BARS-LoRA: Budget-Aware Rank Stabilization for Adaptive Low-Rank Adaptation

Anonymous authors
Paper under double-blind review

Abstract

Adaptive-rank LoRA decides during training where to spend a limited rank budget, and each decision is only as reliable as the evidence behind it. This evidence degrades in two distinct ways. Hard masks, as in AdaLoRA, commit rank from mini-batch salience estimates that remain noisy under short schedules and can leave entire modules without rank. Soft gates, as in SoRA, read their sparsity pattern from a rank-one dictionary that keeps drifting as training continues. We propose BARS-LoRA (Budget-Aware Rank Stabilization), which applies one principle, *stabilize the evidence before committing rank*, through two instantiations. SCALP-LoRA scores each hard channel by its first-order deletion effect, which we show is the common signed quantity underlying all three terms of the AdaLoRA triplet score; it discounts this score by its relative dispersion and allocates rank per Linear module with a guaranteed minimum rank. SAGE-SoRA orthogonalizes the gated dictionary and consolidates its basis before the sparse gate pattern is read out. The two branches specialize by training budget. On GLUE with Qwen3-0.6B, SCALP reaches 84.75% six-task accuracy under a four-epoch schedule, 1.15 points above AdaLoRA. At 800 updates, SAGE reaches 82.55% four-task accuracy, 1.64 points above SoRA. On Qwen3-1.7B, SAGE ranks first among seven methods on every task and exceeds DoRA by 1.74 points.

1 Introduction

Parameter-efficient fine-tuning (PEFT) has become the practical default for adapting large pretrained language models when many downstream tasks or users must be supported (Lialin et al., 2023). Low-rank adaptation (LoRA) is among the most widely adopted PEFT methods: it inserts trainable low-rank matrices into frozen linear transformations and can be merged into the original model at inference time (Hu et al., 2022). The standard LoRA update,

$$W = W^{(0)} + \Delta W = W^{(0)} + BA, \tag{1}$$

builds on the observation that adaptation updates tend to lie in a low-dimensional subspace (Aghajanyan et al., 2021). It also fixes one design choice by hand: every adapted matrix receives the same rank.

Adaptive-rank methods lift this restriction from two directions. AdaLoRA writes the update as an SVD-style residual and prunes singular channels by importance (Zhang et al., 2023); SoRA places sparse gates on rank channels and updates them with a proximal operator (Ding et al., 2023). Both turn rank selection into an online decision: at each mask event or proximal step, a mini-batch estimate determines which channels keep capacity. The final adapter is therefore shaped by how reliable that estimate is at the moment it is used.

The two families face different reliability problems. A hard mask removes channels on the basis of the salience evidence accumulated so far. Under short schedules this evidence is still dominated by mini-batch noise, and global triplet-level pruning can turn the noise into fragmented rank layouts or Linear modules that receive no rank at all. Soft gates avoid irreversible pruning but read their decision from a moving target: the rank-one directions they select keep changing after the gates sparsify, so the final nonzero pattern reflects a basis that is still evolving.

We address both problems with a single principle, *stabilize the evidence before committing rank*, and instantiate it as BARS-LoRA, a budget-aware rank-stabilization framework with two branches.

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

SCALP-LoRA (Section 3.2) stabilizes AdaLoRA-style hard masks: it replaces triplet aggregation with a channel-level Taylor score, discounts the score by its relative dispersion, allocates rank at Linear granularity with a minimum-rank guarantee, and consolidates the final layout. SAGE-SoRA (Section 3.3) stabilizes SoRA-style soft gates: it regularizes the geometry of the rank-one dictionary and consolidates the basis before the gate pattern is read out.

On GLUE with Qwen3-0.6B, stabilizing each family reveals a clear division of labor over the training budget (Figure 2). SAGE-SoRA leads at every fixed horizon from 800 to 3000 updates, while SCALP-LoRA improves monotonically across these uniform budgets and leads at the task-scaled four-epoch endpoint. This pattern is consistent with the two failure modes: soft gating is most effective before the dictionary has had time to drift, and hard masking benefits from the salience evidence that accumulates over longer schedules. The choice of rank-allocation mechanism therefore depends on the training budget.

Our contributions are as follows.

- We cast adaptive rank allocation as committing budget from online evidence, identify noisy salience in hard masks and basis drift in soft gates as two distinct sources of unreliable rank evidence, and show a budget-dependent crossover between the two stabilized families.
- We propose SCALP-LoRA. Its channel score rests on an identity (Eq. (10)) showing that the three terms of the AdaLoRA triplet score are signed contractions of a single channel-deletion derivative. The score is combined with relative-dispersion discounting and Linear-level allocation that keeps at least one rank in every target module.
- We propose SAGE-SoRA, which uses the factorization of the inner product between rank-one maps (Eq. (21)) to motivate dictionary orthogonalization and consolidates the basis before reading out the sparse gates.
- We evaluate both branches on GLUE against eight PEFT baselines and a tuned fixed-rank sweep. SCALP gives the best four-epoch six-task average on Qwen3-0.6B (84.75%), SAGE gives the best 800-update four-task average on both Qwen3-0.6B and Qwen3-1.7B (82.55% and 86.52%), and the complete mechanism attains the highest mean in each component study.

2 Related Work

Parameter-efficient fine-tuning. A long line of work reduces the number of trainable parameters while keeping the backbone frozen. Adapters and AdapterFusion insert small bottleneck modules between transformer sublayers (Houlsby et al., 2019; Pfeiffer et al., 2021); prefix and prompt tuning learn continuous token embeddings (Li & Liang, 2021; Lester et al., 2021; Liu et al., 2022b); BitFit and IA^3 update only biases or learned scaling vectors (Ben Zaken et al., 2022; Liu et al., 2022a); compact hypercomplex adapters share parameters via Kronecker structure (Karimi Mahabadi et al., 2021); QLoRA combines LoRA with quantized backbones (Dettmers et al., 2023). Lialin et al. (2023) provide a broader survey. Because LoRA's low-rank residual can be merged into the pre-trained matrix at inference time, rank is the structural unit of its adaptation capacity (Hu et al., 2022).

The LoRA family and adaptive rank. The two closest prior works are AdaLoRA (Zhang et al., 2023), which prunes singular channels of an SVD-style residual via a triplet importance score, and SoRA (Ding et al., 2023), which sparsifies rank-channel gates with a proximal operator. We build directly on both: Section 3.2 derives a channel-level loss derivative that unifies the coordinate terms of the AdaLoRA triplet, and Section 3.3 stabilizes the rank-one dictionary underlying SoRA's gates. SalientLoRA (Ke et al., 2024) analyzes intrinsic rank through salience and commits allocation directly from it, whereas SCALP filters salience by its reliability before allocation. Another line changes how each low-rank update is parameterized or optimized: LoRA+ uses separate factor learning rates (Hayou et al., 2024), DoRA separates weight magnitude and direction (Liu et al., 2024), PiSSA initializes adaptation in principal singular subspaces (Meng et al., 2024), and rsLoRA changes the rank-dependent scaling (Kalajdzievski, 2023). BARS-LoRA acts instead on how rank is allocated, and our experiments compare it with all of these methods under the four-epoch regime, and with DoRA, PiSSA, and rsLoRA also at 800 updates.

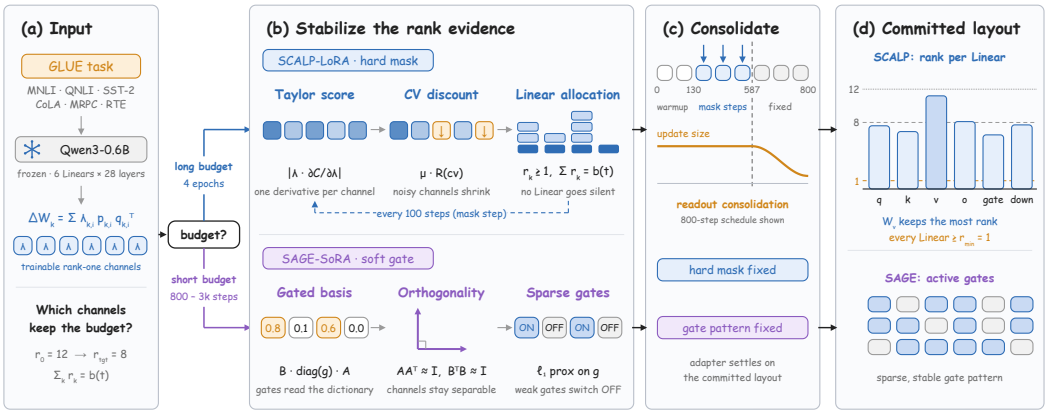


Figure 1: Overview of BARS-LoRA. (a) Each target Linear of the frozen backbone carries a dictionary of trainable rank-one channels. The training budget selects the branch: SCALP-LoRA for the four-epoch schedule and SAGE-SoRA for fixed horizons of 800–3000 updates. (b) Spatial stabilization. SCALP scores channels by Taylor salience, discounts high-dispersion evidence, and allocates the scheduled budget per Linear with $r_{\min} = 1$ at every mask step. SAGE separates the gated dictionary with orthogonality and sparsifies gates with an ℓ_1 proximal step. (c) Temporal stabilization. Readout consolidation settles the adapter once the mask or gate pattern is fixed (800-update schedule shown). (d) The committed output: the per-Linear SCALP rank layout on MNLI (seed 42, four-epoch schedule), where W_v keeps the most rank, and a schematic sparse SAGE gate pattern.

Importance scoring and structured pruning. SCALP-LoRA's scoring view comes from structured pruning. Optimal Brain Damage estimates deletion cost via curvature (LeCun et al., 1989), and first-order Taylor pruning approximates the loss change of removing a structured unit (Molchanov et al., 2017). Later work studies uncertainty in importance estimation and movement-based pruning during fine-tuning (Molchanov et al., 2019; Sanh et al., 2020), and lottery-ticket findings highlight the lasting effect of early structural choices (Frankle & Carbin, 2019). SparseGPT and LLM-Pruner apply these ideas to large language models (Frantar & Alistarh, 2023; Ma et al., 2023). BARS-LoRA brings Taylor salience to online rank reallocation during PEFT, where allocation decisions are repeated throughout training and the temporal reliability of importance estimates becomes a central design constraint.

3 BARS-LoRA Method

BARS-LoRA has two instantiations that share one design principle: stabilize the rank evidence before committing budget. SCALP-LoRA (Section 3.2) stabilizes hard AdaLoRA-style masks through a single Taylor channel score, a relative-dispersion reliability discount, Linear-level allocation with a minimum rank of one, and temporal consolidation before the final layout is read out. SAGE-SoRA (Section 3.3) stabilizes soft SoRA-style gates through orthogonality regularization on the rank-one dictionary and consolidation of the non-gate basis before the gate pattern is read out. Both branches stabilize rank evidence along two axes: *spatially*, by controlling how evidence is scored, filtered, and distributed across channels and modules, and *temporally*, by consolidating that evidence before it is committed. Figure 1 summarizes both branches. We first introduce the SVD-channel parameterization shared with AdaLoRA.

3.1 Preliminaries

SVD-based rank channels. For an adapted matrix indexed by k, AdaLoRA writes the residual as an SVD-like sum of rank-one channels:

$$\Delta W_k = P_k \operatorname{diag}(\lambda_k) Q_k = \sum_i \lambda_{k,i} p_{k,i} q_{k,i}^\top. \tag{2}$$

Setting $\lambda_{k,i} = 0$ deactivates the i-th rank-one map while retaining its basis parameters. The allocation rule determines which channels receive rank budget at each mask event.

The AdaLoRA triplet score. AdaLoRA scores the triplet $\mathcal{G}_{k,i} = \{p_{k,i}, \lambda_{k,i}, q_{k,i}\}$ by aggregating per-entry importances $s(\theta) = |\theta\, \partial C/\partial\theta|$ over the scalar singular value and the two singular-vector blocks:

$$S^{\mathrm{Ada}}_{k,i} = s(\lambda_{k,i}) + \frac{1}{d_1}\sum_j s(P_{k,ji}) + \frac{1}{d_2}\sum_j s(Q_{k,ij}), \tag{3}$$

where d_1, d_2 are the row and column dimensions of P_k and Q_k, respectively. SCALP-LoRA asks how these coordinate-level terms relate to the deletion derivative of the complete rank-one channel.

3.2 SCALP-LoRA: Stabilizing Hard Rank Allocation

Single-channel salience via Taylor expansion. Let C denote the mini-batch task loss and $G_k = \partial C/\partial \Delta W_k$ its gradient with respect to the low-rank residual. Removing the i-th singular channel of ΔW_k (Eq. (2)) corresponds to the perturbation

$$\delta \Delta W_{k,i} = -\lambda_{k,i} p_{k,i} q_{k,i}^\top. \tag{4}$$

A first-order multivariate Taylor expansion of C around the current residual gives

$$\Delta C_{k,i} \approx \langle G_k, \delta \Delta W_{k,i}\rangle_F = -\lambda_{k,i} p_{k,i}^\top G_k q_{k,i}, \tag{5}$$

so the magnitude of the first-order channel-deletion salience is

$$T_{k,i} = \left|\lambda_{k,i} p_{k,i}^\top G_k q_{k,i}\right|. \tag{6}$$

Equation (6) measures the sensitivity of the loss to removing the entire rank-one map at once, rather than to perturbing its $\lambda_{k,i}$, $p_{k,i}$, and $q_{k,i}$ entries independently.

To connect this channel-level view with AdaLoRA-style triplet scoring, we apply the chain rule to the same rank-one channel:

$$\frac{\partial C}{\partial \lambda_{k,i}} = p_{k,i}^\top G_k q_{k,i}, \tag{7}$$

$$\frac{\partial C}{\partial p_{k,i}} = \lambda_{k,i} G_k q_{k,i}, \tag{8}$$

$$\frac{\partial C}{\partial q_{k,i}} = \lambda_{k,i} G_k^\top p_{k,i}. \tag{9}$$

Forming the scalar $\theta \cdot \partial C/\partial\theta$ for each $\theta \in \{\lambda_{k,i}, p_{k,i}, q_{k,i}\}$, with the dot interpreted as an inner product for vectors, yields three identical values:

$$\lambda_{k,i}\frac{\partial C}{\partial \lambda_{k,i}} = p_{k,i}^\top \frac{\partial C}{\partial p_{k,i}} = q_{k,i}^\top \frac{\partial C}{\partial q_{k,i}} = \lambda_{k,i} p_{k,i}^\top G_k q_{k,i}. \tag{10}$$

Equation (10) shows that the three coordinate groups of the AdaLoRA triplet carry the same first-order channel sensitivity, contracted along different coordinates. The entry-wise absolute values and dimension-wise averaging in Eq. (3) discard the signed cancellations within each contraction and make the score depend on how evidence is spread across coordinates. SCALP-LoRA instead scores the scalar channel switch directly:

$$s^{(0)}_{k,i} = \left|\lambda_{k,i}\frac{\partial C}{\partial \lambda_{k,i}}\right|. \tag{11}$$

The rank-one geometry of $p_{k,i}$ and $q_{k,i}$ is retained inside $\partial C/\partial\lambda_{k,i} = p_{k,i}^\top G_k q_{k,i}$, so this single quantity equals the deletion salience of Eq. (6). The score uses task-loss gradients and parameter values evaluated at the same model state. Computing Eq. (11) adds no backward pass: $\partial C/\partial\lambda_{k,i}$ is already produced by the standard gradient computation, so the score costs one elementwise product per channel. Because the three contractions in Eq. (10) coincide, scoring the scalar switch also removes the need to choose relative weights for the singular-value and singular-vector terms.

Relative-dispersion reliability discounting. The instantaneous score in Eq. (11) fluctuates across mini-batches. SCALP-LoRA maintains an exponential moving average of salience and a corresponding absolute-deviation estimate:

$$\mu_{k,i}^{(t)} = \beta_1 \mu_{k,i}^{(t-1)} + (1-\beta_1) s_{k,i}^{(0,t)}, \tag{12}$$

$$u_{k,i}^{(t)} = \beta_2 u_{k,i}^{(t-1)} + (1-\beta_2) \left| s_{k,i}^{(0,t)} - \mu_{k,i}^{(t)} \right|. \tag{13}$$

AdaLoRA multiplies uncertainty into smoothed importance to encourage exploration. SCALP-LoRA uses relative fluctuation for the opposite purpose: to discount unreliable evidence near the pruning boundary. Its CV-style (coefficient-of-variation) statistic is the relative smoothed absolute deviation

$$\mathrm{cv}_{k,i}^{(t)} = \frac{u_{k,i}^{(t)}}{\mu_{k,i}^{(t)} + \varepsilon}. \tag{14}$$

Normalizing by the mean penalizes a channel when its fluctuation is large relative to its estimated signal, independent of the channel's absolute scale. A bounded reliability discount

$$R(\mathrm{cv}) = \rho + (1-\rho)\exp[-\gamma \max(0, \mathrm{cv} - \tau)] \tag{15}$$

gives the final channel score

$$S_{k,i}^{\mathrm{SCALP}} = \mu_{k,i} R(\mathrm{cv}_{k,i}). \tag{16}$$

Stable, high-salience channels retain their mean score, and high-fluctuation channels receive a discount bounded below by ρ. Here $\beta_1, \beta_2 \in [0, 1)$ are smoothing factors, τ is a CV tolerance below which no discount is applied, $\gamma > 0$ controls the discount slope, $\rho \in (0, 1]$ is the floor, and ε is a small stabilizer. The discount satisfies $\rho \le R \le 1$ and is non-increasing in cv, so it can reorder channels whose mean scores are close but cannot overturn a large salience gap; it therefore acts mainly on channels near the pruning boundary, where noisy evidence is most likely to change the mask.

Module-level allocation with a minimum-rank guarantee. AdaLoRA prunes through global competition among channels, which can assign zero rank to an individual Linear module. SCALP-LoRA separates structural allocation from within-module selection by aggregating stabilized evidence to the Linear level:

$$\Phi_k = \frac{1}{r_k} \sum_i S_{k,i}^{\mathrm{SCALP}}. \tag{17}$$

The module potential determines capacity allocation, while the individual scores $S_{k,i}^{\mathrm{SCALP}}$ select channels within each Linear. Given the scheduled rank budget $b(t)$, the module ranks, bounded by the initial rank r_0, satisfy

$$r_{\min} \le r_k \le r_0, \qquad \sum_k r_k = b(t), \qquad r_{\min} = 1. \tag{18}$$

The one-channel minimum keeps an adapter path in every target Linear. Exact-budget repair enforces the total rank budget, and the top r_k channels form each module's hard mask. Once the mask is fixed, readout consolidation (Eq. (23), defined in Section 3.3) is applied so that the adapter settles on the committed layout before it is read out. Algorithm 1 in Appendix A summarizes SCALP-LoRA.

3.3 SAGE-SoRA: Stabilizing Soft Rank Gating

SoRA learns sparse gates over rank channels:

$$\Delta W_k = B_k \,\mathrm{diag}(g_k) A_k = \sum_i g_{k,i} b_{k,i} a_{k,i}^\top. \tag{19}$$

The gradient of the task loss with respect to a gate is

$$\frac{\partial C}{\partial g_{k,i}} = \langle G_k, b_{k,i} a_{k,i}^\top \rangle_F. \tag{20}$$

Each gate reads out one rank-one direction. When two directions are highly correlated, a sparsity update can suppress one gate while the other absorbs its function, so the selected pattern depends on the optimization trajectory.

The Frobenius inner product between two rank-one gated maps factorizes as

$$\langle b_i a_i^\top, b_j a_j^\top \rangle_F = (b_i^\top b_j)(a_i^\top a_j), \tag{21}$$

so reducing redundancy in either the input- or output-side basis reduces cross-channel interference. SAGE-SoRA adds

$$\mathcal{L}_{\text{orth}} = \sum_k \left(\left\| A_k A_k^\top - I \right\|_F + \left\| B_k^\top B_k - I \right\|_F \right). \tag{22}$$

Driving both Gram matrices toward the identity pushes the cross-channel inner products in Eq. (21) toward zero and separates the dictionary elements on which sparsity acts.

The second source of instability is temporal: A_k and B_k may keep moving after the gates become sparse, so the dictionary on which the gates act is still changing when the sparse pattern is read out. SAGE-SoRA therefore applies *readout consolidation*, which contracts the update magnitude of the non-gate basis over the final phase:

$$\eta_t = \eta_{\min} + \tfrac{1}{2}(\eta_0 - \eta_{\min}) \left(1 + \cos(\pi p_t)\right), \tag{23}$$

where $p_t \in [0, 1]$ denotes normalized progress through the consolidation phase and η_t is the step size applied to the non-gate basis, decaying from the base learning rate η_0 to the floor $\eta_{\min}$. Orthogonality reduces the substitutability between channels, and consolidation suppresses basis drift before the gate pattern is finalized. Together they fix both the geometry and the motion of the dictionary before the final sparse adapter is selected. Algorithm 2 in Appendix A summarizes SAGE-SoRA.

4 Experiments

4.1 Setup

Backbone and data. We use Qwen3-0.6B in fp16 as the frozen backbone (Yang et al., 2025), with adapters in all six target Linear projections per layer ($W_q, W_k, W_v, W_o, W_{\text{gate}}, W_{\text{down}}$), initial hard rank $r_0 = 12$, and target rank $r_{\text{tgt}} = 8$. We evaluate on six GLUE tasks (Wang et al., 2019). MNLI, QNLI, and SST-2 are subsampled to 10,000 training examples with a fixed seed, and CoLA, MRPC, and RTE use their full training splits; evaluation uses the full official validation splits. The cap reduces data-scale imbalance across tasks and defines the task-scaled long schedule. All tables report prompt-classification accuracy, including for CoLA and MRPC; Avg-6 and Avg-4 are unweighted task means.

Training budgets. The short regime uses a uniform 800 optimizer updates per task, with warmup 130 and a final consolidation phase of 213 updates; the budget study adds uniform 1600 and 3000 updates. The long regime makes four passes over each task's capped training set: 5000 optimizer updates for MNLI, QNLI, and SST-2, 4300 for CoLA, 1800 for MRPC, and 1200 for RTE. Its endpoint is therefore task-scaled rather than uniformly longer than 3000 updates.

Optimization and baselines. We use AdamW, maximum length 128, effective batch size eight (microbatch one, accumulation eight), and seeds $\{42, 43, 44\}$. The adaptive branches use learning rate (LR) $\eta_0 = 3 \times 10^{-4}$, consolidation floor $\eta_{\min} = 3 \times 10^{-5}$, and mask interval 100. SCALP uses $\beta_1 = \beta_2 = 0.85$, $\rho = 0.85$, $\gamma = 1$, $\tau = 0.3$, and $\varepsilon = 10^{-8}$; SAGE uses $\lambda_{\text{orth}} = 0.1$ and gate proximal threshold $\lambda_{\text{sp}} = 3 \times 10^{-4}$. Baselines are LoRA, AdaLoRA, SoRA, SalientLoRA, LoRA+, DoRA, PiSSA, and rsLoRA. Fixed LoRA is swept over $r \in \{4, 8, 12, 16\}$ and LR $\in \{10^{-4}, 2 \times 10^{-4}, 3 \times 10^{-4}, 5 \times 10^{-4}\}$, with one LR selected globally per rank and regime by the cross-task macro. Rank eight matches the nominal target rank, and ranks four, twelve, and sixteen provide lower- and higher-capacity controls. DoRA, PiSSA, and rsLoRA use their native parameterizations with $r = 8$, $\alpha = 16$, and a global LR selected from $\{10^{-4}, 3 \times 10^{-4}\}$. Selected rates and complete control results appear in Appendix B.

Table 1: Main results on Qwen3-0.6B under the four-epoch task-scaled schedule. All task scores are accuracy; subscripts are three-seed standard deviations. Bold marks the largest mean in each column. Fixed LoRA, DoRA, PiSSA, and rsLoRA use their selected global learning rates; LoRA+ uses base LR 10^{-4} and factor-LR ratio four. Additional fixed ranks appear in Table 5.

Method	MNLI	QNLI	SST-2	CoLA	MRPC	RTE	Avg-6
SCALP-LoRA	$82.87_{0.08}$	$\mathbf{90.11_{0.42}}$	$92.62_{0.63}$	$81.08_{0.53}$	$88.07_{0.62}$	$\mathbf{73.77_{1.99}}$	$\mathbf{84.75_{0.39}}$
SAGE-SoRA	$82.47_{0.33}$	$89.30_{0.37}$	$91.78_{0.29}$	$81.62_{0.53}$	$\mathbf{89.22_{0.88}}$	$73.41_{0.91}$	84.63
SoRA	$82.26_{0.27}$	$89.27_{0.68}$	$92.32_{0.80}$	$81.59_{1.54}$	$87.83_{1.23}$	$\mathbf{73.77_{2.54}}$	84.50
rsLoRA	$82.43_{0.48}$	$88.30_{1.80}$	$91.51_{1.02}$	$81.43_{0.77}$	$89.05_{0.93}$	$73.41_{1.67}$	$84.36_{0.42}$
DoRA	$81.75_{1.23}$	$88.50_{2.08}$	$\mathbf{92.66_{0.20}}$	$81.53_{1.06}$	$87.25_{1.30}$	$72.56_{2.25}$	$84.04_{0.34}$
LoRA ($r = 8$)	$81.76_{0.24}$	$88.38_{0.65}$	$92.09_{0.69}$	$\mathbf{81.85_{1.88}}$	$87.58_{1.39}$	$72.20_{0.36}$	$83.98_{0.50}$
SalientLoRA	$82.63_{0.41}$	$88.51_{1.71}$	$91.55_{0.26}$	$81.24_{0.78}$	$87.42_{1.23}$	$72.20_{3.44}$	83.93
AdaLoRA	$\mathbf{82.99_{0.26}}$	$87.81_{2.85}$	$92.01_{1.04}$	$81.30_{0.48}$	$85.78_{2.83}$	$71.72_{1.16}$	83.60
PiSSA	$79.74_{0.88}$	$87.18_{1.87}$	$91.02_{0.18}$	$79.99_{0.92}$	$84.89_{2.14}$	$\mathbf{73.77_{1.82}}$	$82.76_{0.37}$
LoRA+	$80.87_{0.48}$	$88.44_{0.91}$	$89.41_{2.24}$	$80.44_{1.39}$	$85.95_{0.99}$	$70.88_{2.08}$	82.66

4.2 Main Results: Long-Budget Regime

Table 1 reports the long-budget comparison. SCALP-LoRA attains the highest six-task mean, 84.75 ± 0.39, followed by SAGE-SoRA (84.63) and SoRA (84.50), the strongest external baseline. SCALP improves on AdaLoRA, its hard-mask counterpart, by 1.15 percentage points (pp), and its lead holds against every tuned control: 0.39 pp over rsLoRA (84.36 ± 0.42), 0.57 pp over the best fixed rank ($r = 4$, 84.18 ± 0.71), 0.67 pp over the higher-capacity $r = 12$ (84.08), and 0.77 pp over rank-matched LoRA ($r = 8$, 83.98 ± 0.50). The learning-rate sweep substantially strengthens fixed-rank LoRA, and SCALP remains ahead of every tuned result.

The gains over AdaLoRA concentrate where AdaLoRA is least stable across seeds. SCALP improves QNLI by 2.30 pp while reducing the standard deviation from 2.85 to 0.42, MRPC by 2.29 pp with $2.83 \rightarrow 0.62$, and RTE by 2.05 pp. On MNLI and CoLA, where AdaLoRA already varies little across seeds, the two methods are within 0.25 pp. This task profile is consistent with gains that come from more reliable allocation evidence rather than from additional capacity.

4.3 Budget Crossover

Figure 2 tracks four allocation rules on MNLI, SST-2, CoLA, and RTE from 800 to 3000 uniform updates and at the four-epoch endpoint. The two branches specialize. SAGE-SoRA has the highest macro mean at every uniform budget and peaks at 83.75 at 1600 updates. SCALP-LoRA rises monotonically from 80.50 at 800 updates to 81.40 and 82.28 at 1600 and 3000 updates, and reaches 82.58 at the task-scaled four-epoch endpoint, where it leads all four rules. This profile is consistent with the two failure modes: the hard-mask branch gains from the salience evidence that accumulates with longer training, whereas the soft-gate branch is strongest before its dictionary has had time to drift, and its mean is 82.32 at the four-epoch endpoint, below its 83.75 peak. At 800 updates, SAGE improves on AdaLoRA by 3.58 pp and on SoRA by 1.64 pp.

SAGE's short-budget lead also holds against the tuned controls, including higher-capacity fixed ranks (Table 6). It reaches 82.55 ± 0.22, compared with 82.15 ± 0.35 for the strongest fixed rank ($r = 16$), 81.76 ± 1.00 for $r = 4$, and 81.53 ± 1.39 for rsLoRA, and it exceeds all seven tuned controls at 800 updates.

4.4 Ablations

Table 2 evaluates each mechanism as a whole and through its components. In SCALP, the structural stages determine *which* layout is committed: the Taylor score defines the channel sensitivity being tracked, relative-dispersion filtering weights fluctuating evidence, Linear-level allocation sets module capacity, and $r_{\min} = 1$ keeps a channel in every target module. Temporal consolidation then lets the adapter settle on the committed layout. The components are complementary: the Linear-level scaffold alone improves on AdaLoRA (82.13 vs. 82.01), and consolidation is most effective on

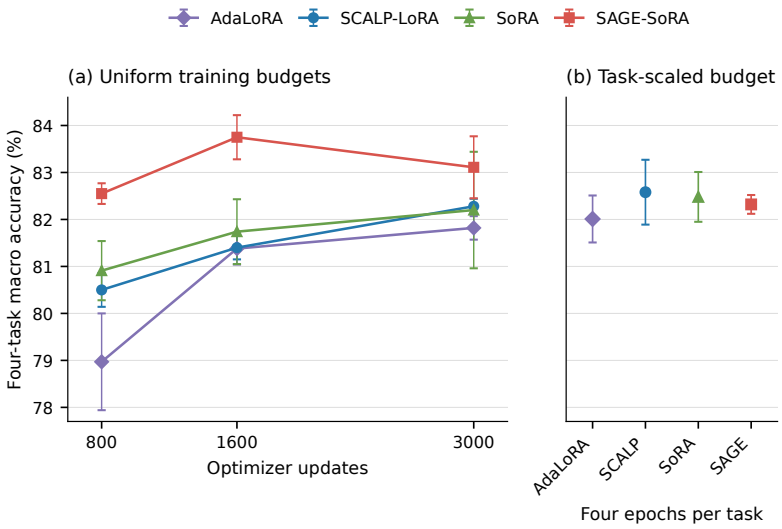


Figure 2: Four-task macro accuracy (MNLI, SST-2, CoLA, RTE) across training budgets. (a) Uniform 800/1600/3000-update schedules. (b) Task-scaled four-epoch endpoint. The soft-gate branch SAGE-SoRA leads at every uniform budget; the hard-mask branch SCALP-LoRA improves monotonically across uniform budgets and leads at the task-scaled four-epoch endpoint. Error bars are three-seed macro standard deviations; values are listed in Table 4.

Table 2: Component studies over three seeds, each using the baseline runs recorded for that study. (a) SCALP rows 1–6 add components cumulatively; the last two rows apply temporal consolidation or the Linear-level scaffold alone to AdaLoRA. (b) SAGE uses a 2×2 split of orthogonality and consolidation. In both studies the full configuration attains the highest macro mean. Figure 5 visualizes both studies.

Stage	MNLI	SST-2	CoLA	RTE	Avg-4
(a) SCALP-LoRA: four-epoch task-scaled					
AdaLoRA	$82.99_{0.26}$	$92.01_{1.04}$	$81.30_{0.48}$	$71.72_{1.16}$	82.01
+ Taylor	$82.78_{0.37}$	$91.90_{1.13}$	$81.14_{0.77}$	$71.48_{2.60}$	81.83
+ CV	$82.86_{0.28}$	$92.24_{0.85}$	$81.02_{0.53}$	$71.36_{1.99}$	81.87
+ Linear	$83.09_{0.31}$	$91.97_{0.61}$	$80.79_{1.60}$	$71.60_{1.99}$	81.86
+ $r_{\min} = 1$	$82.56_{0.10}$	$92.05_{0.75}$	$80.60_{0.56}$	$72.44_{1.27}$	81.91
+ Consolidation (full)	$82.87_{0.08}$	$92.62_{0.63}$	$81.08_{0.53}$	$73.77_{1.99}$	**82.58**
Consolidation only	$82.84_{0.39}$	$92.43_{0.41}$	$81.50_{0.48}$	$73.16_{1.27}$	82.48
Linear+$r_{\min}$ only	$83.02_{0.25}$	$92.05_{0.43}$	$80.63_{1.00}$	$72.80_{1.37}$	82.13

Stage	MNLI	SST-2	CoLA	RTE	Avg-4
(b) SAGE-SoRA: 800 uniform updates					
SoRA	$80.32_{2.47}$	$90.98_{0.67}$	$78.81_{0.66}$	$73.53_{0.91}$	80.91
Orth only	$80.27_{3.20}$	$91.51_{0.64}$	$78.72_{0.60}$	$71.84_{0.96}$	80.58
Consolidation only	$83.01_{0.20}$	$91.86_{0.23}$	$80.28_{0.74}$	$74.37_{2.20}$	82.38
Both (full)	$83.00_{0.26}$	$92.24_{0.24}$	$80.22_{0.56}$	$74.73_{0.96}$	**82.55**

the stabilized layout, adding 0.67 pp there versus 0.47 pp on AdaLoRA's layout. The full pipeline reaches 82.58, the highest mean in the study.

The SAGE study shows the same interplay between geometry and time. Orthogonality is effective in combination with consolidation: adding it on top of consolidation raises the mean from 82.38 to 82.55, the highest in the study and 1.64 pp above SoRA. This is consistent with the view that separating dictionary elements matters most once those elements stop moving. The combination also more than halves the RTE seed spread of consolidation alone (2.20 $\rightarrow$ 0.96) and reduces the MNLI standard deviation of SoRA from 2.47 to 0.26, nearly a tenfold reduction in seed sensitivity.

Table 3: Qwen3-1.7B at 800 uniform updates: seven methods, four tasks, and three seeds (84 runs). Subscripts denote standard deviations.

Method	MNLI	SST-2	CoLA	RTE	Avg-4
SAGE-SoRA	$\mathbf{85.89_{0.57}}$	$\mathbf{95.34_{0.29}}$	$\mathbf{83.38_{0.06}}$	$\mathbf{81.47_{0.91}}$	$\mathbf{86.52_{0.14}}$
DoRA	$85.17_{1.20}$	$92.70_{2.84}$	$83.03_{1.16}$	$78.22_{1.82}$	$84.78_{0.47}$
SCALP-LoRA	$84.59_{0.20}$	$94.27_{0.12}$	$79.42_{0.29}$	$79.54_{0.83}$	$84.46_{0.16}$
LoRA ($r = 8$)	$82.21_{4.64}$	$94.84_{0.99}$	$82.61_{0.71}$	$77.02_{0.55}$	$84.17_{1.20}$
AdaLoRA	$84.43_{0.17}$	$94.19_{0.18}$	$78.81_{1.91}$	$78.70_{1.57}$	$84.03_{0.68}$
PiSSA	$80.18_{0.55}$	$88.95_{1.04}$	$74.75_{5.24}$	$68.95_{9.08}$	$78.21_{2.72}$
rsLoRA	$81.95_{3.29}$	$77.06_{24.51}$	$77.79_{4.30}$	$63.54_{12.92}$	$75.09_{5.16}$

Effects beyond the endpoint mean. Stabilization also controls properties that a macro mean does not capture. The minimum-rank constraint provides a structural guarantee: in a pilot trajectory, the unrepaired allocation would have assigned zero rank to a target Linear with a counterfactual risk of 32.74%, whereas with $r_{\min} = 1$ every module retained at least one channel throughout training. Stabilization also reduces seed sensitivity. Relative to AdaLoRA, SCALP lowers the QNLI standard deviation from 2.85 to 0.42 and the MRPC standard deviation from 2.83 to 0.62, and full SAGE lowers SoRA's MNLI standard deviation from 2.47 to 0.26. Finally, Linear-level allocation yields an interpretable layout. SCALP's ranks change while the budget decays and flatten once the mask is fixed, and the converged layout retains substantially more rank in W_v than in W_{gate} (Figure 6). SAGE's gate magnitudes evolve during the sparse-update phase, and their change rate declines toward the final readout (Figure 7).

4.5 Larger-Backbone Validation

We extend the 800-update comparison to Qwen3-1.7B with seven methods, four tasks, and three seeds, completing all $7 \times 4 \times 3 = 84$ runs (Table 3). SAGE-SoRA reaches 86.52 ± 0.14, ranks first on every task, and exceeds the strongest competing method, DoRA (84.78 ± 0.47), by 1.74 pp. Its margins over SCALP, rank-eight LoRA, and AdaLoRA are 2.06, 2.35, and 2.49 pp, respectively. SAGE also has the smallest macro standard deviation of all seven methods. The short-budget advantage of the soft-gate branch therefore carries over from the 0.6B to the 1.7B backbone.

5 Conclusion

BARS-LoRA treats adaptive rank selection as a problem of committing budget from evidence whose reliability changes during training. Its two branches apply one principle, *stabilize the evidence before committing rank*, to the two ways this evidence fails. SCALP-LoRA scores hard channels with a single Taylor derivative that unifies the AdaLoRA triplet (Eq. (10)), filters it by relative dispersion, and guarantees rank in every module; SAGE-SoRA separates and consolidates the soft-gated dictionary. Each branch leads in its regime against strong PEFT baselines and a tuned fixed-rank sweep: SCALP under the four-epoch schedule, and SAGE at 800 updates on both Qwen3-0.6B and Qwen3-1.7B. The resulting budget crossover makes training budget a design input for adaptive rank allocation, with hard masking favored when evidence has time to accumulate and soft gating favored when it does not.

6 Limitations

Our evaluation covers Qwen3-0.6B and Qwen3-1.7B on GLUE prompt classification with three seeds per configuration; generation, instruction tuning, and multilingual settings are natural next targets. The branch is currently chosen by the training budget. Appendix D describes a paired-probe diagnostic for choosing it automatically; measuring its selection accuracy, regret, and compute overhead across held-out tasks is the next step.

AI USE STATEMENT

In this work, we used generative AI tools for translation, interpretation and consistency checking of experimental summaries, feedback on experimental methodology, and the exposition of the mathematical derivations in Section 3, i.e., wording and presentation of derivations formulated by the authors. We have not used generative AI tools for generating synthetic data, formulating mathematical claims, providing proof ingredients, proposing hypotheses, or cleaning datasets, and qualitative or thematic data analysis is not applicable to this work. Additionally, we used generative AI tools for manuscript editing, literature lookup, figure preparation, and LaTeX formatting. We have reviewed all AI-assisted work: derivations were re-checked by hand, and every reported number was re-checked against the result logs. The experimental results reported here come from the authors' experiments. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work uses the publicly available GLUE benchmark and openly released Qwen3 model weights. It involves no human subjects, personal data, or newly collected data.

REPRODUCIBILITY STATEMENT

Section 3 provides the derivation, and Appendix A gives both algorithms; Section 4.1 describes the training and tuning protocols. Appendix B reports selected learning rates and the complete controls, and Appendix D describes the paired diagnostic protocol.

REFERENCES

Armen Aghajanyan, Sonal Gupta, and Luke Zettlemoyer. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 7319–7328, 2021.

Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. BitFit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 1–9, 2022.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. QLoRA: Efficient finetuning of quantized LLMs. In *Advances in Neural Information Processing Systems*, volume 36, pp. 10088–10115, 2023.

Ning Ding, Xingtai Lv, Qiaosen Wang, Yulin Chen, Bowen Zhou, Zhiyuan Liu, and Maosong Sun. Sparse low-rank adaptation of pre-trained language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 4133–4145, 2023.

Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rJl-b3RcF7.

Elias Frantar and Dan Alistarh. SparseGPT: Massive language models can be accurately pruned in one-shot. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 10323–10337, 2023.

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*, pp. 17783–17806, 2024. URL https://proceedings.mlr.press/v235/hayou24a.html.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2790–2799, 2019.

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

Damjan Kalajdzievski. A rank stabilization scaling factor for fine-tuning with LoRA. *arXiv preprint arXiv:2312.03732*, 2023. URL https://arxiv.org/abs/2312.03732.

Rabeeh Karimi Mahabadi, James Henderson, and Sebastian Ruder. Compacter: Efficient low-rank hypercomplex adapter layers. In *Advances in Neural Information Processing Systems*, volume 34, pp. 1022–1035, 2021.

Wenjun Ke, Jiahao Wang, Peng Wang, Jiajun Liu, Dong Nie, Guozheng Li, and Yining Li. Unveiling LoRA intrinsic ranks via salience analysis. In *Advances in Neural Information Processing Systems*, volume 37, pp. 131575–131595, 2024.

Yann LeCun, John Denker, and Sara Solla. Optimal brain damage. In *Advances in Neural Information Processing Systems*, volume 2, 1989.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 3045–3059, 2021.

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

Vladislav Lialin, Vijeta Deshpande, Xiaowei Yao, and Anna Rumshisky. Scaling down to scale up: A guide to parameter-efficient fine-tuning. *arXiv preprint arXiv:2303.15647*, 2023.

Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin A. Raffel. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. In *Advances in Neural Information Processing Systems*, volume 35, pp. 1950–1965, 2022a.

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

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 61–68, 2022b.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. LLM-Pruner: On the structural pruning of large language models. In *Advances in Neural Information Processing Systems*, volume 36, pp. 21702–21720, 2023.

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, pp. 121038–121072, 2024.

Pavlo Molchanov, Stephen Tyree, Tero Karras, Timo Aila, and Jan Kautz. Pruning convolutional neural networks for resource efficient inference. In *International Conference on Learning Representations*, 2017.

Pavlo Molchanov, Arun Mallya, Stephen Tyree, Iuri Frosio, and Jan Kautz. Importance estimation for neural network pruning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11264–11272, 2019.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. AdapterFusion: Non-destructive task composition for transfer learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pp. 487–503, 2021.

Victor Sanh, Thomas Wolf, and Alexander Rush. Movement pruning: Adaptive sparsity by fine-tuning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 20378–20389, 2020.

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.

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

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

A ALGORITHMS

Algorithms 1 and 2 summarize the two BARS-LoRA branches. In Algorithm 1, the task salience is computed and cached at the current parameters before the optimizer step, so that the parameters and task-loss gradients in Eq. (11) refer to the same model state. In Algorithm 2, the gradient part $C + \lambda_{\text{orth}}\mathcal{L}_{\text{orth}}$ is separated from the ℓ_1 proximal operation: each SparseAdamW step on the gates applies one gradient update followed by one soft-threshold.

Algorithm 1: SCALP-LoRA

Input: Frozen model; target Linear modules; initial rank r_0; budget $b(t)$; $r_{\min} = 1$.

1. Insert $\Delta W_k = P_k \operatorname{diag}(\lambda_k) Q_k$.
2. Initialize λ_k near zero; retain all channels during warmup.
3. **for** each training step t **do**
4. Cache task salience $s_{k,i}^{(0)}$ (Eq. (11)).
5. Update adapter parameters with AdamW.
6. Update μ, u (Eqs. (12)–(13)).
7. **if** t is a mask step **then**
8. Compute $S_{k,i}^{\text{SCALP}}$ (Eqs. (14)–(16)).
9. Aggregate Φ_k (Eq. (17)).
10. Allocate ranks with $r_{\min} = 1$; repair to the exact budget.
11. Keep top r_k channels per Linear.
12. **end if**
13. In the final stage, consolidate (Eq. (23)).
14. **end for**

Algorithm 2: SAGE-SoRA

Input: Frozen model; target Linear modules; gate sparsity λ_{sp}; orthogonality weight λ_{orth}.

1. Insert $\Delta W_k = B_k \operatorname{diag}(g_k) A_k$.
2. **for** each training step t **do**
3. Form $\mathcal{L}_{\text{grad}} = C + \lambda_{\text{orth}}\mathcal{L}_{\text{orth}}$.
4. Update A_k, B_k on $\mathcal{L}_{\text{grad}}$ (AdamW).
5. For g_k, use a SparseAdamW step:
6. step on $\mathcal{L}_{\text{grad}}$, then one ℓ_1 proximal soft-threshold.
7. In the final stage, consolidate the non-gate basis (Eq. (23)).
8. **end for**
9. Remove zero-gate channels; return the compact adapter.

B ADDITIONAL EXPERIMENTAL RESULTS

This appendix lists the budget-curve values and the complete tuned controls. All task columns use accuracy; subscripts denote standard deviations across three seeds. Fixed LoRA uses one global learning rate per rank and regime, selected by the cross-task macro. The LoRA sweep covers four ranks and four rates; DoRA, PiSSA, and rsLoRA each use the two-rate search described in Section 4.1. Figures 3 and 4 visualize the four-epoch per-task and six-task results.

Table 4: Four-task accuracy (MNLI, SST-2, CoLA, RTE) across training budgets, plotted in Figure 2. The first three columns use uniform optimizer-update counts; the final column reports task-scaled four-epoch endpoints. The AdaLoRA and SCALP endpoints follow the SCALP component series in Table 2(a). Subscripts are macro standard deviations across three seeds.

Method	800	1600	3000	4 epochs
AdaLoRA	$78.97_{1.03}$	$81.38_{0.34}$	$81.82_{0.25}$	$82.01_{0.50}$
SCALP-LoRA	$80.50_{0.36}$	$81.40_{0.25}$	$82.28_{0.16}$	$\mathbf{82.58_{0.69}}$
SoRA	$80.91_{0.63}$	$81.74_{0.69}$	$82.20_{1.24}$	$82.48_{0.53}$
SAGE-SoRA	$\mathbf{82.55_{0.22}}$	$\mathbf{83.75_{0.47}}$	$\mathbf{83.11_{0.66}}$	$82.32_{0.20}$

Table 5: Complete tuned long-budget controls on Qwen3-0.6B. The two panels show the six tasks from the same runs, with Avg-6 repeated for reference. A dash marks the adaptive-branch configuration of Section 4.1.

Method	LR	MNLI	QNLI	SST-2	Avg-6
SCALP-LoRA	—	$82.87_{0.08}$	$90.11_{0.42}$	$92.62_{0.63}$	$84.75_{0.39}$
rsLoRA	1×10^{-4}	$82.43_{0.48}$	$88.30_{1.80}$	$91.51_{1.02}$	$84.36_{0.42}$
LoRA ($r=4$)	1×10^{-4}	$82.00_{0.82}$	$89.99_{0.24}$	$92.62_{0.54}$	$84.18_{0.71}$
LoRA ($r=12$)	1×10^{-4}	$81.72_{0.39}$	$89.55_{0.56}$	$92.35_{0.29}$	$84.08_{0.26}$
DoRA	1×10^{-4}	$81.75_{1.23}$	$88.50_{2.08}$	$92.66_{0.20}$	$84.04_{0.34}$
LoRA ($r=8$)	1×10^{-4}	$81.76_{0.24}$	$88.38_{0.65}$	$92.09_{0.69}$	$83.98_{0.50}$
LoRA ($r=16$)	2×10^{-4}	$81.54_{0.99}$	$87.95_{1.98}$	$91.55_{0.35}$	$83.87_{0.63}$
PiSSA	1×10^{-4}	$79.74_{0.88}$	$87.18_{1.87}$	$91.02_{0.18}$	$82.76_{0.37}$

Method	LR	CoLA	MRPC	RTE	Avg-6
SCALP-LoRA	—	$81.08_{0.53}$	$88.07_{0.62}$	$73.77_{1.99}$	$84.75_{0.39}$
rsLoRA	1×10^{-4}	$81.43_{0.77}$	$89.05_{0.93}$	$73.41_{1.67}$	$84.36_{0.42}$
LoRA ($r=4$)	1×10^{-4}	$81.08_{1.03}$	$86.60_{1.74}$	$72.80_{2.18}$	$84.18_{0.71}$
LoRA ($r=12$)	1×10^{-4}	$80.89_{2.14}$	$87.50_{1.36}$	$72.44_{1.37}$	$84.08_{0.26}$
DoRA	1×10^{-4}	$81.53_{1.06}$	$87.25_{1.30}$	$72.56_{2.25}$	$84.04_{0.34}$
LoRA ($r=8$)	1×10^{-4}	$81.85_{1.88}$	$87.58_{1.39}$	$72.20_{0.36}$	$83.98_{0.50}$
LoRA ($r=16$)	2×10^{-4}	$81.88_{0.75}$	$86.68_{1.23}$	$73.65_{1.91}$	$83.87_{0.63}$
PiSSA	1×10^{-4}	$79.99_{0.92}$	$84.89_{2.14}$	$73.77_{1.82}$	$82.76_{0.37}$

Table 6: Complete tuned controls on Qwen3-0.6B at 800 uniform updates. Each fixed rank and added method uses its selected global learning rate; SAGE uses the adaptive-branch configuration of Section 4.1.

Method	LR	MNLI	SST-2	CoLA	RTE	Avg-4
SAGE-SoRA	—	$83.00_{0.26}$	$92.24_{0.24}$	$80.22_{0.56}$	$74.73_{0.96}$	$82.55_{0.22}$
LoRA ($r=16$)	2×10^{-4}	$82.37_{0.19}$	$92.32_{0.64}$	$79.64_{0.87}$	$74.25_{0.42}$	$82.15_{0.35}$
LoRA ($r=4$)	2×10^{-4}	$82.52_{0.37}$	$92.01_{0.54}$	$79.45_{0.87}$	$73.04_{3.15}$	$81.76_{1.00}$
LoRA ($r=12$)	2×10^{-4}	$82.30_{0.36}$	$91.55_{1.42}$	$79.99_{0.58}$	$73.04_{2.21}$	$81.72_{0.80}$
LoRA ($r=8$)	2×10^{-4}	$82.50_{0.26}$	$91.40_{2.41}$	$79.55_{0.31}$	$72.92_{1.44}$	$81.59_{0.66}$
rsLoRA	1×10^{-4}	$81.19_{1.90}$	$91.17_{1.52}$	$79.74_{1.09}$	$74.01_{1.08}$	$81.53_{1.39}$
DoRA	1×10^{-4}	$81.79_{0.41}$	$90.94_{1.93}$	$78.72_{1.01}$	$73.77_{1.46}$	$81.30_{1.06}$
PiSSA	1×10^{-4}	$81.28_{0.70}$	$91.48_{0.53}$	$77.09_{3.07}$	$74.13_{2.90}$	$80.99_{0.19}$

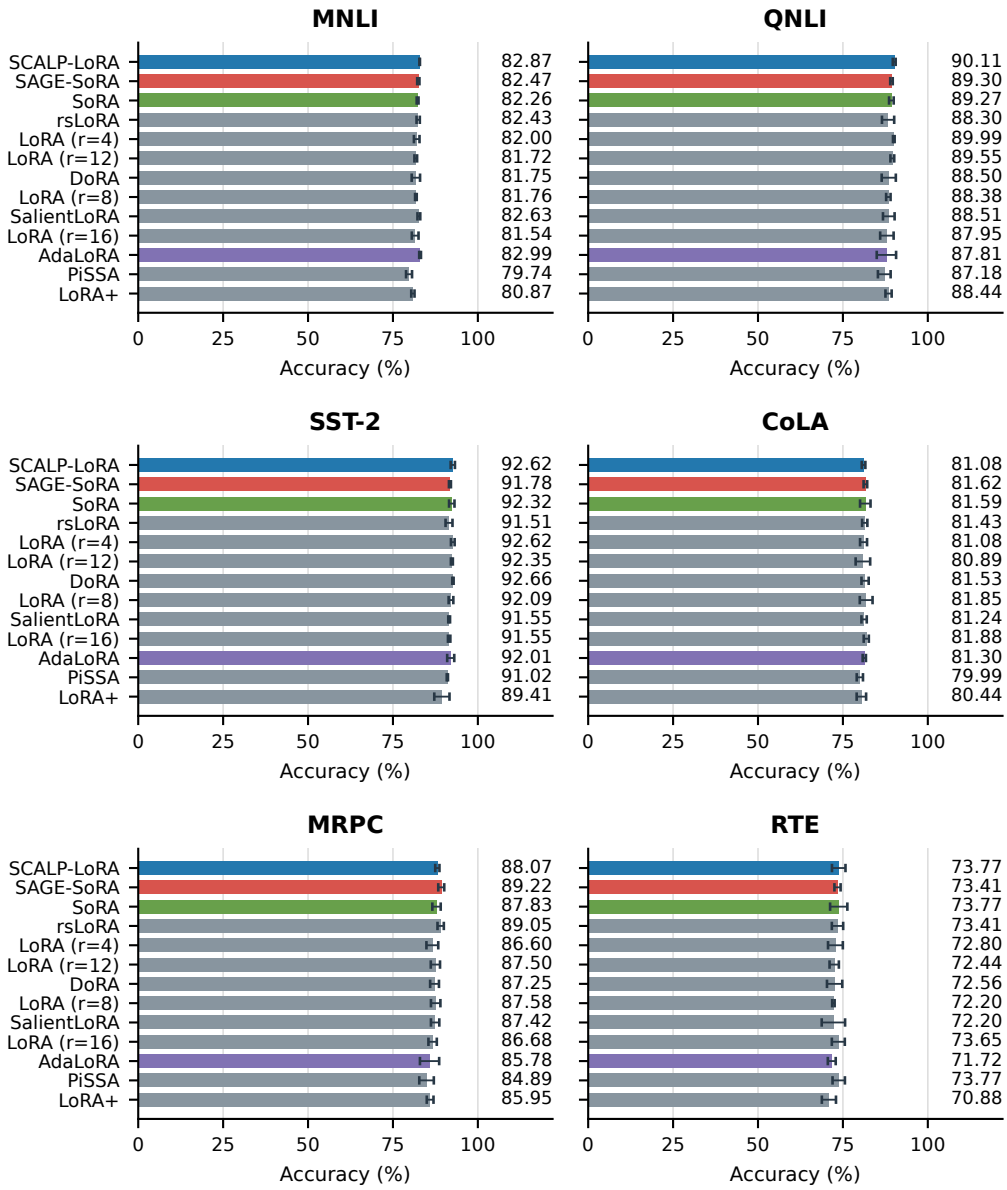


Figure 3: Per-task accuracy under the Qwen3-0.6B four-epoch schedule, including the tuned fixed-rank controls. Error bars are standard deviations across three seeds.

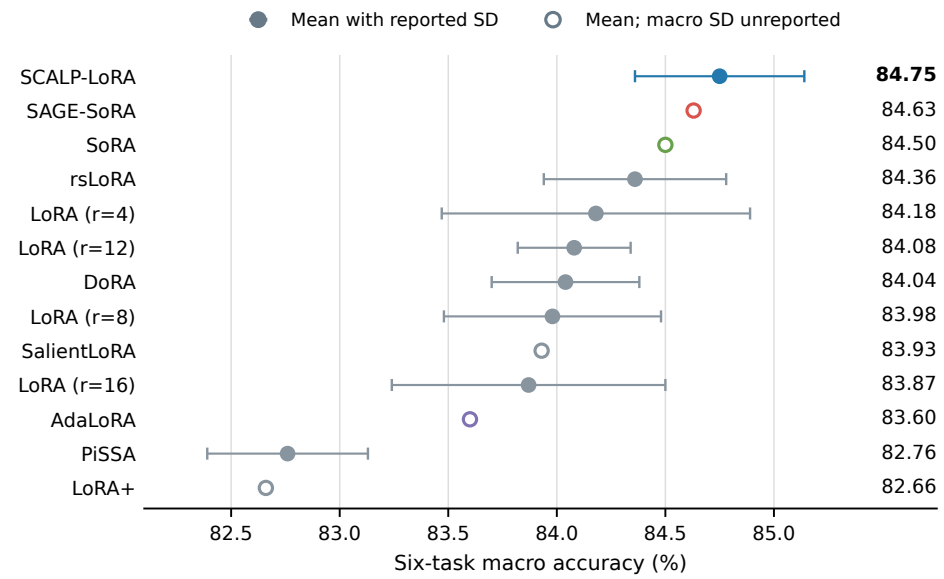


Figure 4: Six-task mean accuracy under the Qwen3-0.6B four-epoch schedule, including the tuned fixed-rank controls. SCALP-LoRA ranks first among all thirteen configurations.

C Component Studies and Allocation Dynamics

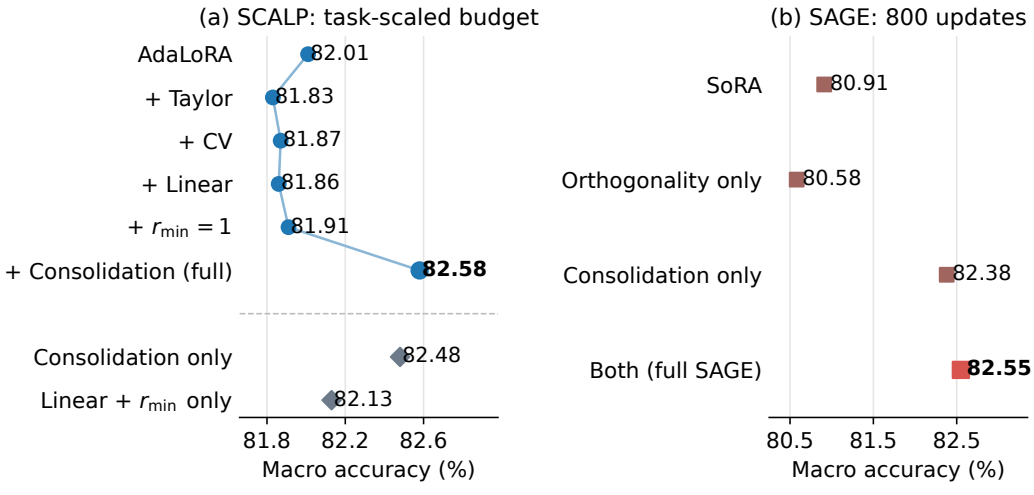


Figure 5: Component studies of Table 2. (a) The first six SCALP rows are cumulative; the two lower rows apply temporal consolidation or the Linear-level scaffold alone. (b) SAGE compares orthogonality and consolidation separately and together. Points are four-task macro means.

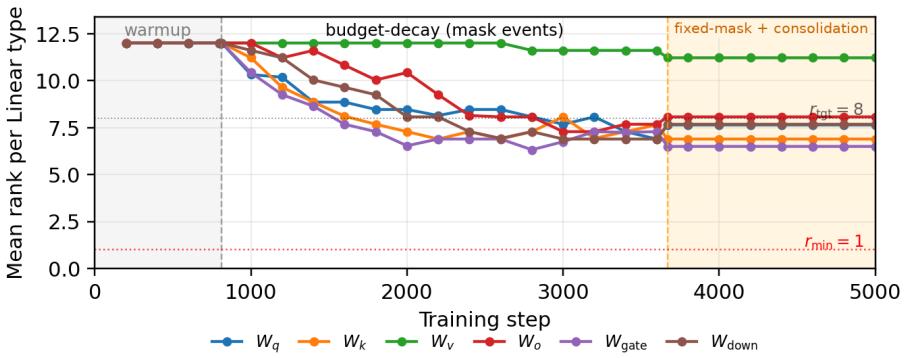


Figure 6: SCALP-LoRA per-Linear mean rank on MNLI (seed 42, $r_0 = 12$, $r_{\text{tgt}} = 8$), averaged across 28 layers. Warmup is followed by budget decay and a fixed-mask consolidation phase. The converged layout assigns more rank to W_v than to W_{gate}.

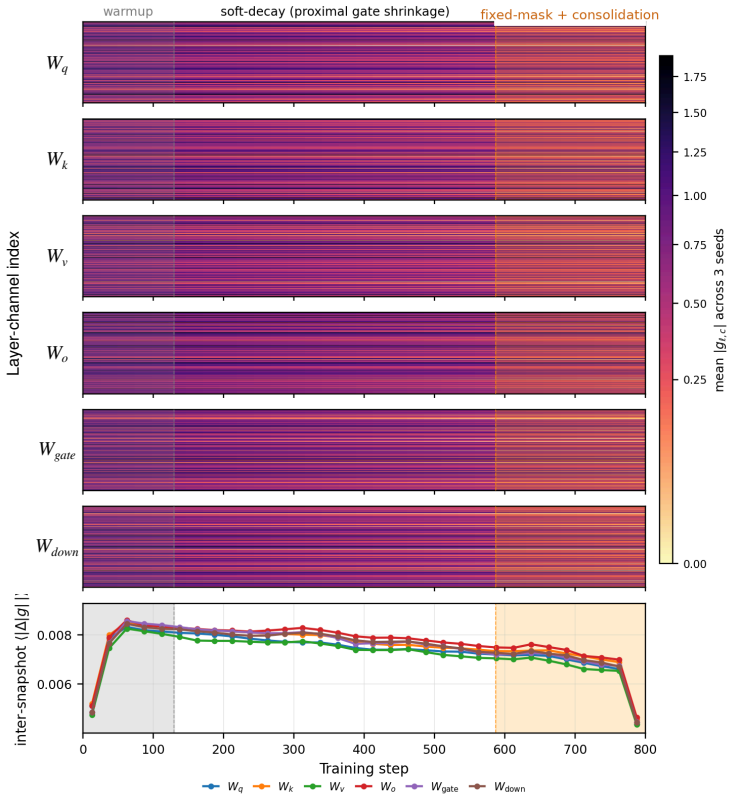


Figure 7: SAGE-SoRA per-channel gate trajectories on MNLI, averaged across three seeds ($r = 8$, 800 updates). Each heatmap stacks 28 layers and eight channels per layer for one Linear type. The lower panel shows the mean inter-snapshot change in gate magnitude, which declines toward the final readout. Phases are warmup (0–130), sparse updates (130–587), and final consolidation (587–800).

D Paired Branch Diagnostics

Because the preferred branch depends on the training budget, a practitioner benefits from choosing it before committing a full run. We designed and implemented BACE, a paired-probe crossover

estimator that ties this choice to observed allocation dynamics. Both probes run with identical settings and data order. The SCALP probe records boundary CV, salience margin, valid-boundary coverage, and rank-layout turnover, which characterize how mature the hard-mask evidence is. The SAGE probe records orthogonality deficit, basis drift, gate turnover, and calibration trend, which characterize how stable the soft-gate readout is.

In a first acceptance test on Qwen3-0.6B, RTE, and an 800-update budget, both 500-update probes ran under identical splits and token counts, for 1000 optimizer updates in total. SAGE led at the final calibration checkpoint by 3.61 pp, agreeing with the winner of the frozen 800-update endpoint comparison (SAGE by 2.53 pp). These diagnostic traces provide the basis for measuring branch-selection accuracy and regret across tasks.