Nikos Karampatziakis, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. AdaLoRA: Adaptive budget allocation for parameter-efficient fine-tuning. In *International Conference on Learning Representations*, 2023. doi: 10.48550/arXiv.2303.10512. URL https://openreview.net/forum?id=lq62uWRJjiY.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*, 2023.

A Extended Theory and Proofs

A.1 Related Work and Scope

Compressed and structured adaptation. LoRA, AdaLoRA, and DoRA use low-rank updates and refinements of their allocation or parameterization (Hu et al., 2022; Zhang et al., 2023; Liu et al., 2024). Tied-LoRA, VeRA, NOLA, PRoLoRA, VB-LoRA, LoRA-XS, Uni-LoRA, and FourierFT further restrict adaptation through sharing, frozen bases, or structured representations (Renduchintala et al., 2024; Kopiczko et al., 2024; Koohpayegani et al., 2024; Wang et al., 2024; Li et al., 2024; Bałazy et al., 2025; Li et al., 2025; Gao et al., 2024). Our fixed linear reconstruction directly models the Uni-LoRA branch; extending its risk conditions to other parameterizations requires checking their estimation spaces and assumptions. Intrinsic-dimension work already connects compact adaptation to compression-based generalization bounds (Aghajanyan et al., 2021). We focus on the relative risk of compressed and full-space estimators: dimensionality and within-subspace isometry alone do not determine excluded signal or retained noise.

Theory and sparse adaptation. Expressivity and optimization analyses address complementary questions (Zeng & Lee, 2024; Liu et al., 2025; Wang et al., 2025; Yen et al., 2025; Wu et al., 2026); practical training-seed variability need not follow an asymptotic statistical-variance law. SparseAdapter, SIFT, SpIEL, and RoSA introduce sparse or hybrid adaptation directions (He et al., 2022; Song et al., 2024; Ansell et al., 2024; Nikdan et al., 2024), but ProLoSA applies its sparse correction in a different space. Let $\theta = \mathrm{vec}(\{A_\ell, B_\ell\})$ denote the LoRA-factor coordinates. ProLoSA parameterizes $\theta = Pz + R_I\alpha$, so that selected factor coordinates are released from the constraint imposed by the compressed subspace P; after reconstruction, each layer still forms the standard low-rank update $\Delta W_\ell = B_\ell A_\ell$ with $\mathrm{rank}(\Delta W_\ell) \le r$. RoSA instead parameterizes the weight update as $\Delta W = BA + S$, with S sparse directly in weight-matrix space, thereby complementing the low-rank component and potentially increasing the rank and expressivity of the resulting update. Thus, ProLoSA is a *factor-space correction of compression-induced bias*, whereas RoSA is a *weight-space low-rank–plus–sparse expansion*. Their downstream performance can be compared under stated resource budgets, but such comparisons alone do not test the matched-budget residual-recovery mechanism analyzed here or isolate the contribution of the support selector. Our design criterion instead follows from the tradeoff between recovered projection bias and the opportunity cost of reallocating a fixed parameter budget, building on classical shrinkage, regression, and compressed estimation (James & Stein, 1961; Hastie et al., 2009; Maillard & Munos, 2009; Kabán, 2014; Thanei et al., 2017); its relevance to neural networks is tested empirically rather than inferred from parameter count.

A.2 Local Risk Decomposition, Including Estimation Bias

Let $\theta \in \mathbb{R}^D$ collect the LoRA factors, and restrict compressed adaptation to $\theta = Pz$, $z \in \mathbb{R}^d$, with fixed P and $\mathcal{S}_U = \mathrm{col}(P)$. For population risk $R(\theta) = \mathbb{E}_{x,y}\ell(f(x;\theta), y)$ and empirical risk $\widehat{R}_n(\theta) = n^{-1}\sum_i \ell(f(x_i;\theta), y_i)$, let $\theta^\star$ be a stationary full-space population optimum. It stays fixed when training samples are resampled. In its neighborhood,

$$R(\theta) - R(\theta^\star) \approx \tfrac{1}{2}\|\theta - \theta^\star\|_H^2, \qquad H = \nabla^2 R(\theta^\star) \succeq 0. \tag{12}$$

This requires local smoothness, not global convexity. Within this quadratic model the restricted optimum $\theta_U^\star = Pz_U^\star$ is the H-weighted projection of $\theta^\star$. Write the fitted estimators, allowing finite-sample or incomplete-optimization bias, as

$$\hat\theta_L = \theta^\star + b_L + \xi_L, \qquad \hat\theta_U = \theta_U^\star + Pb_z + P\zeta, \quad \mathbb{E}\xi_L = \mathbb{E}\zeta = 0, \tag{13}$$

where $b_L = \mathbb{E}\hat\theta_L - \theta^\star$, $b_z = \mathbb{E}\hat z_U - z_U^\star$, $\Sigma_L = \mathrm{Cov}(\xi_L)$, and $\Sigma_z = \mathrm{Cov}(\zeta)$. The identity $\mathbb{E}\|a+\xi\|_H^2 = \|a\|_H^2 + \mathrm{tr}(H\,\mathrm{Cov}\,\xi)$ and projection orthogonality $(\theta_U^\star - \theta^\star)^\top HPb_z = 0$ yield

$$\mathcal{E}_L := \mathbb{E}R(\hat\theta_L) - R(\theta^\star) \approx \tfrac{1}{2}\|b_L\|_H^2 + \tfrac{1}{2}\,\mathrm{tr}(H\Sigma_L), \tag{14}$$

$$\mathcal{E}_U := \mathbb{E}R(\hat\theta_U) - R(\theta^\star) \approx B_{\mathrm{proj}} + \tfrac{1}{2}\|Pb_z\|_H^2 + \tfrac{1}{2}\,\mathrm{tr}(HP\Sigma_z P^\top), \tag{15}$$

where $B_{\mathrm{proj}} = \frac{1}{2}\|\theta_U^\star - \theta^\star\|_H^2$. Setting $b_L = b_z = 0$ gives the centered comparison used in the main text:

$$\mathcal{E}_L \approx \tfrac{1}{2}\,\mathrm{tr}(H\Sigma_L), \qquad \mathcal{E}_U \approx B_{\mathrm{proj}} + \tfrac{1}{2}\,\mathrm{tr}(HP\Sigma_z P^\top). \tag{16}$$

Thus, within this approximation, compression helps when

$$2B_{\mathrm{proj}} + \mathrm{tr}(HP\Sigma_z P^\top) < \mathrm{tr}(H\Sigma_L). \tag{17}$$

With nonzero estimator bias, add $\|Pb_z\|_H^2$ and $\|b_L\|_H^2$ to the respective sides. Quadratic-approximation error remains outside these formulas; a small parameter count alone does not establish this inequality.

A.3 Energy–Noise Condition: Proof of Proposition 1

Fixed-instance identity. Fix $\theta^\star$ and P with $P^\top P = I_d$ and $0 < d < D$. Observe $y = \theta^\star + \sigma_{\mathrm{eff}}\epsilon$, $\epsilon \sim \mathcal{N}(0, I_D)$, and take excess risk $\frac{1}{2}\|\theta - \theta^\star\|_2^2$. Least squares gives $\hat{\theta}_L = y$ and $\hat{\theta}_U = PP^\top y$. With $Q_P = I_D - PP^\top$, $\hat{\theta}_U - \theta^\star = -Q_P\theta^\star + \sigma_{\mathrm{eff}} PP^\top\epsilon$. These terms are orthogonal for every noise draw and $\mathbb{E}\|PP^\top\epsilon\|^2 = d$, hence

$$\mathcal{E}_L = \tfrac{1}{2}D\sigma_{\mathrm{eff}}^2, \qquad \mathcal{E}_U = \tfrac{1}{2}\|Q_P\theta^\star\|^2 + \tfrac{1}{2}d\sigma_{\mathrm{eff}}^2. \tag{18}$$

All expectations here condition on the same target and projection. Subtraction proves the exact threshold

$$\mathcal{E}_L - \mathcal{E}_U = \tfrac{1}{2}(D-d)(\sigma_{\mathrm{eff}}^2 - \hat{v}_\perp), \qquad \hat{v}_\perp = \frac{\|Q_P\theta^\star\|^2}{D-d}. \tag{19}$$

The residual energy $\hat{v}_\perp$ is deterministic for this instance; no target distribution is required.

Additional ensemble average. For generated problems, let $\Theta^\star$ have zero mean and second moment vI_D, independently of P and observation noise. Then $\mathbb{E}[\|Q_P\Theta^\star\|^2 \mid P] = \mathrm{tr}(Q_P v I_D) = (D-d)v$. Averaging Eq. (18) over targets and, if random, projections gives

$$\mathbb{E}_{\mathrm{ens}}\mathcal{E}_L = \tfrac{1}{2}D\sigma_{\mathrm{eff}}^2, \qquad \mathbb{E}_{\mathrm{ens}}\mathcal{E}_U = \tfrac{1}{2}(D-d)v + \tfrac{1}{2}d\sigma_{\mathrm{eff}}^2, \tag{20}$$

so the ensemble advantage is positive exactly when

$$v < \sigma_{\mathrm{eff}}^2. \tag{21}$$

This proves Proposition 1. □

Equal-dimensional target-independent orthonormal projections have equal ensemble risk, but need not have equal bias on a fixed target. Repeated-noise trials estimate Eq. (18), not the additional task average. The theorem is exact for the sequence model, not for general neural fine-tuning. Moreover, $v = D^{-1}\mathbb{E}\|\Theta^\star\|^2$ depends on factor coordinates. Comparisons fix parameterization, initialization scale, and training protocol; the general local object is curvature-weighted B_{proj}, whose empirical proxies require separate validation.

Factor gauge and identifiable coordinates. For any invertible $r \times r$ matrix C, $BA = (BC)(C^{-1}A)$; this includes scaling, rank permutations, and rotations (Yen et al., 2025). Averaging unaligned factors across training seeds can change a residual diagnostic without a corresponding change in the trained functions. Factor-space residual measures need not be invariant to general LoRA factor transformations, even with diagonal empirical-Fisher weighting. This representation dependence can be avoided by defining bias and variance in prediction space, since exact factor transformations that preserve the model's weight updates leave its predictions unchanged.

A coordinate change must transform the compression constraint and metric consistently. Under $\theta' = T\theta$, the same constraint is $P' = TP$ and the same quadratic metric is $H' = T^{-\top}HT^{-1}$; selected directions become TE_I, generally no longer coordinate axes. Canonicalizing the target alone while retaining the original P, E_I changes the comparison. Averaging $B_\ell A_\ell$ avoids factor gauge, but the factor-to-weight map is bilinear, so the old factor-space projection cannot simply be reused in weight space. A replacement diagnostic requires a specified common local update space or a function-space fitting criterion.

A.4 Sparse Recovery and Matched Budgets: Proof of Proposition 2

Geometric preparation. For a support I of size K, set $E_I = [e_{i_1}, \ldots, e_{i_K}]$ and $\mathcal{S}_{U,I} = \mathrm{col}([P, E_I])$. Because $\mathcal{S}_U \subseteq \mathcal{S}_{U,I}$, minimizing the same quadratic error over these spaces proves

$\mathsf{Bias}_{U,I} \le \mathsf{Bias}_U$ for any $H \succeq 0$. This fixed-P comparison spends $d + K$ versus d and is not a matched-budget result. For $H = I_D$, $P^\top P = I_d$, let $q = Q_P\theta^\star$ and $A_I = Q_P E_I$. The best residual correction solves $\min_\alpha \|q - A_I\alpha\|^2$, giving

$$\mathsf{Bias}_{U,I} = \tfrac{1}{2}\|(I_D - \Pi_{A_I})q\|^2 = \tfrac{1}{2}\|q\|^2 - \tfrac{1}{2}\mathcal{E}_I, \qquad \mathcal{E}_I = \|\Pi_{A_I} q\|^2. \tag{22}$$

Thus $\rho_I^2 = \mathcal{E}_I/\|q\|^2 \in [0,1]$ measures recovered bias for $q \neq 0$; if $q = 0$, both biases vanish. Coordinate energy $\sum_{i\in I} q_i^2$ need not equal $\mathcal{E}_I$.

Risk comparison. For any subspace $\mathcal{S}$ of dimension m independent of estimation noise, orthogonal least squares gives

$$\mathbb{E}_{\mathrm{ens}}[R(\Pi_{\mathcal{S}} y) - R(\theta^\star)] = \tfrac{1}{2}\mathbb{E}_{\mathrm{ens}}\|(I_D - \Pi_{\mathcal{S}})\theta^\star\|^2 + \tfrac{1}{2} m\sigma_{\mathrm{eff}}^2. \tag{23}$$

For a target-independent subspace, isotropy makes the first expectation $(D - m)v$ by the trace identity in Proposition 1's proof. Let (P_B, P) be jointly independent of the target, with orthonormal columns of dimensions B and $d = B - K$. Support $\hat{I}$ may depend on the target, but observation noise is independent of $(\Theta^\star, P_B, P, \hat{I})$, and $\mathrm{rank}([P, E_{\hat{I}}]) = B$. Both variances are $B\sigma_{\mathrm{eff}}^2/2$, while Eq. (22) gives hybrid mean squared residual $(D - B + K)v - \mathbb{E}_{\mathrm{ens}}\mathcal{E}_{\hat{I}}$. Subtracting the baseline residual $(D - B)v$ proves

$$\mathbb{E}_{\mathrm{ens}}[R(\hat{\theta}_{\mathrm{hyb}}) - R(\hat{\theta}_{U,B})] = \tfrac{1}{2}(Kv - \mathbb{E}_{\mathrm{ens}}\mathcal{E}_{\hat{I}}). \tag{24}$$

This is the ensemble counterpart of Proposition 2. □

For fixed $\theta^\star, P, P_B, I$ satisfying the same noise-independence and rank conditions, the conditional comparison instead is

$$\mathbb{E}_\epsilon[R(\hat{\theta}_{\mathrm{hyb}}) - R(\hat{\theta}_{U,B}) \mid \theta^\star, P, P_B, I] = \tfrac{1}{2}\left(\|q\|^2 - \mathcal{E}_I - \|(I_D - P_B P_B^\top)\theta^\star\|^2\right). \tag{25}$$

Subtracting the two fixed-task risks proves Proposition 2. Hence Kv is an ensemble opportunity cost, not an exact cost for every realized pair of projections. Define the fixed-task cost

$$C_{\mathrm{task}} = \|q\|^2 - \|(I_D - P_B P_B^\top)\theta^\star\|^2. \tag{26}$$

The hybrid wins on this fixed task exactly when $\mathcal{E}_I > C_{\mathrm{task}}$ under the same estimator assumptions. This cost is nonnegative for nested $\mathrm{col}(P) \subseteq \mathrm{col}(P_B)$, but need not be nonnegative for arbitrary realized spaces. Higher coordinate capture than random support alone establishes neither this inequality nor its ensemble counterpart.

Corollary 1 (Uninformed supports provide no expected gain) *If $(P, \hat{I})$ is jointly independent of the target and observation noise and the augmented rank is B, then $\mathbb{E}_{\mathrm{ens}}\mathcal{E}_{\hat{I}} = Kv$ and the hybrid and budget-B compressed estimator have equal ensemble risk.*

Proof. Conditional on any selected target-independent subspace, its mean squared residual is $(D - B)v$. Comparing with $(D - B + K)v - \mathbb{E}_{\mathrm{ens}}\mathcal{E}_{\hat{I}}$ proves the first identity; Eq. (24) proves the second. □

Recoverability and its limits. The matched-budget advantage is $\frac{1}{2}(\mathbb{E}_{\mathrm{ens}}\mathcal{E}_{\hat{I}} - Kv)$, so gains require target-informed recovery above the opportunity cost. Under this noise-independent support model, the ideal support maximizes the joint residual recovery $\|\Pi_{A_I} q\|^2$. Coordinate-wise top-K selection is a surrogate that can approximate this objective when residualized directions are nearly orthogonal and have comparable norms. For a diffuse Gaussian target, the top-K coordinate energy is an $O((K/D)\log(D/K))$ fraction; under an incoherent projection with $d, K \ll D$, it cannot substantially remove the bias floor. The heavy-tailed synthetic target instead concentrates most realized residual energy on about πD coordinates. A near-tie on one fixed synthetic target does not directly verify the corollary's ensemble equality. Anisotropic curvature, conditioning, and optimization can break the tie in networks. The sensitivity-based diagnostic measures coordinate-level residual capture and its geometric corrections, rather than the population matched-budget condition itself.

Exact sequence risk with observation-dependent support. Fix $\theta^\star, P, P_B$ as in the fixed-task comparison, and allow $\hat{I} = \hat{I}(y)$ to depend on the observation, with $\mathrm{rank}([P, E_{\hat{I}}]) = B$ almost surely. Let $\Pi_{\hat{I}}$ project onto $\mathrm{col}([P, E_{\hat{I}}])$, and fit $\hat{\theta}_{\mathrm{hyb}} = \Pi_{\hat{I}} y$ by exact least squares. For every observation, including adaptively selected supports,

$$\|\hat{\theta}_{\mathrm{hyb}} - \theta^\star\|^2 = \|(I_D - \Pi_{\hat{I}})\theta^\star\|^2 + \sigma_{\mathrm{eff}}^2 \|\Pi_{\hat{I}}\epsilon\|^2. \tag{27}$$

The two error components lie in orthogonal spaces for that realized support; independence is not needed for this identity. Write $\mathcal{E}_{\hat{I}} = \|\Pi_{Q_P E_{\hat{I}}} q\|^2$ and

$$N_{\mathrm{adapt}} := \frac{\sigma_{\mathrm{eff}}^2}{2} \left(\mathbb{E}\|\Pi_{\hat{I}}\epsilon\|^2 - B \right). \tag{28}$$

Expectations condition on the fixed task and projections and include observation noise and any selector randomness. Subtracting from the fixed budget-B baseline gives the exact comparison

$$\mathcal{E}_{U,B} - \mathcal{E}_{\mathrm{hyb}} = \tfrac{1}{2} \left(\mathbb{E}\mathcal{E}_{\hat{I}} - C_{\mathrm{task}} \right) - N_{\mathrm{adapt}}. \tag{29}$$

Thus recovery must offset both the fixed-task opportunity cost and the change in retained noise. For a noise-independent support, $\mathbb{E}\|\Pi_{\hat{I}}\epsilon\|^2 = B$ and $N_{\mathrm{adapt}} = 0$, recovering the previous conditional comparison. No sign is asserted for N_{adapt} under an arbitrary selector. With random support, the expected geometric residual and retained-noise energy in Eq. (27) are not, in general, the squared bias and covariance trace of the complete estimator. In particular, a noisy-support/energy-surrogate risk gap does not isolate selection variance: both recovered signal and retained noise can change. This identity applies to exact sequence least squares, not directly to finite-step nonlinear training.

Estimator and selection variance. For a fixed support with $M_I = [P, E_I]$, a locally centered hybrid with covariance $M_I \Sigma_I M_I^\top$ improves over the fixed-P compressed estimator within the quadratic model when

$$\mathsf{Bias}_U - \mathsf{Bias}_{U,I} > \tfrac{1}{2} \left[\mathrm{tr}(H M_I \Sigma_I M_I^\top) - \mathrm{tr}(HP\Sigma_z P^\top) \right]. \tag{30}$$

Thus, residual recovery must compensate for any increase in estimation variability. When the support is selected from the same data used for fitting, the support becomes data-dependent, introducing additional selection variability and altering the within-support estimation law; consequently, the idealized $B\sigma_{\mathrm{eff}}^2/2$ variance need not hold. Equation (29) gives the corresponding exact recovery–retained-noise decomposition for the sequence model.

A.5 Joint Recovery and Practical Selector Approximations

Curvature-weighted recovery. Consider the exact quadratic objective $R_Q(\theta) = \frac{1}{2}\|\theta - \theta^\star\|_H^2$ with $H \succ 0$ in specified identifiable coordinates and full-column-rank P. Let $\theta_P^\star = P(P^\top HP)^{-1}P^\top H\theta^\star$ and $g = H(\theta_P^\star - \theta^\star)$, so $P^\top g = 0$. For $E = E_I$, define

$$M_I = E^\top HE - E^\top HP(P^\top HP)^{-1}P^\top HE, \qquad g_I = E^\top g. \tag{31}$$

Then joint refitting gives

$$R_Q(\theta_P^\star) - \min_{z,\alpha} R_Q(Pz + E\alpha) = \tfrac{1}{2} g_I^\top M_I^\dagger g_I. \tag{32}$$

To prove this, write the objective change as $g_I^\top \alpha + \frac{1}{2}\|P\delta z + E\alpha\|_H^2$. Minimizing over δz leaves $g_I^\top \alpha + \frac{1}{2}\alpha^\top M_I \alpha$. If $M_I a = 0$, then $Ea \in \mathrm{col}(P)$ and $a^\top g_I = 0$, so g_I lies in the range of M_I. Minimization over α proves Eq. (32), including redundant selected directions. This is a Schur-complement calculation, related to classical curvature-based parameter adjustment (Hassibi & Stork, 1993), applied here to releasing compression constraints. The fixed-P gain does not deduct the capacity displaced at a matched total budget.

For a single nonredundant coordinate the risk reduction is $g_i^2/(2M_{ii})$. Multiple coordinates interact through M_I, so ranking independent coordinate scores need not maximize joint recovery. At a finite warmup iterate, $P^\top g$ need not vanish; the formula above cannot be applied unchanged. In a singular factor parameterization, an identifiable common space must first be specified; a pseudoinverse alone does not justify local identifiability.

From exact recovery to coordinate-wise selection. Equation (32) defines the local gain for a fixed support,

$$\Delta(I) = \frac{1}{2} g_I^\top M_I^\dagger g_I. \tag{33}$$

The corresponding ideal support is

$$I^\star = \arg\max_{|I|=K} g_I^\top M_I^\dagger g_I. \tag{34}$$

This objective is generally impractical at realistic LoRA dimensions: the search is combinatorial in D and K, while each candidate support retains interactions through the $K \times K$ matrix M_I. Consequently, the exact objective provides a design criterion rather than an executable selector.

Diagonal recovery approximation. If interactions among released coordinates are neglected, M_I is approximated by its diagonal and Eq. (34) becomes approximately additive:

$$g_I^\top M_I^{-1} g_I \approx \sum_{i \in I} \frac{g_i^2}{M_{ii}}. \tag{35}$$

This suggests ranking coordinates by g_i^2/M_{ii}. The approximation discards the joint interactions that make Eq. (34) exact, and therefore need not preserve the optimal support.

One-hot sharing with diagonal curvature. For Uni-LoRA's normalized one-hot sharing, let coordinate i belong to group $\mathcal{G}_j$, and approximate

$$H \approx \mathrm{diag}(h_1, \ldots, h_D). \tag{36}$$

Define

$$H_j = \sum_{k \in \mathcal{G}_j} h_k, \qquad S_j = \sum_{k \in \mathcal{G}_j} \bar{g}_k, \tag{37}$$

where $\bar{g}_i$ is the signed gradient averaged over the scoring window. The diagonal Schur complement is

$$M_{ii} = h_i - \frac{h_i^2}{H_j}. \tag{38}$$

At a finite warmup iterate, $P^\top \bar{g}$ need not vanish. We therefore remove the component that can be absorbed by a refit of the compressed group:

$$\tilde{g}_i = \bar{g}_i - \frac{h_i}{H_j} S_j. \tag{39}$$

Equations (38)–(39) yield the curvature-aware coordinate score

$$s_i^{\mathrm{RF}} = \frac{\left(\bar{g}_i - \frac{h_i}{H_j} S_j\right)^2}{h_i - \frac{h_i^2}{H_j} + \lambda}. \tag{40}$$

We call this *Recovery-Fisher* (RF). In our implementation,

$$h_i = \frac{1}{T_s} \sum_{t=1}^{T_s} \left(g_i^{(t)}\right)^2, \tag{41}$$

which is a minibatch-gradient second-moment proxy for diagonal curvature, not a per-example Fisher matrix or the true Hessian. The damping parameter $\lambda > 0$ stabilizes coordinates with small estimated effective curvature.

Residualized Gradient. If curvature is further treated as approximately homogeneous within sharing groups, and the small group-size-dependent denominator is ignored, Eq. (40) reduces to the simpler score

$$s_i^{\mathrm{RG}} = \left(\bar{g}_i - \frac{1}{|\mathcal{G}_j|} \sum_{k \in \mathcal{G}_j} \bar{g}_k \right)^2 . \tag{42}$$

We call this rule *Residualized Gradient* (RG). It removes the gradient component that the shared latent coordinate can approximately realize and prioritizes coordinate-specific disagreement within a sharing group.

Both RG and RF are therefore approximations to the release objective, rather than consequences of the exact theory. RF additionally substitutes noisy finite-window second moments for curvature and diagonalizes the joint interaction matrix. Their practical quality must consequently be established empirically.

Proof of the shared-group formula. For the normalized one-hot reconstruction, write its disjoint nonempty groups as G_j, with sizes m_j. Under $H = I$, the restricted fit subtracts group means, giving residual $q = (I - PP^\top)\theta^\star$ with $\sum_{i \in G_j} q_i = 0$. Let $I_j = I \cap G_j$ and $k_j = |I_j|$. Selected entries can be fitted individually. The remaining entries share a refitted mean, so their squared residual is

$$\sum_{i \in G_j \setminus I_j} q_i^2 - \frac{\left(\sum_{i \in G_j \setminus I_j} q_i \right)^2}{m_j - k_j} \quad (k_j < m_j). \tag{43}$$

Subtracting from $\sum_{i \in G_j} q_i^2$ and using the zero group sum proves Eq. (9). Orthogonality across groups makes their recovery additive. If $k_j = m_j$, the group is fitted exactly and its recovered energy is $\sum_{i \in G_j} q_i^2$; no division by zero is used. The residualized-coordinate Gram matrix is $I_{k_j} - \mathbf{1}\mathbf{1}^\top / m_j$: its rank is k_j if $k_j < m_j$, and $m_j - 1$ otherwise. Hence

$$\operatorname{rank}([P, E_I]) = d + K - \#\{j : I_j = G_j\}. \tag{44}$$

Scalar parameter counts and effective dimension thus differ when an entire group is selected. In isotropic sequence estimation, variance is proportional to this actual rank. The equal-rank premise of Proposition 2 must be checked separately.

For example, take one group with $q = (2, 1, -1, -2)$ and $K = 2$. The coordinate-energy rule selects entries $\{1, 4\}$ and recovers energy 8. Selecting $\{1, 2\}$ captures coordinate energy 5 but recovers $5 + 3^2/2 = 9.5$ after refitting the remaining group. Both supports have the same effective rank. A coordinate-energy surrogate is therefore not necessarily a joint-recovery oracle, even with an exactly known target.

Fixed-task recovery with anisotropic noise. The same calculation extends to $y = \theta^\star + \xi$, with $\mathbb{E}\xi = 0$, covariance C, and exact H-weighted projection estimators. Put $v = H^{1/2}\theta^\star$, $\Omega = H^{1/2} C H^{1/2}$, and let $\Pi_P, \Pi_B, \Pi_{P,I}$ project onto the columns of $H^{1/2}P, H^{1/2}P_B, H^{1/2}[P, E_I]$, respectively. Define

$$\mathcal{E}_{I,H} = \|(I - \Pi_P)v\|^2 - \|(I - \Pi_{P,I})v\|^2, \tag{45}$$

$$C_{\mathrm{task},H} = \|(I - \Pi_P)v\|^2 - \|(I - \Pi_B)v\|^2. \tag{46}$$

For fixed support independent of estimation noise, each risk is $\frac{1}{2}\|(I - \Pi)v\|^2 + \frac{1}{2}\operatorname{tr}(\Pi\Omega)$, yielding

$$\mathcal{E}_{U,B} - \mathcal{E}_{\mathrm{hyb}} = \tfrac{1}{2}\left\{ \mathcal{E}_{I,H} - C_{\mathrm{task},H} - \operatorname{tr}[(\Pi_{P,I} - \Pi_B)\Omega] \right\}. \tag{47}$$

Equal rank cancels the noise term only for isotropic Ω (or other cases where the two traces happen to agree). For observation-dependent support, replace half the trace difference by $\frac{1}{2}\mathbb{E}[\|\Pi_{P,\hat{I}} H^{1/2}\xi\|^2 - \|\Pi_B H^{1/2}\xi\|^2]$ and average recovery over selection. This follows from realized projection orthogonality, not independence. The geometric and retained-noise terms need not be the squared bias and covariance trace of the complete adaptive estimator. These identities concern specified projection estimators, not arbitrary SGD solutions.

Scope of sensitivity selection. Warmup SNIP accumulates removal sensitivities $|\theta_i g_i|$ (Lee et al., 2019); it does not optimize Eq. (32). If a group's constrained optimum is zero while its residual is nonzero, instantaneous SNIP scores vanish although releasing some entries strictly improves the quadratic fit. This static counterexample does not assert that accumulated scores along a nonstationary warmup trajectory vanish. It rules out an unconditional equivalence between sensitivity and recovery. Similarly, the one-hot formulas fix factor coordinates and metric; under a coordinate transformation the target, compression directions, selected directions and metric must all be transported. Their exactness does not make raw factor-space capture a gauge-invariant predictor of neural-network risk.

Computational cost. Naive exact optimization of Eq. (34) requires a combinatorial search over size-K supports and repeatedly accounting for K-way interactions. In contrast, RG and RF use statistics already available during the scoring backward passes. Their additional accumulation cost is $O(T_s D)$, followed by $O(D)$ group reductions and a global top-K operation; RG stores one D-dimensional gradient accumulator, while RF additionally stores one second-moment accumulator.

B ProLoSA Implementation

B.1 ProLoSA Parameterization

Vectorize all adapted layers' LoRA factors as $\theta_D = \mathrm{vec}(\{A_\ell, B_\ell\}) \in \mathbb{R}^D$, where $\Delta W_\ell = B_\ell A_\ell$ and W_0 is frozen. ProLoSA reconstructs

$$\theta_D = P\theta_d + R\alpha, \tag{48}$$

where $\theta_d \in \mathbb{R}^d$ is the compressed latent parameter, $P \in \mathbb{R}^{D\times d}$ is a fixed projection matrix, $R \in \{0,1\}^{D\times K}$ routes sparse residual coefficients to selected full-space coordinates, and $\alpha \in \mathbb{R}^K$ contains the trainable sparse coefficients, for a total budget of $B = d + K$ trainable parameters with $d, K \ll D$.

We instantiate P with the normalized one-hot projection of Uni-LoRA (Li et al., 2025): an assignment $\phi : \{1,\dots,D\} \to \{1,\dots,d\}$, sampled once at initialization and kept fixed, maps each full-space coordinate to a latent coordinate, and

$$P_{ij} = \begin{cases} c_j^{-1/2}, & \text{if } j = \phi(i), \\ 0, & \text{otherwise,} \end{cases} \qquad c_j = |\{i : \phi(i) = j\}|, \tag{49}$$

with nonempty groups, so that each row of P has exactly one nonzero entry and $P^\top P = I_d$. The projected branch ties the D LoRA coordinates into d shared groups, while $R\alpha$ releases selected coordinates. Joint refitting can also improve the remaining shared entries (Eq. (9)). The storage budget is $d + K$; its effective factor-space rank is smaller if entire groups are selected (Eq. (44)). This rank correction is separate from nonlinear factor-gauge identifiability. After reconstruction, θ_D is unpacked into layer-wise factors (A_ℓ, B_ℓ) with $\Delta W_\ell = B_\ell A_\ell$; since the final update consists of standard LoRA factors, it can be merged into the frozen backbone with no additional inference latency.

B.2 Warmup Support Selection and Training

With sparse coefficients inactive, optimize θ_d alone for $T_{\mathrm{pre}} = \lfloor \rho_{\mathrm{w}} T \rfloor$ score-free steps and then for T_{s} scoring steps, accumulating

$$s_i = \sum_{t=T_{\mathrm{pre}}+1}^{T_{\mathrm{pre}}+T_{\mathrm{s}}} |\theta_{D,i}^{(t)} g_i^{(t)}|, \qquad g^{(t)} = \nabla_{\theta_D^{(t)}} \mathcal{L}^{(t)}. \tag{50}$$

The SNIP criterion (Lee et al., 2019; He et al., 2022) follows from the first-order loss change $g_i^{(t)}\delta_i$ with removal perturbation $\delta_i = -\theta_{D,i}^{(t)}$. Select the top-$K$ scores, fix $R = [e_{i_1}, \dots, e_{i_K}]$, initialize $\alpha = 0$, and jointly train θ_d, α thereafter with W_0, P, R fixed. Activation occurs after $T_{\mathrm{pre}} + T_{\mathrm{s}}$ steps. Some runs specify T_{pre} directly rather than through a fraction of training; the scoring window and

optimizer-reset settings are experiment-specific (Appendix D). A score-free warmup of 128 steps and a 128-step scoring window are distinct schedules.

This sensitivity score distinguishes coordinates sharing a latent value through their gradients. Its ability to identify useful residual coordinates is evaluated empirically through coordinate enrichment, joint recovery, and downstream performance (Sections 4 and 5).

Sparse-dominant boundary. The boundary row uses $(d, K) = (1, 23039)$, not a sparse-only estimator. With $P = \mathbf{1}_D/\sqrt{D}$ and the latent scalar initialized uniformly in $[-0.02, 0.02]$, the factors are not both zero, avoiding zero-factor gradient degeneracy. The scalar supplies warmup/SNIP scores and remains trainable after residual activation; exact settings are in Appendix D.6.

B.3 Training Algorithm

Algorithm 1 ProLoSA with Warmup SNIP Support Selection

Require: Frozen W_0; fixed P; sparse budget K; warmup ratio ρ_{w}; scoring steps T_{s}; total steps $T > T_{\mathrm{pre}} + T_{\mathrm{s}}$.

Initialize θ_d, $s \leftarrow 0_D$, and $T_{\mathrm{pre}} \leftarrow \lfloor \rho_{\mathrm{w}} T \rfloor$; keep sparse branch inactive.
for $t = 1$ to $T_{\mathrm{pre}} + T_{\mathrm{s}}$ **do**
  Reconstruct $\theta_D = P\theta_d$, unpack LoRA factors, and compute loss $\mathcal{L}$ and gradient $g = \nabla_{\theta_D}\mathcal{L}$.
  if $t > T_{\mathrm{pre}}$ **then**
    Accumulate $s \leftarrow s + |\theta_D \odot g|$.
  end if
  Update θ_d by back-propagation.
end for
Set $I \leftarrow \mathrm{TopK}(s, K)$, $R \leftarrow [e_{i_1}, \ldots, e_{i_K}]$, and $\alpha \leftarrow 0$.
Apply the benchmark's optimizer-reset and sparse learning-rate settings.
for $t = T_{\mathrm{pre}} + T_{\mathrm{s}} + 1$ to T **do**
  Reconstruct $\theta_D = P\theta_d + R\alpha$, unpack LoRA factors, and compute $\mathcal{L}$.
  Update θ_d, α by back-propagation, keeping W_0, P, R fixed.
end for
return Trained θ_d, α and routing R; merge the reconstructed LoRA update for inference.

C Mechanism Experiments

C.1 Projection-Blind Sequence Experiments

Fixed-target protocol. Independently of the projection, generate each target coordinate with $b_j \sim \mathrm{Bernoulli}(\pi)$, $g_j \sim \mathcal{N}(0,1)$, and $T_j \sim t_\nu$:

$$\Theta_j^\star = (1 - b_j)\tau_0 g_j + b_j \tau_1 \sqrt{(\nu-2)/\nu}, T_j. \tag{51}$$

The normalization makes τ_1 the slab standard deviation. We use this spike-and-heavy-tail mixture as a controlled stress test, motivated by empirical observations that parameter differences can exhibit structured or sparse concentration (Wang et al., 2026) and that delta weights can have long-tailed spectral structure (Ping et al., 2024). The narrow Gaussian represents a concentrated bulk, while the Student-t component introduces rare large deviations; the mixture is not intended as a fitted generative model of real LoRA updates. Appendix C.3 gives all settings; the target law has second moment $v = (1-\pi)\tau_0^2 + \pi\tau_1^2 = 2.52475 \times 10^{-3}$.

Seed 13 draws one target $\theta^\star$ and independent Gaussian-QR projection P, held fixed over every noise level and all $M = 200$ trials. Only $\epsilon^{(m)} \sim \mathcal{N}(0, I_D)$ is resampled in $y^{(m)} = \theta^\star + (\sigma_y/\sqrt{n_{\mathrm{eff}}})\epsilon^{(m)}$. Each estimator solves least squares in its represented space. The noise-dependent support uses top-K of $|y - PP^\top y|$, an estimation-model analog of warmup SNIP rather than the actual network score.

Measurement and conditioning. Population excess risk is $\frac{1}{2}\|\hat\theta - \theta^\star\|^2$. With $\bar\theta = M^{-1}\sum_m \hat\theta^{(m)}$, empirical bias $\frac{1}{2}\|\bar\theta - \theta^\star\|^2$ plus variance $(2M)^{-1}\sum_m \|\hat\theta^{(m)} - \bar\theta\|^2$ equals mean risk exactly; the bias includes finite-M noise. Each decomposition conditions on one target and projection. Different

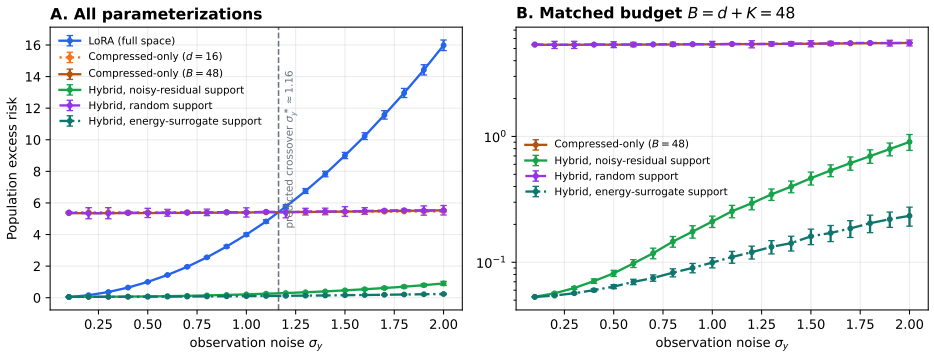

Figure 4: **Matched-budget comparison in the projection-blind sequence model** ($B = d + K = 48$, heavy-tailed target). **A:** population excess risk of all parameterizations; the dashed vertical line marks the fixed-instance full-space/compressed crossover $\sigma_y^* \approx 1.16$ predicted by Eq. (18). **B:** matched-budget methods only (log scale). Random support nearly ties budget-B compression on this instance; this is not a task-ensemble test of Corollary 1. The growing gap between noisy-residual and target-informed top-K supports combines recovery loss and retained-noise effects (Eq. (29)). Error bars show trial standard deviations, not standard errors. The noisy-residual selector is a SNIP analog, while the energy surrogate uses target residual energy and is not the joint-recovery optimum.

targets require separate conditional decompositions before any task averaging; pooling estimator dispersion across targets would include variation of their means. Random-support variation belongs to the estimator variance within a fixed-target experiment.

Matched-budget results. Figure 4 compares budget-$B = 48$ compression with $d = 16$, $K = 32$ hybrids using random, noisy-residual, or target-informed energy-surrogate supports. The energy surrogate ranks coordinates by their residual energy and does not optimize joint recovery over all size-K subsets.

All methods share the target and per-trial noise. For the B-dimensional projection, $\hat{v}_{\perp,B} = 2.6427 \times 10^{-3}$, so the conditional threshold is $\sigma_y^\star = \sqrt{n_{\text{eff}} \hat{v}_{\perp,B}} = 1.163$, matching the crossing near 1.16. The random hybrid nearly ties compression (risks $5.37/5.36$ at $\sigma_y = 0.5$, $5.46/5.45$ at 1.5); this single-instance agreement is not a direct estimate of the ensemble corollary. The realized target contains 33 slab coordinates, close to the sparse budget $K = 32$: top-32 coordinates contain 98.8% of residual coordinate energy, and the augmented space removes 99.0% of projection bias. This is a favorable fixed instance, not evidence of typical recovery across target draws. Noise-dependent supports achieve risk 0.08 at $\sigma_y = 0.5$, but their gap to the energy surrogate grows to 0.47 versus 0.16 at 1.5. Equation (29) separates changes in recovered signal from changes in retained noise; this gap alone does not identify selection variance.

The earlier $d = 16$ compressed-only comparison uses fewer parameters than the hybrid and is archived as a bias-recovery diagnostic. Its residual threshold is 1.165; at $\sigma_y = 0.5/1.5$, full-space risks are $1.00/9.00$, compressed risks $5.41/5.44$, and hybrid risks $0.08/0.47$. It does not establish equal-budget superiority.

C.2 The Bias-Dominated Failure Regime

Replace the target law by $\Theta_j^\star \sim \mathcal{N}(0, 0.125^2)$, holding one independently generated target and projection fixed as above. Its realized $\hat{v}_\perp = 1.5685 \times 10^{-2}$ exceeds effective noise variance throughout the tested range (at most 7.8×10^{-3}); other settings are unchanged. Figure 5 shows a compressed bias floor of 32.0, above the full-space risk even at $\sigma_y = 2$ (16.0). The conditional crossing $\sigma_y^\star = 2.83$ is outside the sweep. Top-K support carries only 6.9% of target energy: at $\sigma_y = 1.5$, the tested hybrid still has risk 30.4 versus 32.0 for compression. Sparse correction under this protocol therefore removes little of the error. The diffuse second moment 0.015625 is 6.19 times the mix-

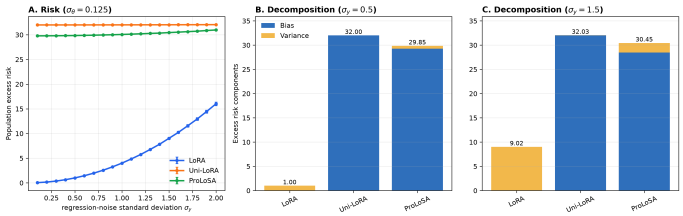

Figure 5: **The bias-dominated failure regime.** With a fixed dense Gaussian target ($\tau = 0.125$, with realized $\hat{v}_\perp > \sigma_{\text{eff}}^2$ throughout the tested range), compression remains near its bias floor of 32.0 and loses to full-space estimation (**A**), as predicted by Proposition 1. The tested hybrid recovers little error (**B**, **C**); its recovery is a separate observation, not a conclusion of Proposition 1.

ture's, so magnitude and concentration both change. This is a failure example, not a matched-energy isolation of concentration or evidence that all downstream updates have the favorable profile.

C.3 Synthetic Experiment Configuration

The matched-budget run adds the dimension-$B = 48$ projection and the three support rules described in Appendix C.1, reusing the same target and noise trials.

Sequence settings. $D = 4096$, $d = 16$, $K = 32$, $n_{\text{eff}} = 512$, $M = 200$, seed 13; σ_y spans $[0.1, 2.0]$. The mixture uses $\pi = 0.01$, Gaussian std. $\tau_0 = 0.005$, and Student-t_3 slab std. $\tau_1 = 0.5$; the failure target is $\mathcal{N}(0, 0.125^2)$. Projection is Gaussian-QR orthonormal, drawn independently of the target and fixed across trials. Support uses top-K of $|y - PP^\top y|$; evaluation is population excess risk. Target-independent normalized one-hot and QR projections have equal isotropic ensemble risk, not necessarily equal fixed-instance risk.

C.4 Predictive Decomposition with a Weight-Space Spike–Slab Teacher

Teacher and supervision. The frozen model is a single Transformer block with hidden width 64, four attention heads, feed-forward width 256, sequence length 32, and output dimension $p = 16$. We perturb the 49,152 weight entries of its query, key, value, attention-output, and two feed-forward matrices. Using one fixed draw (seed 13), independently of the compression maps and support selection, we set

$$\Delta W_j^\star = c\left[(1 - b_j)\,0.005 g_j + b_j \frac{0.5}{\sqrt{3}} T_j\right], \quad b_j \sim \text{Bernoulli}(0.01), \quad g_j \sim \mathcal{N}(0,1), \quad T_j \sim t_3. \tag{52}$$

This uses the same mixture parameters as the sequence experiment, now in physical weight coordinates. It is a synthetic construction, not a fitted distribution of downstream updates. An independent 1024-example calibration set fixes $c = 0.7042098345$ so that the mean squared output displacement is 0.1 times the frozen model's output variance. The resulting displacement RMS is $s_\Delta = 0.05466752028$. The largest approximately 1% of weight updates carry 98.756% of their squared energy; the other entries remain nonzero almost surely. Each update matrix has numerical rank 64, so rank-4 LoRA cannot exactly reconstruct the teacher weights. This does not establish a lower bound on its prediction error, nor does weight concentration imply sparsity in LoRA-factor coordinates.

Training inputs are independent standard-Gaussian 32×64 sequences. Writing $f_\star = f_{W_0 + \Delta W^\star}$, the labels and objective are

$$y_i = f_\star(x_i) + \lambda s_\Delta \epsilon_i, \qquad \epsilon_i \sim \mathcal{N}(0, I_p), \qquad \min_\phi \frac{1}{np} \sum_{i=1}^{n} \|f_\phi(x_i) - y_i\|_2^2, \tag{53}$$

with $n = 256$ and $\lambda \in \{0, 0.5, 1, 2, 4\}$. Teacher weights and slab membership are never used as training targets or support labels. All methods adapt the same six matrices using rank-4 factors; biases, LayerNorm, and readout are frozen.

Table 3: **Clean prediction risk and matched-budget improvement.** Risk values are multiplied by 10^3. Relative reduction is $100(1 - \widehat{\mathcal{R}}_{\mathrm{Pro}}/\widehat{\mathcal{R}}_{\mathrm{Uni}})$; brackets are approximate pointwise 95% paired-bootstrap intervals.

λ	LoRA	Uni-LoRA	ProLoSA	Relative reduction (%)
0	0.3232	1.1918	0.8801	26.15 [24.90, 27.31]
0.5	0.4809	1.2519	0.9422	24.74 [23.68, 25.79]
1	0.9098	1.4010	1.0875	22.38 [21.07, 23.69]
2	2.4350	1.8154	1.6259	10.44 [6.15, 14.51]
4	7.3399	2.7879	2.4909	10.66 [7.40, 13.96]

Training protocol. LoRA trains 4608 factor entries, whereas Uni-LoRA uses a 576-entry shared bank. ProLoSA allocates the same 576 trainable scalars between the shared bank and residual coordinates. The fixed configurations use learning rates of 2×10^{-4} for LoRA and 5×10^{-4} for the shared banks of Uni-LoRA and ProLoSA. ProLoSA uses $(d, K) = (512, 64)$, a sparse learning rate of 0.002, and 256 warmup steps followed by 128 sum-SNIP scoring steps; the zero-initialized sparse branch is activated after step 384.

All methods are fitted with AdamW for 1000 total steps, using zero weight decay, batch size 128, and gradient-norm clipping at 1. LoRA and Uni-LoRA share their initial factor values, while ProLoSA's different bank dimension induces a different fixed map and initialization. For each method, the initialization, map, and minibatch-index random stream are held fixed across repeats. The 20 repeats independently resample training inputs and label noise and are paired across methods; standardized noise is additionally shared across noise levels. The teacher and 1024 noiseless test inputs remain fixed. Effective trainable-scalar counts refer to the optimized parameter budget and do not imply equal storage or equal functional dimension.

Predictive decomposition and uncertainty. Let $\|u\|_{\mathcal{T}}^2 = (Np)^{-1} \sum_{i=1}^{N} \|u(x_i)\|_2^2$ on the fixed test set, $M = 20$, and $\bar{f} = M^{-1} \sum_{m=1}^{M} \widehat{f}^{(m)}$. We report

$$\widehat{\mathcal{R}} = \frac{1}{M} \sum_{m=1}^{M} \|\widehat{f}^{(m)} - f_\star\|_{\mathcal{T}}^2, \qquad \widehat{\mathcal{V}} = \frac{1}{M-1} \sum_{m=1}^{M} \|\widehat{f}^{(m)} - \bar{f}\|_{\mathcal{T}}^2, \tag{54}$$

$$\widehat{\mathcal{B}^2} = \|\bar{f} - f_\star\|_{\mathcal{T}}^2 - \frac{\widehat{\mathcal{V}}}{M}, \qquad \widehat{\mathcal{R}} = \widehat{\mathcal{B}^2} + \widehat{\mathcal{V}}. \tag{55}$$

The correction removes the finite-replicate contribution to squared bias; negative corrected values, if any, are retained. We use 10,000 paired dataset-bootstrap resamples and approximate basic intervals centered at the empirical population functional. Intervals are pointwise, without a multiple-comparison correction, and condition on the locked configuration search and fixed test set. They do not include uncertainty from rerunning selection or drawing new teachers, projections, or test sets.

Observed tradeoffs. Both compressed methods have higher corrected squared bias than LoRA at every noise level. At $\lambda = 0.5$ and 1, their lower variance does not offset this bias increase; at 2 and 4, it does, and the bias increase, variance saving, and net risk reduction each have positive pointwise intervals. At zero label noise, Uni-LoRA instead has higher variance than LoRA, illustrating variation from random training inputs and finite fitting. Against equal-budget Uni-LoRA, ProLoSA lowers both squared bias and variance through $\lambda = 2$, with paired intervals supporting both reductions. At $\lambda = 4$, the squared-bias reduction is 0.0583×10^{-3} with interval $[-0.0616, 0.1814] \times 10^{-3}$, while the variance reduction is 0.2388×10^{-3} with interval $[0.1437, 0.3245] \times 10^{-3}$. Thus the highest-noise improvement cannot be attributed solely to bias recovery.

Optimization controls and interpretation. For the first five paired repeats at $\lambda = 1$ and 4, extending the same fits to 3000 steps retains ProLoSA's relative risk reductions of 8.81% (95% interval $[3.15, 14.25]$) and 9.10% ($[4.93, 11.78]$), respectively. These five-repeat controls are separate from the 20-repeat main estimates. A further noiseless control uses three new paired datasets of size 4096 and the same locked hyperparameters (Table 4). All methods improve with longer fitting; Pro-

Table 4: **Large-sample noiseless control.** Mean clean test MSE multiplied by 10^3, over three new paired training datasets with $n = 4096$. Configurations remain fixed.

Method	1000 steps	4000 steps
LoRA	0.2468	0.0984
Uni-LoRA	1.0623	0.5361
ProLoSA	0.8434	0.4616

LoSA retains a 13.90% mean risk reduction over Uni-LoRA at 4000 steps. Consequently, 1000-step squared bias is not a pure representational or projection-bias floor.

This comparison evaluates complete validation-selected training procedures. Their learning rates differ, and the 576-dimensional compressed bank and 512-plus-64 hybrid do not define nested factor spaces. Equal trainable counts therefore do not imply equal predictive variance. The measured bias includes representation, finite-sample fitting, optimization, and adaptive selection; the weight-space teacher and output-displacement calibration do not certify the local factor-space assumptions. This single fixed teacher provides conditional evidence of predictive tradeoffs, not a causal isolation of concentration or a universal advantage of compression or sparse correction.

D Downstream Benchmarks and Settings

This appendix provides implementation details omitted from the main text for reproducibility. Unless otherwise specified, the pretrained backbone is frozen and only PEFT parameters are updated. Controlled comparisons share the settings specified for that experiment. Published baseline rows retain their original protocols and are distinguished from paired reruns; a shared benchmark or parameter budget alone does not establish identical training and selection procedures. For **ProLoSA**, the sparse support is selected after compressed warmup using the accumulated sensitivity score defined in Appendix B.2. When warmup is specified as a fraction, $T_{\mathrm{pre}} = \lfloor \rho_{\mathrm{w}} T \rfloor$ is the score-free phase; activation follows an additional T_{s} scoring steps. Other runs specify T_{pre} directly. The selected support is fixed during the remaining training stage, and the sparse coefficients are initialized to zero. All reported training time and memory statistics include method-specific stages, including the warmup stage of **ProLoSA**.

D.1 GLUE Results

Table 5 retains the published-baseline comparison for consistency with prior work. Because those published results use source-specific aggregation and training realizations, they should not be interpreted as paired method comparisons. We therefore additionally rerun Uni-LoRA and ProLoSA using the same fixed splits and five matched seeds for a controlled statistical comparison. These matched reruns are used for the paired confidence-interval analysis reported in Section 5, whereas the published rows are retained only as external reference baselines.

Under the matched five-seed protocol, ProLoSA achieves an average GLUE score of 88.734, compared with 88.222 for Uni-LoRA (Table 6). The paired average improvement is $+0.511$ points, with a 95% confidence interval of $[0.042, 0.981]$. At the task level, the gains are heterogeneous. MRPC shows a positive paired difference of $+0.997$ with a 95% confidence interval excluding zero, whereas the intervals for SST-2, CoLA, QNLI, RTE, and STS-B include zero. These matched reruns therefore support a positive aggregate difference between ProLoSA and Uni-LoRA, while indicating that the improvement is not uniform across individual GLUE tasks.

D.2 Full Mathematical Reasoning Results

Evaluation protocol. We evaluate LoRA, Uni-LoRA, and ProLoSA on GSM8K and MATH500 using three training seeds, 42–44. All methods are trained on MetaMathQA. Hyperparameters and training configurations are selected using held-out validation data only, without using GSM8K or MATH500 performance. After configuration selection, the resulting configurations are fixed and

Table 5: Results with RoBERTa$_{\text{base}}$ and RoBERTa$_{\text{large}}$ on GLUE (Wang et al., 2018): Matthews correlation (CoLA), Pearson correlation (STS-B), and accuracy (others). The median performance over 5 runs with different random seeds is reported. Baseline results are taken from Uni-LoRA (Li et al., 2025).

Backbone	Method	Trainable Params	SST-2	MRPC	CoLA	QNLI	RTE	STS-B	Avg.
BASE	FT	125M	94.8	90.2	63.6	92.8	78.7	91.2	85.2
	LoRA	0.295M	**95.1**$_{\pm 0.2}$	89.7$_{\pm 0.7}$	63.4$_{\pm 1.2}$	**93.3**$_{\pm 0.3}$	**86.6**$_{\pm 0.7}$	**91.5**$_{\pm 0.2}$	**86.6**
	VeRA	0.043M	94.6$_{\pm 0.1}$	89.5$_{\pm 0.5}$	**65.6**$_{\pm 0.8}$	91.8$_{\pm 0.2}$	78.7$_{\pm 0.7}$	90.7$_{\pm 0.2}$	85.2
	Tied-LoRA	0.043M	94.4$_{\pm 0.5}$	88.5$_{\pm 1.0}$	61.9$_{\pm 1.6}$	92.0$_{\pm 0.4}$	76.2$_{\pm 1.3}$	89.8$_{\pm 0.3}$	83.8
	VB-LoRA	0.075M†	94.4$_{\pm 0.2}$	89.5$_{\pm 0.5}$	63.3$_{\pm 0.7}$	92.2$_{\pm 0.2}$	82.3$_{\pm 1.3}$	90.8$_{\pm 0.1}$	85.4
	FourierFT	0.024M	94.2$_{\pm 0.3}$	**90.0**$_{\pm 0.8}$	63.8$_{\pm 1.6}$	92.2$_{\pm 0.1}$	79.1$_{\pm 0.5}$	90.8$_{\pm 0.2}$	85.0
	Uni-LoRA	0.023M	94.5$_{\pm 0.2}$	89.7$_{\pm 0.4}$	64.6$_{\pm 1.0}$	92.3$_{\pm 0.4}$	79.8$_{\pm 1.1}$	90.9$_{\pm 0.2}$	85.3
	ProLoSA	0.023M	94.9$_{\pm 0.3}$	89.5$_{\pm 0.6}$	63.5$_{\pm 0.2}$	92.5$_{\pm 0.2}$	81.7$_{\pm 0.6}$	90.8$_{\pm 0.2}$	85.5
LARGE	LoRA	0.786M	96.2$_{\pm 0.5}$	90.2$_{\pm 1.0}$	68.2$_{\pm 1.9}$	**94.8**$_{\pm 0.3}$	85.2$_{\pm 1.1}$	**92.3**$_{\pm 0.5}$	87.8
	VeRA	0.061M	96.1$_{\pm 0.1}$	90.9$_{\pm 0.7}$	68.0$_{\pm 0.8}$	94.4$_{\pm 0.2}$	85.9$_{\pm 0.7}$	91.7$_{\pm 0.8}$	87.8
	Tied-LoRA	0.066M	94.8$_{\pm 0.6}$	89.7$_{\pm 1.0}$	64.7$_{\pm 1.2}$	94.1$_{\pm 0.1}$	81.2$_{\pm 0.1}$	90.8$_{\pm 0.3}$	85.9
	VB-LoRA	0.162M†	96.1$_{\pm 0.2}$	91.4$_{\pm 0.6}$	68.3$_{\pm 0.7}$	94.7$_{\pm 0.5}$	86.6$_{\pm 1.3}$	91.8$_{\pm 0.1}$	88.2
	LoRA-XS	0.025M	95.9$_{\pm 0.3}$	90.7$_{\pm 0.4}$	67.0$_{\pm 1.2}$	93.9$_{\pm 0.1}$	**88.1**$_{\pm 0.3}$	92.0$_{\pm 0.1}$	87.9
	FourierFT	0.048M	96.0$_{\pm 0.2}$	90.9$_{\pm 0.3}$	67.1$_{\pm 1.4}$	94.4$_{\pm 0.4}$	87.4$_{\pm 1.6}$	92.0$_{\pm 0.4}$	88.0
	Uni-LoRA	0.023M	96.3$_{\pm 0.2}$	91.3$_{\pm 0.6}$	68.5$_{\pm 1.1}$	94.6$_{\pm 0.4}$	86.6$_{\pm 1.6}$	92.1$_{\pm 0.1}$	88.3
	ProLoSA	0.023M	**96.5**$_{\pm 0.2}$	**91.7**$_{\pm 0.6}$	**70.3**$_{\pm 0.6}$	94.6$_{\pm 0.1}$	87.4$_{\pm 0.6}$	91.9$_{\pm 0.2}$	**88.7**

Table 6: **Matched Uni-LoRA–ProLoSA comparison on RoBERTa-large.** Both methods use the same fixed splits and five matched seeds. Δ denotes ProLoSA minus Uni-LoRA task score, with paired 95% t confidence intervals. CoLA uses Matthews correlation, STS-B uses Pearson correlation, and the remaining tasks use accuracy.

Task	Uni-LoRA / ProLoSA	Δ (95% CI)
SST-2	96.330 / 96.470	$+0.139$ $[-0.326,\ 0.604]$
MRPC	90.686 / 91.683	$+0.997$ $[0.021,\ 1.973]$
CoLA	68.885 / 70.274	$+1.389$ $[-1.257,\ 4.034]$
QNLI	94.578 / 94.572	-0.006 $[-0.242,\ 0.230]$
RTE	86.936 / 87.520	$+0.584$ $[-1.603,\ 2.771]$
STS-B	91.919 / 91.884	-0.035 $[-0.204,\ 0.134]$
Average	88.222 / 88.734	$+\mathbf{0.511}$ $[0.042,\ 0.981]$

evaluated once on both benchmarks for each training seed. Reported uncertainties are standard deviations across the three training seeds.

Table 7 reports the resulting three-seed benchmark performance. At the matched budget of 2.62M trainable parameters, Uni-LoRA obtains 76.12 ± 0.89 on GSM8K and 26.87 ± 0.12 on MATH500, whereas ProLoSA obtains 76.17 ± 1.03 and 27.33 ± 1.20, respectively. Their two-task averages are 51.45 and 51.75. Standard LoRA with rank 4 uses 10.49M trainable parameters and obtains a two-task average of 51.22.

Interpretation. The three-seed results show that ProLoSA remains competitive with Uni-LoRA at the same trainable-parameter budget, with nearly identical GSM8K accuracy and a higher mean MATH500 accuracy in this experiment. The observed differences are small relative to the across-seed variability, particularly on MATH500, and we therefore treat them as descriptive rather than evidence of statistically significant superiority. Moreover, because GSM8K and MATH500 are evaluated using the same MetaMathQA-trained update, these results cannot isolate benchmark-specific residual energy or directly test the task-specific recovery mechanism predicted by the theory.

D.3 GLUE Hyperparameters

Table 8 reports the main hyperparameters and computational costs for GLUE. These are the reported ProLoSA settings; published baseline protocols follow their source. For CoLA, STS-B, and the remaining tasks, we report Matthews correlation, Pearson correlation, and accuracy, respectively. The quoted baseline rows use their source's median-and-standard-deviation convention. Runtime and peak-memory measurements correspond to single-GPU ProLoSA runs.

Table 7: **Three-seed mathematical reasoning results.** Results are mean ± standard deviation over training seeds 42–44. Configurations are selected using held-out validation data only; GSM8K and MATH500 are used only for final evaluation. Both benchmarks evaluate the same MetaMathQA-trained update. Best results are shown in bold.

Method	Trainable params.	GSM8K	MATH500	Avg.
LoRA	10.49M	76.04 ± 0.55	26.40 ± 1.06	51.22
Uni-LoRA	2.62M	76.12 ± 0.89	26.87 ± 0.12	51.45
ProLoSA	2.62M	$\mathbf{76.17} \pm 1.03$	$\mathbf{27.33} \pm 1.20$	**51.75**

Backbone	Task	d	K	Epochs	BS	LR	Time	Peak Mem.	GPU
RoBERTa-base	SST-2	22320	720	60	32	5e-4	13.64 h	11,116 MiB	1
RoBERTa-base	MRPC	22320	720	30	32	1e-3	23.9 min	11,118 MiB	1
RoBERTa-base	CoLA	14400	8640	80	32	2e-2	2.42 h	11,116 MiB	1
RoBERTa-base	QNLI	22320	720	25	32	1e-3	8.97 h	11,116 MiB	1
RoBERTa-base	RTE	22320	720	160	32	1e-2	1.42 h	11,118 MiB	1
RoBERTa-base	STS-B	22320	720	80	32	2e-4	1.75 h	11,118 MiB	1
RoBERTa-large	SST-2	22320	720	20	32	1e-3	1.5 h	8,343 MiB	1
RoBERTa-large	MRPC	22320	720	40	32	2e-4	21.7 min	8,350 MiB	1
RoBERTa-large	CoLA	14400	8640	40	32	5e-3	49.3 min	8,372 MiB	1
RoBERTa-large	QNLI	22320	720	20	32	5e-4	2.6 h	8,329 MiB	1
RoBERTa-large	RTE	22320	720	40	32	5e-3	15.6 min	8,376 MiB	1
RoBERTa-large	STS-B	22320	720	40	32	2e-4	15.1 min	8,338 MiB	1
Optimizer	AdamW								
Scheduler	Linear								
Seeds	5								
ProLoSA warmup ratio	10%								
SNIP score-collection window	128 steps								

Table 8: Core hyperparameter and computational-cost settings for GLUE experiments. BS denotes batch size. The d and K columns denote the compressed latent dimension and sparse residual budget used by ProLoSA. Time denotes total wall-clock training time, Peak Mem. denotes peak GPU memory usage, and all reported runs use a single GPU. Dashes indicate computational-cost measurements that were not recorded.

D.4 Mathematical Reasoning Hyperparameters

Table 9 summarizes the method-specific configurations and computational costs for the mathematical reasoning experiments in Table 1. All methods use the same training data, target modules, precision, and decoding protocol described below. Exact-match scores use the same answer-extraction rule.

Llama-3.1-8B configuration. We fine-tune the Llama-3.1-8B base model (NousResearch mirror) on the first 100,000 MetaMathQA training examples for two epochs, using BF16 without weight quantization, a maximum training length of 512, batch size 1, gradient accumulation 64, cosine learning-rate decay, a 2% scheduler warmup, and zero weight decay. All methods target the attention and MLP linear projections. LoRA uses $r = 4$, $\alpha = 4$, and learning rate 2×10^{-4}; Uni-LoRA and ProLoSA also use $r = 4$.

Method-specific configurations are selected using held-out validation data only; GSM8K and MATH500 are not used for hyperparameter or checkpoint selection. After fixing one configuration per method, we train three seeds (42–44) and evaluate each resulting checkpoint on both GSM8K and MATH500. Evaluation uses greedy decoding (temperature 0, top-$p = 1$), batch size 32, and at most 1,024/2,048 generated tokens for GSM8K/MATH500, which contain 1,319 and 500 examples, respectively. Full MATH results are unavailable for this backbone.

Table 9 reports the selected method-specific learning rates, parameter budgets, support schedules, and computational costs. ProLoSA incurs a modest training-time overhead from support scoring and sparse activation, while its peak GPU-memory usage remains close to LoRA and Uni-LoRA.

Hyperparameter	LoRA	Uni-LoRA	ProLoSA
Rank r	4	4	4
Trainable parameters	10,485,760	2,621,440	2,621,440
Projected dimension d	–	2,621,440	2,330,169
Sparse budget K	–	–	291,271
Optimizer base learning rate	2×10^{-4}	2×10^{-3}	2×10^{-4}
Projected-vector learning rate	–	2×10^{-3}	1.6×10^{-3}
Sparse learning-rate multiplier	–	–	0.2
Score-free support warmup (steps)	–	–	512
SNIP scoring window (steps)	–	–	8
Sparse activation step	–	–	520
Projected initialization bound	–	0.02	0.02
Reset optimizer at activation	–	–	Yes
Sparse decay after activation	–	–	Yes
Training seeds	42, 43, 44	42, 43, 44	42, 43, 44
Training time (h)	10.18	12.04	13.28
Peak allocated GPU memory (GiB)	16.00	16.00	16.28
Peak process GPU memory (GiB)	16.80	16.75	16.93

Table 9: Llama-3.1-8B configurations and computational costs for mathematical reasoning. All methods train for two epochs with an effective batch size of 64 and use three seeds (42–44). Support warmup and SNIP windows count optimizer updates; the shared 2% learning-rate warmup is separate. Uni-LoRA and ProLoSA each use one quarter of the LoRA trainable-parameter budget, with ProLoSA using an approximately $8{:}1$ projected-to-sparse allocation. Training times are estimated in hours; GPU-memory statistics report peak allocated and peak process memory in GiB.

D.5 Instruction-Tuning Experiments

We evaluate LoRA, Uni-LoRA, and ProLoSA on Llama-3.1-8B instruction tuning under a common training and evaluation protocol. All methods start from the same frozen Llama-3.1-8B base model and are trained on the same 50,000 instruction-tuning examples using three training seeds, $0, 21, 42$. Table 10 summarizes the method-specific configurations and computational costs, and Table 11 reports the final IFEval and IFBench results.

Training setup. We fine-tune `NousResearch/Meta-Llama-3.1-8B` on 50,000 conversations from the `smol-magpie-ultra` subset of SmolTalk. The pretrained backbone, embeddings, and language-model head are frozen, and adapters are applied to the `q_proj`, `k_proj`, `v_proj`, `o_proj`, `gate_proj`, `up_proj`, and `down_proj` modules of all 32 Transformer layers. Conversations are serialized with plain role headers, and the causal language-modeling loss is computed only on assistant responses and their EOS tokens. Sequences longer than 2,048 tokens after serialization are excluded rather than truncated or packed.

All methods use one epoch, a micro-batch size of 1, gradient accumulation over 32 steps, and hence an effective batch size of 32, for 1,563 optimizer updates. Training uses AdamW with $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$, zero weight decay, cosine learning-rate decay, and a 3% scheduler warmup corresponding to 47 optimizer steps. Gradients are clipped to global norm 1.0 and adapter dropout is zero. The frozen backbone is stored in BF16, whereas trainable adapter and optimizer parameters are maintained in FP32 under BF16 autocast. We use SDPA attention, non-reentrant gradient checkpointing, and TF32. Every run uses a single NVIDIA L20 GPU.

All three methods use rank 4. Standard LoRA uses $\alpha = 16$ and a learning rate of 5×10^{-5}. Uni-LoRA uses a compressed-parameter learning rate of 3×10^{-3} and an initialization bound of 0.02. ProLoSA uses the same initialization bound, with learning rates 2.4×10^{-3} and 4.8×10^{-4} for the compressed and sparse parameters, respectively. Sparse support is activated after 128 warmup steps using SNIP scores summed over eight scoring steps, after which the optimizer state is reset.

Evaluation and results. We evaluate the final model from each training seed on IFEval and IFBench, reporting prompt-strict accuracy for IFEval and prompt-loose accuracy for IFBench. Neither

Table 10: **Llama-3.1-8B instruction-tuning configurations and computational cost.** All methods use rank 4, the same 50K training examples, seeds $\{0, 21, 42\}$, and one NVIDIA L20 GPU. Params. denotes the trainable/effective adapter parameter count under each method's convention. Comp. and Sparse denote the compressed and sparse parameter counts, respectively. Training time is reported as mean $\pm$ standard deviation over the three seeds.

Method	Params.	Budget	Comp.	Sparse	Time (h)	Peak Mem. (GiB)
LoRA	10.49M	100%	–	–	3.296 ± 0.018	18.644
Uni-LoRA	2.62M	25%	2,621,440	–	4.772 ± 0.128	18.645
ProLoSA	2.62M	25%	2,419,791	201,649	4.899 ± 0.018	18.929

Table 11: **Instruction-tuning results on Llama-3.1-8B.** Results are mean $\pm$ standard deviation over seeds $\{0, 21, 42\}$. IFEval reports prompt-strict accuracy and IFBench reports prompt-loose accuracy. Uni-LoRA and ProLoSA use 25% of the effective adapter-parameter budget of standard LoRA. Neither benchmark is used for model or hyperparameter selection.

Method	Params.	IFEval	IFBench
LoRA	10.49M	46.64 ± 2.17	16.89 ± 1.07
Uni-LoRA	2.62M	46.64 ± 4.22	15.78 ± 0.96
ProLoSA	2.62M	46.89 ± 2.69	18.11 ± 1.17

benchmark is used for hyperparameter selection or checkpoint selection. Table 11 reports the mean and standard deviation across the three training seeds.

At one quarter of LoRA's effective adapter-parameter budget, Uni-LoRA attains a similar mean IFEval score but a lower mean IFBench score. Under the same 2.62M budget, ProLoSA obtains a similar mean IFEval score to both baselines and a higher mean IFBench score in this comparison. Because these results are based on only three training seeds and exhibit overlapping variability, we treat the differences as descriptive empirical results rather than evidence of statistical significance.

D.6 Ablation Details

Table 12 reports the full budget-allocation sweep between the projected branch and the sparse residual branch. All variants use the same total trainable-parameter budget $B = d + K = 23040$. The compressed-only variant has $K = 0$. At the opposite allocation extreme, the sparse-dominant hybrid retains $d = 1$ shared latent parameter and uses $K = 23039$ residual coordinates; it is not a strictly sparse-only baseline. Intermediate configurations retain both branches. Since B is fixed, increasing the sparse budget K necessarily reduces the compressed latent dimension d.

For the sparse-dominant configuration, the nonzero scalar initialization in Appendix B.2 supplies the warmup factors. Its saved settings specify 128 score-free optimizer steps, one scoring step, and optimizer reset at residual activation. The latent learning rate is 5×10^{-3} and the sparse rate is 10^{-3}. The support is selected once and fixed; zero initialization of the residual preserves the current prediction. These settings describe this confirmed boundary configuration and do not determine the schedules of other reported rows.

D.7 Vision Transformer Experiments

Table 13 reports the published vision-transfer results. Published LP, full fine-tuning, FourierFT, and Uni-LoRA results are quoted from Uni-LoRA (Li et al., 2025); ProLoSA results use five seeds (42–46). Because the published baselines use unrecovered original splits, Table 13 is a cross-study comparison. We therefore additionally rerun Uni-LoRA on the same reconstructed splits and matched seeds as ProLoSA in Table 15.

ViT configuration and sweep. We fine-tune ImageNet-21k-pretrained ViT-B/16 and ViT-L/16, using rank-4 adapters on the query and value projections of all Transformer blocks. The ProLoSA budgets are $d + K = 72{,}000$ for ViT-Base and $144{,}000$ for ViT-Large. CIFAR10, CIFAR100, OxfordPets, and StanfordCars use 10% of the training set for validation; EuroSAT, RESISC45, and

Study	Variant	d	K	CoLA	MRPC
Budget	Uni-LoRA	23040	0	$68.89_{\pm 0.58}$	$90.93_{\pm 0.30}$
	ProLoSA	22680	360	$68.29_{\pm 1.40}$	$90.11_{\pm 0.50}$
	ProLoSA	22320	720	$69.62_{\pm 1.84}$	$\mathbf{91.69}_{\pm 0.40}$
	ProLoSA	21600	1440	$67.21_{\pm 0.84}$	$90.87_{\pm 0.46}$
	ProLoSA	20160	2880	$68.72_{\pm 0.45}$	$83.74_{\pm 9.65}$
	ProLoSA	18720	4320	$67.51_{\pm 0.51}$	$90.03_{\pm 0.42}$
	ProLoSA	14400	8640	$\mathbf{70.27}_{\pm 0.60}$	$90.77_{\pm 0.42}$
	ProLoSA	5760	17280	$68.75_{\pm 1.27}$	$89.71_{\pm 0.20}$
	Sparse-dominant	1	23039	$59.99_{\pm 1.40}$	$84.89_{\pm 0.83}$
Support	Random	22320	720	$66.58_{\pm 0.63}$	$90.28_{\pm 0.61}$
	Accum. gradient	22320	720	$67.95_{\pm 0.59}$	$90.03_{\pm 0.31}$
	Warmup SNIP	22320	720	$\mathbf{69.62}_{\pm 1.84}$	$\mathbf{91.69}_{\pm 0.40}$
Warmup	0%	22320	720	$67.32_{\pm 0.47}$	$90.36_{\pm 0.20}$
	5%	22320	720	$69.05_{\pm 0.51}$	$91.12_{\pm 0.70}$
	10%	22320	720	$\mathbf{69.62}_{\pm 1.84}$	$\mathbf{91.69}_{\pm 0.40}$
	20%	22320	720	$67.49_{\pm 0.96}$	$90.60_{\pm 0.81}$

Table 12: Ablation details on budget allocation, sparse support selection, and warmup length. CoLA reports Matthews correlation and MRPC reports accuracy. The sparse-dominant row uses $(d, K) = (1, 23039)$; it retains a shared scalar for initialization and support selection.

FGVC Aircraft use their provided splits; DTD uses a fixed 4,060/450/1,130 train/validation/test split. Training uses random resized crops and horizontal flips at 224×224, with deterministic resize and center crop for validation and testing. All runs use AdamW for 20 epochs, effective batch size 512, weight decay 0.01, linear learning-rate decay, and zero adapter dropout. Hyperparameters are selected by validation accuracy from projected learning rates $\eta_z \in \{0.002, 0.01, 0.05\}$, head learning rates $\eta_h \in \{0.002, 0.01\}$, and allocations $d : K \in \{8 : 1, 12 : 1\}$. The sparse learning rate is $0.2\eta_z$; SNIP support is selected once after the first 10% of optimization steps. Selected configurations are given in Table 16. Wall-clock training time and peak allocated GPU memory are reported separately in Table 14.

Table 13: **Image classification with ViT-Base and ViT-Large.** Accuracy (%, mean ± standard deviation); Avg. is the unweighted mean over eight tasks. Published baselines are quoted from Uni-LoRA (Li et al., 2025); ProLoSA uses five seeds (42–46) on reconstructed fixed splits.

Model	Method	Params.	OxfordPets	StanfordCars	CIFAR10	DTD	EuroSAT	FGVC	RESISC45	CIFAR100	Avg.
ViT-Base	LP	–	$90.28_{\pm 0.43}$	$25.76_{\pm 0.28}$	$96.41_{\pm 0.02}$	$69.77_{\pm 0.67}$	$88.72_{\pm 0.13}$	$17.44_{\pm 0.43}$	$74.22_{\pm 0.10}$	$84.28_{\pm 0.11}$	68.36
	FF	85.8M	$93.14_{\pm 0.40}$	$79.78_{\pm 1.15}$	$98.92_{\pm 0.05}$	$77.68_{\pm 1.21}$	$99.05_{\pm 0.09}$	$54.84_{\pm 1.23}$	$96.13_{\pm 0.13}$	$92.38_{\pm 0.13}$	86.49
	FourierFT	72K	$93.21_{\pm 0.26}$	$46.11_{\pm 0.24}$	$98.58_{\pm 0.07}$	$75.09_{\pm 0.37}$	$98.29_{\pm 0.04}$	$27.51_{\pm 0.64}$	$91.97_{\pm 0.31}$	$91.20_{\pm 0.14}$	77.75
	FourierFT	239K	$93.05_{\pm 0.34}$	$56.36_{\pm 0.66}$	$98.69_{\pm 0.06}$	$77.30_{\pm 0.61}$	$98.78_{\pm 0.11}$	$32.44_{\pm 0.99}$	$94.26_{\pm 0.20}$	$91.45_{\pm 0.18}$	80.29
	Uni-LoRA	72K	$94.00_{\pm 0.13}$	$76.06_{\pm 0.23}$	$98.77_{\pm 0.03}$	$76.99_{\pm 0.96}$	$98.86_{\pm 0.10}$	$50.36_{\pm 0.63}$	$94.08_{\pm 0.19}$	$92.10_{\pm 0.25}$	85.15
	ProLoSA	72K	$94.03_{\pm 0.11}$	$78.35_{\pm 0.26}$	$98.75_{\pm 0.10}$	$77.96_{\pm 0.35}$	$98.98_{\pm 0.11}$	$56.79_{\pm 0.75}$	$94.32_{\pm 0.36}$	$91.90_{\pm 0.10}$	**86.38**
ViT-Large	LP	–	$91.11_{\pm 0.30}$	$37.91_{\pm 0.27}$	$97.78_{\pm 0.04}$	$73.33_{\pm 0.26}$	$92.64_{\pm 0.08}$	$24.62_{\pm 0.24}$	$82.02_{\pm 0.11}$	$84.28_{\pm 0.11}$	72.96
	FF	303.3M	$94.43_{\pm 0.56}$	$88.90_{\pm 0.26}$	$99.15_{\pm 0.04}$	$81.79_{\pm 1.01}$	$99.04_{\pm 0.08}$	$68.25_{\pm 1.63}$	$96.43_{\pm 0.07}$	$93.58_{\pm 0.19}$	90.20
	FourierFT	144K	$94.46_{\pm 0.28}$	$69.56_{\pm 0.30}$	$99.10_{\pm 0.04}$	$80.83_{\pm 0.43}$	$98.65_{\pm 0.09}$	$39.92_{\pm 0.68}$	$93.86_{\pm 0.14}$	$93.31_{\pm 0.09}$	83.71
	FourierFT	480K	$94.84_{\pm 0.05}$	$79.14_{\pm 0.67}$	$99.08_{\pm 0.05}$	$81.88_{\pm 0.50}$	$98.66_{\pm 0.03}$	$51.28_{\pm 0.66}$	$95.20_{\pm 0.07}$	$93.37_{\pm 0.11}$	86.68
	Uni-LoRA	144K	$94.65_{\pm 0.23}$	$83.16_{\pm 0.62}$	$98.77_{\pm 0.03}$	$81.35_{\pm 0.48}$	$98.89_{\pm 0.07}$	$58.89_{\pm 0.62}$	$95.24_{\pm 0.12}$	$93.08_{\pm 0.11}$	88.00
	ProLoSA	144K	$94.79_{\pm 0.14}$	$83.13_{\pm 0.39}$	$99.05_{\pm 0.05}$	$80.76_{\pm 0.28}$	$98.84_{\pm 0.13}$	$59.82_{\pm 0.40}$	$95.29_{\pm 0.16}$	$93.38_{\pm 0.16}$	**88.13**

Training efficiency. Table 14 reports measured wall-clock training time and peak allocated GPU memory for LoRA, Uni-LoRA, and ProLoSA under the same training setup. ProLoSA incurs only a small runtime and memory overhead relative to Uni-LoRA. Averaged across the eight tasks, ProLoSA takes 9.43 versus 9.33 minutes for Uni-LoRA on ViT-Base, and 28.16 versus 27.80 minutes on ViT-Large. Peak allocated memory remains essentially unchanged, at about 4.03 GiB for ViT-Base and 6.22 GiB for ViT-Large.

Matched Uni-LoRA reruns. To control for split and seed differences, we rerun Uni-LoRA using the same fixed splits and five matched seeds as ProLoSA. Table 15 reports ProLoSA minus Uni-LoRA paired differences with 95% paired t confidence intervals. Averaged over the eight tasks,

Table 14: **Training efficiency on the vision benchmarks.** Each entry reports wall-clock training time in minutes / peak allocated GPU memory in GiB. All values are measured under the same training setup.

Model	Dataset	LoRA	Uni-LoRA	ProLoSA
ViT-Base	OxfordPets	1.70 / 4.02	1.80 / 4.02	1.81 / 4.03
	StanfordCars	3.70 / 4.02	4.00 / 4.02	4.03 / 4.03
	CIFAR10	20.30 / 4.02	21.60 / 4.02	21.87 / 4.03
	DTD	2.10 / 4.02	2.30 / 4.02	2.30 / 4.03
	EuroSAT	8.00 / 4.02	8.50 / 4.02	8.61 / 4.03
	FGVC	4.30 / 4.02	4.60 / 4.02	4.67 / 4.03
	RESISC45	9.60 / 4.02	10.20 / 4.02	10.28 / 4.03
	CIFAR100	20.30 / 4.02	21.60 / 4.03	21.86 / 4.03
ViT-Large	OxfordPets	5.00 / 6.20	5.30 / 6.21	5.36 / 6.22
	StanfordCars	11.70 / 6.20	12.30 / 6.21	12.44 / 6.22
	CIFAR10	62.60 / 6.20	65.70 / 6.21	66.59 / 6.22
	DTD	6.10 / 6.20	6.40 / 6.21	6.51 / 6.22
	EuroSAT	25.00 / 6.20	26.20 / 6.21	26.53 / 6.22
	FGVC	7.80 / 6.20	8.20 / 6.21	8.24 / 6.22
	RESISC45	29.60 / 6.20	31.10 / 6.21	31.49 / 6.22
	CIFAR100	64.00 / 6.20	67.20 / 6.21	68.09 / 6.22

Table 15: **Matched Uni-LoRA–ProLoSA comparison.** Both methods use the same fixed splits and five matched seeds. Δ denotes ProLoSA minus Uni-LoRA accuracy, with paired 95% t confidence intervals.

	ViT-Base		ViT-Large	
Dataset	Uni / ProLoSA	Δ (95% CI)	Uni / ProLoSA	Δ (95% CI)
OxfordPets	93.992 / 94.026	$+0.034$ $[-0.098, 0.165]$	94.636 / 94.794	$+0.158$ $[-0.003, 0.320]$
StanfordCars	75.874 / 78.346	$+2.472$ $[2.160, 2.784]$	82.664 / 83.134	$+0.470$ $[-0.082, 1.022]$
CIFAR10	98.794 / 98.754	-0.040 $[-0.159, 0.079]$	98.794 / 99.050	$+0.256$ $[0.196, 0.316]$
DTD	77.596 / 77.965	$+0.369$ $[-0.153, 0.890]$	80.212 / 80.761	$+0.549$ $[-0.045, 1.142]$
EuroSAT	98.824 / 98.978	$+0.154$ $[-0.082, 0.389]$	98.933 / 98.844	-0.089 $[-0.275, 0.097]$
FGVC	50.200 / 56.790	$+6.590$ $[4.944, 8.235]$	58.734 / 59.820	$+1.086$ $[0.322, 1.850]$
RESISC45	94.198 / 94.321	$+0.123$ $[-0.316, 0.561]$	95.222 / 95.292	$+0.070$ $[-0.145, 0.285]$
CIFAR100	92.046 / 91.900	-0.146 $[-0.387, 0.095]$	93.058 / 93.376	$+0.318$ $[0.089, 0.547]$
Average	85.191 / 86.385	$+\mathbf{1.194}$ $[0.899, 1.489]$	87.782 / 88.134	$+\mathbf{0.352}$ $[0.193, 0.511]$

the paired differences are $+1.194$ points for ViT-Base (95% CI $[0.899, 1.489]$) and $+0.352$ for ViT-Large (95% CI $[0.193, 0.511]$).

D.8 RoSA Reference Runs

We additionally evaluate RoSA (Nikdan et al., 2024) on RoBERTa-large for CoLA and MRPC. These experiments provide an independent-budget reference to a related sparse–low-rank adaptation method and are not intended as matched-budget tests of Proposition 2.

The evaluated RoSA configuration uses rank $r = 2$, with 884,736 trainable low-rank-factor parameters and 884,736 trainable sparse weight-space parameters, for 1,769,472 adapter parameters in total. Classification-head parameters are excluded from this count. By comparison, the ProLoSA experiments considered here use 23,040 adapter parameters. RoSA therefore operates at a substantially larger adapter budget. The methods also use different sparse-update spaces and support-selection procedures: RoSA selects sparse weight-space coordinates using accumulated squared gradients, whereas ProLoSA selects residual factor-space coordinates using warmup SNIP sensitivity.

Configuration and checkpoint selection use a split constructed from the training data, while the official development set is reserved for final evaluation. The study contains 48 tuning runs and 10 final runs, and predictions from all ten final runs were recomputed before reporting the results.

Table 16: **Selected ProLoSA configurations.** η_z and η_h denote projected-vector and classification-head learning rates. The sparse learning rate is $0.2\eta_z$.

	ViT-Base: $d + K = 72{,}000$			ViT-Large: $d + K = 144{,}000$		
Dataset	η_z	η_h	$d : K$	η_z	η_h	$d : K$
OxfordPets	0.01	0.01	$8:1$	0.01	0.002	$8:1$
StanfordCars	0.01	0.01	$12:1$	0.01	0.01	$12:1$
CIFAR10	0.01	0.01	$8:1$	0.01	0.01	$8:1$
DTD	0.01	0.01	$12:1$	0.01	0.01	$8:1$
EuroSAT	0.01	0.01	$8:1$	0.01	0.01	$8:1$
FGVC	0.05	0.01	$8:1$	0.01	0.01	$8:1$
RESISC45	0.01	0.01	$12:1$	0.01	0.01	$12:1$
CIFAR100	0.01	0.01	$12:1$	0.01	0.002	$8:1$

Table 17: **RoSA reference runs on two GLUE tasks.** Results are mean $\pm$ sample standard deviation over five seeds, with pointwise 95% t intervals. These runs use an independent parameter budget and tuning protocol and should not be interpreted as paired or matched-budget comparisons with ProLoSA.

Task	Score	95% t interval
CoLA	64.31 ± 3.77	$[59.63, 68.99]$
MRPC	89.26 ± 0.72	$[88.37, 90.16]$

These runs do not constitute paired comparisons with ProLoSA, even when the same seed identifiers are used. The experiment batches differ in parameter budget, initialization, regularization, configuration selection, sparse-update space, and support construction. Their score differences therefore cannot be attributed solely to the adaptation parameterization.

A native equal-budget RoSA comparison is also unavailable under the current target-layer coverage. Even rank-1 standard LoRA factors require 442,368 trainable parameters, already exceeding the 23,040-parameter ProLoSA budget before adding any sparse weight-space parameters. Matching the ProLoSA budget would therefore require changing either the target-layer coverage or the RoSA parameterization itself, introducing an additional comparison factor. We consequently report these results only as an independent-budget reference.

D.9 Recovery-oriented selector diagnostics

We evaluate the recovery-oriented selectors derived in Appendix A.5 as practical alternatives to the SNIP support rule used by ProLoSA. Specifically, we instantiate Recovery-Fisher (RF) using Eq. (40) and Residualized Gradient (RG) using Eq. (42). These selectors are treated as theory-oriented diagnostics rather than as consequences of the theory: the recovery analysis motivates the desired residual-recovery objective, but does not imply that either approximation, or SNIP, is an optimal finite-sample selector.

For both RG and RF, scores are accumulated from minibatch gradients and ranked globally across layers and LoRA A/B coordinates using the Uni-LoRA sharing groups. RG and RF configurations are selected by the three-seed mean validation score over the evaluated K values.

For CoLA, ProLoSA uses $(d, K) = (14{,}400, 8{,}640)$, while the best-performing RG and RF configurations use $(20{,}160, 2{,}880)$ and $(18{,}720, 4{,}320)$, respectively. For MRPC, ProLoSA uses $(d, K) = (22{,}320, 720)$, while the best-performing RG and RF configurations use $(20{,}160, 2{,}880)$ and $(22{,}320, 720)$, respectively. These configurations maintain total adapter budgets of 23,040 for both tasks, excluding classification heads. ProLoSA follows the standard task-specific SNIP configuration reported in Appendix D.3. RG and RF use 128 score-free warmup steps followed by 32 scoring steps. Thus, the comparison maintains the same total adapter budget but differs in budget allocation and scoring window.

Neither recovery-oriented approximation improves upon the SNIP-based ProLoSA mean in these experiments. Relative to ProLoSA, RG is lower by 1.56 CoLA MCC points and 1.58 MRPC accu-

Table 18: **Recovery-oriented selector diagnostics on RoBERTa-large.** Results report mean ± sample standard deviation over 5 seeds using validation-selected checkpoints. ProLoSA uses the standard SNIP configuration, whereas RG and RF replace the support selector while retaining the compressed-plus-sparse parameterization. Because their scoring windows are not identical, the comparison should be interpreted as a diagnostic of practical selector behavior rather than as a strictly controlled selector-only ablation.

Method	CoLA MCC ↑	MRPC Acc. ↑
Uni-LoRA	68.89 ± 0.58	90.93 ± 0.30
ProLoSA	$\mathbf{70.27 \pm 0.60}$	$\mathbf{91.69 \pm 0.40}$
RG	68.72 ± 1.09	90.11 ± 0.37
RF	68.43 ± 1.03	89.54 ± 0.62

racy points, while RF is lower by 1.84 and 2.15 points, respectively; RF also does not improve over RG. These results separate the theoretical recovery criterion from its practical estimation: a selector designed to approximate the local recovery objective more directly need not perform better under finite-sample gradient estimation and subsequent nonlinear optimization. Possible sources include diagonalization error, noisy curvature estimation, departure from the local constrained optimum, and interactions with the later training trajectory. The theory therefore motivates recovery-aware support selection, while SNIP remains an empirical design choice in ProLoSA supported by the downstream ablations.

E Additional Mechanism Controls

This appendix provides additional controls for the mechanisms studied in Sections A.2–A.5. The experiments address five questions left open by the main text: whether residual recovery persists in a nonlinear Transformer; how data-dependent support selection affects retained noise and realized gain; whether the concentrated-versus-diffuse comparison persists after matching residual magnitude; whether the local quadratic and linearized approximations remain informative over the scale of the studied updates; and whether the matched-budget conclusions persist across independently sampled tasks rather than a small number of fixed targets. Together, these controls test the robustness and empirical scope of the proposed mechanism rather than establishing additional claims of universal superiority.

E.1 Recoverability and Selector Realizability in a Fixed-Teacher Transformer

From recoverable geometry to realized prediction gain. We construct two fixed teachers on the same frozen Transformer, with concentrated and diffuse update directions calibrated to 10% of the flattened base-output variance using calibration inputs disjoint from evaluation. We vary $n \in \{64, 256, 1024, 4096\}$ and the observation-noise standard deviation in $\{0.25, 0.5, 1\}$, with 20 paired training seeds per condition. The adapted query/value parameterization contains $D = 8192$ coordinates. Budget-128 compression is compared with hybrids having $d = K = 64$ and verified joint rank 128.

For a selected support I, we report the exact Euclidean recovery

$$r_I = \frac{\mathcal{E}_I}{\|q\|^2},$$

the realized matched-budget margin

$$m_I = \mathcal{E}_I - C_{\text{task}},$$

and the paired predictive gain

$$g_I = \text{MSE}_{U,B} - \text{MSE}_{\text{hyb}}.$$

The first two quantities are projection-space diagnostics, whereas g_I measures nonlinear predictive performance; they need not coincide.

Each hybrid uses a 256-step pilot, accumulates its support score over the final 128 pilot steps, and is then refit from a fresh initialization for 744 steps. The compressed baseline receives 1000 fitting steps. Random and coordinate-energy supports are charged the same pilot cost. This controlled protocol isolates support quality and is not the end-to-end ProLoSA training schedule. The coordinate-energy selector ranks coordinates by q_i^2 and serves only as a surrogate for joint recovery; it is not a globally optimal selector.

Across the 12 training/noise conditions for each teacher, SNIP's pointwise paired 95% confidence intervals for g_I are positive, negative, or overlap zero in $4/7/1$ conditions for the concentrated teacher and $3/8/1$ conditions for the diffuse teacher, respectively. The coordinate-energy surrogate improves over budget-matched compression in all 12 conditions for both teachers, although it does not uniformly outperform the other informed selectors. Thus geometric recoverability can persist in a nonlinear model while the practical benefit of a particular selector remains condition dependent.

Table 19: Fixed-teacher recovery and paired prediction gain at $n = 4096$ and noise standard deviation 0.25 (20 paired seeds). r_I and m_I are means; g_I is reported with a pointwise paired 95% t interval. Positive g_I favors the hybrid.

Teacher	Selector	r_I	m_I	$10^3 g_I$ (95% CI)
Concentrated	Random	0.0027	-0.3655	-0.077 $[-0.095, -0.058]$
	Gradient	0.5685	26.23	2.051 $[2.032, 2.069]$
	SNIP	0.4636	21.30	1.733 $[1.670, 1.796]$
	Energy surrogate	0.9991	46.46	2.265 $[2.256, 2.274]$
Diffuse	Random	0.0079	0.0195	-0.015 $[-0.029, -0.00015]$
	Gradient	0.0142	0.2613	0.495 $[0.463, 0.527]$
	SNIP	0.0093	0.0709	0.281 $[0.245, 0.316]$
	Energy surrogate	0.0689	2.375	0.199 $[0.191, 0.207]$

Independent support selection and data cost. We next separate the data used for support selection from the data used for fitting. Selecting on one half of the data and freshly fitting on the other transfers neither pilot parameters nor optimizer state. Relative to selecting and fitting on the fitting half alone, this independent protocol has positive paired gain intervals in $11/12$ conditions for each teacher, with one interval overlapping zero. However, this comparison uses twice as many unique examples. When total unique data are instead held fixed, independent selection uses half of the examples for fitting and yields $4/7/1$ positive/negative/overlapping intervals for each teacher. Therefore, the apparent benefit of independent selection depends strongly on whether fitting-data size or total data cost is held fixed.

E.2 Adaptive Selection and Retained Noise

The matched-budget identity in Eq. (29) separates recovered task signal from the change in retained noise under an observation-dependent support. We test these two terms directly in the sequence model.

We use $D = 4096$, $d = 16$, $K = 32$, $B = 48$, and $n_{\text{eff}} = 512$, using the original target, Gaussian-QR projections P, P_B, and paired noise streams. We additionally repeat the construction with two further independently generated target/projection pairs. For each task we use the same 20 noise levels and 200 paired noise draws. Every realized augmented rank is verified.

For each realized support, direct fitted risk agrees numerically with the sum of unrecovered signal and retained projected noise, and the paired gain agrees with recovered signal minus the corresponding noise-energy difference. This verifies the sequence-model decomposition itself; it does not identify the full adaptive estimator's squared bias and variance.

The target-energy selector recovers similar geometric signal to the noisy-residual selector but incurs almost no additional retained-noise cost. Using the same data for selection and fitting produces a substantially larger noise penalty. Independent selection nearly removes this penalty when fitting size is held fixed, but this comparison uses more unique data. Under an equal-total-data split, fitting noise increases because fewer observations are available to the final estimator. These comparisons

Table 20: Adaptive-support decomposition for the original sequence task at $\sigma_y = 1.5$. Geometric gain is $\frac{1}{2}(\mathbb{E}\mathcal{E}_{\hat{f}} - C_{\text{task}})$. $\widehat{N}_{\text{adapt}}$ denotes retained noise relative to the theoretical budget-48 baseline. Units are one-half of the squared parameter error.

Selector	Geometric gain	$\widehat{N}_{\text{adapt}}$	Paired gain
Fixed random	-0.05401	0.00078	-0.05716
Fresh random	-0.00739	0.00371	-0.01348
Target-energy surrogate	5.29631	0.00276	5.29118
Noisy residual	5.22526	0.23674	4.98614
Independent, equal fit size	5.22430	0.00139	5.22054
Independent, equal total data	5.09778	0.10685	4.98857
Same-half selection/fit	5.09634	0.68619	4.40778

illustrate why adaptive support selection cannot be assessed from recovered signal alone. Thus selector quality has two distinct components: identifying a high-recovery support and avoiding excessive coupling between support estimation and fitting noise. The former concerns geometry; the latter concerns statistical realizability.

E.3 Matched-Residual-Energy Sequence Controls

The concentrated and diffuse targets in the main sequence experiment differ in both residual structure and overall magnitude. To separate these effects, we retain the two target directions and the same projections P, P_B, but rescale each direction so that the realized residual energy per excluded dimension,

$$\frac{\|(I - P_B P_B^\top)\theta^\star\|^2}{D - B},$$

matches either the original concentrated scale (0.0026427) or the original diffuse scale (0.015685). We evaluate $\sigma_y \in \{0.5, 1.5\}$ with 200 paired repetitions and random, noisy-residual, and coordinate-energy supports. This is a fixed-instance normalization experiment rather than an ensemble statement.

Table 21: Matched-residual-energy sequence control at $\sigma_y = 1.5$. Entries are paired mean gains over budget-48 compression. Within each residual scale, the concentrated and diffuse directions have the same realized residual energy per excluded dimension.

Profile	Residual scale	Random	Noisy residual	Energy surrogate
Concentrated	0.002643	0.00515	4.98985	5.29862
Diffuse	0.002643	-0.01268	-0.23988	0.30092
Concentrated	0.015685	-0.03287	31.25278	31.43705
Diffuse	0.015685	-0.06711	1.22996	1.79846

Matching residual magnitude preserves a large difference in sparse recoverability between the two tested directions. In particular, at the lower residual scale the energy surrogate yields a gain of 5.30 for the concentrated direction but only 0.30 for the diffuse direction. The control therefore removes the magnitude confound present in the original comparison, while remaining a fixed-instance result rather than evidence for a universal concentration law.

E.4 Amplitude and Local-Linearity Diagnostic

To probe the scope of the local approximation used in the theory, we rescale the same fixed Transformer teacher directions from 10% to 100% of the frozen model's base-output variance. Calibration and held-out evaluation use disjoint inputs. We report the relative first-order linearization error

$$\frac{\mathbb{E}\|f(W_0 + \delta) - f(W_0) - J_0\delta\|^2}{\mathbb{E}\|f(W_0 + \delta) - f(W_0)\|^2}.$$

The larger-displacement teachers depart more strongly from the local linear approximation. Accordingly, changes in hybrid performance at larger amplitudes should not be interpreted as direct

Table 22: Transformer amplitude diagnostic. Increasing target displacement increases the measured linearization error for both teacher directions.

Profile	Target displacement	Relative linearization error
Concentrated	10%	0.01452
Concentrated	100%	0.21112
Diffuse	10%	0.00503
Diffuse	100%	0.07630

tests of the exact projection identities. This control motivates the conditional interpretation of the Transformer results in the main text.

E.5 Independent Tasks and Allocation Endpoints

The original sequence figures condition on a small number of fixed targets. We therefore repeat the matched-budget comparison over 30 independently generated target/projection pairs, with 200 noise draws per task, total budget $B = 48$, $\sigma_y = 1$, and

$$K \in \{0, 8, 16, 24, 32, 40, 48\}, \qquad d = B - K.$$

Nested Gaussian-QR spaces $P_d \subset P_B$ ensure a nonnegative realized opportunity cost. Both the compression-only endpoint ($K = 0$) and the sparse-only endpoint ($d = 0$) are included.

We compare random support with noisy-residual, gradient, SNIP, and target-energy-surrogate supports. In this quadratic control, the gradient and noisy-residual scores induce the same ranking and are therefore reported jointly. The sequence SNIP score instead accumulates $|\theta_i g_i|$ over steps 128–255 of a 256-step quadratic pilot initialized at $P_d P_d^\top y$. Although SNIP closely tracks the noisy-residual/gradient selector in this setting, it is not identical and is therefore reported separately.

Table 23: Mean paired gain across 30 independently generated tasks from the original target law ($\pi = 0.01$). Noise is averaged within task before averaging across tasks. Gradient and noisy-residual scores induce the same ranking in this quadratic control and are reported jointly; SNIP is evaluated separately.

K	d	Random	Noisy residual / gradient	SNIP	Energy surrogate
0	48	-0.0000	-0.0000	-0.0000	-0.0000
8	40	0.0004	4.0871	4.0872	4.1001
16	32	-0.0037	4.7561	4.7562	4.7731
24	24	0.0009	5.0047	5.0055	5.0381
32	16	-0.0046	5.0535	5.0541	5.1395
40	8	-0.0054	5.0220	5.0222	5.1678
48	0	0.0073	4.9769	4.9769	5.1756

Random supports remain close to the zero-gain matched-budget reference after averaging across independent tasks, whereas target-informed supports recover substantial excluded signal over a broad range of allocations. The noisy-residual/gradient and SNIP selectors yield very similar gains in this quadratic setting, while the energy surrogate provides stronger target-informed reference. For example, at $K = 32$ the mean realized opportunity cost is 0.08393, compared with the ensemble quantity $Kv = 0.080792$, and random support has mean gain -0.00462. The difference between the realized and ensemble costs reflects finite-task variation rather than an imposed equality. Because the heavy-tailed target law does not have a finite fourth moment, we report these cross-task means descriptively rather than attach Gaussian standard-error claims.