When Does Subspace Direction Matter for LoRA? Regime Analysis of the Magnitude Principle in Few-Shot Adaptation

Nischal Subedi
Univ. of Delaware
nsubedi@udel.edu

Cencheng Shen
Microsoft Research
c.shen@microsoft.com

Peng Zhao
Univ. of Delaware
pzhao@udel.edu

Abstract

We show that the importance of a LoRA adapter's initial input-subspace direction is regime-dependent, and we characterize this dependence in theory and experiment. This refines the recent view that spectral initializers help through update magnitude alone. Direction matters most when labeled data are scarce relative to the representation dimension and the activation subspace used for initialization captures supervised task structure. In this regime, the adapter's row span remains sensitive to its starting point, making task-aligned initialization especially useful. A spiked-covariance perturbation analysis characterizes the finite-sample stability of the estimated subspace. A first-step analysis of LoRA training explains why the subspace's alignment with the task matters and motivates an alignment coefficient ρ that closely tracks gains across classification tasks. We test this account with PivotLoRA, which initializes adapters from activation PCA at prediction-relevant positions and applies one output-preserving orthogonal pivot early in training. In few-shot BERT-base classification, PivotLoRA achieves the best average rank among 13 methods and improves over vanilla LoRA in 29 of 30 settings, even though it starts from a smaller input-factor initialization norm. Additional classification experiments extend the evaluation to 7B and 8B Llama decoders. Classification gains diminish with more labeled data, consistent with the proposed regime dependence. A targeted extension to completion-only reasoning also shows improvements on GSM8K and StrategyQA. These findings identify conditions under which initialization direction matters, with subspace–task alignment offering a low-cost diagnostic before training for classification.

1 Introduction

Low-Rank Adaptation [Hu et al., 2022] efficiently adapts pretrained models by parameterizing weight updates with low-rank factors. The input factor determines the feature subspace through which adaptation initially proceeds. Existing initialization methods derive directions for this subspace from singular components of pretrained weights [Meng et al., 2024, Wang et al., 2025], task gradients [Wang et al., 2024, Zhang et al., 2025a], or activation statistics [Paischer et al., 2025, Yang et al., 2024]. Recent work attributes gains from pretrained-weight spectral initialization primarily to amplified update magnitudes instead of the selected singular directions themselves [Zhang et al., 2025b]. This leaves open when initialization direction materially affects adaptation. We study this question primarily in classification, with a targeted extension to generation, by applying PCA to frozen-model activations from the adaptation set at positions associated with the supervised prediction. These include the `[CLS]` position in encoder classification, the mean over input positions in decoder classification, where the label describes the whole sequence, and hidden states that predict supervised response tokens in completion-only generation.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

Our central claim is that the value of initialization direction is regime-dependent. Direction matters most when labeled data are scarce relative to the representation dimension and the estimated subspace captures supervised task structure. A spiked-covariance Davis–Kahan analysis shows what limits such an estimate, since its error depends on the effective dimension-to-sample ratio and the spectral gap. The same qualitative pattern appears in synthetic experiments and transformer activations, where the ambient ratio d/N provides a simple proxy for the dimensional regime. For BERT-base, $d = 768$ and $N = 16$ gives $d/N = 48$; for 7B and 8B decoders with $d = 4096$, the high-dimensional regime persists at $N = 64$. A large d/N is not sufficient, however, because the selected subspace must also contain structure relevant to the supervised objective.

The role of alignment also appears in the first useful LoRA update. Standard LoRA initializes the output factor with $B = 0$, so initially $\Delta W = 0$ and the gradient with respect to the input factor A is zero. The first nonzero adapter signal enters through the projected features $Ah(x)$ in the update to B. Under a class-conditional model, the squared norm of the initial population gradient depends on the supervised structure captured by the row span of A. We quantify this structure with an alignment coefficient ρ, defined as the fraction of between-class scatter captured by a candidate subspace. For BERT-base with rank $r = 8$, a random subspace captures approximately $r/d \approx 1\%$ of this signal in expectation, whereas prediction-position PCA captures up to 84% on high-alignment tasks. Across seven classification tasks, ρ tracks PivotLoRA's shared-protocol gain with Pearson $r = 0.93$ and exact permutation $p = 0.002$. The coefficient ρ thus offers a low-cost measure of subspace–task alignment for classification.

Motivated by this analysis, we propose **PivotLoRA**. The method initializes the LoRA input factor from prediction-relevant activation PCA and allows early task gradients to refine its span. As the factors evolve, however, the input basis can acquire unequal scales and correlations that affect subsequent optimization. We therefore separate learning the span from normalizing its coordinates. After K epochs, PivotLoRA performs one output-preserving orthogonal pivot. Under a full-row-rank condition, the pivot replaces the learned input basis with an orthonormal basis for the same row span and re-encodes the output factor to preserve the adapter update. This retains the task-adapted directions while resetting $AA^\top$ to the identity, allowing training to continue in normalized coordinates without changing the model function at the pivot.

Our primary evaluation focuses on few-shot classification, covering 30 BERT-base settings and 13 methods, with additional classification experiments on Llama-2-7B and Llama-3.1-8B. PivotLoRA ranks first overall and improves over vanilla LoRA in 29 of 30 primary BERT settings, with the largest shared-protocol gains concentrated in low-sample, high-alignment tasks. PivotLoRA achieves these gains despite starting from a smaller initialization norm than vanilla LoRA. Separately, restoring LoRAM's intended initialization magnitude does not eliminate its performance deficit relative to vanilla LoRA. In full-data classification, the gains from initialization direction diminish, consistent with the proposed dependence on labeled sample size. Beyond classification, we conduct a targeted evaluation of the activation construction for generation. GPT-2 summarization shows that a large d/N alone is not enough, as PivotLoRA is competitive with LoRA there but not decisively better. We also evaluate response-predictor activation initialization in completion-only adaptation with Llama-3.1-8B. Under a matched protocol, this extension improves generated-answer accuracy on GSM8K and StrategyQA by 4.5 and 3.6 percentage points, respectively.

Contributions. First, we identify a regime in which LoRA initialization direction matters, combining a high dimension-to-sample ratio with strong subspace–task alignment. Second, we support this view through a spiked-covariance Davis–Kahan analysis, a local first-step LoRA analysis, and validation on synthetic and transformer activations. Third, we propose **PivotLoRA**, which combines prediction-relevant activation-subspace initialization with one output-preserving orthogonal reparameterization. Fourth, we introduce ρ as a frozen-forward-pass diagnostic for classification and evaluate PivotLoRA primarily through few-shot classification across encoder and decoder models, complemented by full-data and summarization controls and a targeted extension to generated-answer reasoning.

2 Related Work

LoRA initialization. Initializers for LoRA [Hu et al., 2022] draw the adapter subspace from three sources. Weight-based methods use singular components of the pretrained matrix, either the principal ones [Meng et al., 2024] or the minor ones [Wang et al., 2025]. Gradient-based methods take the SVD of a task gradient [Wang et al., 2024, Zhang et al., 2025a] or of a Fisher-weighted

variant [Zhang et al., 2026]. Activation-based methods estimate directions from downstream inputs, either from all-token covariances [Paischer et al., 2025] or by modulating the weight decomposition with activation statistics [Yang et al., 2024]. PivotLoRA belongs to the third family but estimates its subspace at prediction-relevant positions, and we test when direction matters beyond scale.

Magnitude and optimization accounts. Zhang et al. [2025b] argue that the gains of weight-spectral initializers largely reflect amplified update magnitude, and Lee et al. [2026] find that vanilla LoRA matches many of its variants once each method's learning rate is tuned. Differential factor learning rates [Hayou et al., 2024] change the optimization without changing the adapter subspace, whereas periodic re-initialization or resampling [Yang et al., 2025, Hao et al., 2024] replaces directions during training. The PivotLoRA pivot instead adds no new direction and preserves the adapter update exactly, changing only the coordinates in which optimization continues. Our analysis complements these accounts by characterizing a regime in which direction matters, with controls that separate direction from initialization scale and learning rate (extended review in Appendix A).

3 When Does Initialization Direction Matter?

We argue that LoRA initialization direction matters most in a regime that combines a high dimension-to-sample ratio with strong alignment between the estimated activation subspace and supervised task structure. A spiked-covariance perturbation analysis characterizes the finite-sample stability of activation PCA, while an alignment coefficient ρ quantifies the supervised structure captured by the resulting subspace. The analysis is most direct for classification settings in which predictions are formed from a designated representation, such as the `[CLS]` representation in BERT.

Notation. Let N denote the total number of labeled examples in the few-shot training set, sampled approximately class-balanced across C classes, so each class has roughly $n_c \approx N/C$ examples. PivotLoRA estimates its PCA subspace from the global target-token covariance over all N examples. The between-class signal used by the alignment diagnostic depends on class means estimated from approximately n_c examples each. Since C is fixed within a task, replacing N by n_c changes the dimension-to-sample ratio only by a task-dependent constant. We therefore express the theoretical regime in terms of d_{eff}/N and use d/N empirically as a simple architecture-level proxy, not as the literal quantity in the perturbation bound.

Subspace estimation. We model target-token activations at an adapted layer as draws from a spiked covariance distribution

$$\Sigma = \sum_{i=1}^{r^*} (\lambda_i - \sigma^2) v_i v_i^\top + \sigma^2 I, \tag{1}$$

where $\lambda_i > \sigma^2$, the vectors v_i are orthonormal signal directions, and $\sigma^2 I$ is an isotropic noise floor. This model is idealized, but it approximates the variance concentration observed in transformer activations. Appendix B.1 shows that the top eight components of BERT-base `[CLS]` activations capture 38–46% of the total variance.

Under a sub-Gaussian sampling assumption, combining Davis–Kahan perturbation theory [Davis and Kahan, 1970, Yu et al., 2015] with effective-rank covariance concentration gives the regime-level scaling

$$\mathbb{E}\left[\|\sin\Theta(\hat{V}, V^*)\|_F\right] \lesssim \min\left\{\sqrt{r^*}, \sqrt{r^*}\,\frac{\|\Sigma\|_{\text{op}}}{\delta}\left(\sqrt{\frac{d_{\text{eff}}}{N}} + \frac{d_{\text{eff}}}{N}\right)\right\}, \tag{2}$$

where $\delta = \lambda_{r^*} - \sigma^2$ is the eigengap and

$$d_{\text{eff}} = \frac{\text{Tr}(\Sigma)}{\|\Sigma\|_{\text{op}}}$$

is the effective rank. The quantity d_{eff} reflects activation anisotropy more faithfully than the ambient dimension d. Appendix B.2 gives the derivation. The bound provides stronger control when N is large relative to d_{eff} and the eigengap is large. When either weakens, the estimated subspace becomes more sensitive to finite-sample variation.

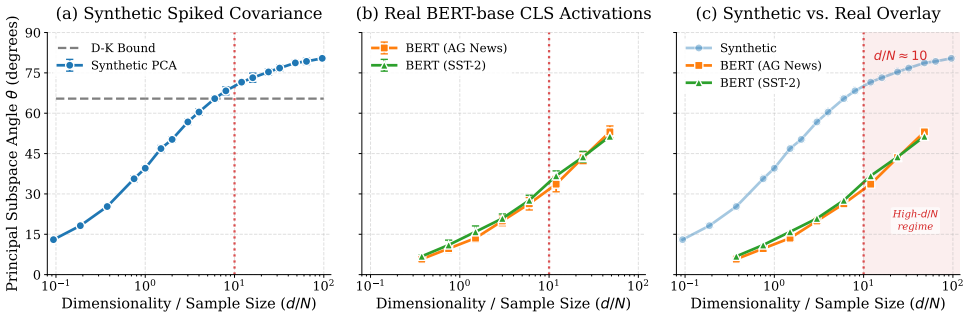

Figure 1: PCA estimation error vs. d/N. **(a)** Synthetic spiked covariance. **(b)** BERT-base `[CLS]` activations on AG News and SST-2. **(c)** Overlay. The shaded region marks the empirical high-d/N regime associated with the largest direction-aware gains in our experiments. Error bars show ± 1 std.

Empirical validation. Figure 1 evaluates this prediction using PCA estimation error from both a synthetic spiked model and real BERT-base activations. For the real-activation experiment, we compute PCA on few-shot `[CLS]` embeddings from AG News and SST-2 and measure principal angles to a reference subspace estimated from the full training set. Real activations follow the same qualitative pattern as the synthetic model, with stable estimation at low d/N and rapid degradation near $d/N \approx 10$ in the evaluated configurations. The onset of degradation depends on the covariance spectrum and sampling conditions and is not determined by d/N alone.

Subspace-task alignment. A reliably estimated activation subspace is useful only if it also captures supervised task structure. Let Σ_{task} denote the between-class scatter of target-token representations,

$$\Sigma_{\text{task}} = \frac{1}{N} \sum_{c=1}^{C} n_c (\mu_c - \bar{\mu})(\mu_c - \bar{\mu})^\top,$$

where n_c is the number of examples in class c, μ_c is the class mean, and $\bar{\mu}$ is the global mean.

Definition 1 (Subspace-task alignment)**.** *For a rank-r subspace with orthonormal basis $U \in \mathbb{R}^{d \times r}$, define*

$$\rho(U, \Sigma_{\text{task}}) = \frac{\text{Tr}(U^\top \Sigma_{\text{task}} U)}{\text{Tr}(\Sigma_{\text{task}})} \in [0, 1]. \tag{3}$$

The value ρ measures the fraction of between-class signal captured by the subspace. It equals one when the column span of U contains the range of Σ_{task}. The metric requires discrete class labels and is not defined for open-ended generation.

Why direction affects the first LoRA update. Consider a LoRA contribution

$$\Delta z = sBAh(x), \qquad s = \alpha/r,$$

where $h(x) \in \mathbb{R}^d$ is the target-token representation. Let $U \in \mathbb{R}^{d \times r}$ have orthonormal columns and initialize $A = U^\top$, so that the adapter observes the projected features $U^\top h(x)$. At standard zero-B initialization, $\Delta z = 0$ and $\nabla_A \mathcal{L} = 0$ exactly, while $\nabla_B \mathcal{L}$ can be nonzero. The first adapter update therefore acts on B, and its dependence on the initial subspace enters through the projected features.

Let

$$G_U = \nabla_B \mathcal{L}(0; U)$$

denote the initial population gradient. Under the class-conditional mean, cross-covariance, centering, and proportionality assumptions stated in Appendix B.3, its squared Frobenius norm satisfies

$$\|G_U\|_F^2 = s^2 \gamma \ \text{Tr}\left(U^\top \Sigma_{\text{task}} U\right) \tag{4}$$

for a task-dependent constant $\gamma > 0$. Thus, the first-order loss decrease from an update to B is controlled by the supervised structure captured by the row span of A. Appendix B.3 gives the derivation and states the assumptions explicitly.

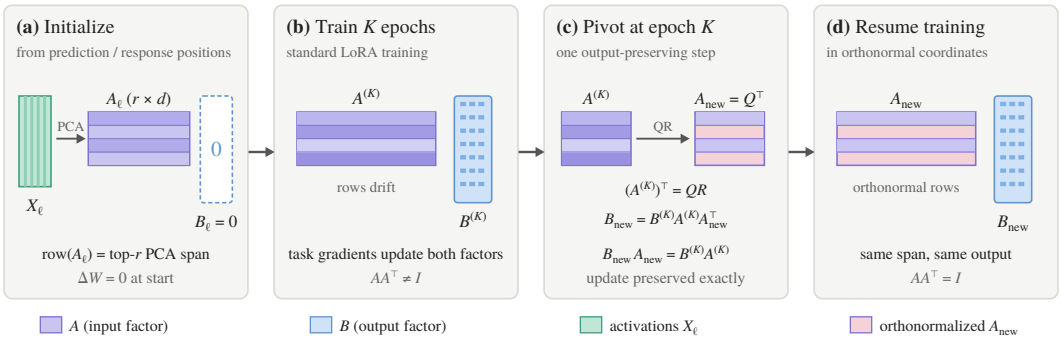

Figure 2: Overview of PivotLoRA. Activations at supervised prediction positions initialize the row span of A_ℓ, with $B_\ell = 0$ so that $\Delta W_\ell = 0$. For classification, the activation source is the `[CLS]` representation (BERT) or the sequence mean (decoders); for completion-only reasoning, it comprises hidden states that predict supervised response tokens. After K epochs, PivotLoRA replaces the learned input basis with an orthonormal basis for the same row span and re-encodes B_ℓ so that $B_\ell A_\ell$ is preserved.

If U is sampled uniformly from rank-r subspaces independently of the fixed task matrix, then its expected captured fraction of between-class signal is r/d. For BERT-base with $r = 8$, the random-subspace expectation is $r/d \approx 0.010$, while the most aligned task reaches $\rho \approx 0.84$ in our measurements, as reported in Appendix E.7. Direction-aware initialization is therefore expected to be useful when ρ substantially exceeds r/d and the relevant activation structure can be estimated from the few-shot sample.

The initial span also changes slowly. The gradient with respect to A is proportional to B, which starts at zero, so A moves slowly; in our classification runs its norm stays within 5% of its initial value until the pivot (Appendix F.6).

Remark 1 (Two-condition view)**.** *Direction-aware initialization yields meaningful gains when (i) N is small relative to the representation dimension, so the adapter row span is sensitive to its initialization, and (ii) the estimated subspace captures between-class signal, which requires a spectral gap large enough to control the error in Equation* (2) *and a moderate to high ρ. In our BERT-base measurements the advantage is largest above $d/N \approx 10$. It shrinks as either condition weakens, and vanilla LoRA is competitive once data are abundant or the source is misaligned. There direction adds little, consistent with the magnitude account of Zhang et al. [2025b].*

4 Method

PivotLoRA operationalizes the two conditions of Section 3 by initializing the LoRA input factor from activations at prediction-relevant positions and performing one output-preserving orthogonal pivot after early training drift (Figure 2).

Prediction positions depend on the objective. For BERT classification, we use the `[CLS]` representation. Decoder-scale classification, whose label describes the whole sequence, uses the mean of adapter inputs over input positions. Completion-only reasoning uses hidden states that predict supervised response tokens. For the GPT-2 summarization control we compare the sequence mean with response positions (Appendix E.4).

All methods use the standard LoRA parameterization

$$\Delta W = \frac{\alpha}{r} BA, \tag{5}$$

where $A \in \mathbb{R}^{r \times d}$ stores input directions as rows and $B \in \mathbb{R}^{m \times r}$ maps low-rank coordinates to the output space. Unless stated otherwise, we use rank $r = 8$ and scaling $\alpha = 16$.

4.1 Supervised-Position Subspace Initialization

Activation initializers differ in the positions used to estimate their subspaces. All-token pooling provides more activation vectors and can estimate shared structure more reliably, whereas prediction-

position pooling targets the representation directly connected to the supervised objective. Activation-source selection is therefore an alignment–estimation trade-off, and no single source need be optimal. Our primary controlled classification construction uses prediction-position PCA. A matched ablation shows that all-token and per-projection sources can score higher (Appendix E.12).

Let $\mathcal{D}$ contain N training examples. A frozen forward pass produces one activation vector $x_{\ell,n} \in \mathbb{R}^d$ for each example n and adapted layer ℓ. For classification, $x_{\ell,n}$ is the adapter input at the `[CLS]` position (BERT) or its mean over input positions (decoders). For completion-only reasoning, it is the pooled response-predictor representation defined in Section 4.2. We form

$$X_\ell = \begin{bmatrix} x_{\ell,1}^\top \\ \vdots \\ x_{\ell,N}^\top \end{bmatrix} \in \mathbb{R}^{N\times d}, \qquad \mu_\ell = \frac{1}{N} X_\ell^\top \mathbf{1}, \qquad \bar{X}_\ell = X_\ell - \mathbf{1}\mu_\ell^\top .$$

The empirical covariance is

$$\widehat{\Sigma}_\ell = \frac{1}{N} \bar{X}_\ell^\top \bar{X}_\ell. \tag{6}$$

Let $Q_\ell \in \mathbb{R}^{d\times r}$ contain the leading r eigenvectors of $\widehat{\Sigma}_\ell$, ordered by decreasing eigenvalue:

$$\widehat{\Sigma}_\ell Q_\ell = Q_\ell \Lambda_\ell, \qquad Q_\ell^\top Q_\ell = I_r.$$

We initialize

$$A_\ell^{(0)} = c\, R_\ell^\top Q_\ell^\top, \qquad B_\ell^{(0)} = \mathbf{0}, \qquad c = \sqrt{5/d}, \tag{7}$$

where $R_\ell \in \mathbb{R}^{r\times r}$ is a fixed random orthogonal matrix drawn once per adapter. Because R_ℓ is orthogonal, the row span of $A_\ell^{(0)}$ is exactly the leading PCA subspace; R_ℓ only mixes coordinates within it. The scale gives $\|A_\ell^{(0)}\|_F = \sqrt{5r/d}$, smaller than the approximately $\sqrt{r/3}$ of the Kaiming-uniform initialization used by vanilla LoRA (about $7\times$ smaller for $d = 768$ and $17\times$ for $d = 4096$). The initialization is therefore not norm-matched to vanilla LoRA, and because it is smaller, PivotLoRA's gains do not come from a larger initialization norm. Since $B_\ell^{(0)} = \mathbf{0}$, initialization leaves the pretrained model function unchanged. We call this initialization without the pivot TT-PCA (target-token PCA). We prefer PCA to LDA for three reasons. PCA needs no labels, so the same construction applies to completion-only reasoning, where there are no classes. It keeps ρ an independent measure of how much supervised structure an unsupervised subspace captures. LDA also yields at most $C - 1 \le 3$ directions here, fewer than $r = 8$. The TT-LDA ablation (Appendix C.2) has a better average rank than TT-PCA but uses a different learning-rate structure (Appendix F.9).

In the main classification protocol, the PCA eigenvalues, in decreasing order, determine an allocation of the learning rate across the r columns of B. Because of R_ℓ, the input direction paired with each column mixes several principal directions. The average B-to-A learning-rate ratio is $\lambda_{\text{ratio}} = 16$, matching the LoRA+ scale of Hayou et al. [2024]. This allocation is an optimization heuristic and is not part of the subspace definition. Classification-specific learning-rate equations appear in Appendix C.1.

4.2 Response-Position Initialization for Generative Reasoning

For completion-only fine-tuning, each sequence contains a prompt followed by a supervised response, with prompt tokens excluded from the language-modeling loss. Because hidden state t predicts token $t + 1$, let $M_{n,t+1} \in \{0, 1\}$ indicate whether token $t + 1$ is supervised. The corresponding predictor positions are

$$\mathcal{P}_n = \{t : M_{n,t+1} = 1\}.$$

For adapted layer ℓ, let $h_{\ell,n,t} \in \mathbb{R}^d$ denote the adapter input at predictor position t. PivotLoRA forms one activation vector per training example:

$$x_{\ell,n} = \frac{1}{|\mathcal{P}_n|} \sum_{t\in\mathcal{P}_n} h_{\ell,n,t}. \tag{8}$$

We then center the resulting example-level activation matrix and compute its leading r principal directions using Eq. (6). For the reasoning experiments, these directions initialize $A_\ell^{(0)} = \sqrt{5/d}\, Q_\ell^\top$, with $B_\ell^{(0)} = \mathbf{0}$, the same scale as Eq. (7) without the rotation. Appendix D gives the prompt templates, masking rules, generation settings, and remaining training configuration.

4.3 Output-Preserving Orthogonal Pivot

Because A remains trainable, its initially orthogonal rows can become anisotropic during early optimization. PivotLoRA performs a single reparameterization after K epochs while preserving both the learned adapter update and the model function.

Let $A^{(K)} \in \mathbb{R}^{r\times d}$ and $B^{(K)} \in \mathbb{R}^{m\times r}$ denote the factors at the pivot. Assuming that $A^{(K)}$ has full row rank, we compute the reduced QR decomposition of its transpose in double precision:

$$\big(A^{(K)}\big)^\top = QR, \qquad Q^\top Q = I_r, \tag{9}$$

and define

$$A^{\mathrm{new}} = Q^\top.$$

The rows of A^{new} are orthonormal and have the same row span as $A^{(K)}$. We re-encode the output factor as

$$B^{\mathrm{new}} = B^{(K)} A^{(K)} \big(A^{\mathrm{new}}\big)^\top. \tag{10}$$

Because A^{new} is an orthonormal basis for the row span of $A^{(K)}$, we have $A^{(K)} = A^{(K)}(A^{\mathrm{new}})^\top A^{\mathrm{new}}$, and therefore

$$B^{\mathrm{new}} A^{\mathrm{new}} = B^{(K)} A^{(K)}. \tag{11}$$

The fixed scale α/r in Eq. (5) is unchanged. The pivot therefore preserves the LoRA update, and hence the model function, up to floating-point error.

Although the represented update is unchanged, subsequent factorized optimization need not be invariant to the reparameterization. In particular, if $G = \nabla_{\Delta W}\mathcal{L}$, then

$$\nabla_B \mathcal{L} = \frac{\alpha}{r} G A^\top.$$

The geometry of the B-factor update therefore depends on the Gram matrix $AA^\top$. The pivot restores

$$A^{\mathrm{new}} \big(A^{\mathrm{new}}\big)^\top = I_r,$$

removing coordinate anisotropy from this Gram matrix without replacing the learned row span. Concretely, a plain gradient step on B changes the update by $-\eta(\alpha/r)^2 G A^\top A$, so before the pivot each direction in the row span is scaled by the corresponding squared singular value of A. After the pivot, $A^\top A$ is the orthogonal projector onto that span, and every learned direction receives the same step. The pivot also returns the rows of A to unit norm, which enlarges the effective step on B. In classification, the norm of A changes little before the pivot, so the pivot raises it about twelvefold. In reasoning, A grows about sixfold and its Gram matrix has an effective rank of about two or fewer out of eight, and the pivot restores all eight (Appendix F.6).

Optimizer-state handling is distinct from the algebraic preservation guarantee. In the classification implementation, we reset the Adam moments for A and retain those for B. In the reasoning implementation, we reset the optimizer states for both transformed factors while leaving the global learning-rate schedule unchanged. In the classification ablation (Appendix F.7), the QR pivot itself accounts for most of PivotLoRA's improvement over TT-PCA, while the optimizer-state reset produces smaller, task-dependent differences; we therefore regard the reset as an optional stabilizer. The pivot itself takes under a second, and most of PivotLoRA's 5.8% wall-time overhead on BERT-base comes from the per-rank optimizer groups (Appendix G.3).

For each pivoted adapter we record the relative update-preservation error and the row-orthogonality error of A^{new}; both stay below 10^{-6} in every run (Appendix F.6). Unless stated otherwise, PivotLoRA pivots at $K = 3$.

5 Experiments

We evaluate PivotLoRA on classification and generation tasks with four pretrained models. BERT-base provides the main controlled regime analysis, decoder-scale classification tests transfer to larger causal models, GPT-2 summarization tests whether a large d/N suffices, and GSM8K and StrategyQA test the response-position extension for generated-answer reasoning.

5.1 Setup

Models, tasks, and protocol. We evaluate BERT-base [Devlin et al., 2019] on five GLUE tasks (SST-2, CoLA, MNLI, QNLI, and RTE) [Wang et al., 2019] and AG News [Zhang et al., 2015] at $N \in \{16, 32, 64, 128, 256\}$ total labeled examples, sampled approximately class-balanced, with BoolQ as an auxiliary seventh task for the alignment analysis. We evaluate Llama-2-7B [Touvron et al., 2023] and Llama-3.1-8B [Grattafiori et al., 2024] on SST-2 and AG News at $N \in \{64, 256\}$, and additionally evaluate Llama-3.1-8B on BoolQ [Clark et al., 2019]. We evaluate GPT-2 [Radford et al., 2019] on XSum [Narayan et al., 2018] and CNN/DailyMail [Hermann et al., 2015, Nallapati et al., 2016] with $N \in \{64, 256\}$ training sequences. Finally, we evaluate completion-only reasoning with Llama-3.1-8B on GSM8K [Cobbe et al., 2021] and StrategyQA [Geva et al., 2021] using $N = 128$ training sequences, three paired seeds, and the complete test split of each dataset.

The primary BERT benchmark contains 30 task–N settings over three seeds. All methods use rank $r = 8$, query and value adapters, AdamW, 15 epochs, and batch size 16. The shared BERT base learning-rate anchor is $\eta_A = 2 \times 10^{-4}$. Unless a baseline specifies otherwise, $\alpha_{\mathrm{LoRA}} = 16$. We compare LoRA, DoRA, AdaLoRA, PiSSA, LoRA-GA, LoRA-One, LoRA+, EVA, LoRAM, and SRLoRA, plus the TT-PCA and TT-LDA ablations. Method-specific learning-rate allocations, the shared initialization scaling, other implementation settings, sampling, hardware, and evaluation details appear in Appendix C.

For generated-answer reasoning, paired LoRA and PivotLoRA runs share the sampled examples, base AdamW configuration, learning-rate schedule, tokenization, completion mask, training budget, decoding procedure, and answer scoring. Response-position PivotLoRA additionally applies PCA initialization, one output-preserving pivot after epoch $K = 3$, and its prescribed post-pivot optimizer-state reset. Unlike the classification protocol, neither reasoning method uses differential or eigenvalue-scaled learning rates.

5.2 Few-Shot Classification on BERT-Base

Table 1 summarizes all six primary tasks, reporting each method's mean score across tasks at each N (accuracy, or Matthews correlation for CoLA), its average rank over the 30 task–N settings, and the number of settings in which it exceeds vanilla LoRA. Per-task results with standard deviations appear in Appendix E.1. Under the shared protocol, no prior initialization method consistently outperforms vanilla LoRA over all 30 settings. TT-PCA surpasses the published baselines on average, supporting activation-based initialization in this regime.

PivotLoRA achieves the best average rank and improves over vanilla LoRA in 29 of 30 primary settings (Appendix E.2). It is strongest at $N \leq 64$, with an average rank of 1.39 and the best or tied-best result in 13 of 18 settings. The margins narrow at $N \in \{128, 256\}$ as d/N decreases. On AG News at $N = 64$, for example, PivotLoRA reaches about 89% of the full-data accuracy, compared with 68% for LoRA (Appendix E.11).

Sensitivity and full fine-tuning. At $N = 64$, a representative learning-rate sweep narrows some shared-protocol margins, but PivotLoRA remains strongest on average over AG News, SST-2, and CoLA among the swept methods, with 0.667 versus 0.655 for TT-PCA, 0.636 for LoRA+, and 0.622 for LoRA (Appendices F.9 and F.4). At the same N, PivotLoRA exceeds the mean full-fine-tuning score on all six primary tasks while training approximately 0.30M adapter parameters (Appendix E.6).

Direction versus learning-rate allocation. At $N = 64$, a random orthonormal basis given the same eigenvalue-scaled learning rates reaches 0.444, 0.571, and 0.028 on AG News, SST-2, and CoLA, versus 0.751, 0.769, and 0.268 for TT-PCA, while TT-PCA with uniform or permuted rates is essentially unchanged (Appendix F.10). The PCA directions, not the rate allocation, carry the gain.

Table 1: BERT-base, mean over six tasks and three seeds at each N; rank and >LoRA over 30 settings. **Bold**/underline: best/second.

	Labeled examples N						
Method	16	32	64	128	256	Rank↓	>LoRA
LoRA	.402	.428	.488	.566	.618	5.23	–
DoRA	.404	.423	.486	.561	.614	5.83	10/30
AdaLoRA	.379	.381	.407	.433	.458	11.30	2/30
PiSSA	.383	.395	.426	.499	.581	9.87	2/30
LoRA-GA	.383	.391	.427	.505	.601	9.30	5/30
LoRA-One	.379	.388	.431	.523	.606	8.47	6/30
LoRA+	**.459**	.481	.438	.425	.415	8.43	9/30
EVA	.381	.391	.429	.520	.606	8.47	5/30
LoRAM	.379	.387	.434	.517	.608	8.43	6/30
SRLoRA	.401	.423	.472	.554	.617	6.93	4/30
TT-PCA	.415	.445	.546	.606	.631	3.77	24/30
TT-LDA	.415	.455	.555	**.607**	.634	3.13	25/30
PivotLoRA	.451	**.500**	**.575**	.606	**.641**	**1.83**	**29/30**

Table 2: Decoder-scale classification at $N = 64$, mean over three seeds.

	Llama-2-7B		Llama-3.1-8B	
Method	SST-2	AG News	SST-2	AG News
LoRA	.708	.748	.704	.759
PiSSA	.635	.700	.622	.715
LoRAM	.763	.742	.662	.748
EVA	.713	.745	.643	.743
PivotLoRA	**.769**	**.784**	**.709**	**.799**

Table 3: Llama-3.1-8B reasoning at N=128: accuracy (%) over three seeds, $\pm$ one sample SD. McNemar tests are exact and pooled over seeds.

Method	GSM8K	StrategyQA
Base (no adapter)	25.6	54.4
LoRA	28.7 ± 0.2	64.1 ± 2.3
PivotLoRA	**33.2** ± 2.5	**67.7** ± 1.4
Paired gain	$+4.5 \pm 2.3$	$+3.6 \pm 3.1$
Seeds improved	3/3	2/3 (1 tie)
McNemar p	$< 10^{-6}$	0.004

Contribution of the pivot. Applied to the same TT-PCA initialization without the optimizer-state reset, the pivot alone improves the score in 12 of 15 task–N settings, by 0.037 on average and by up to 0.128 on AG News at $N = 32$ (Appendix F.7). Each tested pivot epoch, $K \in \{1, 2, 3, 5, 7, 10\}$, outperforms LoRA at every $N \leq 64$ on SST-2, CoLA, and AG News, so the default $K = 3$ is not a finely tuned choice (Appendix F.5). The gain also does not depend on how the B learning rate is handled after the pivot, since leaving it unchanged gives $+0.036$ over TT-PCA (Appendix F.8).

Robustness. A two-sided sign test on the 29-of-30 record gives $p = 5.8 \times 10^{-8}$, and five-seed reruns of seven representative settings yield bootstrap 95% intervals for the paired gain over LoRA that exclude zero in every case, with mean gains from $+0.032$ on QNLI to $+0.231$ on AG News (Appendices F.1 and F.2). Restoring LoRAM's intended initialization magnitude does not close its gap to LoRA (Appendix C), so initialization magnitude alone does not explain the gains.

5.3 Decoder-Scale Classification

The activation-subspace pattern also appears in decoder-scale classification (Table 2). At $N = 64$, PivotLoRA ranks first on both backbones, while EVA does not close the aggregate gap under the same per-layer rank budget. At $N = 256$, PivotLoRA trails LoRA in three of the four SST-2 and AG News settings, whereas on BoolQ with Llama-3.1-8B PivotLoRA leads LoRA at both $N = 64$ and $N = 256$ (.602 versus .554 and .618 versus .576). Complete results appear in Appendix E.3. Because the decoder activation source is a sequence mean, these gains suggest that a coarse source suffices when the label describes the whole sequence. At $N = 256$, the all-token EVA initializer becomes competitive, consistent with the alignment–estimation trade-off, since a larger sample favors the source with more activation vectors.

5.4 Generated-Answer Reasoning

Table 3 reports completion-only fine-tuning with Llama-3.1-8B on GSM8K (mathematical word problems with worked solutions) and StrategyQA (yes/no questions supervised with supporting facts). Within each dataset and seed, LoRA and PivotLoRA share the same $N = 128$ training examples and are evaluated on the complete test split (1,319 and 687 questions). PivotLoRA improves on LoRA on both datasets, at every seed on GSM8K, and an exact item-level McNemar test pooled over seeds rejects equal accuracy on both. Pooled over the three seeds, PivotLoRA alone answers 655 GSM8K test items correctly against 475 for LoRA alone, and 350 against 276 on StrategyQA. Both methods

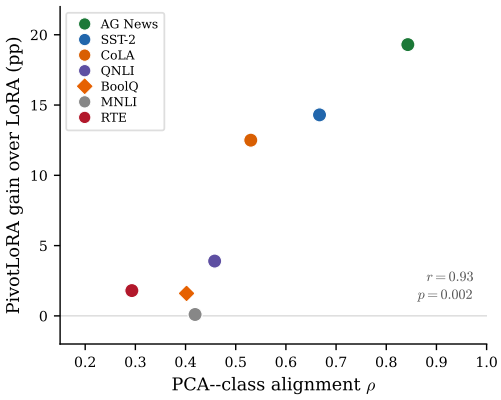

Figure 3: Alignment ρ versus PivotLoRA's gain over LoRA across seven BERT tasks at $N = 64$ (CoLA in Matthews correlation points).

Table 4: Regime map. Mean gain of PivotLoRA over LoRA in points of each task's primary metric, and settings won (details in Table 28). d/N is comparable within a backbone.

Setting	d/N	Source	Gain	Wins
BERT, $N \leq 64$	12–48	`[CLS]`	$+6.9$	18/18
BERT, $N \geq 128$	3–6	`[CLS]`	$+3.2$	11/12
BERT, full data	<0.4	`[CLS]`	$+0.4$	3/6
Llama-2/3.1, $N = 64$	64	seq. mean	$+3.8$	5/5
Llama-2/3.1, $N = 256$	16	seq. mean	$+0.4$	2/5
Llama-3.1, reasoning	32	response	$+4.1$	2/2
GPT-2, summarization	3–12	response	$+0.3$	3/4

follow the trained answer format in at least 95% of generations, so formatting does not explain the gains. These comparisons evaluate the full procedure (Appendix D). In these runs, early training concentrates each input factor on about two or fewer of its eight directions before the pivot, and the pivot restores an isotropic basis for the same span (Appendix F.6).

5.5 Regime Summary and Alignment Diagnostic

Table 4 organizes every evaluated setting by d/N and activation source, following Remark 1. Within each backbone the gain shrinks as d/N falls, and with full training sets PivotLoRA, LoRA, and PiSSA perform comparably (Appendix E.5). On GPT-2 summarization, PivotLoRA is competitive with LoRA but not decisively better under the single learning rate used for reasoning ($+0.2$ to $+0.3$ ROUGE-L points), and it trails under the classification learning-rate recipe whichever positions it pools (Appendix E.4). Because the same response-position source helps reasoning, the difference may reflect the backbone, the adapted modules, or how much loss-relevant structure the leading directions capture in summarization. A large d/N is therefore not sufficient by itself.

Across the six primary BERT tasks plus auxiliary BoolQ, the classification alignment coefficient ρ at $N = 64$ is strongly associated with PivotLoRA's shared-protocol gain over vanilla LoRA, with Pearson $r = 0.93$ and exact permutation $p = 0.002$ as shown in Figure 3. This is consistent with Equation (4), in which ρ is the share of between-class signal available to the first update of B. Spearman correlation is 0.86, leave-one-task-out Pearson correlations stay within $[0.87, 0.96]$, and a cross-split estimate preserves the task ordering (Appendices E.7 and E.8). Both d/N and ρ are available before training. When d/N is large and ρ stays high across resampled subsets, direction-aware initialization is likely to help, and when either is small, a well-tuned vanilla LoRA is a strong default.

6 Conclusion

Initialization direction is not uniformly consequential for LoRA. It matters when few labeled examples leave the adapter's row span sensitive to its starting point and the estimated activation subspace carries supervised structure, and it fades once data are abundant or the activation source is poorly aligned with the objective. An alignment measure that does not need class labels, such as one built from loss gradients at initialization, would extend the diagnostic to generative tasks, and calibrating it across backbones would turn the regime account into a decision rule. Because the pivot applies to any full-row-rank starting basis, it could also test whether these regime boundaries hold for weight- or gradient-based initializers.

Acknowledgments and Disclosure of Funding

Funding and support. We thank the University of Delaware for providing computational resources. Peng Zhao's work was partially supported by a research gift from Microsoft Research.

Competing interests. The authors declare no competing interests.

References

Jinho Baik, Gérard Ben Arous, and Sandrine Péché. Phase transition of the largest eigenvalue for nonnull complex sample covariance matrices. *The Annals of Probability*, 33(5):1643–1697, 2005. doi: 10.1214/009117905000000233.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2924–2936, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1300. URL https://aclanthology.org/N19-1300/.

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. doi: 10.48550/arXiv.2110.14168. URL https://arxiv.org/abs/2110.14168.

Ido Dagan, Oren Glickman, and Bernardo Magnini. The PASCAL recognising textual entailment challenge. In Joaquin Quiñonero-Candela, Ido Dagan, Bernardo Magnini, and Florence d'Alché Buc, editors, *Machine Learning Challenges: Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Textual Entailment*, volume 3944 of *Lecture Notes in Computer Science*, pages 177–190. Springer, 2006. doi: 10.1007/11736790_9. URL https://link.springer.com/chapter/10.1007/11736790_9.

Chandler Davis and William Morton Kahan. The rotation of eigenvectors by a perturbation. III. *SIAM Journal on Numerical Analysis*, 7(1):1–46, 1970. doi: 10.1137/0707001.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423/.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. Did Aristotle use a laptop? A question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361, 2021. doi: 10.1162/tacl_a_00370. URL https://aclanthology.org/2021.tacl-1.21/.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. doi: 10.48550/arXiv.2407.21783. URL https://arxiv.org/abs/2407.21783.

Yongchang Hao, Yanshuai Cao, and Lili Mou. Flora: Low-rank adapters are secretly gradient compressors. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 17554–17571. PMLR, 2024. URL https://proceedings.mlr.press/v235/hao24a.html.

Soufiane Hayou, Nikhil Ghosh, and Bin Yu. LoRA+: Efficient low rank adaptation of large models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 17783–17806. PMLR, 2024. URL https://proceedings.mlr.press/v235/hayou24a.html.

Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*, volume 28, 2015. URL https://proceedings.neurips.cc/paper/2015/hash/afdec7005cc9f14302cd0474fd0f3c96-Abstract.html.

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. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Iain M. Johnstone. On the distribution of the largest eigenvalue in principal components analysis. *The Annals of Statistics*, 29(2):295–327, 2001. doi: 10.1214/aos/1009210544.

Yu-Ang Lee, Ching-Yun Ko, Pin-Yu Chen, and Mi-Yen Yeh. Learning rate matters: Vanilla LoRA may suffice for LLM fine-tuning. In *Advances in Neural Information Processing Systems*, 2026. URL https://arxiv.org/abs/2602.04998.

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*, pages 32100–32121. PMLR, 2024. URL https://proceedings.mlr.press/v235/liu24bn.html.

Fanxu Meng, Zhaohui Wang, and Muhan Zhang. PiSSA: Principal singular values and singular vectors adaptation of large language models. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-3846. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/db36f4d603cc9e3a2a5e10b93e6428f2-Abstract-Conference.html.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gülçehre, and Bing Xiang. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/K16-1028. URL https://aclanthology.org/K16-1028/.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium, 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1206. URL https://aclanthology.org/D18-1206/.

Fabian Paischer, Lukas Hauzenberger, Thomas Schmied, Benedikt Alkin, Marc Deisenroth, and Sepp Hochreiter. Parameter efficient fine-tuning via explained variance adaptation. In *Advances in Neural Information Processing Systems*, volume 38, pages 46215–46267. Curran Associates, Inc., 2025. doi: 10.52202/085713-1539. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/41d33bd41fd44bd9dba0e092047cf213-Abstract-Conference.html.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. Technical report, OpenAI, 2019. URL https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1264. URL https://aclanthology.org/D16-1264/.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL https://aclanthology.org/D13-1170/.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. doi: 10.48550/arXiv.2307.09288. URL https://arxiv.org/abs/2307.09288.

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 *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rJ4km2R5t7.

Hanqing Wang, Yixia Li, Shuo Wang, Guanhua Chen, and Yun Chen. MiLoRA: Harnessing minor singular components for parameter-efficient LLM finetuning. In Luis Chiruzzo, Alan Ritter, and Lu Wang, editors, *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4823–4836, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.naacl-long.248. URL https://aclanthology.org/2025.naacl-long.248/.

Shaowen Wang, Linxi Yu, and Jian Li. LoRA-GA: Low-rank adaptation with gradient approximation. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-1741. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/62c4718cc334f6a0a62fb81c4a2095a1-Abstract-Conference.html.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641, 2019. doi: 10.1162/tacl_a_00290. URL https://aclanthology.org/Q19-1040/.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1101. URL https://aclanthology.org/N18-1101/.

Haodong Yang, Lei Wang, and Md Zakir Hossain. SRLoRA: Subspace recomposition in low-rank adaptation via importance-based fusion and reinitialization. *arXiv preprint arXiv:2505.12433*, 2025. doi: 10.48550/arXiv.2505.12433. URL https://arxiv.org/abs/2505.12433.

Yibo Yang, Xiaojie Li, Zhongzhu Zhou, Shuaiwen Leon Song, Jianlong Wu, Liqiang Nie, and Bernard Ghanem. CorDA: Context-oriented decomposition adaptation of large language models for task-aware parameter-efficient fine-tuning. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-2292. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/83f95bb0ac5046338ea2afe3390e9f4b-Abstract-Conference.html.

Yi Yu, Tengyao Wang, and Richard J. Samworth. A useful variant of the Davis–Kahan theorem for statisticians. *Biometrika*, 102(2):315–323, 2015. doi: 10.1093/biomet/asv008.

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

Qingyue Zhang, Chang Chu, Tianren Peng, Qi Li, Xiangyang Luo, Zhihao Jiang, and Shao-Lun Huang. LoRA-DA: Data-aware initialization for low-rank adaptation via asymptotic analysis. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2510.24561.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*, volume 28, 2015. URL https://proceedings.neurips.cc/paper/2015/hash/250cf8b51c773f3f8dc8b4be867a9a02-Abstract.html.

Yuanhe Zhang, Fanghui Liu, and Yudong Chen. LoRA-One: One-step full gradient could suffice for fine-tuning large language models, provably and efficiently. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 75513–75574. PMLR, 2025a. URL https://proceedings.mlr.press/v267/zhang25ax.html.

Zicheng Zhang, Haoran Li, Yifeng Zhang, Guoqiang Gong, Jiaxing Wang, Junxing Hu, Pengzhang Liu, and Qixia Jiang. The primacy of magnitude in low-rank adaptation. In *Advances in Neural Information Processing Systems*, volume 38, 2025b. URL https://arxiv.org/abs/2507.06558.

A Extended Related Work

LoRA initialization. Vanilla LoRA [Hu et al., 2022] initializes A with Kaiming-uniform noise and $B = 0$, so that $\Delta W = 0$ at the start of training. Subsequent work has explored several sources of initialization signal.

Weight-based methods derive the adapter subspace from the pretrained weight matrix W_0. PiSSA [Meng et al., 2024] initializes A and B from the top-r singular components of W_0 and freezes the residual $W^{\text{res}} = W_0 - W_0^{(r)}$. MiLoRA [Wang et al., 2025] takes the complementary approach and initializes from minor singular components to preserve the dominant directions of W_0. Both methods are task-agnostic and derive their subspaces from pretrained weight geometry alone. Because our benchmark already includes PiSSA as a representative weight-spectral baseline, we do not evaluate MiLoRA separately.

Gradient-based methods use the downstream task loss. LoRA-GA [Wang et al., 2024] computes a forward–backward pass and initializes from the SVD of $\nabla_W \mathcal{L}$. LoRA-One [Zhang et al., 2025a] uses balanced singular-value scaling between A and B. LoRA-DA [Zhang et al., 2026] augments the gradient with a diagonal Fisher estimate and solves a quadratic program for initialization. Appendix F.3 reports our LoRA-DA implementation under the few-shot protocol used in this work.

Activation-based methods estimate a subspace from the input distribution. EVA [Paischer et al., 2025] collects activations from all token positions in a calibration minibatch, computes an SVD of their covariance, and initializes A from the resulting right singular vectors. The original method also permits adaptive redistribution of rank across layers. In our controlled comparison, we disable redistribution and use uniform rank $r = 8$ at every adapted layer. CorDA [Yang et al., 2024] modulates the weight SVD by an activation covariance, thereby orienting the decomposition toward downstream input statistics. CorDA requires covariance inversion, which was numerically unstable in our smallest-N setting, so it is not included in the primary comparison.

Activation-based initializers also differ in which token positions supply the activation statistics. All-token pooling, as used by EVA, provides more activation observations, whereas PivotLoRA's prediction-position pooling targets representations associated with the supervised objective. This distinction introduces an alignment–estimation trade-off, with no universal advantage for either source.

LoRAM [Zhang et al., 2025b] provides a magnitude-based perspective. It argues that gains from pretrained weight-spectral initializers can often be explained by update-magnitude amplification, not by the semantic content of the selected singular directions. LoRAM approximates these gains using magnitude-scaled orthogonal bases without storing spectral components. Our setting is complementary. PivotLoRA studies directions estimated from downstream activation covariance and includes controls that separate prediction-position direction from eigenvalue-scaled learning rates.

Dynamic adapter methods. SRLoRA [Yang et al., 2025] periodically fuses low-importance rank-one pairs into the frozen weights and reinitializes from unused SVD directions. Flora [Hao et al., 2024] resamples random projections at fixed intervals. PivotLoRA instead performs one output-preserving orthogonal reparameterization. It replaces the learned input basis with an orthonormal basis for the same row span and transforms B so that the represented update is preserved up to numerical precision. The pivot is therefore a coordinate change, unlike periodic rank replacement or random resampling.

Structural and optimization variants. DoRA [Liu et al., 2024] decomposes the adapted weight into magnitude and direction components. AdaLoRA [Zhang et al., 2023] parameterizes adapters as SVD triplets and prunes singular values during training. LoRA+ [Hayou et al., 2024] retains the standard factorization but uses different learning rates for the two factors, typically with $\eta_B/\eta_A \approx 16$. These methods alter the adapter parameterization, rank allocation, or factor-wise optimization rule. PivotLoRA retains the standard low-rank update and instead changes how its input-side coordinates are initialized and, once during training, orthogonally reparameterized.

Learning-rate selection can materially affect comparisons among LoRA methods. Lee et al. [2026] systematically re-evaluate vanilla LoRA and nine variants over learning rate, batch size, rank, and training duration. They find that different methods favor different learning-rate ranges and that, after broad tuning, their peak performance is often similar. This result cautions against interpreting

improvements obtained under one shared configuration as evidence of an intrinsic methodological advantage. It also motivates our method-specific learning-rate sensitivity analysis (Appendix F.9), which tests whether the principal comparisons persist over a prespecified range under our few-shot protocol.

Subspace estimation in high dimensions. The Davis–Kahan $\sin\Theta$ theorem [Davis and Kahan, 1970, Yu et al., 2015] bounds the angle between estimated and population eigenspaces under spectral perturbation. Under spiked covariance models [Johnstone, 2001], PCA recovery depends on sample size, effective dimension, and spectral separation. In proportional asymptotic regimes, eigenvectors can also undergo a phase transition near the BBP threshold [Baik et al., 2005]. We use these results as regime-level motivation for dependence on d_{eff}/N, not as a complete probabilistic model of transformer fine-tuning.

B Additional Theory

B.1 Empirical Variance Concentration in `[CLS]` Activations

Section 3 models prediction-position activations through the spiked covariance form

$$\Sigma = \sum_{i=1}^{r^*} (\lambda_i - \sigma^2) v_i v_i^\top + \sigma^2 I,$$

where $\lambda_i > \sigma^2$, the vectors $v_1, \ldots, v_{r^*}$ are orthonormal signal directions, and $\sigma^2 I$ is an isotropic noise floor. Under this parameterization, the population eigenvalue along v_i is λ_i, and each remaining population eigenvalue is σ^2.

The model is idealized but captures the possibility that substantial activation variance is concentrated in a low-dimensional subspace. We examine this premise using frozen BERT-base `[CLS]` representations.

We run the frozen encoder on $N = 64$ total labeled examples from AG News and SST-2, sampled approximately class-balanced. We extract the final-layer `[CLS]` representation, construct

$$\bar{X} = X - \mathbf{1}\mu^\top \in \mathbb{R}^{N \times 768}, \qquad \mu = N^{-1} X^\top \mathbf{1},$$

and compute

$$\widehat{\Sigma} = N^{-1} \bar{X}^\top \bar{X}.$$

Figure 4 reports the leading eigenvalues and cumulative explained variance.

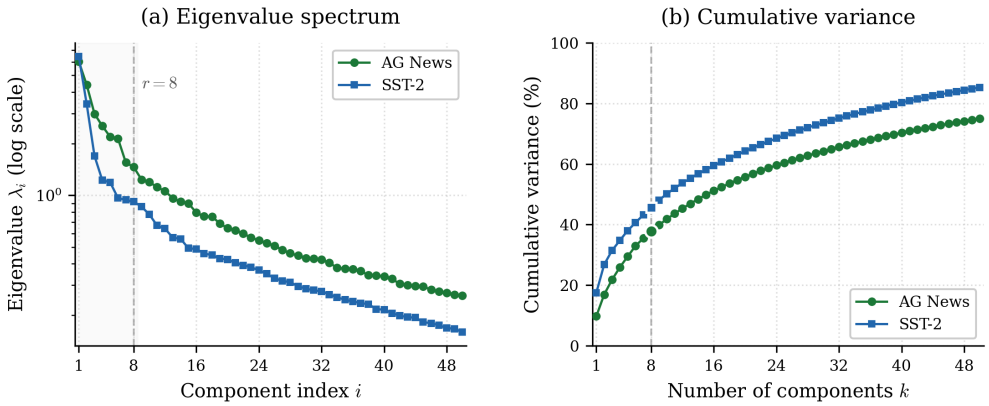

Figure 4: Eigenvalue spectrum and cumulative variance of frozen BERT-base `[CLS]` activations at $N = 64$. The top $r = 8$ components capture 38% of the total variance for AG News and 46% for SST-2. The spectra decay continuously, with no sharp finite-rank cutoff.

The top eight components use approximately one percent of the 768-dimensional representation space while capturing 38–46% of total variance. Because $N = 64$, the sample covariance has rank at most 63, so this share is measured over at most 63 nonzero sample eigenvalues. This indicates substantial low-dimensional concentration without an exact finite-rank cutoff, so we use the spiked model and the perturbation calculation below as regime-level guidance.

B.2 Derivation of Equation (2)

Let $X_1, \ldots, X_N \in \mathbb{R}^d$ be independent, centered, sub-Gaussian activation vectors with covariance Σ. Let $\widehat{\Sigma}$ be their sample covariance, let V^* contain the leading r^* population eigenvectors, and let $\widehat{V}$ contain the corresponding sample eigenvectors. We take the adapter rank equal to the number of spikes, $r = r^*$. For $r \neq r^*$ the gap is $\lambda_r - \lambda_{r+1}$. Under the spiked model, the eigengap is

$$\delta = \lambda_{r^*} - \sigma^2 > 0.$$

An operator-norm Davis–Kahan bound gives

$$\left\| \sin \Theta(\widehat{V}, V^*) \right\|_{\mathrm{op}} \lesssim \frac{\|\widehat{\Sigma} - \Sigma\|_{\mathrm{op}}}{\delta}.$$

Effective-rank covariance concentration gives

$$\mathbb{E}\left[\|\widehat{\Sigma} - \Sigma\|_{\mathrm{op}} \right] \lesssim \|\Sigma\|_{\mathrm{op}} \left(\sqrt{\frac{d_{\mathrm{eff}}}{N}} + \frac{d_{\mathrm{eff}}}{N} \right),$$

where

$$d_{\mathrm{eff}} = \frac{\mathrm{Tr}(\Sigma)}{\|\Sigma\|_{\mathrm{op}}}.$$

For a principal-angle matrix of rank at most r^*,

$$\|\sin \Theta\|_F \leq \sqrt{r^*}\, \|\sin \Theta\|_{\mathrm{op}}, \qquad \|\sin \Theta\|_F \leq \sqrt{r^*}.$$

Combining these inequalities yields

$$\mathbb{E}\left[\|\sin \Theta(\widehat{V}, V^*)\|_F \right] \lesssim \min\left\{ \sqrt{r^*}, \sqrt{r^*}\, \frac{\|\Sigma\|_{\mathrm{op}}}{\delta} \left(\sqrt{\frac{d_{\mathrm{eff}}}{N}} + \frac{d_{\mathrm{eff}}}{N} \right) \right\},$$

which is Equation (2).

The bound provides stronger control when N is large relative to d_{eff} and when the eigengap δ is large. If either condition deteriorates, the bound becomes weaker, which is consistent with greater finite-sample sensitivity of the estimated subspace. Transformer activations need not be independent or exactly sub-Gaussian, and their covariance need not follow a literal spiked model, so the calculation is not a finite-sample theorem for transformer fine-tuning.

The PCA covariance uses all N examples. By contrast, the alignment diagnostic depends on class means estimated from approximately N/C examples per class. We therefore express PCA estimation through d_{eff}/N while recognizing the additional uncertainty in class-conditional quantities.

B.3 First-Step LoRA Signal and Task Alignment

This section derives the local first-step relationship used in Section 3. Consider

$$\Delta z = s\, BU^\top h(x), \qquad s = \alpha/r,$$

where $U \in \mathbb{R}^{d\times r}$ has orthonormal columns and $B \in \mathbb{R}^{m\times r}$. This corresponds to $A = U^\top$.

Define

$$q(x, y) = \nabla_{\Delta z}\ell(z_0(x) + \Delta z, y)|_{\Delta z=0}$$

and

$$\mathcal{L}(B; U) = \mathbb{E}_{(x,y)}\left[\ell\big(z_0(x) + sBU^\top h(x), y\big)\right].$$

Proposition 1 (First-step signal in a fixed LoRA subspace)**.** *Assume that $\mathcal{L}(B; U)$ is twice differentiable near $B = 0$ with a locally bounded Hessian. Let*

$$G_U = \nabla_B\mathcal{L}(0; U).$$

A gradient step $B_1 = -\eta G_U$ from $B_0 = 0$ satisfies

$$\mathcal{L}(B_1; U) - \mathcal{L}(0; U) = -\eta\|G_U\|_F^2 + O\big(\eta^2\|G_U\|_F^2\big).$$

Suppose further that, conditional on class $y = c$,

$$h(x) = \mu_c + \varepsilon, \qquad q(x, y) = g_c + \xi,$$

where

$$\mathbb{E}[\varepsilon \mid y = c] = 0, \qquad \mathbb{E}[\xi \mid y = c] = 0,$$

and the class-conditional cross-covariance vanishes:

$$\mathbb{E}\left[\xi\varepsilon^\top \mid y = c\right] = 0.$$

Let the class priors be π_c, define

$$\bar{\mu} = \sum_{c=1}^{C} \pi_c\mu_c,$$

and assume the class-gradient codes are centered:

$$\sum_{c=1}^{C} \pi_c g_c = 0.$$

Then

$$\|G_U\|_F^2 = s^2 \operatorname{Tr}\left(U^\top H_{\text{grad}} U\right),$$

where

$$H_{\text{grad}} = \sum_{c,c'=1}^{C} \pi_c \pi_{c'} \langle g_c, g_{c'} \rangle (\mu_c - \bar{\mu})(\mu_{c'} - \bar{\mu})^\top.$$

If the gradient-weighted class geometry satisfies

$$H_{\text{grad}} = \gamma \Sigma_{\text{task}}$$

for some task-dependent $\gamma > 0$, then

$$\|G_U\|_F^2 = s^2 \gamma \operatorname{Tr}\left(U^\top \Sigma_{\text{task}} U\right).$$

Proof. Differentiating through the adapter at $B = 0$ gives

$$G_U = s\, \mathbb{E}\left[q(x,y)(U^\top h(x))^\top\right].$$

Conditioning on $y = c$ and expanding $q = g_c + \xi$ and $h = \mu_c + \varepsilon$ gives

$$\mathbb{E}[qh^\top \mid y = c] = g_c \mu_c^\top,$$

because the conditional residual means and cross-covariance vanish. Hence

$$G_U = s \sum_{c=1}^{C} \pi_c g_c (U^\top \mu_c)^\top.$$

The centering condition on the class-gradient codes permits subtraction of the global representation mean:

$$G_U = s \sum_{c=1}^{C} \pi_c g_c \left(U^\top (\mu_c - \bar{\mu})\right)^\top.$$

Expanding the squared Frobenius norm and rewriting it as a trace yields

$$\|G_U\|_F^2 = s^2 \operatorname{Tr}\left(U^\top H_{\text{grad}} U\right).$$

The special case $H_{\text{grad}} = \gamma \Sigma_{\text{task}}$ gives the final equality.

For the loss-decrease statement, a Taylor expansion at $B = 0$ gives

$$\mathcal{L}(-\eta G_U; U) = \mathcal{L}(0; U) - \eta \|G_U\|_F^2 + O\left(\eta^2 \|G_U\|_F^2\right).$$

□

Interpretation. The most general quantity in the proposition is H_{grad}, not the unweighted between-class scatter. The reduction to Σ_{task} requires the stated proportionality assumption. The proposition is therefore a local explanatory calculation and not a convergence theorem. It takes $A = U^\top$. For the implemented initialization $A = cR^\top Q^\top$ of Eq. (7), the rotation R cancels and $\|G\|_F^2$ gains a factor c^2, so its dependence on $\operatorname{Tr}(Q^\top \Sigma_{\text{task}} Q)$ is unchanged. The proposition uses population priors π_c, and the empirical Σ_{task} of Section 3 replaces them with n_c/N.

If U is sampled uniformly from the Grassmannian of rank-r subspaces and is independent of a fixed positive-semidefinite task matrix, then

$$\mathbb{E}_U\left[\frac{\operatorname{Tr}(U^\top \Sigma_{\text{task}} U)}{\operatorname{Tr}(\Sigma_{\text{task}})}\right] = \frac{r}{d}.$$

For BERT-base with $r = 8$ and $d = 768$, this expectation is approximately 0.010. The largest measured target-token alignment is approximately $\rho = 0.84$, as reported in Appendix E.7.

B.4 Output Preservation Under the Orthogonal Pivot

Proposition 2 (Exact preservation under orthogonal reparameterization)**.** *Let $A \in \mathbb{R}^{r\times d}$ have full row rank and let $B \in \mathbb{R}^{m\times r}$. Let $A^{new} \in \mathbb{R}^{r\times d}$ be an orthonormal row basis for row(A), so that*

$$A^{new}(A^{new})^\top = I_r$$

and

$$row(A^{new}) = row(A).$$

Define

$$B^{new} = BA(A^{new})^\top.$$

Then

$$B^{new}A^{new} = BA.$$

Consequently, the LoRA update $\Delta W = (\alpha/r)BA$ is preserved exactly in exact arithmetic.

Proof. The matrix

$$P = (A^{\mathrm{new}})^\top A^{\mathrm{new}}$$

is the orthogonal projector onto $\mathrm{row}(A)$. Every row of A lies in that subspace, so $AP = A$. Therefore

$$B^{\mathrm{new}}A^{\mathrm{new}} = BA(A^{\mathrm{new}})^\top A^{\mathrm{new}} = BAP = BA.$$

□

Because the construction depends only on the current full-row-rank adapter factors, the pivot can follow any initializer, whether PCA-, weight-, activation-, or gradient-based. In finite precision, the equality is affected by QR and matrix-multiplication error. Appendix F.6 reports these errors for the classification and reasoning runs.

For $G = \nabla_{\Delta W}\mathcal{L}$ and, up to the factor α/r, $\Delta W = BA$,

$$\nabla_B\mathcal{L} = GA^\top, \qquad \nabla_A\mathcal{L} = B^\top G.$$

The B-gradient geometry therefore depends on $AA^\top$. The pivot replaces the current row coordinates with an orthonormal basis, making $AA^\top = I_r$ while preserving BA. Adam and finite-step SGD are not generally invariant to this factor reparameterization, so subsequent optimization can change even though the adapter output is unchanged at the pivot.

C Experimental Details

Code, configurations, sampled-example manifests, result summaries, selected run artifacts, and available prediction artifacts are released at `https://github.com/Nischal1011/pivotlora`.

Software and hardware. The BERT-base and GPT-2 experiments were run on a single NVIDIA T4 with 16 GB of memory, while decoder-scale classification used a single NVIDIA L4 with 24 GB. Generated-answer reasoning used one NVIDIA GeForce RTX 5090 with 31.8 GB of memory, PyTorch `2.9.0+cu128`, and CUDA 12.8, with the base model in bf16 and the adapter factors in float32.

Reproducibility. The classification implementation enables deterministic PyTorch and cuDNN settings. Main classification results use seeds 42, 43, and 44 with approximately class-balanced sampling unless otherwise noted. Deterministic configuration reduces avoidable variation but does not guarantee bitwise identity across different hardware, driver, and library versions.

Asset licenses. BERT-base is released under Apache-2.0, GPT-2 under a modified MIT license, Llama-2-7B under the Llama 2 Community License, and Llama-3.1-8B under the Llama 3.1 Community License; we accessed the Llama checkpoints after accepting these terms. GSM8K and StrategyQA are released under MIT, BoolQ under CC BY-SA 3.0, QNLI (derived from SQuAD [Rajpurkar et al., 2016]) under CC BY-SA 4.0, MNLI [Williams et al., 2018] under the OANC terms with CC BY-SA portions, and CNN/DailyMail under Apache-2.0 as distributed on Hugging Face. SST-2 [Socher et al., 2013], CoLA [Warstadt et al., 2019], RTE [Dagan et al., 2006], AG News, and XSum are distributed for research use without an explicit license. All assets are used only for research evaluation, and no model weights or dataset contents are redistributed with our code.

Adapter targets. For BERT-base, we adapt `attention.self.query` and `attention.self.value`, giving 24 adapters. For Llama-2-7B and Llama-3.1-8B classification, we adapt `self_attn.q_proj` and `self_attn.v_proj`, giving 64 adapters. For GPT-2, we adapt `attn.c_proj` and `mlp.c_fc`.

BERT classification details. The task-specific linear classification head is trained jointly with the adapter parameters. Its initialization is matched across methods and seeds. The reported adapter parameter count excludes the classification head. Few-shot examples are sampled from the training split. Evaluation uses the validation split when available, the matched validation split for MNLI, and the test split for AG News in the dataset distribution used by our implementation.

Decoder-scale classification. Decoder-scale experiments use sequence-classification models with a task-specific head trained jointly with the adapters. Sentence-pair fields are concatenated before tokenization. The activation source is the mean of each adapter's inputs over all positions of the padded input sequence, computed in one frozen forward pass. EVA provides an activation-based comparison, although it differs from PivotLoRA in components beyond activation pooling.

Few-shot sampling convention. The BERT columns labeled 16, 32, 64, 128, and 256 denote the total number of labeled examples N, sampled approximately class-balanced. Thus, a C-class task contains approximately N/C examples per class. The PCA covariance is estimated from all N examples.

Baseline configurations. PiSSA [Meng et al., 2024] uses the top-r SVD of W_0 with a frozen residual and $\alpha_{\mathrm{LoRA}} = r$. LoRAM [Zhang et al., 2025b] estimates $\sigma_{\max}$ with 10 power iterations, constructs a fixed-seed random orthogonal basis, and scales it by $\log(1 + \sigma_{\max})$. LoRA-GA [Wang et al., 2024] uses an SVD of a one-step gradient with residual subtraction. LoRA-One [Zhang et al., 2025a] uses balanced SVD scaling without its optional Scaled-Adam preconditioner. EVA [Paischer et al., 2025] uses all-token activation SVD. We disable adaptive rank redistribution and set $\rho_{\mathrm{EVA}} = 1.0$, so each adapted layer uses rank $r = 8$. LoRA+ [Hayou et al., 2024] uses Kaiming initialization and $\eta_B = 16\eta_A$. DoRA [Liu et al., 2024] uses its magnitude-direction parameterization. AdaLoRA [Zhang et al., 2023] uses SVD triplets with rank pruning. SRLoRA [Yang et al., 2025] fuses rank-one pairs every three epochs. The LoRA-DA implementation uses diagonal Fisher estimation with residual subtraction.

Baseline fidelity. In our classification implementation, every supplied initialization of A, including those of PiSSA, LoRA-GA, LoRA-One, LoRAM, and EVA, passes through the same LoRA layer, which multiplies it by $\sqrt{5/d}$ (Eq. (7)). Because this rescales the baselines' intended initialization magnitudes, we audited the two baselines for which magnitude is most central at $N = 64$ with three seeds (Table 5). The reference PEFT implementation of PiSSA matches our implementation to within 0.012 on all three tasks. Restoring LoRAM's intended $\log(1 + \sigma_{\max})$ magnitude changes its score by $+0.019$, -0.017, and -0.035, and it remains below vanilla LoRA on every task. The rescaling therefore does not change the baseline conclusions in Table 1.

Table 5: Baseline-fidelity audit at $N = 64$, mean over three seeds. Each block is a separate matched run; compare within blocks. CoLA uses Matthews correlation.

Method	AG News	SST-2	CoLA
LoRA	.634	.633	.183
PiSSA (our implementation)	.477	.552	.099
PiSSA (reference PEFT)	.472	.548	.087
LoRA	.636	.639	.201
LoRAM (our implementation)	.476	.597	.141
LoRAM (intended magnitude)	.495	.580	.106

Matched BERT protocol. The main BERT comparison uses rank $r = 8$, AdamW, 15 epochs, batch size 16, query and value adapters, and a shared base learning-rate anchor of $\eta_A = 2 \times 10^{-4}$. Unless a baseline's standard parameterization requires otherwise, $\alpha_{\mathrm{LoRA}} = 16$. The learning-rate sweeps in Appendix F.9 test sensitivity to this anchor.

C.1 Classification Learning-Rate Allocation

Here, η_A denotes the base learning-rate anchor. Methods with shared factor-wise rates apply this rate directly to both LoRA factors, whereas methods with differential rates derive their B-factor rates from it. At initialization, TT-PCA and PivotLoRA assign the rank-i coordinate of B a learning rate proportional to the corresponding PCA eigenvalue:

$$\eta_{B,i} = \frac{\lambda_i}{\bar{\lambda}} \lambda_{\mathrm{ratio}} \eta_A, \qquad \bar{\lambda} = \frac{1}{r} \sum_{j=1}^{r} \lambda_j, \qquad \lambda_{\mathrm{ratio}} = 16. \tag{12}$$

Normalizing by $\bar{\lambda}$ makes the average B-to-A learning-rate ratio equal to 16. This allocation is an optimizer heuristic and is not part of the activation-subspace definition. The controls in Appendix F.10 separate the selected directions from the eigenvalue-to-coordinate rate assignment.

TT-PCA uses these rates throughout training. PivotLoRA uses them except for the 10 optimizer steps after the pivot, during which a common B rate rises from η_A to $16\eta_A$ as a precaution, because the pivot resets the Adam state of A while the state of B still refers to the previous coordinates. Appendix F.8 shows that this precaution is not needed. Post-pivot learning-rate policies differ by at most 0.011 in average score, and the pivot with no learning-rate change performs as well. The reasoning experiments instead use a single learning rate of 2×10^{-4} for both factors, with no differential or eigenvalue-scaled rates and no post-pivot change.

C.2 TT-LDA Ablation

TT-LDA uses the same prediction-position representations as TT-PCA but constructs supervised linear-discriminant directions. Because a C-class problem supplies at most $C - 1$ discriminant directions, residual PCA directions fill the unused coordinates when $C - 1 < r$. The combined basis is orthogonalized to rank $r = 8$. TT-LDA uses the uniform rate $\eta_B = 16\eta_A$, with no eigenvalue scaling across ranks. We use TT-LDA as an ablation of the activation statistic, not as the default method. Comparative results appear in Appendix F.9.

C.3 Classification Pivot Configuration

The primary classification configuration uses pivot epoch $K = 3$. At the pivot, reduced QR is applied to $A^\top$, and B is re-encoded according to Eq. (10). Adam first- and second-moment states for A are reset. The state for B is retained, and its learning rate is handled as described in Appendix C.1. The reset policy is an implementation choice and is separate from the algebraic output-preservation guarantee.

Algorithm 1 Classification PivotLoRA

Require: Frozen model $\mathcal{M}$, labeled data $\mathcal{D}$, rank r, pivot epoch K
Collect prediction-position activations X_ℓ at each adapted layer
for each adapted layer ℓ **do**
    Center X_ℓ and compute its top-r PCA basis
    Initialize $\mathrm{row}(A_\ell)$ from the PCA basis
    Set $B_\ell \leftarrow 0$
    Assign the B_ℓ coordinate rates using Eq. (12)
end for
Train through epoch K
for each adapted layer ℓ **do**
    $A_{\mathrm{old}} \leftarrow A_\ell$ and $B_{\mathrm{old}} \leftarrow B_\ell$
    $Q, R \leftarrow \mathrm{QR}(A_{\mathrm{old}}^\top)$
    $A_{\mathrm{new}} \leftarrow Q^\top$
    $B_{\mathrm{new}} \leftarrow B_{\mathrm{old}} A_{\mathrm{old}} A_{\mathrm{new}}^\top$
    Replace (A_ℓ, B_ℓ) by $(A_{\mathrm{new}}, B_{\mathrm{new}})$
    Reset the Adam state for A_ℓ
end for
Ramp the B rate from η_A to $16\eta_A$ over 10 steps, then resume Eq. (12)
Train for the remaining epochs

GPT-2 summarization configuration. The loss is applied only to the supervised summary, and the tied language-modeling head remains frozen. We cross two activation sources with two learning-rate recipes. The sources are the mean over all positions of the padded sequence and the mean over response positions (those predicting supervised summary tokens). The recipes are the classification recipe of Eq. (12) with $\lambda_{\mathrm{ratio}} = 16$ and a single rate of 2×10^{-4} for both factors, as in the reasoning experiments (Appendix E.4).

D Generated-Answer Reasoning Details

We evaluate response-position adaptation on GSM8K and StrategyQA. LoRA and PivotLoRA use matched data, training, and evaluation protocols, and PivotLoRA additionally uses activation-based initialization and one output-preserving pivot. Exact commands and lower-level implementation details are provided with the released code.

StrategyQA was selected with a screening step fixed in advance that ran only the base model and LoRA. A candidate dataset was kept if LoRA was clearly above chance, followed the trained answer format in at least 90% of generations, and rarely hit the generation limit. No PivotLoRA run was made on a candidate before it passed. We also considered MATH-500 but stopped at this stage. Across three pipeline checks, each method solved between 1 and 10 of 50 problems in our 128-example fine-tuning setting, and fewer than 80% of trained-model generations followed the boxed-answer format, so a comparison would have been dominated by noise and format failures.

D.1 Data and Response-Position Activations

GSM8K contains 7,473 training examples and 1,319 test examples, and StrategyQA (the `ChilleD/StrategyQA` split) contains 1,603 training examples and 687 test examples. For StrategyQA, the one training question that also appears in the test split is removed from the training pool. For each dataset, we sample $N = 128$ training examples without replacement using seeds $\{42, 43, 44\}$. LoRA and PivotLoRA use the same subset at each seed.

The GSM8K prompt is

```
Question: {question}
Answer:
```

with a single space followed by the dataset's original answer field as the completion. StrategyQA uses the GSM8K prompt with completion

```
{facts}
```

```
#### {yes|no}
```

where `facts` is the dataset's supporting-facts field.

Each prompt ends at `Answer:` and each completion begins with a space, so the prompt used at inference is an exact token-level prefix of every training sequence, which the implementation asserts for every training example. Generation therefore starts from the same token boundary seen in training.

Prompt tokens are excluded from the language-modeling loss, whereas completion tokens and the appended EOS token are supervised. Sequences are jointly tokenized and truncated to a maximum length of 512 tokens. Because hidden state t predicts token $t+1$, the response-predictor positions for example n are

$$\mathcal{P}_n = \{t : M_{n,t+1} = 1\},$$

where M denotes the completion mask. At adapted layer ℓ, the activation used for PCA is

$$x_{\ell,n} = \frac{1}{|\mathcal{P}_n|} \sum_{t \in \mathcal{P}_n} h_{\ell,n,t}.$$

The resulting example-level activation matrix is centered before PCA. Averaging within each example prevents response length from determining its weight in the covariance estimate.

D.2 Training and Pivot Configuration

Table 6: Configuration for the reasoning experiments, shared by both datasets.

Field	Value
Checkpoint	`meta-llama/Llama-3.1-8B`
Target modules	`q_proj`, `v_proj`
Adapted modules	64
Rank	$r = 8$
LoRA scaling	$\alpha_{\mathrm{LoRA}} = 16$
Optimizer	AdamW
Learning rate	2×10^{-4}
Weight decay	0.01
Schedule	Linear decay, no warm-up
Epochs	15
Per-device batch size	4
Gradient accumulation	2
Effective batch size	8
Optimizer steps	240
Maximum sequence length	512
Base-model precision	bf16
Adapter precision	float32
Pivot epoch	3
Pivot step	48

The PCA basis is orthogonalized and scaled by $\sqrt{5/d}$, with $d = 4096$, before assignment to the LoRA input factor A, with no random rotation. PCA is computed in float32. PivotLoRA performs one pivot at epoch 3, corresponding to global step 48, using float64 QR, and the transformed factors are stored in float32. All 64 query and value adapters are transformed.

Immediately after the pivot, the AdamW states for both A and B are reset, while the learning-rate schedule continues from the current global step. Resetting both states differs from the classification policy (Appendix C.3).

The $\sqrt{5/d}$ basis scale gives A about $17\times$ smaller Frobenius norm than vanilla Kaiming-uniform initialization, so it is not norm-matched. These experiments therefore evaluate the complete initialization-and-pivot procedure, not direction in isolation.

D.3 Generation and Evaluation

Evaluation uses greedy decoding with batch size 1 and at most 512 new tokens, terminating earlier at EOS. The model uses EOS for padding and `sdpa` attention.

The evaluator first extracts a numeric value following the final `####` delimiter. If none is present, it uses the final numeric value in the generated response. Predictions and references are normalized as numeric strings and compared by exact equality. Signed integers, comma-grouped numbers, and decimals are supported; fractions and percentages are not converted. Outputs without a valid numeric value are counted as incorrect. For StrategyQA, the evaluator instead extracts the first standalone `yes` or `no` after the final `####`, falling back to the last standalone `yes` or `no` in the response, and compares it with the label.

We additionally report format compliance, the fraction of generations that follow the trained answer format, which ends in `#### <number>` for GSM8K and in `#### yes` or `#### no` for StrategyQA. As a reference, the frozen base model without adapters is evaluated once with the same prompt, decoding, and scorer.

D.4 Per-Seed Results

Table 7: GSM8K results. Accuracy is numeric exact match on the 1,319-example test set. Format is the percentage of generations following the trained answer format. The base model has no adapter and is evaluated once.

Method	Seed	Correct	Accuracy (%)	Parse rate (%)	Format (%)
Base model	–	338	25.63	100.00	0.0
LoRA	42	376	28.51	100.00	98.0
LoRA	43	381	28.89	100.00	96.0
LoRA	44	378	28.66	100.00	95.8
PivotLoRA	42	421	31.92	100.00	98.2
PivotLoRA	43	476	36.09	100.00	97.6
PivotLoRA	44	418	31.69	100.00	97.4

PivotLoRA improves on LoRA at every seed, by $+3.41$, $+7.20$, and $+3.03$ points (mean $+4.55 \pm 2.31$, sample SD over seeds), and even the smallest gain exceeds LoRA's own seed-to-seed spread (± 0.19). Pooled over the three seeds, PivotLoRA alone answers 655 test items correctly against 475 for LoRA alone (exact McNemar $p = 9.5 \times 10^{-8}$), and the test is significant at each seed ($p \leq 0.036$).

Table 8: StrategyQA results. Accuracy is yes/no exact match on the 687-example test split. Columns as in Table 7.

Method	Seed	Correct	Accuracy (%)	Parse rate (%)	Format (%)
Base model	–	374	54.44	85.15	0.0
LoRA	42	439	63.90	98.11	95.2
LoRA	43	457	66.52	99.85	99.4
LoRA	44	426	62.01	99.42	96.7
PivotLoRA	42	476	69.29	99.85	99.6
PivotLoRA	43	457	66.52	99.71	99.4
PivotLoRA	44	463	67.39	99.85	98.0

PivotLoRA improves on LoRA by $+5.39$ points at two seeds and ties at the third (mean $+3.59 \pm 3.11$). The tie does not reflect identical outputs, since each method alone answers 98 questions correctly. Pooled over the three seeds, PivotLoRA alone answers 350 test items correctly against 276 for LoRA alone (exact McNemar $p = 0.004$).

E Extended Experimental Results

E.1 Complete BERT-Base Results

Tables 9–15 report mean and sample standard deviation over three seeds. The six primary tasks contain 30 task–N settings. BoolQ is an auxiliary classification task used in the alignment analysis. All methods follow the shared benchmark protocol together with the method-specific configurations documented in Appendix C.

Table 9: SST-2 accuracy, reported as mean ± standard deviation over three seeds. Columns denote the total number of labeled examples N.

Method	16	32	64	128	256
LoRA	.524±.011	.546±.037	.652±.014	.772±.006	.844±.012
DoRA	.528±.014	.544±.036	.647±.018	.761±.010	.841±.008
AdaLoRA	.502±.019	.500±.008	.548±.016	.580±.026	.633±.019
PiSSA	.503±.023	.500±.008	.548±.027	.665±.058	.823±.002
LoRA-GA	.503±.018	.500±.007	.565±.033	.664±.077	.849±.004
LoRA-One	.497±.015	.501±.007	.565±.032	.711±.059	.854±.010
LoRA+	.551±.033	.569±.042	.538±.038	.509±.000	.509±.000
EVA	.503±.017	.500±.007	.561±.031	.683±.032	.855±.010
LoRAM	.498±.015	.500±.007	.568±.030	.702±.060	.849±.005
SRLoRA	.524±.011	.536±.036	.632±.014	.750±.014	.843±.011
TT-PCA	.520±.024	.530±.011	.755±.007	.817±.003	.862±.007
TT-LDA	.523±.020	.542±.014	.783±.007	.828±.014	.859±.011
PivotLoRA	.543±.038	.633±.018	.795±.016	.823±.017	.850±.010

Table 10: CoLA Matthews correlation, reported as mean ± standard deviation over three seeds.

Method	16	32	64	128	256
LoRA	.081±.056	.131±.047	.163±.046	.264±.072	.407±.010
DoRA	.085±.060	.128±.041	.171±.042	.257±.070	.393±.007
AdaLoRA	.096±.055	.081±.040	.100±.029	.152±.033	.183±.023
PiSSA	.090±.064	.111±.040	.111±.083	.206±.059	.338±.022
LoRA-GA	.086±.061	.097±.045	.103±.078	.214±.059	.382±.003
LoRA-One	.087±.060	.089±.039	.111±.062	.240±.067	.387±.007
LoRA+	.071±.023	.105±.041	.038±.012	.000±.000	.000±.000
EVA	.089±.061	.095±.036	.101±.070	.243±.058	.387±.017
LoRAM	.089±.061	.085±.036	.112±.070	.239±.058	.390±.017
SRLoRA	.081±.058	.129±.030	.161±.051	.240±.065	.405±.008
TT-PCA	.103±.044	.137±.046	.274±.094	.381±.049	.397±.019
TT-LDA	.106±.036	.150±.041	.281±.100	.377±.071	.412±.016
PivotLoRA	.124±.015	.171±.049	.288±.098	.365±.071	.411±.025

Table 11: MNLI matched-validation accuracy, reported as mean ± standard deviation over three seeds.

Method	16	32	64	128	256
LoRA	.350±.008	.356±.006	.366±.003	.356±.008	.392±.011
DoRA	.349±.008	.355±.006	.366±.002	.359±.008	.392±.012
AdaLoRA	.350±.004	.354±.004	.347±.013	.352±.009	.356±.013
PiSSA	.352±.003	.354±.002	.349±.014	.354±.009	.383±.023
LoRA-GA	.350±.003	.356±.003	.348±.013	.351±.009	.397±.014
LoRA-One	.350±.004	.355±.004	.348±.013	.357±.010	.401±.012
LoRA+	.353±.012	.355±.001	.352±.019	.355±.000	.355±.000
EVA	.350±.004	.356±.003	.347±.014	.352±.008	.397±.015
LoRAM	.350±.004	.356±.003	.349±.014	.356±.008	.404±.015
SRLoRA	.349±.007	.356±.005	.367±.003	.357±.007	.390±.009
TT-PCA	.349±.005	.359±.007	.367±.003	.378±.011	.424±.025
TT-LDA	.352±.007	.357±.005	.366±.003	.379±.007	.425±.024
PivotLoRA	.356±.010	.362±.008	.367±.005	.389±.009	.454±.041

Table 12: QNLI accuracy, reported as mean ± standard deviation over three seeds.

Method	16	32	64	128	256
LoRA	.519±.037	.521±.040	.555±.042	.626±.061	.662±.040
DoRA	.517±.039	.519±.040	.556±.042	.622±.055	.663±.035
AdaLoRA	.513±.029	.514±.029	.527±.023	.537±.029	.546±.024
PiSSA	.510±.027	.512±.041	.528±.021	.574±.035	.616±.038
LoRA-GA	.512±.027	.514±.035	.528±.024	.567±.045	.641±.043
LoRA-One	.511±.028	.516±.031	.529±.023	.574±.049	.644±.045
LoRA+	.539±.033	.576±.052	.554±.040	.515±.012	.505±.000
EVA	.511±.028	.516±.031	.532±.024	.575±.052	.645±.050
LoRAM	.512±.028	.516±.031	.530±.024	.567±.052	.644±.050
SRLoRA	.512±.038	.519±.039	.551±.040	.618±.059	.662±.034
TT-PCA	.517±.034	.525±.031	.574±.059	.640±.057	.677±.024
TT-LDA	.517±.035	.528±.033	.572±.048	.638±.055	.681±.029
PivotLoRA	.533±.026	.547±.036	.594±.049	.645±.050	.691±.013

Table 13: RTE accuracy, reported as mean ± standard deviation over three seeds.

Method	16	32	64	128	256
LoRA	.508±.040	.549±.036	.551±.022	.569±.015	.549±.003
DoRA	.511±.040	.546±.024	.555±.015	.568±.018	.540±.007
AdaLoRA	.510±.035	.526±.011	.551±.022	.540±.012	.524±.014
PiSSA	.505±.033	.528±.032	.539±.019	.555±.025	.528±.015
LoRA-GA	.516±.036	.525±.018	.552±.016	.554±.015	.527±.014
LoRA-One	.515±.037	.528±.017	.561±.021	.561±.007	.540±.013
LoRA+	.545±.028	.552±.011	.527±.000	.527±.000	.527±.000
EVA	.513±.037	.534±.024	.555±.017	.563±.013	.539±.024
LoRAM	.513±.037	.530±.024	.563±.017	.566±.013	.551±.024
SRLoRA	.513±.046	.552±.031	.539±.021	.567±.018	.549±.009
TT-PCA	.522±.051	.563±.033	.555±.025	.561±.022	.552±.005
TT-LDA	.516±.055	.567±.023	.558±.020	.561±.028	.555±.007
PivotLoRA	.551±.019	.584±.017	.569±.025	.549±.010	.563±.012

Table 14: AG News accuracy, reported as mean ± standard deviation over three seeds.

Method	16	32	64	128	256
LoRA	.431±.016	.466±.050	.640±.033	.807±.006	.855±.006
DoRA	.432±.015	.446±.041	.619±.016	.801±.010	.857±.010
AdaLoRA	.304±.027	.308±.016	.368±.038	.438±.031	.504±.028
PiSSA	.338±.031	.364±.007	.479±.024	.643±.036	.795±.024
LoRA-GA	.330±.022	.354±.012	.466±.038	.679±.042	.812±.030
LoRA-One	.314±.025	.338±.012	.475±.033	.696±.055	.808±.032
LoRA+	.695±.011	.730±.044	.619±.136	.646±.281	.594±.123
EVA	.317±.024	.345±.012	.481±.040	.702±.070	.815±.029
LoRAM	.311±.024	.333±.012	.482±.040	.672±.070	.809±.029
SRLoRA	.426±.021	.446±.045	.584±.015	.795±.013	.851±.004
TT-PCA	.478±.028	.558±.012	.753±.041	.858±.005	.874±.006
TT-LDA	.475±.026	.587±.015	.769±.027	.861±.009	.871±.008
PivotLoRA	.600±.049	.705±.047	.834±.018	.866±.004	.878±.001

Table 15: BoolQ accuracy, reported as mean ± standard deviation over three seeds.

Method	16	32	64	128	256
LoRA	.602±.005	.603±.010	.551±.048	.567±.039	.601±.019
DoRA	.601±.005	.606±.007	.553±.043	.573±.032	.596±.022
AdaLoRA	.593±.018	.603±.012	.549±.030	.542±.024	.576±.017
PiSSA	.597±.015	.603±.012	.549±.029	.563±.030	.591±.035
LoRA-GA	.596±.015	.602±.011	.565±.006	.562±.030	.584±.017
LoRA-One	.593±.018	.602±.011	.538±.032	.549±.010	.580±.017
LoRA+	.591±.042	.622±.000	.621±.001	.622±.000	.622±.000
EVA	.594±.017	.603±.012	.537±.033	.538±.014	.579±.020
LoRAM	.593±.018	.603±.012	.545±.029	.549±.009	.576±.017
SRLoRA	.602±.005	.603±.010	.549±.050	.566±.039	.599±.021
TT-PCA	.596±.016	.601±.014	.568±.014	.574±.006	.612±.010
TT-LDA	.596±.016	.602±.013	.569±.016	.567±.009	.612±.005
PivotLoRA	.596±.016	.601±.014	.567±.029	.589±.011	.611±.010

E.2 Full Aggregate Rankings

Table 16 aggregates the 30 primary BERT task–N settings. PivotLoRA has average rank 1.83 across all 13 methods and has the strictly highest mean in 19 of 30 settings, counting TT-PCA and TT-LDA, and ties for highest on MNLI at $N = 64$. It exceeds vanilla LoRA in 29 of the 30 settings.

Table 16: Aggregate statistics across the 30 primary BERT-base few-shot settings. "Wins vs. PivotLoRA" counts settings in which the listed method has a strictly higher mean than PivotLoRA.

Method	Average rank (↓)	Wins vs. PivotLoRA
PivotLoRA (Ours)	**1.83**	N/A
TT-LDA	3.13	5/30
TT-PCA	3.77	3/30
LoRA	5.23	1/30
DoRA	5.83	1/30
SRLoRA	6.93	1/30
LoRA+	8.43	5/30
LoRAM	8.43	1/30
LoRA-One	8.47	2/30
EVA	8.47	2/30
LoRA-GA	9.30	1/30
PiSSA	9.87	1/30
AdaLoRA	11.30	0/30

E.3 Decoder-Scale Classification Results

The decoder-scale experiments test whether the activation-subspace pattern extends beyond BERT-base. Tables 17 and 18 report SST-2 and AG News, while Table 19 reports the BoolQ extension. PivotLoRA ranks first at $N = 64$ on SST-2 and AG News for both backbones. At $N = 256$, the differences are smaller and PivotLoRA is not uniformly best.

EVA is included as an activation-based baseline under the same per-layer rank budget. It treats every token activation as a separate sample, whereas PivotLoRA uses one sequence-mean vector per example.

Table 17: Llama-2-7B few-shot classification, reported as mean ± standard deviation over three seeds. Bold marks the best mean among methods evaluated in the corresponding setting.

Method	SST-2 64	SST-2 256	AG News 64	AG News 256
LoRA	.708±.051	.924±.007	.748±.027	**.879**±.004
EVA	.713±.015	.928±.016	.745±.027	**.879**±.004
PiSSA	.635±.058	.922±.010	.700±.014	.865±.010
LoRAM	.763±.025	**.929**±.014	.742±.027	.876±.003
PivotLoRA	**.769**±.030	.912±.020	**.784**±.029	.861±.020

Table 18: Llama-3.1-8B few-shot classification, reported as mean ± sample standard deviation over three seeds. EVA entries at $N = 64$ report population standard deviations because per-seed values were not retained. Bold marks the best mean among methods evaluated in the corresponding setting.

Method	SST-2 64	SST-2 256	AG News 64	AG News 256
LoRA	.704±.037	.883±.030	.759±.023	**.880**±.009
EVA	.643±.050	**.909**±.017	.743±.034	.876±.009
PiSSA	.622±.043	.876±.030	.715±.027	.864±.010
LoRAM	.662±.050	.901±.012	.748±.047	.872±.010
PivotLoRA	**.709**±.085	.899±.012	**.799**±.001	.870±.030

Table 19: Llama-3.1-8B classification on BoolQ, reported as mean ± population standard deviation over three seeds (per-seed values were not retained).

Method	$N = 64$	$N = 256$
LoRA	.554±.025	.576±.020
PiSSA	.527±.023	.536±.021
LoRAM	.534±.032	.573±.014
PivotLoRA	**.602**±.009	**.618**±.004

E.4 Summarization Control

GPT-2 summarization tests whether a large d/N alone produces large gains (Table 20; XSum and CNN/DailyMail, ROUGE-L on 500 validation examples, three seeds, one matched sweep). Under the single learning rate used for reasoning, PivotLoRA exceeds LoRA by only $+0.2$ points with the sequence mean and $+0.3$ with response positions, winning 8 and 9 of the 12 seed-level comparisons, respectively. These gains are much smaller than the classification and reasoning gains at comparable d/N. Under the classification recipe, PivotLoRA trails LoRA by 1.5 to 5.1 points with either source, so the learning-rate recipe matters more than the activation source. Response positions score above the sequence mean in all eight comparisons, by $+0.1$ to $+0.2$ points under the single rate and by $+0.2$ to $+1.2$ under the classification recipe.

Table 20: GPT-2 few-shot summarization, ROUGE-L means over three seeds. "classification" uses eigenvalue-scaled rates with $\lambda_{\text{ratio}} = 16$; "single rate" uses 2×10^{-4} for both factors, as in the reasoning experiments.

		XSum		CNN/DailyMail	
Method	Source / recipe	64	256	64	256
LoRA	—	.185	.184	**.212**	.227
PivotLoRA	sequence mean / classification	.135	.164	.193	.208
PivotLoRA	response / classification	.147	.168	.198	.210
PivotLoRA	sequence mean / single rate	.188	.191	.209	.228
PivotLoRA	response / single rate	**.191**	**.192**	.211	**.229**

E.5 Full-Data Reference

Table 21 compares PivotLoRA and LoRA when the full BERT training split is used. The differences range from -0.009 to $+0.022$ in the task's primary score.

Table 21: BERT-base on the full training split using seed 42. CoLA uses Matthews correlation, and the remaining tasks use accuracy.

Method	SST-2	CoLA	MNLI	QNLI	RTE	AG News
LoRA	.923	.533	.806	.893	.610	.940
PivotLoRA	.914	.555	.817	.889	.610	.944
Δ	$-.009$	+.022	+.011	$-.004$	.000	+.004

PiSSA under full data. Because weight-spectral initializers such as PiSSA are often motivated by full-data fine-tuning, we ran a separate matched single-seed audit on the full training splits using the reference PEFT implementation of PiSSA [Meng et al., 2024]. LoRA, PiSSA, and PivotLoRA share the data, seed, optimizer, and training budget within this audit (Table 22). Because it is a separate execution, its LoRA and PivotLoRA values differ slightly from Table 21. PiSSA is within $+0.018$ of LoRA on five tasks and 0.021 below it on CoLA, and PivotLoRA is at or above PiSSA on all six tasks. With one seed, the three methods are comparable in the full-data regime. The audit does not measure convergence speed, where PiSSA reports its advantages.

Table 22: Single-seed full-data audit with the reference PEFT implementation of PiSSA (seed 42). CoLA uses Matthews correlation; the other tasks use accuracy.

Method	SST-2	CoLA	MNLI	QNLI	RTE	AG News
LoRA	.911	.515	.809	.876	.545	.941
PiSSA	.915	.494	.812	.879	.563	.941
PivotLoRA	.916	.565	.815	.902	.606	.943

E.6 Full Fine-Tuning Reference

Table 23 compares PivotLoRA with full fine-tuning at $N = 64$. Full fine-tuning is a strong reference but does not match the adapter parameter budget. PivotLoRA exceeds the mean full fine-tuning score on all six tasks while training about $370\times$ fewer parameters (approximately 0.30M).

Table 23: Full fine-tuning, LoRA, and PivotLoRA on BERT-base at $N = 64$. Full fine-tuning is reported as mean $\pm$ standard deviation over three seeds. LoRA and PivotLoRA entries are the corresponding three-seed means from the primary tables.

Task	Full FT (110M)	LoRA (0.30M)	PivotLoRA (0.30M)
SST-2	.700±.053	.652	**.795**
CoLA	.286±.043	.163	**.288**
AG News	.791±.042	.640	**.834**
MNLI	.345±.007	.366	**.367**
QNLI	.555±.004	.555	**.594**
RTE	.520±.046	.551	**.569**

E.7 Alignment Estimation Procedure and Task-Level Gains

For each BERT classification task, we run the frozen encoder on all N labeled examples and extract final-layer `[CLS]` representations. The few-shot set is sampled approximately class-balanced. Let n_c be the number of examples in class c, let μ_c be its mean representation, and let $\bar{\mu}$ be the global mean. We compute

$$\Sigma_{\text{task}} = \frac{1}{N} \sum_{c=1}^{C} n_c (\mu_c - \bar{\mu})(\mu_c - \bar{\mu})^\top.$$

If U contains the top-r PCA directions computed from the same prediction-position representations, the alignment coefficient is

$$\rho = \frac{\text{Tr}\left(U^\top \Sigma_{\text{task}} U\right)}{\text{Tr}\left(\Sigma_{\text{task}}\right)}.$$

This diagnostic requires no gradients and is computed from one frozen forward pass followed by covariance and trace-ratio calculations.

Table 24 contains the seven values used for the main correlation analysis. CoLA gains are measured in Matthews correlation points, and all other gains are accuracy points. Across the seven tasks, the Pearson correlation is $r = 0.93$. An exact two-sided permutation test evaluates all $7! = 5{,}040$ assignments of the gain values to the fixed alignment values and gives $p = 0.002$.

Table 24: Alignment ρ at $N = 64$ and PivotLoRA's shared-protocol gain over vanilla LoRA.

Task	ρ	Fisher ratio	PivotLoRA gain (points)
AG News	0.843	0.14	+19.4
SST-2	0.667	0.02	+14.3
CoLA	0.530	0.02	+12.5
QNLI	0.458	0.02	+3.9
MNLI	0.419	0.03	+0.1
BoolQ	0.402	0.01	+1.6
RTE	0.293	0.01	+1.8

The class-based definition of ρ requires discrete labels. It is not directly defined for open-ended generated responses and is not used to predict the reasoning gains.

E.8 Robustness of the Alignment Diagnostic

The seven-task Spearman correlation is 0.86. Leave-one-task-out Pearson correlations range from 0.87 to 0.96. Removing AG News still gives $r = 0.87$, so the association is not determined only by the largest-gain task.

As a same-sample-bias check, we compute a cross-split estimate on AG News, SST-2, and CoLA. The PCA subspace is estimated using one half of the $N = 64$ set, while Σ_{task} is estimated using

the other half. We then exchange the halves and average the two estimates. The absolute values decrease because each estimate uses fewer examples, but the ordering AG News $>$ SST-2 $>$ CoLA is preserved.

Table 25: Cross-split robustness check at $N = 64$. CoLA gains are in Matthews correlation points, and the other gains are in accuracy points.

Task	Original ρ	Cross-split ρ	PivotLoRA gain
AG News	0.843	0.390	+19.4
SST-2	0.667	0.285	+14.3
CoLA	0.530	0.259	+12.5

Subset variability. To estimate the sampling uncertainty of ρ, we recomputed it on ten independently drawn class-balanced $N = 64$ subsets per task (Table 26). AG News is consistently high, and SST-2 is high in eight of ten subsets, whereas QNLI, MNLI, RTE, and BoolQ exceed 0.6 in only one subset each. In this BERT-base, rank-8, $N = 64$ configuration, values near or above 0.6 therefore separate the two consistently high-alignment tasks from the low-gain group. CoLA, at $\rho = 0.53$, lies between the two groups on both alignment and gain.

Table 26: ρ over ten class-balanced $N = 64$ subsets per task, with the shared-protocol PivotLoRA gain from Table 24. CoLA was not included in this resampling.

Task	Mean $\pm$ SD	Range	Subsets with $\rho \geq 0.6$	PivotLoRA gain (points)
AG News	0.843 ± 0.017	0.823–0.874	10/10	+19.4
SST-2	0.673 ± 0.185	0.191–0.824	8/10	+14.3
QNLI	0.440 ± 0.079	0.360–0.623	1/10	+3.9
BoolQ	0.457 ± 0.137	0.306–0.749	1/10	+1.6
MNLI	0.444 ± 0.120	0.289–0.647	1/10	+0.1
RTE	0.391 ± 0.109	0.255–0.606	1/10	+1.8

Class imbalance. On AG News we drew 30 $N = 64$ subsets under each of three class-count profiles. Balanced sampling (16/16/16/16) gives $\rho = 0.869 \pm 0.027$, moderate imbalance (28/16/12/8) gives 0.866 ± 0.024, and severe imbalance (40/12/8/4) gives 0.818 ± 0.028, with a minimum of 0.755. Severe imbalance lowers the estimate but does not change the high-alignment conclusion in this experiment. We did not test label noise. Because ρ is built from class means, noisy labels can distort it directly.

For practical use, we recommend estimating uncertainty by recomputing ρ over class-stratified resamples of the cached activations. A low or unstable estimate should be treated as weak evidence for direction-aware initialization, and method selection should still be validated under matched tuning when feasible. We do not recommend a universal threshold. Instead, ρ should be treated as a comparative screening diagnostic under a fixed model, rank, representation, and sampling protocol.

E.9 Encoder-Specific Class Structure

Table 27 reports BERT-specific measurements from pretrained `[CLS]` representations. Fisher's ratio summarizes overall class separation, while ρ measures the fraction of between-class scatter captured by the top-r PCA subspace. These quantities need not produce identical task orderings because one measures the amount of between-class separation and the other measures its orientation relative to the leading covariance directions.

Table 27: Class structure in frozen BERT [CLS] representations, averaged over three seeds. Columns denote the total number of labeled examples.

	$N = 16$	$N = 32$	$N = 64$	$N = 128$	$N = 256$
Fisher's discriminant ratio $\mathrm{tr}(S_B)/\mathrm{tr}(S_W)$					
SST-2	0.07	0.04	0.02	0.01	0.01
CoLA	0.09	0.05	0.02	0.01	0.01
AG News	0.37	0.18	0.14	0.11	0.10
MNLI	0.16	0.07	0.03	0.02	0.01
RTE	0.07	0.03	0.01	0.01	0.00
QNLI	0.07	0.04	0.02	0.01	0.01
PCA-class alignment ρ					
SST-2	0.803	0.653	0.673	0.495	0.582
CoLA	0.824	0.721	0.530	0.403	0.566
AG News	0.911	0.821	0.843	0.876	0.901
MNLI	0.760	0.536	0.423	0.310	0.351
RTE	0.705	0.478	0.299	0.345	0.280
QNLI	0.756	0.560	0.453	0.424	0.437

Table 27 comes from a separate run of the class-structure analysis that drew its few-shot subsets independently, so its $N = 64$ column differs from Table 24 by at most 0.006 (0.673 versus 0.667 on SST-2). The correlation analysis uses Table 24.

E.10 Per-Setting Gains Behind the Regime Map

Table 28 gives the range of per-setting gains behind Table 4. BERT rows cover the six primary tasks. Llama classification rows cover SST-2 and AG News on Llama-2-7B and Llama-3.1-8B and BoolQ on Llama-3.1-8B. The reasoning row covers GSM8K and StrategyQA on Llama-3.1-8B at $N = 128$ and evaluates the complete response-position procedure (Appendix D). The GPT-2 row covers XSum and CNN/DailyMail at $N \in \{64, 256\}$ with response positions and the single learning rate (Appendix E.4).

Table 28: PivotLoRA gain over vanilla LoRA by regime, in points of each task's primary metric (ROUGE-L for GPT-2).

Setting	d	d/N	Mean gain	Range
BERT-base, $N \leq 64$	768	12–48	$+6.9$	$+0.1$ to $+23.9$
BERT-base, $N \geq 128$	768	3–6	$+3.2$	-2.0 to $+10.1$
BERT-base, full data	768	0.002–0.31	$+0.4$	-0.9 to $+2.2$
Llama classification, $N = 64$	4096	64	$+3.8$	$+0.5$ to $+6.1$
Llama classification, $N = 256$	4096	16	$+0.4$	-1.8 to $+4.2$
Llama-3.1-8B reasoning	4096	32	$+4.1$	$+3.6$ to $+4.5$
GPT-2 summarization	768	3–12	$+0.3$	-0.1 to $+0.8$

E.11 Data-Efficiency Illustration

Figure 5 provides a task-specific illustration on AG News. At $N = 64$, PivotLoRA reaches approximately 89% of the displayed full-data accuracy ceiling, compared with approximately 68% for vanilla LoRA.

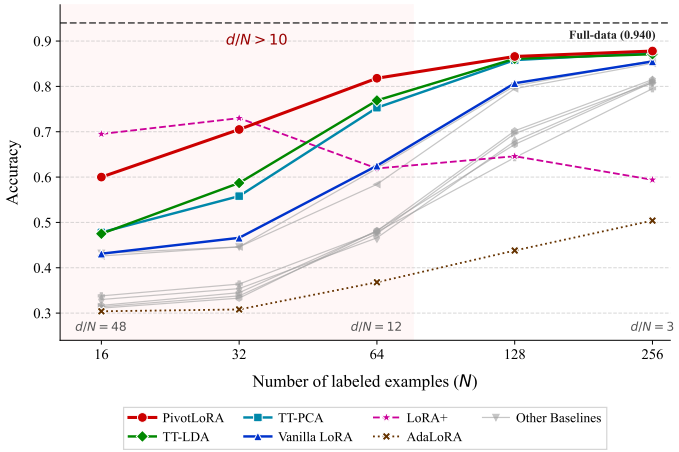

Figure 5: Data efficiency on AG News with BERT-base. The shaded region marks the high-d/N few-shot regime. The dashed line denotes the displayed full-data accuracy reference.

E.12 Activation-Source Ablation

We compare activation sources for estimating the PivotLoRA initialization subspace. Target-position initialization uses one `[CLS]` activation per example, whereas all-token initialization uses all non-padding token activations. The first two configurations use the same source for query and value adapters. The last two use the target position for query adapters and either all tokens or attention-weighted pooling for value adapters. All other components, including the sampled examples, rank, scale, learning-rate allocation, pivot timing, optimizer-state policy, and training budget, are held fixed within each paired comparison.

This ablation is a separate matched sweep, so its absolute values should be compared within Table 29.

Under this matched comparison, all-token initialization has a higher mean by 3.15 points on AG News and 2.33 points on SST-2, with seed-level wins in three of three and two of three comparisons, respectively. The SST-2 difference is small relative to seed variation.

Assigning the target position to query adapters and all tokens to value adapters also improves on the uniform target-only configuration on AG News (+2.1 points, 3/3 seeds) and slightly on SST-2 (+0.6 points, 1/3 seeds), showing that sources can be selected per projection type. The tested attention-weighted value source is neutral on AG News (-0.1 points) and lower on SST-2 (-11.2 points), so distributed-token pooling is not automatically beneficial.

Target-position PCA uses exactly 64 activation vectors per layer, whereas all-token PCA uses about 3,400 vectors on AG News and 830 on SST-2. The corresponding top-eight explained-variance ratios, averaged over the inputs of the twelve adapted layers (Appendix B.1 reports the final layer), are 55.88% and 67.75% for target-position PCA, compared with 15.14% and 21.16% for all-token PCA. More broadly, the comparison illustrates the alignment–estimation trade-off. Prediction-position pooling targets the representation used for supervision, whereas all-token pooling estimates its subspace from a substantially larger activation sample.

F Robustness and Ablation Studies

Several analyses below were run as separate sweeps under the primary protocol. Their absolute scores differ slightly from the primary tables, so each is compared within its own table.

F.1 Aggregate Paired Robustness Analysis

Each individual task–N setting contains only three seeds, so we do not emphasize per-setting significance tests. Instead, we summarize the direction of the mean differences across the 30 primary BERT settings.

Table 29: Activation-source ablation for BERT-base PivotLoRA at $N = 64$. Values are mean $\pm$ sample standard deviation over three paired seeds. Paired gains are computed as all-token minus target-position accuracy. The last two rows use different sources for query and value adapters.

Activation source	AG News	SST-2
Target position (`[CLS]`)	.814±.015	.778±.063
All non-padding tokens	.845±.008	.802±.048
Target for query, all tokens for value	.835±.007	.784±.024
Target for query, attention-weighted for value	.813±.017	.666±.045
Paired gain of all-token	+.032	+.023
All-token seed-level wins	3/3	2/3

PivotLoRA exceeds vanilla LoRA in 29 of the 30 settings, with the only negative difference at RTE $N = 128$. A two-sided exact sign test on these 30 setting-level comparisons gives

$$p = 5.8 \times 10^{-8}$$

under a null model in which positive and negative differences are equally likely. Because the settings share a backbone and tasks, they are not independent, and the test summarizes the benchmark without a strict inferential guarantee. The five-seed experiments below provide complementary seed-level evidence for selected settings.

F.2 Five-Seed Confirmation

All seven bootstrap intervals for the paired gain exclude zero, with mean gains from $+0.032$ on QNLI to $+0.231$ on AG News at $N = 32$. We reran seven representative BERT settings with seeds 42–46. Table 30 reports the mean, standard deviation, and bootstrap 95% confidence interval for the mean paired difference. The intervals were computed from seed-aligned LoRA and PivotLoRA runs.

Table 30: Five-seed confirmation on representative BERT-base settings using seeds 42–46. The confidence interval is a bootstrap interval for the mean paired difference, PivotLoRA minus LoRA.

Task	N	LoRA	PivotLoRA	Δ	95% CI
AG News	16	.397±.044	.577±.035	+.180	[+.145, +.217]
AG News	32	.490±.016	.721±.047	+.231	[+.191, +.265]
AG News	64	.650±.032	.816±.022	+.166	[+.128, +.205]
SST-2	32	.525±.026	.615±.064	+.090	[+.056, +.125]
SST-2	64	.655±.032	.790±.022	+.135	[+.112, +.156]
CoLA	64	.148±.061	.265±.095	+.117	[+.070, +.169]
QNLI	64	.564±.035	.596±.037	+.032	[+.010, +.047]

F.3 Fisher-Gradient Baseline

We implement a Fisher-gradient variant inspired by LoRA-DA [Zhang et al., 2026]. For per-example gradients g_i, it estimates

$$F_{\text{diag}} = \frac{1}{N} \sum_{i=1}^{N} g_i \odot g_i$$

and performs an SVD of the Fisher-weighted mean gradient

$$\bar{G} \odot \sqrt{F_{\text{diag}}}.$$

Tables 31 and 32 report the resulting scores, with vanilla LoRA rerun in the same sweep. LoRA-DA is below vanilla LoRA in all 10 settings, by 0.006 to 0.131.

Table 31: LoRA-DA on AG News. Values are accuracy means and standard deviations over three seeds. LoRA is rerun in the same sweep.

Method	16	32	64	128	256
LoRA	.410±.029	.477±.078	.624±.054	.774±.033	.858±.010
LoRA-DA	.317±.034	.352±.015	.493±.056	.679±.066	.814±.030

Table 32: LoRA-DA on SST-2. Values are accuracy means and standard deviations over three seeds. LoRA is rerun in the same sweep.

Method	16	32	64	128	256
LoRA	.541±.017	.527±.026	.639±.063	.777±.010	.847±.007
LoRA-DA	.528±.016	.510±.020	.598±.028	.702±.056	.841±.001

F.4 LoRA+ Behavior at Low Labeled-Example Counts

LoRA+ has the best mean in 5 of the 18 primary settings with $N \leq 64$, all at $N \leq 32$, where activation-based estimates are noisiest. It retains a random LoRA input basis and changes only the relative learning rates of the two factors.

Under the shared learning-rate protocol, LoRA+ collapses to a constant prediction on most tasks at $N \geq 128$ (zero standard deviation across seeds), and its AG News scores vary widely. Appendix F.9 includes a separate learning-rate and differential-ratio sensitivity analysis.

F.5 Pivot Timing

We vary the pivot epoch over

$$K \in \{1, 2, 3, 5, 7, 10\}$$

on SST-2, CoLA, and AG News, with vanilla LoRA rerun in the same sweep. Every tested pivot epoch outperforms LoRA at every $N \leq 64$ on all three tasks, and the default $K = 3$ is competitive but not uniformly optimal. An early pivot allows more training in the orthonormal coordinates, while a later pivot permits more task-gradient drift before the coordinate change.

Table 33: Pivot timing on SST-2. Values are accuracy means over three seeds.

K	16	32	64	128	256
1	.570	.601	.762	.818	.854
2	.568	.630	.779	.821	.854
3 (default)	.567	.608	.786	.829	.844
5	.565	.619	.794	.825	.851
7	.569	.659	.797	.820	.851
10	.561	.612	.783	.813	.850
LoRA	.543	.526	.641	.777	.846

Table 34: Pivot timing on CoLA. Values are Matthews correlation means over three seeds.

K	16	32	64	128	256
1	.127	.183	.285	.403	.412
2	.126	.191	.296	.378	.400
3 (default)	.119	.182	.289	.366	.393
5	.132	.172	.276	.355	.388
7	.128	.183	.279	.365	.403
10	.125	.175	.270	.369	.401
LoRA	.100	.113	.200	.339	.400

Table 35: Pivot timing on AG News. Values are accuracy means over three seeds.

K	16	32	64	128	256
1	.589	.708	.799	.857	.877
2	.593	.717	.802	.861	.878
3 (default)	.604	.707	.816	.866	.878
5	.595	.724	.817	.865	.879
7	.583	.719	.821	.865	.877
10	.516	.701	.813	.865	.879
LoRA	.411	.477	.631	.774	.857

F.6 Pivot Preservation and Orthogonality Diagnostics

For each classification adapter, we measure the relative preservation error

$$\epsilon_{\text{pivot}} = \frac{\left\| B^{(K)}A^{(K)} - B^{\text{new}}A^{\text{new}} \right\|_F}{\left\| B^{(K)}A^{(K)} \right\|_F + 10^{-12}}.$$

The factor α/r cancels from this ratio. We also measure the residual between the old and new row spans, the row-orthogonality error

$$\epsilon_{\text{orth}}(A) = \|AA^\top - I_r\|_F,$$

and the change in the layerwise alignment score. The classification values in Table 36 are aggregated over three seeds and 72 adapter instances per task, comprising 24 adapted modules for each seed.

Table 36: Classification output-preservation and orthogonality diagnostics. The alignment change excludes layer 0 because its embedding-level `[CLS]` scatter is near zero.

Task	Mean ϵ_{pivot}	Max. ϵ_{pivot}	Mean span residual	$\epsilon_{\text{orth}}(A^{(K)})$	$\epsilon_{\text{orth}}(A^{\text{new}})$	Mean $\Delta\rho$
AG News	3.14×10^{-7}	8.82×10^{-7}	2.94×10^{-7}	2.809	4.84×10^{-7}	2.77×10^{-9}
CoLA	3.00×10^{-7}	5.07×10^{-7}	2.87×10^{-7}	2.809	4.69×10^{-7}	-2.31×10^{-9}
SST-2	2.79×10^{-7}	7.91×10^{-7}	2.89×10^{-7}	2.809	4.76×10^{-7}	2.56×10^{-10}

The output and span residuals remain below 10^{-6}. The post-pivot orthogonality errors are also below 10^{-6}, and the alignment change is near numerical zero. These measurements confirm that the classification pivot preserves the represented update and learned row span while changing their coordinate parameterization. In the reasoning runs, the maximum relative preservation error is 1.44×10^{-7} and the maximum post-pivot orthogonality error is 3.51×10^{-8}.

What the pivot changes. Writing $\sigma_1, \ldots, \sigma_r$ for the singular values of $A^{(K)}$, $\epsilon_{\text{orth}}(A^{(K)})^2 = r - 2\sum_i \sigma_i^2 + \sum_i \sigma_i^4$. When $\|A^{(K)}\|_F^2 = \sum_i \sigma_i^2$ is also logged, as in the reasoning runs, these two quantities determine the effective rank $r_{\text{eff}} = (\sum_i \sigma_i^2)^2 / \sum_i \sigma_i^4$ of the pre-pivot Gram matrix. In classification, the value 2.809 is what a scaled orthonormal matrix $\sqrt{5/d}\,U^\top$ would give (2.810 for $d = 768$), which places $\|A^{(K)}\|_F$ within about 5% of its initial value. The pivot therefore raises the row norms of A about twelvefold, although the three-decimal value cannot resolve whether A also became anisotropic. In the reasoning runs, the median adapter's $\|A^{(K)}\|_F$ grows to $5.7\times$ its initial value, and the median r_{eff} is 1.9–2.1 on GSM8K and 1.4–1.5 on StrategyQA across seeds, out of $r = 8$. Early training therefore concentrates the learned input basis on a few directions, and the pivot restores an isotropic basis for the same span.

F.7 Orthogonal Pivot and Optimizer-State Ablation

We separate the output-preserving QR pivot from the classification optimizer-state policy. QR-PIVOT-NORESET applies the pivot but keeps the Adam state of A, whereas PIVOTLORA resets it. Both use the same post-pivot learning-rate handling (Appendix C.1).

Table 37: Classification optimizer-state ablation on SST-2. Values are accuracy means over three seeds.

Method	$N = 16$	$N = 32$	$N = 64$	$N = 128$	$N = 256$
TT-PCA	.520	.530	.755	.817	.858
QR-Pivot-NoReset	.552	.623	.795	.829	.858
PivotLoRA	.541	.628	.784	.828	.847

Table 38: Classification optimizer-state ablation on CoLA. Values are Matthews correlation means over three seeds.

Method	$N = 16$	$N = 32$	$N = 64$	$N = 128$	$N = 256$
TT-PCA	.104	.137	.270	.383	.402
QR-Pivot-NoReset	.117	.172	.301	.381	.398
PivotLoRA	.117	.174	.304	.381	.393

Table 39: Classification optimizer-state ablation on AG News. Values are accuracy means over three seeds.

Method	$N = 16$	$N = 32$	$N = 64$	$N = 128$	$N = 256$
TT-PCA	.478	.561	.756	.857	.874
QR-Pivot-NoReset	.589	.689	.811	.859	.878
PivotLoRA	.597	.703	.811	.857	.878

The QR pivot accounts for most of the improvement over TT-PCA. Without the reset, it improves on TT-PCA in 12 of 15 settings, by 0.037 on average and by up to 0.128 on AG News at $N = 32$. The additional reset produces smaller, task-dependent differences and is not uniformly superior to QR-PIVOT-NORESET. We therefore treat the classification reset as a stabilization policy and not as the main source of the gain.

This ablation does not apply directly to the reasoning experiments, which reset the optimizer states of both A and B, and no reasoning reset-only or no-reset ablation was conducted.

F.8 Post-Pivot Learning-Rate Policy

The pivot's gain does not depend on how the B learning rate is handled afterward. All pivot variants share the TT-PCA initialization, the eigenvalue-scaled rates before the pivot, the QR pivot at $K = 3$, and the Adam reset for A, and differ only in the B rate after the pivot. The implemented policy raises a common rate from η_A to $16\eta_A$ over 10 steps and then resumes the eigenvalue-scaled rates. The alternatives keep the eigenvalue-scaled rates unchanged, switch to a uniform $16\eta_A$ for the rest of training, or ramp to that uniform rate and keep it. The sweep covers SST-2, CoLA, and AG News at $N \in \{16, 32, 64, 128, 256\}$ with three seeds.

Table 40: Post-pivot learning-rate policy. Task columns average over the five values of N (CoLA in Matthews correlation); "Wins" counts the 15 task–N settings in which a variant exceeds TT-PCA.

Variant	SST-2	CoLA	AG News	Average	Wins
TT-PCA (no pivot)	.697	.257	.717	.557	–
Pivot, no rate change	.734	.275	.770	.593	11/15
Pivot, implemented policy	.728	.276	.772	.592	12/15
Pivot, uniform rate	.744	.266	.777	.596	13/15
Pivot, ramp then uniform rate	.723	.265	.766	.585	12/15

Every policy improves on TT-PCA by 0.028–0.039 on average (0.036 with no rate change), and the policies differ by at most 0.011.

F.9 Learning-Rate Structure in the Ablation Ladder

TT-PCA and TT-LDA differ in both initialization direction and learning-rate structure. TT-PCA uses the eigenvalue-scaled rates in Eq. (12), whereas TT-LDA uses the uniform differential rate $\eta_B = 16\eta_A$. Their direct comparison is consequently an ablation ladder and does not isolate PCA from LDA.

Across the full BERT benchmark, TT-LDA, TT-PCA, and PivotLoRA have average ranks 3.13, 3.77, and 1.83, respectively. LoRA+ retains random input directions and applies a $16\times$ differential rate but has average rank 8.43 under the shared protocol. This aggregate comparison argues against an explanation based only on the average B-to-A learning-rate ratio. The direct controls in Appendix F.10 isolate the PCA directions from the eigenvalue-to-coordinate rate assignment.

Learning-rate sensitivity. We sweep learning rates on AG News, SST-2, and CoLA at $N = 64$.

For each method, we sweep the base learning-rate anchor over

$$\eta_A \in \left\{5 \times 10^{-5}, 2 \times 10^{-4}, 8 \times 10^{-4}\right\}.$$

The primary shared-protocol anchor is $\eta_A = 2\times10^{-4}$. Differential-rate methods derive their B-factor rates from this anchor. For LoRA+, model selection additionally ranges over

$$\eta_B/\eta_A \in \{4, 8, 16, 32\}.$$

Table 41 reports the average score at the primary anchor and the average of each method's best swept score. Tuning reduces several margins, but PivotLoRA remains strongest on average among the swept methods.

Table 41: Learning-rate sensitivity at $N = 64$. Scores are unweighted averages of each task's primary metric over AG News, SST-2, and CoLA. "Avg. anchor" uses the primary base learning-rate anchor $\eta_A = 2 \times 10^{-4}$, together with each method's prescribed factor-wise allocation. "Avg. best" averages each method's best score over the base-rate sweep. For LoRA+, selection additionally includes $\eta_B/\eta_A \in \{4, 8, 16, 32\}$. Gaps are measured relative to PivotLoRA under the corresponding comparison.

Method	Avg. anchor	Avg. best	Pivot anchor gap	Pivot best gap
LoRA	.522	.622	+.139	+.045
LoRA+	.326	.636	+.335	+.031
PiSSA	.420	.606	+.241	+.061
LoRAM	.428	.617	+.233	+.050
EVA	.424	.618	+.237	+.048
TT-PCA	.643	.655	+.017	+.012
PivotLoRA	.661	.667	.000	.000

Table 42: Best swept scores by task at $N = 64$. Each score is selected over the base learning-rate anchors $\{5 \times 10^{-5}, 2 \times 10^{-4}, 8 \times 10^{-4}\}$. For LoRA+, selection additionally ranges over the differential ratios $\eta_B/\eta_A \in \{4, 8, 16, 32\}$.

Method	AG News	SST-2	CoLA
LoRA	.781	.784	.301
LoRA+	.833	.757	.318
PiSSA	.754	.733	.330
LoRAM	.732	.789	.330
EVA	.742	.786	.327
TT-PCA	.817	.821	.328
PivotLoRA	.843	.822	.335

F.10 Direction versus Learning-Rate Scaling Controls

We evaluate three controls at $N = 64$ to separate target-token PCA directions from the associated learning-rate allocation.

- RANDOM+EIGENLR uses a random orthonormal input basis with the same eigenvalue-scaled per-rank B learning rates.
- TT-PCA+UNIFORMLR retains the target-token PCA basis and uses a uniform differential learning rate.
- TT-PCA+SHUFFLEDLR retains the target-token PCA basis and randomly permutes the eigenvalue-to-coordinate rate assignment.

The TT-PCA row in Table 43 comes from the same control sweep and is the reference for the three variants.

Table 43: Direction versus learning-rate scaling controls at $N = 64$ using BERT-base. Values are mean $\pm$ standard deviation over three seeds from the separate diagnostic sweep.

Method	AG News	SST-2	CoLA
LoRA	.640±.033	.647±.040	.188±.042
TT-PCA	.751±.025	.769±.038	.268±.092
Random+EigenLR	.444±.268	.571±.031	.028±.035
TT-PCA+UniformLR	.752±.019	.769±.035	.276±.100
TT-PCA+ShuffledLR	.754±.028	.765±.028	.281±.088

A random basis does not become competitive merely by receiving the eigenvalue-scaled rates. Conversely, the target-token PCA basis retains its performance when the rates are made uniform or their assignments are permuted. Within this control sweep, the selected directions therefore explain more of the advantage than the eigenvalue-to-coordinate rate assignment. Because the classification implementation applies a fixed random rotation within the PCA span (Eq. (7)), each rate-scaled row of B already corresponds to a mixture of principal directions, so the shuffled control therefore tests the rate allocation itself, not a pairing between individual directions and rates. This experiment separates direction from learning-rate allocation but not from initialization norm.

G Per-Seed and Cost Details

G.1 Per-Seed Decoder-Scale Results

Tables 44 and 45 provide the per-seed decoder-scale results. PivotLoRA has the highest $N = 64$ aggregate mean on SST-2 and AG News for both backbones, although individual seed-level exceptions occur.

Table 44: Llama-2-7B per-seed classification results in bf16 on an NVIDIA L4.

	SST-2, $N = 64$			SST-2, $N = 256$			AG News, $N = 64$			AG News, $N = 256$		
	s42	s43	s44	s42	s43	s44	s42	s43	s44	s42	s43	s44
LoRA	.711	.757	.655	.931	.917	.922	.724	.743	.777	.877	.884	.876
PiSSA	.683	.571	.650	.911	.927	.930	.706	.709	.684	.867	.874	.854
LoRAM	.765	.787	.737	.927	.944	.915	.719	.771	.737	.878	.877	.872
PivotLoRA	.802	.742	.764	.927	.921	.889	.809	.791	.753	.876	.869	.839

Table 45: Llama-3.1-8B per-seed classification results in bf16 on an NVIDIA L4.

	SST-2, $N = 64$			SST-2, $N = 256$			AG News, $N = 64$			AG News, $N = 256$		
	s42	s43	s44	s42	s43	s44	s42	s43	s44	s42	s43	s44
LoRA	.728	.723	.661	.912	.885	.851	.758	.783	.737	.870	.885	.886
PiSSA	.626	.578	.663	.911	.865	.853	.710	.744	.691	.865	.874	.854
LoRAM	.720	.640	.627	.907	.909	.888	.735	.800	.708	.877	.879	.861
PivotLoRA	.735	.778	.614	.888	.912	.899	.800	.799	.798	.889	.884	.836

G.2 Initialization Overhead

Table 46 reports initialization time and additional persistent storage for representative methods. PivotLoRA performs one activation pass and PCA initialization before training and one QR-based coordinate change at the pivot. The timing is small relative to the approximately 69-second classification training run used in the cost comparison below.

Table 46: Initialization overhead for BERT-base with 12 layers and rank $r = 8$ on an NVIDIA T4.

Method	Initialization time	Additional persistent storage
LoRA	< 0.1 s	None
PivotLoRA	< 1 s plus < 1 s at pivot	None
PiSSA	$\sim$2 s, layerwise SVD	Residual W^{res}
LoRA-GA	$\sim$3 s, gradient plus SVD	None
LoRA-One	$\sim$3 s, gradient plus SVD	None
EVA	$\sim$1 s, activation SVD	None
LoRAM	< 0.5 s, power iteration	None

G.3 Computational Cost

Tables 47 and 48 report end-to-end training time and peak memory. On the BERT measurement, PivotLoRA adds 4.0 seconds relative to LoRA, corresponding to 5.8% wall-time overhead. Most of this difference comes from the per-rank optimizer groups for the eigenvalue-scaled learning rates, and the QR pivot itself takes under a second. Peak memory is nearly unchanged.

Table 47: BERT-base training time and peak memory at $N = 64$ on SST-2, averaged over two seeds on one NVIDIA T4.

Method	Wall time (s)	Peak VRAM (MB)	Overhead vs. LoRA
LoRA	69.4	1,266	N/A
DoRA	69.5	1,399	+0.1 s (+0.2%)
AdaLoRA	102.2	1,264	+32.9 s (+47.4%)
PiSSA	68.6	1,267	-0.8 s (-1.2%)
LoRA-GA	68.0	1,272	-1.4 s (-1.9%)
LoRA-One	69.0	1,272	-0.4 s (-0.6%)
LoRA+	68.9	1,267	-0.5 s (-0.7%)
EVA	69.8	1,267	+0.4 s (+0.6%)
LoRAM	68.9	1,267	-0.5 s (-0.7%)
SRLoRA	89.4	1,267	+20.0 s (+28.8%)
TT-PCA	73.6	1,267	+4.2 s (+6.0%)
TT-LDA	69.3	1,267	-0.1 s (-0.1%)
PivotLoRA	73.4	1,268	+4.0 s (+5.8%)

TT-PCA and PivotLoRA use per-rank optimizer groups for their eigenvalue-scaled rates. TT-LDA uses the uniform differential rate $\eta_B = 16\eta_A$. The 0.2-second difference between TT-PCA and PivotLoRA is within run-to-run timing variation.

Table 48: Llama-2-7B classification training cost on SST-2 at $N = 64$, using bf16 on one NVIDIA L4.

Method	Wall time (s)	Peak VRAM (GB)
LoRA	$\sim$74	14.6
PiSSA	$\sim$73	14.6
PivotLoRA	$\sim$73	14.6

H Limitations

Activation-source and task scope. Activation-based initialization works best with a source matched to the supervision, such as the `[CLS]` representation for BERT classification, the sequence mean for decoder classification, and hidden states that predict supervised response tokens for completion-only reasoning. Target-position pooling emphasizes representations tied to prediction, whereas all-token pooling provides more observations but includes variation less specific to the target, and neither should be expected to dominate across tasks. Query and value adapters share one source rule in our main configuration. Sources can be chosen per projection type (Appendix E.12), but we do not study this systematically.

Theoretical and diagnostic scope. The coefficient ρ is defined through between-class covariance, so it applies most directly to labeled classification, and its association with gains is measured across seven BERT tasks under one protocol. Generation and continuous targets would require a different definition of supervised structure. The perturbation result assumes independent sub-Gaussian observations, an approximately spiked covariance, and a positive eigengap, and the first-step analysis adds a class-conditional mean model and an assumption relating gradient-weighted geometry to between-class scatter. Both give a regime-level account and are not convergence guarantees for transformer fine-tuning.

Model and task coverage. The decoder classification experiments cover fewer tasks and baselines than the BERT study and check whether the main pattern persists at larger scale, not full rankings of methods for 7B and 8B models. The reasoning experiments cover mathematical and yes/no reasoning with one 8B model and do not support claims about instruction tuning, multilingual adaptation, or unrestricted generation.

Reasoning attribution and statistical scope. The reasoning procedure combines response-position PCA, a fixed $\sqrt{5/d}$ basis scale, one output-preserving pivot, and a reset of the AdamW state for both factors. Without norm-matched, reset-only, or direction-versus-rate controls in this setting, the results evaluate the complete procedure and do not isolate subspace direction as the cause. The McNemar tests in Appendix D.4 treat test items as the unit of variation and do not capture variation across training runs.

Pivot scope. The pivot orthonormalizes coordinates within the row span learned before epoch K and adds no new directions. Replacing low-utility rank directions during training while preserving the accumulated update is a natural extension.

Broader impacts. Parameter-efficient adaptation lowers the cost of few-shot specialization, which may also lower the cost of adapting pretrained models for harmful or deceptive uses. PivotLoRA should be used under the same safety, data-governance, and misuse-mitigation practices that apply to the underlying pretrained models.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and Section 1 state the paper's main scope, which comprises a regime-based account of when LoRA initialization direction matters, the PivotLoRA method, and the alignment coefficient ρ as a classification diagnostic. These claims are developed theoretically in Section 3, operationalized in Section 4, and evaluated in Section 5. The quantitative claims match the reported results. Across the 30 primary BERT-base few-shot settings, PivotLoRA achieves the best overall average rank of 1.83 and improves over vanilla LoRA in 29 settings; it obtains the best or tied-best result in 13 of the 18 settings with $N \leq 64$ (Section 5.2). The alignment analysis is reported in Section 5.5 and Appendix E.7. Decoder-scale classification is reported in Section 5.3 and Appendix E.3, while summarization and full-data controls are reported in Appendices E.4 and E.5. Generated-answer reasoning is reported in Section 5.4 and Appendix D. Limitations are discussed in Appendix H.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Appendix H discusses limitations of activation-source and task scope, theoretical and diagnostic scope, model and task coverage, reasoning attribution and statistical scope, and pivot scope. In particular, ρ applies most directly to labeled classification, the theory is a regime-level idealization, the decoder experiments are narrower than the BERT study, and the reasoning experiments do not isolate direction as the sole causal factor. The paper also reports boundary cases with little or no benefit, including full-data classification (Appendix E.5) and GPT-2 summarization (Appendix E.4).

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.

- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Section 3 states the spiked-covariance model, defines the eigengap, and presents the Davis–Kahan regime-level scaling. Appendix B.2 derives this scaling from Davis–Kahan perturbation theory and effective-rank covariance concentration while stating the sub-Gaussian and idealized covariance assumptions. Appendix B.3 states and proves the first-step LoRA signal result, including the assumptions needed to relate the gradient-weighted class geometry to between-class scatter. Appendix B.4 states and proves the output-preservation property of the orthogonal pivot. The results are presented as regime-level motivation and local optimization analysis, not as finite-sample or convergence guarantees for transformer fine-tuning.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Section 5.1 specifies the evaluated models, tasks, sample sizes, shared training configuration, optimizer, learning rate, epochs, batch size, adapter rank, adapter targets, and method-specific LoRA scaling conventions. Appendix C provides hardware, precision, deterministic settings, seeds, baseline configurations, learning-rate allocations, and PivotLoRA implementation details. Algorithm 1 gives the classification PivotLoRA procedure. Appendix D provides the complete training, pivot, generation, evaluation, and per-seed protocols for generated-answer reasoning. Appendix E reports the complete BERT-base results, aggregate rankings, decoder-scale results, summarization and full-data controls, full-fine-tuning comparisons, and alignment analyses. Appendices F and G report robustness checks, ablations, per-seed values, and computational costs. Code, configurations, sampled-example manifests, result summaries, and available prediction artifacts are provided in the public repository identified in Appendix C.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Code, configurations, sampled-example manifests, result summaries, selected run artifacts, and available prediction artifacts are publicly available at `https://github.com/Nischal1011/pivotlora`. Appendix C documents the principal implementation settings, including hyperparameters, adapter targets, seeds, hardware, numerical precision, baseline configurations, and learning-rate policies. Appendix D documents the generated-answer reasoning protocol, and Algorithm 1 specifies the classification PivotLoRA procedure. The experiments use publicly accessible benchmark datasets and pretrained model checkpoints subject to their respective access conditions.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.

- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 5.1 reports the shared classification configuration, including rank $r = 8$, AdamW, the base learning-rate anchor, 15 epochs, batch size 16, and query and value adapter targets. Unless a baseline's standard parameterization specifies otherwise, the LoRA scale is $\alpha_{\mathrm{LoRA}} = 16$. Section 5.1 also specifies the evaluated models, datasets, and sample sizes and clarifies that N denotes the total number of labeled examples. Appendix C gives classification implementation details, including deterministic seeding with seeds 42, 43, and 44, adapter targets for each model family, baseline configurations, pivot epoch $K = 3$, eigenvalue-based learning-rate allocation, and the post-pivot learning-rate handling. Appendix D specifies the separate training, masking, generation, scoring, precision, and optimizer-state protocol used for generated-answer reasoning.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Appendix E.1 reports means and sample standard deviations across three seeds for all BERT-base methods and tasks. Appendix E.3 reports aggregate decoder-scale classification results, and Appendix G.1 provides the corresponding per-seed values. Appendix D.4 reports paired reasoning differences and their sample standard deviations across three paired seeds. Appendix F.1 reports an aggregate paired sign test over the 30 BERT-base task–N settings, with the stated caveat that these settings are not independent trials. Appendix F.2 provides five-seed confirmations and bootstrap confidence intervals for representative settings. Appendix E.7 reports the exact permutation test for the alignment analysis, while Appendix E.8 reports Spearman correlation, leave-one-task-out Pearson correlations, and a cross-split robustness check. Table 3 and Appendix D.4 report exact McNemar tests on per-question outcomes pooled over seeds; the paper does not claim seed-level significance for the three-seed reasoning experiments. Two decoder-scale entries (Llama-3.1-8B EVA and BoolQ) report population standard deviations because per-seed values were not retained, as stated in their captions.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix C reports the hardware and numerical precision used for the classification and summarization experiments. BERT-base and GPT-2 experiments use one NVIDIA T4 GPU, while decoder-scale classification uses one NVIDIA L4 GPU. Appendix C also reports that generated-answer reasoning uses one NVIDIA GeForce RTX 5090 with 31.8 GB of memory, together with the PyTorch and CUDA versions and model and adapter precision. Appendix G.2 reports initialization overhead, and Appendix G.3 reports representative end-to-end wall-clock time and peak memory. These measurements quantify the computational overhead of PivotLoRA relative to vanilla LoRA.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The work uses established pretrained models and NLP benchmarks accessed under their providers' terms. It does not involve human-subject experiments, private data, surveillance data, or the collection of sensitive personal information. The contribution is an initialization and reparameterization procedure for parameter-efficient model adaptation. The method's technical limitations and scope conditions are discussed in Appendix H.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: Parameter-efficient adaptation reduces the cost of few-shot specialization, which may also lower the cost of adapting pretrained models for harmful or deceptive uses. Appendix H discusses these impacts and recommends the same safety and misuse-mitigation practices that apply to the underlying pretrained models.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The paper does not introduce or release a new pretrained model, dataset, or other artifact identified as having a high risk of misuse. PivotLoRA is an initialization and reparameterization method applied to existing pretrained models. The released code does not contain model weights or datasets and does not alter the access restrictions or usage terms of the underlying assets.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The creators of the pretrained models and benchmark datasets are credited through their corresponding publications, including BERT [Devlin et al., 2019], GPT-2 [Radford et al., 2019], Llama-2 [Touvron et al., 2023], Llama-3.1 [Grattafiori et al., 2024], GLUE [Wang et al., 2019] and its constituent tasks [Socher et al., 2013, Warstadt et al., 2019, Williams et al., 2018, Rajpurkar et al., 2016, Dagan et al., 2006], AG News [Zhang et al., 2015], BoolQ [Clark et al., 2019], XSum [Narayan et al., 2018], CNN/DailyMail [Hermann et al., 2015, Nallapati et al., 2016], GSM8K [Cobbe et al., 2021], and StrategyQA [Geva et al., 2021]. The experiments use these assets for research evaluation under their respective access conditions. Appendix C lists the license or terms of use of each checkpoint and dataset, and no model weights or dataset contents are redistributed.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The paper does not introduce a new dataset or pretrained model, but it releases an implementation of PivotLoRA at `https://github.com/Nischal1011/pivotlora`. The method is specified in Section 4 and Algorithm 1, while implementation details are provided in Appendix C. The repository includes configurations, sampled-example manifests, result summaries, selected run artifacts, and available prediction artifacts.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing, human-subject experiments, user studies, annotation studies, or participant compensation.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The paper does not involve human-subject research, crowdsourcing, user studies, or the collection of data from participants. Institutional review board approval is therefore not applicable.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: Pretrained language models are the experimental subjects of the adaptation method, but no external LLM is used as an important, original, or non-standard component for generating data, labels, evaluations, or scientific results. Any LLM assistance limited to writing, editing, or formatting does not affect the methodology, experiments, or scientific claims and therefore does not require a separate declaration under the checklist guidance.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.