WHAT MAKES A LAYER ADAPTABLE? A MECHANISTIC STUDY OF LANGUAGE MODEL FINE-TUNING

Anonymous authors
Paper under double-blind review

ABSTRACT

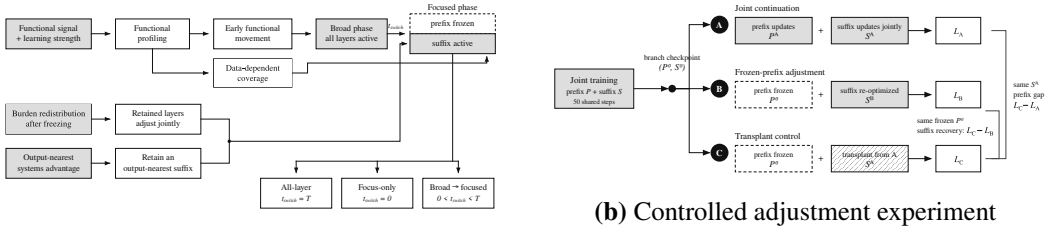
What makes one Transformer layer more useful to fine-tune than another? We judge a layer by the loss-reducing output change, or correction, it can produce. Our first characterization separates the correction a layer can express from the spectral strength that determines how readily it is learned. These factors can trend in opposite directions with depth and produce a nearly flat gradient profile, so gradient size can hide clear differences in what layers can correct. Our second characterization shows how joint training redistributes the remaining error and available output directions across layers; as a result, when earlier layers are frozen, the retained layers can absorb part of the correction the frozen layers would otherwise have made. Controlled GPT-2 experiments test these mechanisms, and a systems study establishes the cost advantage of placing trainable layers nearest the output. Across 15 datasets on each of Qwen3-8B and LLaMA2-7B, most new loss-relevant output change occurs early in training, and the share of accumulated change covered by each output-nearest layer set remains stable over training within a dataset but differs strongly across datasets. These results yield a roadmap parameterized by when to stop broad training and how many output-nearest layers to retain, spanning all-layer, focus-only, and broad-to-focused tuning under different resource budgets. One implementation, Suffix95, trains across all layers until a chosen step and then retains the smallest output-nearest layer set covering 95% of the measured all-layer change. Across 39 model and dataset combinations, it achieves competitive median performance relative to full-depth baselines while reducing total training compute by 8.76–25.80% and post-transition memory by 11.47–19.12%.

1 INTRODUCTION

Fine-tuning often faces resource constraints of different forms. On-device adaptation and memory-limited GPUs impose hard limits: a run that exceeds available memory cannot execute. Cloud services need flexible plans that let users choose how much computation to spend for a desired level of quality. Shared clusters impose softer constraints: every job can run, but shorter jobs improve queue throughput. Partial tuning, which updates only a subset of layers, is a natural way to serve these needs. Existing selective methods sample layers randomly (Pan et al., 2024) or rank them by gradient magnitude (Liu & Litman, 2025), but offer no principled roadmap for deciding which layers to train, when to freeze them, or how these choices should change with the resource budget. Building such a roadmap requires answering the central question of this paper: what determines a layer's adaptability? We ask what loss-reducing output change a layer can produce, how readily that change is learned, and how the layer's role evolves as other layers train or freeze. These questions are coupled with a systems asymmetry: the earliest active layer determines the backward path and activation-storage cost.

Our analysis gives two theoretical characterizations. *Theoretical Characterization 1* separates functional signal, the useful correction a layer can express, from spectral learning strength, which measures how easily gradient descent produces it. These quantities can trend in opposite directions with depth, leaving the gradient norm nearly flat and concealing different functional roles; gradient magnitude can therefore misrank layers. *Theoretical Characterization 2* shows that joint training further changes those roles by redistributing the remaining error and available output directions. The same redistribution helps explain why a broad-to-focused switch can preserve progress: after the

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.



(a) Layer-adaptability roadmap

(b) Controlled adjustment experiment

Figure 1: Layer-adaptability roadmap and controlled adjustment experiment. **(a)** Mechanistic and empirical findings guide the tuning regime. **(b)** Three continuations isolate the loss gap from freezing the prefix and the part recovered by retraining the suffix.

earlier layers (the prefix) are frozen, the retained output-nearest layers (the suffix) can absorb part of the correction the frozen layers would otherwise have made (panel **(b)** of Figure 1).

Three empirical findings complete the picture. *Empirical Finding 1*: on GPT-2 Medium, moving four trainable layers from the input end to the output end shortens the backward path, reducing training time by 30–34% and peak memory by about 56%. *Empirical Finding 2*: across 15 datasets on each of Qwen3-8B and LLaMA2-7B, most new loss-relevant output change occurs early, falling by 64–76% between early and late training; this favors broad early training. *Empirical Finding 3*: the share of accumulated change covered by each output-nearest layer set varies 7.6–8.1× as much across datasets as over training, so retained depth can be estimated early but must be chosen per dataset.

Together, these results yield a configurable roadmap (panel **(a)** of Figure 1): train broadly while most change occurs, then train only a data-dependent number of output-nearest layers. The switch time and retained depth map onto the constraints above: a hard memory limit calls for focus-only training from the start, a shared cluster or cloud budget for a mid-training switch that trades cost for quality, and an unconstrained run for all-layer tuning. At the switch, our Suffix95 implementation retains the smallest output-nearest layer set covering 95% of the measured all-layer change. Across 39 model and dataset combinations, it achieves competitive median performance against full-depth LISA and LoRA baselines while reducing total training compute by 8.76–25.80% and post-transition memory by 11.47–19.12%.

Contributions. First, we separate functional signal from spectral learning strength, explaining why parameter movement and gradients can obscure layer behavior. Second, we characterize joint redistribution and adjustment after freezing. Third, we combine depth-dependent cost, early change, and data-dependent coverage into a roadmap evaluated across 39 model and dataset combinations.

2 Related Work

Layerwise structure and representation quality. Probing shows that linguistic and predictive information varies across depth (Tenney et al., 2019; Wallat et al., 2020; Fayyaz et al., 2021). Geometric, ablation, attribution, and interaction methods study frozen representations or post-training importance (Skean et al., 2025; Zhang et al., 2022; 2024; Cheng et al., 2024). We instead ask which loss-reducing output changes become available through each layer and how other updates alter them.

Theory of fine-tuning and feature learning. Kernel and spectral analyses connect task alignment and parameter geometry to learning (Jacot et al., 2018; Malladi et al., 2023; Rahimi Afzal et al., 2025; Amari, 1996; Martens, 2020; Ali et al., 2019; Bordelon et al., 2020; Canatar et al., 2021); feature-learning, attention, and LoRA theory moves beyond fixed kernels (Atanasov et al., 2022; Bordelon & Pehlevan, 2022; Yang et al., 2021; Vyas et al., 2023; Oymak et al., 2023; Kim & Suzuki, 2024; Jang et al., 2024; Dayi & Chen, 2025). We factor each finite-width layer's output change into its input and downstream route, separating functional signal, learning strength, and joint redistribution.

Efficient and selective fine-tuning. Backward computation and activation storage remain costly even with few trainable parameters (Li et al., 2025; Parthasarathy et al., 2024; Xia et al., 2024).

Parameter-efficient methods constrain updates (Houlsby et al., 2019; Hu et al., 2022; Ben Zaken et al., 2022; Liu et al., 2022; Li & Liang, 2021); selective methods sample layers, rank gradients, or allocate rank (Pan et al., 2024; Liu & Litman, 2025; Wei et al., 2025). We measure functional change and overlap while accounting for the extra backward path and stored activations caused by early trainable layers.

3 A Functional Lens on Layer Adaptability

Layer adaptability poses two questions: what requested correction can a layer express, and how costly is it to learn? Characterization 1 separates correction from learning strength; Characterization 2 tracks its redistribution during joint training. We then establish output-nearest depth as a systems constraint. Assumptions and proofs are in Appendix A.

Transformer and data setting. For an example $z \sim \mathcal{D}$, $F_\theta(z)$ collects the evaluated logits. A residual Transformer obeys $h_{\ell+1} = h_\ell + \mathcal{T}_\ell(h_\ell; \theta_\ell)$ and $F_\theta(z) = \mathcal{O}(h_L)$. We index trainable matrices by b and group them by layer when reporting depth.

At training time t, let $A_{b,t}(z)$ be the activation entering matrix b and $\mathcal{R}_{b,t}(z)$ the derivative from that matrix output through the remaining network to the task logits. Then

$$J_{b,t}[\Delta W_b] = \mathcal{R}_{b,t}[\Delta W_b A_{b,t}]. \tag{1}$$

A weight change acts on the current activation and the remaining network carries it to the logits; both may evolve, while LoRA restricts ΔW_b.

Let $L(F)$ be the task loss, $q_t = \nabla_F L$ its logit gradient, and $M_t = \nabla_F^2 L \succeq 0$ its Hessian. For trainable set $\mathcal{A}$, let $W_{\mathcal{A},t}$ and $J_{\mathcal{A},t}$ collect its parameters and Jacobians, and set $K_{\mathcal{A},t} = J_{\mathcal{A},t} J_{\mathcal{A},t}^*$. Gradient flow satisfies

$$\dot{W}_{\mathcal{A},t} = -J_{\mathcal{A},t}^* q_t, \qquad -\dot{L}_t = q_t^\top K_{\mathcal{A},t} q_t. \tag{2}$$

Equation 2 applies while the Jacobian evolves, exposing available output change, learning cost, and redistribution.

Table 1 tests the analysis with progressively less isolation. Panels A and B compare parameter movement, output correction, and spectral cost. Panel C restores interaction: C1 changes model context, C2 compares joint and isolated training, and C3 changes the active set after shared training.

3.1 Theoretical Characterization 1: Functional Signal and Learning Strength

Available correction and learning strength are distinct. Each layer reaches only the output directions allowed by its activation and downstream route. Projection onto these directions measures available correction; singular strength determines learning speed. Adding layers expands the reachable set; removing one may delete a task-relevant direction.

Parameter distance and gradient magnitude combine output contribution with parameter scaling and direction strength. At a fixed checkpoint, omit t, set $r = -M^{\dagger/2} q$, and define $\Phi_\mathcal{A} = M^{1/2} J_\mathcal{A}$, $g_\mathcal{A} = J_\mathcal{A}^* q$, and $H_\mathcal{A} = \Phi_\mathcal{A}^* \Phi_\mathcal{A}$. Here r is the correction requested by the loss, and $\mathrm{range}(\Phi_\mathcal{A})$ is the data-dependent set of corrections that $\mathcal{A}$ can produce. Functional signal is the projection of r onto this range.

Proposition 3.1 (Functional signal and gradient reweighting)**.** *Assume $q \in \mathrm{range}(M)$ and let $P_\mathcal{A}$ project onto $\mathrm{range}(\Phi_\mathcal{A})$. The functional signal accessible to $\mathcal{A}$ is*

$$\mathcal{S}_\mathcal{D}(\mathcal{A}) := \frac{1}{2} g_\mathcal{A}^\top H_\mathcal{A}^\dagger g_\mathcal{A} = \frac{1}{2} \|P_\mathcal{A} r\|_2^2. \tag{3}$$

It is invariant to invertible parameter coordinates and is nondecreasing as trainable matrices are added. If $\Phi_b = U_b \Sigma_b V_b^\top$, $c_{b,i} = u_{b,i}^\top r$, and $D_\mathcal{D}(b) = \|g_b\|_F^2$, then

$$2\mathcal{S}_\mathcal{D}(b) = \sum_{\sigma_{b,i} > 0} c_{b,i}^2, \qquad D_\mathcal{D}(b) = \sum_i \sigma_{b,i}^2 c_{b,i}^2. \tag{4}$$

Transformer routing also imposes

$$\mathcal{S}_\mathcal{D}(b) \le \frac{1}{2} \left\| P_{\mathrm{range}(M^{1/2} \mathcal{R}_b)} r \right\|_2^2. \tag{5}$$

Table 1: Controlled mechanism tests on GPT-2 Large with E2E.

A. Parameter movement and functional contribution separate (Characterization 1)

	Step 45			Step 500		
Quantity	Early-4	Middle-4	Late-4	Early-4	Middle-4	Late-4
Parameter fraction	**0.359**	0.325	0.308	**0.414**	0.329	0.231
Functional norm fraction	0.089	0.229	**0.358**	0.117	0.307	**0.348**
Functional projection fraction	0.053	0.131	**0.239**	0.069	0.212	**0.248**
Loss-recovery fraction	0.095	0.215	**0.381**	0.110	0.269	**0.314**
Functional efficiency	0.25	0.71	**1.16**	0.28	0.93	**1.51**
Loss efficiency	0.27	0.66	**1.24**	0.27	0.82	**1.36**

B. Task-weighted activation strength controls time and parameter cost (Characterization 1)

Quantity	Low band	High band	Contrast	Control	Setting	Reading
Feature-normalized signal	0.00378	0.00289	Low/high $= 1.31\times$	Equal dimension	Band-isolated	Comparable low-band signal
Steps to loss target	**50**	**25**	Low/high $= \mathbf{2.00}\times$	Same target	Band-isolated	Low band is slower
Parameter norm at target	**0.343**	**0.172**	Low/high $= \mathbf{1.99}\times$	Same target	Band-isolated	Low band moves farther
Final loss drop	0.0261	0.0651	High/low $= 2.50\times$	Same 150 steps	Band-isolated	High band learns more
Squared update-norm share	23.7%	**76.3%**	High/low $= 3.22\times$	Same adapter	Unrestricted	Updates favor high modes
Loss-contribution share	18.7%	**81.4%**	High/low $= 4.36\times$	Same adapter	Unrestricted	High modes explain progress

C. Joint redistribution, specialization, and partial compensation (Characterization 2)

	Step 5		Step 50		Step 300	
Squared functional overlap	Adj.	Far	Adj.	Far	Adj.	Far
C1. Same updates applied alone and in the complete model						
$\cos^2(s_i, s_j)$: applied separately	0.857	0.382	0.868	0.351	0.791	0.318
$\cos^2(m_i, m_j)$: additional effect in full model	0.858	0.385	0.551	0.086	0.294	0.077
Overlap reduction	-0.001	-0.003	0.317	0.265	0.497	0.241
C2. Functional specialization under separate and joint training						
$\cos^2(u_i, u_j)$: trained separately	0.892	0.687	0.897	0.720	0.962	0.879
$\cos^2(m_i^{ij}, m_j^{ij})$: trained jointly	0.893	0.688	0.834	0.636	0.880	0.527
Overall specialization difference	-0.001	-0.001	0.063	0.083	0.082	0.352
C3. Retained output-nearest layers adjust after preceding layers are frozen						
Retained set	$\mathbf{L_A}$	$\mathbf{L_C}$	$\mathbf{L_B}$	$\mathbf{L_C - L_A}$	$\mathbf{L_C - L_B}$	$\mathbf{(L_C - L_B)/(L_C - L_A)}$
Last 4	2.838	3.090	3.063	0.251	0.027	10.6%
Last 8	2.838	3.051	3.005	0.213	0.046	21.6%

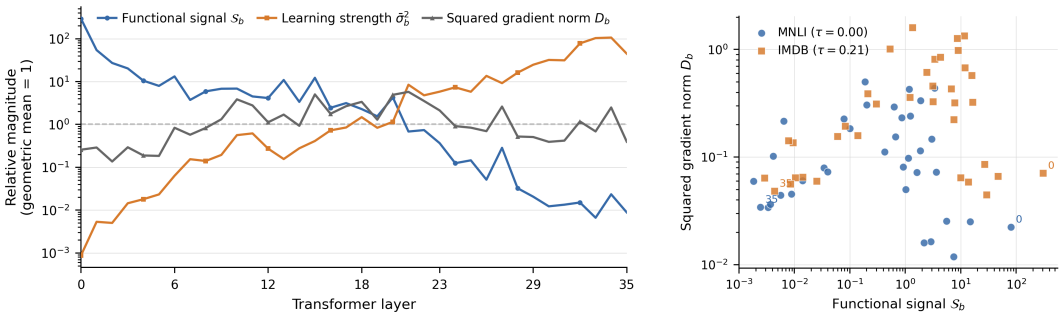


Figure 2: On Qwen3-8B/MNLI, functional signal falls with depth, learning strength rises, and their product, the squared gradient norm, is nearly flat.

The proposition separates cases that parameter norms blur. An orthogonal correction is inaccessible; one aligned with a weak singular direction is accessible but costly. Functional signal measures reachable correction, while the gradient weights each direction by $\sigma_{b,i}^2$. Uniformly rescaling A_b changes these singular values while leaving $c_{b,i}^2$ unchanged, so $\mathcal{S}_{\mathcal{D}}(b)$ is coordinate invariant and $D_{\mathcal{D}}(b)$ carries learning strength.

A measurable ranking reversal. For any trainable matrix with nonzero signal, the same measured singular modes define its task-weighted learning strength

$$\bar{\sigma}_{\mathcal{D},t}^2(b) := \frac{\sum_i \sigma_{b,i,t}^2 c_{b,i,t}^2}{\sum_{\sigma_{b,i,t}>0} c_{b,i,t}^2}, \qquad D_{\mathcal{D},t}(b) = 2\bar{\sigma}_{\mathcal{D},t}^2(b)\mathcal{S}_{\mathcal{D},t}(b). \tag{6}$$

Two layers can satisfy $\mathcal{S}_{\mathcal{D},t}(i) > \mathcal{S}_{\mathcal{D},t}(j)$ but $D_{\mathcal{D},t}(i) < D_{\mathcal{D},t}(j)$ whenever $\bar{\sigma}_{\mathcal{D},t}^2(i)\mathcal{S}_{\mathcal{D},t}(i) < \bar{\sigma}_{\mathcal{D},t}^2(j)\mathcal{S}_{\mathcal{D},t}(j)$. The gradient then favors less signal in stronger directions. A new dataset changes both the residual governing $\mathcal{S}_{\mathcal{D},t}$ and the activations and route governing $\bar{\sigma}_{\mathcal{D},t}^2$.

Figure 2 shows earlier-depth signal and later-depth learning strength producing a nearly flat gradient profile. Across three batches, Kendall τ between signal and gradient magnitude is at most 0.02 in absolute value on MNLI and 0.18–0.21 on IMDB, while signal rankings remain stable ($\tau \geq 0.98$). Thus a flat gradient can hide clear differences in what layers can correct.

The routing bound captures that early-layer changes pass through all downstream attention, residual, normalization, and output operations before reaching the logits. Panel A isolates the resulting separation. At step 500, Early-4 contain 41.4% of parameter movement but only 11.0% of loss-reducing output change, whereas Late-4 contain 23.1% of movement but produce 31.4% of that change. This supports direct measurement at the model output alongside parameter movement and gradient size.

Spectral learning cost. Learning cost complements expressible correction. A weak direction requires more steps and parameter movement, whereas a strong direction produces the same output change faster. Functional signal measures possibility; direction strength determines cost.

Stack contexts from $\mathcal{D}$ (or an empirical batch), and let $y_{b,t} = \mathcal{R}^*_{b,t} q_t$ be the loss gradient reaching matrix b from the output. From equation 1, $g_{b,t} = y_{b,t} A_{b,t}^\top$. Let $s_t = \|M_t^{\dagger/2} q_t\|_2^2$ measure task scale, $\rho_{b,t} = \|y_{b,t}\|_F^2 / s_t$ routing gain, and

$$a_{b,t} = \frac{\|y_{b,t} A_{b,t}^\top\|_F^2}{\|y_{b,t}\|_F^2}, \tag{7}$$

where $a_{b,t}$ is the task-gradient-weighted activation spectrum and varies with the current data and model. We use $\rho_{b,t} = a_{b,t} = 0$ when $s_t = 0$ and $a_{b,t} = 0$ when $y_{b,t} = 0$. In particular,

$$D_{\mathcal{D},t}(b) = \|g_{b,t}\|_F^2 = s_t \rho_{b,t} a_{b,t}, \qquad -\dot{L}_t = \sum_{b \in \mathcal{A}} D_{\mathcal{D},t}(b). \tag{8}$$

The resulting squared gradient norm combines functional signal and learning strength and is also the layer's immediate contribution to loss reduction.

A trainable layer set with a greater total loss-rate factor $s_t \sum_{b \in \mathcal{A}} \rho_{b,t} a_{b,t}$ throughout the same loss interval reaches a matched loss target sooner and requires no greater parameter speed under gradient flow. Appendix A.4 states the comparison over the training trajectory and its assumptions. Available correction and current learning speed can favor different layer sets, so both matter under a fixed training budget.

The loss-reduction rate is task demand $\times$ how much error reaches the layer $\times$ activation strength in output directions relevant to that error.

For one isolated direction of strength σ, let $e(t)$ be its remaining error and $T(\epsilon)$ the time to reach $|e(T)| = \epsilon |e(0)|$ for $0 < \epsilon < 1$. The fixed-geometry benchmark is

$$e(t) = e(0) e^{-\sigma^2 t}, \qquad T(\epsilon) = \frac{\log(1/\epsilon)}{\sigma^2}, \qquad \|\Delta W\|_F = \frac{d}{\sigma}, \tag{9}$$

where d is the desired output displacement. A stronger matched direction therefore needs less time and parameter movement. Panel B isolates this comparison. Although the low band begins with $1.31\times$ more normalized signal, it takes $2.00\times$ as many steps and $1.99\times$ the parameter norm to reach the same loss target; after 150 steps, the high band reduces loss $2.50\times$ more. Available signal and learning cost are distinct.

3.2 Theoretical Characterization 2: Context Dependence, Specialization, and Redistribution

Joint training changes layer function through context dependence (C1), specialization (C2), and burden redistribution after freezing (C3).

Changing error and available output directions. A layer's role depends on the current jointly trained model rather than depth alone. Let $r_t = -M_t^{\dagger/2} q_t$, and let $P_{b,t}$ project onto $\mathrm{range}(M_t^{1/2} J_{b,t})$. When this range has locally constant rank,

$$\dot{\mathcal{S}}_{\mathcal{D},t}(b) = r_t^\top P_{b,t} \dot{r}_t + \frac{1}{2} r_t^\top \dot{P}_{b,t} r_t. \tag{10}$$

The first term measures changes in the remaining error r_t; the second measures changes in the output directions available through layer b as activations, downstream routes, or loss geometry evolve.

Because layers can share output directions, resolving one error can lower several signals, while an upstream update can expose new directions downstream. When $\sum_j D_{\mathcal{D},t}(j) > 0$, define $p_{b,t} = D_{\mathcal{D},t}(b)/\sum_j D_{\mathcal{D},t}(j)$, let $H_{bj,t}$ be the cross-block Hessian, and set $\kappa_{b,t} = \sum_j \langle g_{b,t}, H_{bj,t} g_{j,t}\rangle / \|g_{b,t}\|_F^2$ for $g_{b,t} \neq 0$ and zero otherwise; let $\bar{\kappa}_t = \sum_j p_{j,t}\kappa_{j,t}$. Then

$$\dot{p}_{b,t} = 2p_{b,t}(\bar{\kappa}_t - \kappa_{b,t}), \qquad \frac{p_{b,T}}{p_{j,T}} = \frac{p_{b,0}}{p_{j,0}} \exp\left(2\int_0^T [\kappa_{j,t} - \kappa_{b,t}]\, dt\right). \tag{11}$$

The ratio assumes $p_{b,t}, p_{j,t} > 0$ on $[0, T]$. This specialization law tracks how joint training reallocates loss-reducing work. If $\kappa_{b,t} < \kappa_{j,t}$ on average, block b's share grows relative to j; a persistent gap $\kappa_{j,t} - \kappa_{b,t} \geq \delta > 0$ amplifies the ratio by at least $e^{2\delta T}$. Thus initially similar blocks can diverge as their gradients change at different rates. We call this *functional specialization*: the concentration of loss reduction in different blocks. The full statement and proof are in Theorem A.6.

Short-time adjustment after freezing. At a branching checkpoint, partition the trainable parameters into preceding layers P and retained output-nearest layers S. Starting from the same checkpoint, continuation J updates both P and S, while continuation F freezes P and updates only S. To isolate what S learns, we evaluate the S parameters from both runs behind the same original prefix ϑ_P^0, discarding the updated prefix from J. Let g_P and g_S be their checkpoint gradients, H_{SP} their cross-Hessian block, and $\chi_{S,P} = \langle g_S, H_{SP} g_P\rangle$.

Proposition 3.2 (Short-time adjustment after freezing)**.** *Under twice-continuous differentiability and Euclidean gradient flow,*

$$L(\vartheta_P^0, \vartheta_S^J(t)) - L(\vartheta_P^0, \vartheta_S^F(t)) = \frac{t^2}{2}\chi_{S,P} + o(t^2). \tag{12}$$

Thus, for sufficiently short continuation time, $\chi_{S,P} > 0$ makes adjustment of the retained layers better than transplanting the update learned while all layers continue jointly.

At the branch point, the retained layers have the same gradient in both continuations, so their first steps coincide. Their trajectories separate at second order because the prefix update changes the retained gradient during joint continuation. The sign of $\chi_{S,P}$ determines whether retraining the retained set improves on transplanting its jointly learned update.

Controlled setting. Panels C1 and C2 use GPT-2 Large on E2E and evaluate the same layer pairs after 5, 50, and 300 optimizer steps. We measure the squared cosine between their loss-relevant output changes. High overlap indicates similar corrections; decreasing overlap indicates greater functional separation. Adj. denotes neighboring layers that share much of their downstream route, and Far denotes layers separated in depth.

C1: Conditional layer function. Let $s_{b|\mathcal{A}}$ be the output change from applying layer b's jointly learned update alone, $m_{b|\mathcal{A}}$ its additional change after the other jointly learned updates are present, and $u_{b|b}$ the change learned and applied when only b is trained. Their difference decomposes as

$$m_{b|\mathcal{A}} - u_{b|b} = \underbrace{m_{b|\mathcal{A}} - s_{b|\mathcal{A}}}_{\text{changed model context}} + \underbrace{s_{b|\mathcal{A}} - u_{b|b}}_{\text{changed learned update}}. \tag{13}$$

The first term measures a change in application context; the second measures a change in the learned update. C1 holds the jointly learned updates fixed. The vectors s_i and s_j apply each update to the common reference model, while m_i and m_j measure their additional effects after the surrounding updates are installed. The reduction from $\cos^2(s_i, s_j)$ to $\cos^2(m_i, m_j)$ measures how model context alters layer function.

C2: Functional specialization over time. C2 compares isolated and joint training: u_i and u_j come from single-layer runs, whereas $m_i^{ij} := m_{i|\{i,j\}}$ and $m_j^{ij} := m_{j|\{i,j\}}$ come from joint training. The specialization gap

$$\cos^2(u_i, u_j) - \cos^2(m_i^{ij}, m_j^{ij})$$

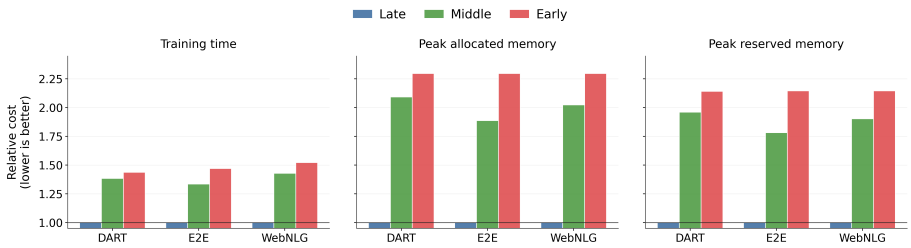


Figure 3: Depth-dependent cost under a fixed GPT-2 Medium adapter budget.

combines the changed context and changed learned update in equation 13. Separate training lets both layers pursue the same dominant correction; joint training lets each respond to corrections already made by other layers.

The C2 gap is essentially zero at step 5 and grows by steps 50 and 300. At step 300, it reaches 0.082 for adjacent pairs and 0.352 for far pairs, showing stronger separation for the far pair. Equation 11 characterizes how instantaneous loss-reduction shares can separate, while C2 shows increasing separation in the accumulated output changes under joint training.

C3: Burden redistribution after freezing. Panel C3 uses the branches in panel **(b)** of Figure 1. After 50 shared steps, A continues all layers, B freezes the prefix and retrains the suffix, and C freezes the prefix but transplants the suffix update learned in A. Let L_A, L_B, and L_C be their evaluation losses. Then $L_C - L_A$ is the removed-prefix gap, while $L_C - L_B$ is the part recovered by suffix retraining. The recovery ratio is 10.6% for the last four layers and 21.6% for the last eight. The appendix decomposes $\chi_{S,P}$ into changes in shared output error, activations, and downstream routes.

While every layer is active, the prefix performs part of the available correction. Freezing it leaves part of that correction unresolved, and continued suffix training transfers some of this burden to the retained layers. This partial recovery explains how a broad-to-focused transition can preserve functional progress.

3.3 Empirical Finding 1: Output-Nearest Placement Reduces Systems Cost

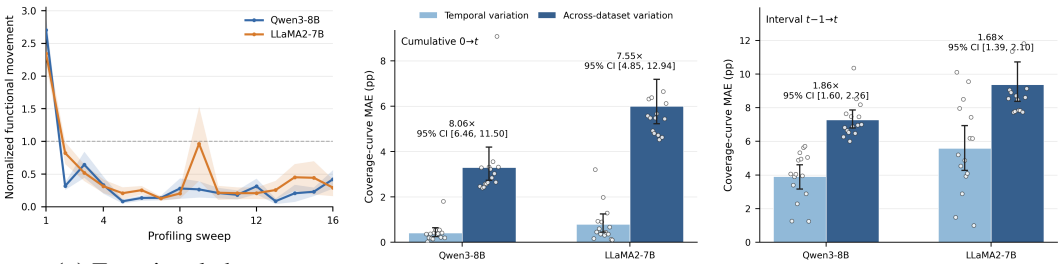
We compare Early-4, Middle-4, and Late-4 on GPT-2 Medium with adapter number and size fixed. Moving the budget from Early-4 to Late-4 reduces training time by about 30–34% and peak memory by about 56% (Figure 3) by shortening the backward path. Thus, once training uses fewer layers, placing them near the output provides the compute and memory benefit. Retained depth determines how many layers remain active; the earliest retained layer determines the backward path and stored activations.

4 A Configurable Roadmap for Efficient Partial Tuning

Characterization 1 motivates direct measurement because gradients and parameter movement mix useful correction with learning strength. Characterization 2 explains how retained layers can absorb part of the missing correction after freezing. Finding 1 fixes the systems direction: among equally sized active sets, output-nearest placement shortens the backward path. The remaining choices are when to focus training and how much output-nearest depth to retain.

4.1 Empirical Finding 2: Functional Change Concentrates Early

We train all layers of Qwen3-8B and LLaMA2-7B on 15 tasks (Wang et al., 2023); one sweep is three joint optimizer steps. Panel (a) of Figure 4 measures new loss-relevant output change from sweep $t-1$ to t. This change decreases from sweeps 2–5 to 13–16 by 76.4% for Qwen3-8B and 63.9% for LLaMA2-7B, with smaller late change on all 15 Qwen3 datasets and 14 of 15 LLaMA2 datasets. Broad training therefore covers the largest functional changes. Proposition A.8 gives a sufficient decay bound under standard local assumptions.



(a) Functional change **(b)** Cumulative profile **(c)** Interval profile

Figure 4: Functional profiling across 15 datasets. **(a)** New functional change per sweep. **(b,c)** Within-dataset temporal and across-dataset variation of cumulative and interval coverage profiles for output-nearest layer sets; error bars are 95% paired bootstrap confidence intervals.

4.2 Empirical Finding 3: Functional Coverage Is Data Dependent

After each sweep t, we also record interval change from $t-1$ to t and cumulative change from the initial adapter state to t on a fixed early accounting panel. For each measurement, the layerwise change Gram matrix yields a coverage curve over output-nearest layer sets: every point reports the share of combined all-layer change reproduced after shared directions are accounted for.

Panels (b) and (c) of Figure 4 compare adjacent-sweep variation of each dataset's coverage curve with its distance from the other datasets at matched sweeps. Each dataset forms one paired observation, and the bootstrap resamples these 15 dataset units. For the cumulative profile used to allocate suffix depth, across-dataset variation is $8.06\times$ temporal variation on Qwen3-8B and $7.55\times$ on LLaMA2-7B. The interval ratios are $1.86\times$ and $1.68\times$. The cumulative profile therefore settles within a task while its location remains strongly data dependent. Proposition A.2 gives the matching structural result: whenever nested suffixes have strictly different reachable spaces, some task correction is representable by the larger suffix and missed by the shorter one.

The cumulative measurement summarizes stable accumulated depth, while the interval measurement exposes noisier local changes. Together they support data-dependent suffix selection.

4.3 Broad-to-Focused as a Configurable Roadmap

The five results guide four practical decisions. Characterization 1 motivates choosing layers by their functional behavior. Finding 2 identifies the early phase for broad training. Characterization 2 explains how retained layers compensate after freezing. Finding 3 supplies a data-dependent depth signal, while Finding 1 places the active layers nearest the output to save compute and memory.

This roadmap spans common tuning regimes through two explicit controls: broad-phase duration t_{switch} and retained suffix depth d. All-layer tuning sets t_{switch} to the full training horizon. A hard efficiency constraint can use focus-only training with $t_{\text{switch}} = 0$ and an output-nearest suffix; the depth-cost result shows why moving the same layer budget from the input side to the output side can turn an expensive selective run into an efficient one. A soft constraint uses $0 < t_{\text{switch}} < T$, preserving broad early adaptation before concentrating the remaining work. Both controls can be planned as ordinary hyperparameters for a queue or compute budget, while functional coverage provides a one-run data-dependent choice of d.

4.4 One Broad-to-Focused Implementation

The procedure has two phases: keep all layers eligible, training them jointly when $K = L$ or in rotating groups when $K < L$, until the user-specified transition step t_{switch}; then freeze the preceding layers and train an output-nearest suffix. In this coverage-based realization, $t_{\text{switch}} > 0$ lies at a completed-sweep boundary. Ordinary training sweeps up to that step measure the layer behavior used to choose suffix depth. After sweep k, $u_{\ell,k}$ records layer ℓ's effect on the loss over the measured tokens. After rescaling, the vectors $\widehat{u}_{\ell,k}$ form a running overlap matrix $\bar{G}_k$. For output-nearest suffix

Table 2: Suffix95 versus matched full-depth baselines across 39 model and dataset combinations. Positive relative change favors Suffix95; savings are task means ± standard errors.

Model	K	Tasks	Metric	Median rel. Δ (%) ↑	Mean rel. Δ (%) ↑	Active/Total	Bwd. PF ↓	Train PF ↓	Post Mem. ↓
GPT-2-M	5	3	Loss	-0.04	-0.24	17–19 / 24	$(23.29 \pm 2.49)\%$	$(14.46 \pm 1.55)\%$	$(19.12 \pm 2.01)\%$
GPT-2-L	5	3	Loss	$+0.08$	$+0.10$	29–32 / 36	$(14.06 \pm 1.97)\%$	$(8.76 \pm 1.22)\%$	$(12.06 \pm 1.88)\%$
GPT-2-M	24	3	Loss	-0.77	-0.69	19–21 / 24	$(16.71 \pm 2.58)\%$	$(10.88 \pm 1.68)\%$	$(13.25 \pm 2.04)\%$
GPT-2-L	36	3	Loss	-0.31	-0.39	29–32 / 36	$(14.18 \pm 2.33)\%$	$(9.33 \pm 1.54)\%$	$(11.47 \pm 1.90)\%$
Qwen3-8B	5	7	Accuracy	$+2.32$	$+3.81$	16–31 / 36	$(33.04 \pm 5.03)\%$	$(20.64 \pm 3.14)\%$	$(17.57 \pm 4.40)\%$
Qwen3-8B	36	12	Accuracy	$+1.02$	$+0.65$	12–36 / 36	$(39.25 \pm 6.08)\%$	$(25.80 \pm 4.00)\%$	$(18.37 \pm 3.44)\%$
LLaMA2-7B	32	8	Accuracy	$+0.14$	$+3.41$	19–32 / 32	$(25.60 \pm 4.79)\%$	$(17.01 \pm 3.18)\%$	$(15.15 \pm 3.00)\%$

S_c, let $v_c = (\bar{G}_k)_{S_c,:}\mathbf{1}$ and define

$$C(S_c) = \frac{v_c^\top (\bar{G}_k)^\dagger_{S_c S_c} v_c}{\mathbf{1}^\top \bar{G}_k \mathbf{1}}. \tag{14}$$

Coverage is the share of combined realized change reproduced by the suffix after shared directions are accounted for. The profiler retains the shortest suffix reaching target γ. Theoretical signal $\mathcal{S}_{\mathcal{D},t}(\ell)$ measures available correction in loss-weighted output space. The empirical vector $u_{\ell,k}$ instead records the per-token effect of a completed optimizer update, including learning strength and changes accumulated during the sweep. We use it to choose how many layers to retain and evaluate accuracy separately.

At t_{switch}, the accumulated overlap matrix determines the retained depth. We report $\gamma = 95\%$ as Suffix95, with 80% and 90% targets in the appendix. Appendix A.8 gives sufficient conditions for recovering the observed boundary when it is separated by a clear population gap. Exact measurement and pseudocode details appear in Appendix B.4; all reported costs include the broad measurement phase.

4.5 Validation Across 39 Model and Dataset Combinations

We evaluate Suffix95 on 39 model and dataset combinations in seven GPT-2, Qwen3-8B, and LLaMA2-7B configurations. Qwen3 uses eight classification tasks (Wang et al., 2023) and four reasoning tasks from ResLoRA (Shi et al., 2024). Baselines are full-depth LISA with $K = 5$ and all-layer LoRA with $K = L$. Relative change is $100(A_{\text{suffix}}/A_{\text{base}} - 1)$ for accuracy and $100(1 - L_{\text{suffix}}/L_{\text{base}})$ for loss, so positive values favor Suffix95. PFLOPs include the broad measurement phase; Post Mem. is peak allocated memory after the transition, and Active/Total is retained over total depth. Full configurations and wall-clock results appear in the appendix.

Performance and efficiency. Across 39 model and dataset combinations, the seven configuration performance medians in Table 2 range from -0.77% to $+2.32\%$, showing competitive performance across model scales and active-layer budgets. Suffix95 reduces mean backward PFLOPs by 14.06–39.25%, total training PFLOPs by 8.76–25.80%, and post-transition memory by 11.47–19.12%. On AG News, the 38.12% training-PFLOP reduction corresponds to a 37.47% wall-clock reduction.

Coverage–efficiency trade-off. Across the 33 model and dataset combinations evaluated at all three coverage targets, Suffix95, Suffix90, and Suffix80 save 19.99%, 25.33%, and 34.26% of total training PFLOPs and 16.91%, 21.76%, and 29.70% of post-transition memory, respectively. Their median relative performance changes are $+0.35\%$, -0.42%, and -0.89%. Lower coverage therefore provides progressively greater savings with a modest median performance trade-off. Complete task-level results and selection settings appear in Appendix B.

5 Conclusion

Layer adaptability depends on the correction a layer can express, how readily it is learned, and redistribution during joint training. Opposing depth trends make precise layer ranking difficult, while retained layers can absorb part of the missing correction after freezing. Together with early change, data-dependent coverage, and the cost advantage of output-nearest placement, these results motivate a configurable broad-to-focused roadmap, which Suffix95 implements at competitive median performance and lower training cost.

AI USE STATEMENT

Generative AI tools were used for text editing and revision and for code debugging. The authors reviewed and verified all AI-assisted changes and take responsibility for the complete content of this work.

ETHICS STATEMENT

This work studies the mechanisms and computational efficiency of language-model fine-tuning using existing pretrained models and public benchmark datasets. Its societal effects follow those of the underlying models and datasets, while its practical aim is to reduce computation and memory after layer selection.

REPRODUCIBILITY STATEMENT

The main text specifies the analysis assumptions, mechanism probes, evaluation measures, and cost accounting. The appendix provides complete proofs, the coverage-based suffix-selection procedure, full results, and implementation details. Experimental configurations, data processing, prompts, and code are included in the supplementary material.

REFERENCES

Alnur Ali, J. Zico Kolter, and Ryan J. Tibshirani. A continuous-time view of early stopping for least squares regression. In Kamalika Chaudhuri and Masashi Sugiyama (eds.), *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pp. 1370–1378. PMLR, 16–18 Apr 2019. URL https://proceedings.mlr.press/v89/ali19a.html.

Shun-ichi Amari. Neural learning in structured parameter spaces - natural riemannian gradient. In M.C. Mozer, M. Jordan, and T. Petsche (eds.), *Advances in Neural Information Processing Systems*, volume 9. MIT Press, 1996. URL https://proceedings.neurips.cc/paper_files/paper/1996/file/39e4973ba3321b80f37d9b55f63ed8b8-Paper.pdf.

Alexander Atanasov, Blake Bordelon, and Cengiz Pehlevan. Neural networks as kernel learners: The silent alignment effect. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=1NvflqAdoom.

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. doi: 10.18653/v1/2022.acl-short.1. URL https://aclanthology.org/2022.acl-short.1/.

Blake Bordelon and Cengiz Pehlevan. Self-consistent dynamical field theory of kernel evolution in wide neural networks. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 32240–32256. Curran Associates, Inc., 2022. doi: 10.52202/068431-2336. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/d027a5c93d484a4312cc486d399c62c1-Paper-Conference.pdf.

Blake Bordelon, Abdulkadir Canatar, and Cengiz Pehlevan. Spectrum dependent learning curves in kernel regression and wide neural networks. In Hal Daumé, III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 1024–1034. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/bordelon20a.html.

Abdulkadir Canatar, Blake Bordelon, and Cengiz Pehlevan. Spectral bias and task-model alignment explain generalization in kernel regression and infinitely wide neural networks. *Nature Communications*, 12(1):2914, May 2021. ISSN 2041-1723. doi: 10.1038/s41467-021-23103-1. URL http://dx.doi.org/10.1038/s41467-021-23103-1.

Xu Cheng, Lei Cheng, Zhaoran Peng, Yang Xu, Tian Han, and Quanshi Zhang. Layerwise change of knowledge in neural networks. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 8038–8059. PMLR, 2024. URL https://proceedings.mlr.press/v235/cheng24b.html.

Arif Kerem Dayi and Sitan Chen. Low-rank fine-tuning lies between lazy training and feature learning. In Nika Haghtalab and Ankur Moitra (eds.), *Proceedings of Thirty Eighth Conference on Learning Theory*, volume 291 of *Proceedings of Machine Learning Research*, pp. 1415–1471. PMLR, 30 Jun–04 Jul 2025. URL https://proceedings.mlr.press/v291/dayi25a.html.

Mohsen Fayyaz, Ehsan Aghazadeh, Ali Modarressi, Hosein Mohebbi, and Mohammad Taher Pilehvar. Not all models localize linguistic knowledge in the same place: A layer-wise probing on bertoids' representations. In Jasmijn Bastings, Yonatan Belinkov, Emmanuel Dupoux, Mario Giulianelli, Dieuwke Hupkes, Yuval Pinter, and Hassan Sajjad (eds.), *Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pp. 375–388, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.blackboxnlp-1.29. URL https://aclanthology.org/2021.blackboxnlp-1.29/.

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. PMLR, 2019. URL https://proceedings.mlr.press/v97/houlsby19a.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.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/5a4be1fa34e62bb8a6ec6b91d2462f5a-Paper.pdf.

Uijeong Jang, Jason D. Lee, and Ernest K. Ryu. LoRA training in the NTK regime has no spurious local minima. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 21306–21328. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/jang24d.html.

Juno Kim and Taiji Suzuki. Transformers learn nonlinear features in context: nonconvex mean-field dynamics on the attention landscape. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 24527–24561. PMLR, 2024. URL https://proceedings.mlr.press/v235/kim24af.html.

Liang Li, Xingke Yang, Wen Wu, Hao Wang, Tomoaki Ohtsuki, Xin Fu, Miao Pan, and Xuemin Shen. MobiLLM: Enabling on-device fine-tuning of billion-sized LLMs via server-assisted side-tuning. *IEEE Journal of Selected Topics in Signal Processing*, 19(7):1251–1265, 2025. doi: 10.1109/JSTSP.2025.3633550.

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. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.acl-long.353. URL https://aclanthology.org/2021.acl-long.353/.

Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin 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, 2022. doi: 10.52202/068431-0142. URL https://doi.org/10.52202/068431-0142.

Zhexiong Liu and Diane Litman. Efficient layer-wise LLM fine-tuning for revision intention prediction. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 15319–15334, 2025. doi: 10.18653/v1/2025.findings-emnlp.829. URL https://aclanthology.org/2025.findings-emnlp.829/.

Sadhika Malladi, Alexander Wettig, Dingli Yu, Danqi Chen, and Sanjeev Arora. A kernel-based view of language model fine-tuning. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 23610–23641. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/malladi23a.html.

James Martens. New insights and perspectives on the natural gradient method. *Journal of Machine Learning Research*, 21(146):1–76, 2020. URL http://jmlr.org/papers/v21/17-678.html.

Samet Oymak, Ankit Singh Rawat, Mahdi Soltanolkotabi, and Christos Thrampoulidis. On the role of attention in prompt-tuning. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 26724–26768. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/oymak23a.html.

Rui Pan, Xiang Liu, Shizhe Diao, Renjie Pi, Jipeng Zhang, Chi Han, and Tong Zhang. LISA: Layerwise importance sampling for memory-efficient large language model fine-tuning. In *Advances in Neural Information Processing Systems*, volume 37, pp. 57018–57049, 2024. doi: 10.52202/079017-1817. URL https://doi.org/10.52202/079017-1817.

Venkatesh Balavadhani Parthasarathy, Ahtsham Zafar, Aafaq Khan, and Arsalan Shahid. The ultimate guide to fine-tuning LLMs from basics to breakthroughs: An exhaustive review of technologies, research, best practices, applied research challenges and opportunities, 2024. URL https://arxiv.org/abs/2408.13296.

Zahra Rahimi Afzal, Tara Esmaeilbeig, Mojtaba Soltanalian, and Mesrob I Ohannessian. Linearization explains fine-tuning in large language models. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, pp. 131794–131823. Curran Associates, Inc., 2025. doi: 10.52202/085713-4388. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/becc00fe2e0ade58213cff16a166fa25-Paper-Conference.pdf.

Shuhua Shi, Shaohan Huang, Minghui Song, Zhoujun Li, Zihan Zhang, Haizhen Huang, Furu Wei, Weiwei Deng, Feng Sun, and Qi Zhang. ResLoRA: Identity residual mapping in low-rank adaption. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 8870–8884, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.525. URL https://aclanthology.org/2024.findings-acl.525/.

Oscar Skean, Md Rifat Arefin, Dan Zhao, Niket Nikul Patel, Jalal Naghiyev, Yann LeCun, and Ravid Shwartz-Ziv. Layer by layer: Uncovering hidden representations in language models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 55854–55875. PMLR, 2025. URL https://proceedings.mlr.press/v267/skean25a.html.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. BERT rediscovers the classical NLP pipeline. In Anna Korhonen, David Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4593–4601, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1452. URL https://aclanthology.org/P19-1452/.

Nikhil Vyas, Alexander Atanasov, Blake Bordelon, Depen Morwani, Sabarish Sainathan, and Cengiz Pehlevan. Feature-learning networks are consistent across widths at realistic scales. In *Advances in Neural Information Processing Systems*, volume 36, pp. 1036–1060. Curran Associates,

Inc., 2023. doi: 10.52202/075280-0050. URL https://openreview.net/forum?id=LTdfYIvbHc.

Jonas Wallat, Jaspreet Singh, and Avishek Anand. BERTnesia: Investigating the capture and forgetting of knowledge in BERT. In Afra Alishahi, Yonatan Belinkov, Grzegorz Chrupała, Dieuwke Hupkes, Yuval Pinter, and Hassan Sajjad (eds.), *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pp. 174–183, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.blackboxnlp-1.17. URL https://aclanthology.org/2020.blackboxnlp-1.17/.

Xiao Wang, Tianze Chen, Qiming Ge, Han Xia, Rong Bao, Rui Zheng, Qi Zhang, Tao Gui, and Xuanjing Huang. Orthogonal subspace learning for language model continual learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 10658–10671. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.findings-emnlp.715. URL https://aclanthology.org/2023.findings-emnlp.715/.

Chenxing Wei, Yao Shu, Ying Tiffany He, and Fei Yu. Flexora: Flexible low-rank adaptation for large language models. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14643–14682, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.713. URL https://aclanthology.org/2025.acl-long.713/.

Yuchen Xia, Jiho Kim, Yuhan Chen, Haojie Ye, Souvik Kundu, Cong Callie Hao, and Nishil Talati. Understanding the performance and estimating the cost of LLM fine-tuning. In *2024 IEEE International Symposium on Workload Characterization (IISWC)*, pp. 210–223. IEEE Computer Society, 2024. doi: 10.1109/IISWC63097.2024.00027. URL https://doi.org/10.1109/IISWC63097.2024.00027.

Greg Yang, Edward J. Hu, Igor Babuschkin, Szymon Sidor, Xiaodong Liu, David Farhi, Nick Ryder, Jakub Pachocki, Weizhu Chen, and Jianfeng Gao. Tensor programs v: Tuning large neural networks via zero-shot hyperparameter transfer. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 17084–17097. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/8df7c2e3c3c3be098ef7b382bd2c37ba-Paper.pdf.

Chiyuan Zhang, Samy Bengio, and Yoram Singer. Are all layers created equal? *Journal of Machine Learning Research*, 23(67):1–28, January 2022. ISSN 1532-4435. URL https://jmlr.org/papers/v23/20-069.html.

Yang Zhang, Yanfei Dong, and Kenji Kawaguchi. Investigating layer importance in large language models. In *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pp. 469–479, 2024. doi: 10.18653/v1/2024.blackboxnlp-1.29. URL https://aclanthology.org/2024.blackboxnlp-1.29/.

APPENDIX

A THEORY DETAILS AND PROOFS

This appendix provides the definitions and proofs omitted from Section 3. The analysis concerns the exact finite-width model along continuous-time gradient flow. The Jacobian and its spectrum may change at every time; no fixed kernel, fixed feature map, or stable eigenbasis is assumed.

A.1 MODEL, LOSS, AND NOTATION

Let $W = (W_b)_b$ denote the trainable matrix blocks, grouped by Transformer depth; pretrained parameters remain fixed. For a trainable block set $\mathcal{A}$, parameters outside $\mathcal{A}$ are held constant.

Table A.1: Notation used in the theoretical analysis.

Symbol	Meaning
$F(W)$	Selected task logits of the finite-width Transformer
q_t	Loss gradient with respect to the selected logits
M_t	Cross-entropy Hessian in logit space
$A_{b,t}$	Activation matrix entering Transformer matrix block b
$\mathcal{R}_{b,t}$	Downstream derivative from block b to the selected logits
$J_{b,t}$	Block Jacobian $\Delta W_b \mapsto \mathcal{R}_{b,t}[\Delta W_b A_{b,t}]$
$\Phi_{\mathcal{A},t}$	Loss-weighted functional map for block set $\mathcal{A}$
$\mathcal{S}_{\mathcal{D},t}(b)$	Functional signal available through block b
$D_{\mathcal{D},t}(b)$	Squared block-gradient norm and instantaneous loss-rate contribution
$\rho_{b,t}, a_{b,t}$	Route and task gain; task-weighted activation strength
$p_{b,t}, \kappa_{b,t}$	Normalized gradient contribution and cross-block competition rate
$\chi_{S,P}$	Interaction between the prefix and retained layers
$G_{\mathcal{D},t}$	Population functional-overlap matrix under dataset $\mathcal{D}$
S_c	Layers $c, \dots, L-1$, forming a set nearest the model output
$C(S_c)$	Estimated functional coverage of suffix S_c
t_{switch}	Input optimizer step at which training changes from broad to focused

For independent sequences $z \sim \mathcal{D}$, let $F(W)$ collect the selected logits and let $L(F)$ be their population cross-entropy loss. Prediction positions within one sequence may be dependent; the independent unit in the statistical result is the sequence or context. We assume that F is continuously differentiable and the gradient-flow fields are locally Lipschitz. Define

$$q_t := \nabla_F L(F(W_t)), \qquad M_t := \nabla_F^2 L(F(W_t)), \qquad J_{b,t} := D_{W_b} F(W_t). \tag{15}$$

All adjoints and norms are Euclidean. Square roots and pseudoinverses of positive-semidefinite matrices act on their nonzero eigenspaces.

For softmax cross-entropy, each token contributes

$$q = p - e_y, \qquad M = \operatorname{diag}(p) - pp^\top.$$

Because every softmax probability is positive, the nullspace of M is the constant-logit direction and its range is the zero-sum subspace. Since $\mathbf{1}^\top(p - e_y) = 0$, we have

$$M_t \succeq 0, \qquad q_t \in \operatorname{range}(M_t). \tag{16}$$

The same statement holds blockwise when the loss averages multiple tokens and examples.

For block set $\mathcal{A}$, concatenate the corresponding Jacobians and define

$$J_{\mathcal{A},t} := [J_{b,t}]_{b\in\mathcal{A}}, \qquad K_{\mathcal{A},t} := J_{\mathcal{A},t} J_{\mathcal{A},t}^*. \tag{17}$$

Euclidean gradient flow restricted to $\mathcal{A}$ is

$$\dot{W}_{\mathcal{A},t} = -J_{\mathcal{A},t}^* q_t.$$

Applying the chain rule gives

$$\dot{F}_t = J_{\mathcal{A},t}\dot{W}_{\mathcal{A},t} = -K_{\mathcal{A},t} q_t, \qquad -\dot{L}_t = q_t^\top K_{\mathcal{A},t} q_t. \tag{18}$$

This proves equation 2. It is an exact instantaneous identity for the nonlinear finite-width model because $J_{\mathcal{A},t}$ is evaluated at the current parameters.

A.2 Exact Transformer-Block Factorization

Write a pre-layer-normalized residual block as

$$\begin{aligned} U_\ell &= X_\ell + \mathrm{Attn}_\ell\big(\mathrm{LN}_\ell^{\mathrm{att}}(X_\ell); \theta_\ell^{\mathrm{att}}\big), \\ X_{\ell+1} &= U_\ell + \mathrm{MLP}_\ell\big(\mathrm{LN}_\ell^{\mathrm{mlp}}(U_\ell); \theta_\ell^{\mathrm{mlp}}\big) =: \mathcal{T}_\ell(X_\ell; \theta_\ell). \end{aligned} \tag{19}$$

Consider any trainable matrix block W_b inside attention or the MLP whose local operation is $W_b A_{b,t}$, using a homogeneous coordinate when a bias is present. Let $\mathcal{R}_{b,t}$ be the derivative of the

remaining computation, including the rest of the current Transformer block, later residual blocks, final normalization, output head, and selected logits. A perturbation of that matrix produces the exact differential

$$J_{b,t}[\Delta W_b] = \mathcal{R}_{b,t}[\Delta W_b A_{b,t}], \tag{20}$$

which proves equation 1. For query and key matrices, the attention softmax is part of $\mathcal{R}_{b,t}$; for LoRA, ΔW_b is restricted to the current low-rank update space. Thus the factorization retains causal attention, normalization, residual connections, and all later finite-width nonlinearities.

A.3 Data-Dependent Functional Signal

At a fixed Transformer checkpoint, suppress time and define

$$r = -M^{\dagger/2}q, \qquad \Phi_{\mathcal{A}} = M^{1/2}J_{\mathcal{A}}, \qquad g_{\mathcal{A}} = J_{\mathcal{A}}^{*}q, \qquad H_{\mathcal{A}} = \Phi_{\mathcal{A}}^{*}\Phi_{\mathcal{A}}. \tag{21}$$

Lemma A.1 (Least-squares projection)**.** *For any linear map Φ and vector r,*

$$\min_x \frac{1}{2}\|r - \Phi x\|_2^2 = \frac{1}{2}\|(I - P_{\mathrm{range}(\Phi)})r\|_2^2,$$

and $\Phi(\Phi^*\Phi)^{\dagger}\Phi^* = P_{\mathrm{range}(\Phi)}$.

Proof. Take a singular-value decomposition of Φ; its nonzero left singular vectors span $\mathrm{range}(\Phi)$, and the pseudoinverse removes the orthogonal component. □

Proof of Proposition 3.1. Since $q \in \mathrm{range}(M)$, $M^{1/2}M^{\dagger/2}q = q$, and therefore $\Phi_{\mathcal{A}}^{*}r = -g_{\mathcal{A}}$. Applying Lemma A.1 gives

$$g_{\mathcal{A}}^{\top}H_{\mathcal{A}}^{\dagger}g_{\mathcal{A}} = \|P_{\mathrm{range}(\Phi_{\mathcal{A}})}r\|_2^2,$$

proving equation 3. An invertible reparameterization right-multiplies $\Phi_{\mathcal{A}}$ without changing its range, while adding blocks enlarges that range, proving invariance and monotonicity. Using equation 20,

$$\mathrm{range}(M^{1/2}J_b) \subseteq \mathrm{range}(M^{1/2}\mathcal{R}_b),$$

which proves equation 5. □

For one block, write $\Phi_b = U_b\Sigma_b V_b^{\top}$ and $c_{b,i} = u_{b,i}^{\top}r$. Then $g_b = -V_b\Sigma_b U_b^{\top}r$ and $H_b = V_b\Sigma_b^2 V_b^{\top}$, so

$$\|g_b\|_F^2 = \sum_i \sigma_{b,i}^2 c_{b,i}^2, \qquad g_b^{\top}H_b^{\dagger}g_b = \sum_{\sigma_{b,i}>0} c_{b,i}^2.$$

This proves equation 4. It also makes the gradient mismatch explicit: the correctable residual coefficient $c_{b,i}^2$ is multiplied by $\sigma_{b,i}^2$ before it appears in the gradient norm.

Proposition A.2 (Task dependence of sufficient depth)**.** *For output-side sets $S_c = \{c, \dots, L-1\}$, let $\mathcal{V}_{\mathcal{D},c} = \mathrm{range}(\Phi_{S_c})$. If $\mathcal{V}_{\mathcal{D},c+1} \subsetneq \mathcal{V}_{\mathcal{D},c}$, some admissible task direction is fully covered by S_c and completely missed by S_{c+1}. Architecture alone therefore cannot determine one minimal sufficient depth for every task.*

Proof. Strict nesting supplies a nonzero $r \in \mathcal{V}_{\mathcal{D},c} \cap \mathcal{V}_{\mathcal{D},c+1}^{\perp}$. After normalization, its coverage is one under S_c and zero under S_{c+1}. □

When $P_{b,t}$ is differentiable on an interval of constant rank,

$$\mathcal{S}_{\mathcal{D},t}(b) = \frac{1}{2}r_t^{\top}P_{b,t}r_t \quad \Longrightarrow \quad \dot{\mathcal{S}}_{\mathcal{D},t}(b) = r_t^{\top}P_{b,t}\dot{r}_t + \frac{1}{2}r_t^{\top}\dot{P}_{b,t}r_t,$$

which proves equation 10. The factorization equation 20 shows that $\dot{P}_{b,t}$ is induced by activation drift $\dot{A}_{b,t}$, route drift $\dot{\mathcal{R}}_{b,t}$, and any change in M_t; meanwhile $\dot{r}_t$ depends on the updates of all active blocks.

For proposed block changes $f_b = \Phi_b\Delta W_b$, direct expansion also gives

$$\frac{1}{2}\left\|r - \sum_{b\in\mathcal{A}} f_b\right\|_2^2 = \frac{1}{2}\|r\|_2^2 - \sum_{b\in\mathcal{A}}\left(\langle r, f_b\rangle - \frac{1}{2}\|f_b\|_2^2\right) + \sum_{i<j}\langle f_i, f_j\rangle. \tag{22}$$

The individual terms are block gains and the cross terms measure redundant or opposing Transformer-block changes.

A.4 TRANSFORMER-FACTORIZED SPECTRAL COST

For matrix block b, let $y_{b,t} = \mathcal{R}^*_{b,t} q_t$ and $K_{b,t} = J_{b,t} J^*_{b,t}$. Taking the adjoint of equation 20 gives

$$J^*_{b,t} q_t = y_{b,t} A^\top_{b,t}, \qquad q_t^\top K_{b,t} q_t = \|y_{b,t} A^\top_{b,t}\|_F^2. \tag{23}$$

If $\|y_{b,t}\|_F > 0$ and $A^\top_{b,t} A_{b,t} = \sum_i \sigma^2_{b,i,t} v_{b,i,t} v^\top_{b,i,t}$, the trace identity gives

$$\frac{\|y_{b,t} A^\top_{b,t}\|_F^2}{\|y_{b,t}\|_F^2} = \sum_i \frac{\|y_{b,t} v_{b,i,t}\|_2^2}{\|y_{b,t}\|_F^2} \sigma^2_{b,i,t},$$

expanding equation 7. This is the activation spectrum weighted by the loss gradient that reaches each direction through the remaining Transformer. For $y_{b,t} = 0$, we use the convention $a_{b,t} = 0$ stated in the main text.

For comparison with the loss-space form, along the trajectory for block set $\mathcal{A}$, define

$$C_{\mathcal{A},t} := M_t^{1/2} K_{\mathcal{A},t} M_t^{1/2}, \qquad z_t := M_t^{\dagger/2} q_t, \qquad c_t := \|z_t\|_2^2 = s_t. \tag{24}$$

For $c_t > 0$, let $(\lambda_{i,t}, \psi_{i,t})$ be a complete orthonormal eigensystem of $C_{\mathcal{A},t}$, including its zero eigenvalues, and write

$$\alpha_{i,t} := \langle \psi_{i,t}, z_t \rangle, \qquad \bar{\lambda}_{\mathcal{A},t} := \frac{\sum_i \lambda_{i,t} \alpha^2_{i,t}}{\sum_i \alpha^2_{i,t}}. \tag{25}$$

Equivalently, if $E_{\mathcal{A},t}$ is the spectral resolution of $C_{\mathcal{A},t}$, define

$$\mu_{\mathcal{A},t}(\Omega) := \frac{\|E_{\mathcal{A},t}(\Omega) z_t\|_2^2}{\|z_t\|_2^2}, \qquad \bar{\lambda}_{\mathcal{A},t} = \int \lambda \, d\mu_{\mathcal{A},t}(\lambda). \tag{26}$$

This notation does not match eigenvectors across time. It only summarizes the spectrum seen by the current task direction at each state. When $c_t = 0$, the loss-rate identity below is trivial, and we set $\bar{\lambda}_{\mathcal{A},t} = 0$.

Proposition A.3 (Instantaneous loss rate)**.** *Along the exact gradient-flow trajectory,*

$$-\dot{L}_t = q_t^\top K_{\mathcal{A},t} q_t = z_t^\top C_{\mathcal{A},t} z_t = c_t \bar{\lambda}_{\mathcal{A},t} = s_t \sum_{b \in \mathcal{A}} \rho_{b,t} a_{b,t}. \tag{27}$$

Proof. From equation 16, $M_t^{1/2} z_t = q_t$. Therefore,

$$z_t^\top C_{\mathcal{A},t} z_t = z_t^\top M_t^{1/2} K_{\mathcal{A},t} M_t^{1/2} z_t = q_t^\top K_{\mathcal{A},t} q_t.$$

Expanding z_t in the instantaneous eigenbasis of $C_{\mathcal{A},t}$ yields

$$z_t^\top C_{\mathcal{A},t} z_t = \sum_i \lambda_{i,t} \alpha^2_{i,t} = c_t \bar{\lambda}_{\mathcal{A},t}.$$

Because parameter blocks are orthogonal coordinates, equation 23 gives $q_t^\top K_{\mathcal{A},t} q_t = \sum_{b \in \mathcal{A}} \|y_{b,t} A^\top_{b,t}\|_F^2 = s_t \sum_b \rho_{b,t} a_{b,t}$. Combining the identities with equation 18 proves equation 8 and the result. □

Proposition A.4 (Transformer learning time and parameter cost)**.** *For trainable set $\mathcal{A}$, if $T_{\mathcal{A}}(\Delta)$ is the first time the loss falls from L_0 to $L_0 - \Delta$, then*

$$T_{\mathcal{A}}(\Delta) = \int_{L_0 - \Delta}^{L_0} \frac{du}{s_{\mathcal{A}}(u) \sum_{b \in \mathcal{A}} \rho_b(u) a_b(u)}. \tag{28}$$

The smallest joint parameter velocity producing loss-decrease rate $v > 0$ is

$$\min_{V : -q_t^\top J_{\mathcal{A},t} V \geq v} \|V\|_F = \frac{v}{\sqrt{s_t \sum_{b \in \mathcal{A}} \rho_{b,t} a_{b,t}}}. \tag{29}$$

Proof of Proposition A.4. Assume the loss decreases strictly from L_0 to $L_0 - \Delta$. It can then be used as a coordinate along the trajectory. By Proposition A.3,

$$dt = -\frac{dL_t}{s_t \sum_{b\in\mathcal{A}} \rho_{b,t} a_{b,t}}.$$

Integrating between the two loss values gives

$$T_{\mathcal{A}}(\Delta) = \int_{L_0-\Delta}^{L_0} \frac{du}{s_{\mathcal{A}}(u) \sum_{b\in\mathcal{A}} \rho_b(u) a_b(u)},$$

which proves equation 28. The quantities are evaluated on the actual $\mathcal{A}$-trajectory when its loss equals u.

For a candidate joint block velocity V, the instantaneous decrease in loss is

$$-q_t^\top J_{\mathcal{A},t} V = -\langle g_{\mathcal{A},t}, V\rangle, \qquad g_{\mathcal{A},t} := J_{\mathcal{A},t}^* q_t.$$

The Cauchy-Schwarz inequality implies

$$v \le \|g_{\mathcal{A},t}\|_F \|V\|_F$$

for every V producing decrease at least v. Equality is attained by

$$V^\star = -\frac{v}{\|g_{\mathcal{A},t}\|_F^2} g_{\mathcal{A},t}.$$

Moreover,

$$\begin{aligned}\|g_{\mathcal{A},t}\|_F^2 &= q_t^\top J_{\mathcal{A},t} J_{\mathcal{A},t}^* q_t \\ &= q_t^\top K_{\mathcal{A},t} q_t = s_t \sum_{b\in\mathcal{A}} \rho_{b,t} a_{b,t}. \end{aligned} \tag{30}$$

Hence

$$\min_{V:\, -q_t^\top J_{\mathcal{A},t} V \ge v} \|V\|_F = \frac{v}{\sqrt{s_t \sum_{b\in\mathcal{A}} \rho_{b,t} a_{b,t}}},$$

proving equation 29. Gradient flow uses $V = -g_{\mathcal{A},t}$ and attains the minimum for its own instantaneous rate $v = \|g_{\mathcal{A},t}\|_F^2$.

Gradient flow has $\|\dot{W}_{\mathcal{A},t}\|_F^2 = -\dot{L}_t$, so its path length to the target is

$$\mathcal{P}_{\mathcal{A}}(\Delta) = \int_{L_0-\Delta}^{L_0} \frac{du}{\sqrt{s_{\mathcal{A}}(u) \sum_b \rho_b(u) a_b(u)}}.$$

Pointwise dominance of the total factor therefore orders the time integral, this path-length integral, and equation 29, proving the comparison claim.

□

Remark A.5 (Interpretation of the spectral result). The comparison follows the actual trajectory of each Transformer-block set. Activations, routes, task weights, and the loss metric may all change with time; no initial activation spectrum is assumed to predict the complete run.

A.5 Transformer-Block Loss-Reduction Share Dynamics

The coupling can be read in two equivalent ways. From $g_{b,t} = J_{b,t}^* q_t$,

$$\dot{g}_{b,t} = \dot{J}_{b,t}^* q_t + J_{b,t}^* \dot{q}_t, \qquad \dot{J}_{b,t}[\Delta W_b] = \dot{\mathcal{R}}_{b,t}[\Delta W_b A_{b,t}] + \mathcal{R}_{b,t}[\Delta W_b \dot{A}_{b,t}]. \tag{31}$$

Thus gradient activity changes because the shared loss residual changes, because upstream representations change, and because the downstream route changes. The full Hessian expression below collects exactly these effects.

Whenever $\sum_j D_{\mathcal{D},t}(j) > 0$, let $p_{b,t} = D_{\mathcal{D},t}(b)/\sum_j D_{\mathcal{D},t}(j)$ be block b's share of immediate loss reduction and let $H_{bj,t} = \nabla^2_{W_b W_j} L$. For a block with nonzero gradient, define $\kappa_{b,t} = \sum_j \langle g_{b,t}, H_{bj,t} g_{j,t}\rangle / \|g_{b,t}\|_F^2$; set $\kappa_{b,t} = 0$ when $g_{b,t} = 0$, and let $\bar{\kappa}_t = \sum_j p_{j,t}\kappa_{j,t}$. Thus $\kappa_{b,t}$ summarizes how all block updates change block b's gradient.

Theorem A.6 (Coupled loss-reduction-share dynamics)**.** *On any interval where the shares of blocks b and j remain positive, exact finite-width gradient flow gives*

$$\dot{p}_{b,t} = 2p_{b,t}(\bar{\kappa}_t - \kappa_{b,t}), \qquad \frac{p_{b,T}}{p_{j,T}} = \frac{p_{b,0}}{p_{j,0}} \exp\left(2\int_0^T [\kappa_{j,t} - \kappa_{b,t}]dt\right). \tag{32}$$

A persistent gap $\kappa_{j,t} - \kappa_{b,t} \geq \delta > 0$ therefore produces at least $e^{2\delta T}$ relative separation of the two gradient shares. Cross-block curvature thereby determines how gradient activity is reassigned during training.

Proof of Theorem A.6. Under joint gradient flow, $\dot{W}_{j,t} = -g_{j,t}$, hence

$$\dot{g}_{b,t} = \sum_j H_{bj,t}\dot{W}_{j,t} = -\sum_j H_{bj,t}g_{j,t}, \qquad \frac{d}{dt}D_{\mathcal{D},t}(b) = -2\kappa_{b,t}D_{\mathcal{D},t}(b).$$

Writing $D_t = \sum_j D_{\mathcal{D},t}(j)$ gives $\dot{D}_t = -2\bar{\kappa}_t D_t$. Differentiating $p_{b,t} = D_{\mathcal{D},t}(b)/D_t$ yields $\dot{p}_{b,t} = 2p_{b,t}(\bar{\kappa}_t - \kappa_{b,t})$. Subtracting the logarithmic equations for blocks b and j and integrating proves equation 32; the exponential lower bound follows immediately. □

For the functional measurement used in Panel C, let $\Delta_{b|\mathcal{A}}$ be the update learned by Transformer block b when blocks $\mathcal{A}$ are trained jointly, let $\Delta_{-b|\mathcal{A}}$ collect the other jointly learned updates, and let $\Delta_{b|b}$ be the isolated update. Define

$$\begin{aligned} s_{b|\mathcal{A}} &:= F(\theta_0 + \Delta_{b|\mathcal{A}}) - F(\theta_0), \\ m_{b|\mathcal{A}} &:= F(\theta_0 + \Delta_{-b|\mathcal{A}} + \Delta_{b|\mathcal{A}}) - F(\theta_0 + \Delta_{-b|\mathcal{A}}), \\ u_{b|b} &:= F(\theta_0 + \Delta_{b|b}) - F(\theta_0). \end{aligned} \tag{33}$$

Lemma A.7 (Finite-increment identity)**.** *For continuously differentiable F,*

$$F(a+h) - F(a) = \int_0^1 DF(a+vh)h\,dv. \tag{34}$$

Proof. Apply the fundamental theorem of calculus to $F(a+vh)$. □

Applying Lemma A.7 with and without the other block updates gives

$$m_{b|\mathcal{A}} - s_{b|\mathcal{A}} = \int_0^1 \Big[J_b(\theta_0 + \Delta_{-b|\mathcal{A}} + v\Delta_{b|\mathcal{A}}) - J_b(\theta_0 + v\Delta_{b|\mathcal{A}})\Big]\Delta_{b|\mathcal{A}}\,dv, \tag{35}$$

$$s_{b|\mathcal{A}} - u_{b|b} = \int_0^1 J_b\big(\theta_0 + \Delta_{b|b} + v(\Delta_{b|\mathcal{A}} - \Delta_{b|b})\big)\,(\Delta_{b|\mathcal{A}} - \Delta_{b|b})\,dv. \tag{36}$$

Adding the two expressions proves equation 13. The first vanishes if the Jacobian of block b is unchanged by the other Transformer blocks; the second vanishes if joint and isolated training learn the same block update. Panel C1 isolates the first comparison. Panel C2 compares the endpoints m and u and therefore contains both terms; a direct s-versus-u comparison would be required to isolate the second.

A.6 Local Compensation After Freezing

Let $\vartheta = (\vartheta_P, \vartheta_S)$ denote the actual trainable coordinates, including LoRA factor coordinates when applicable, at a branching checkpoint translated to $t = 0$. Write $g_P = \nabla_{\vartheta_P} L$, $g_S = \nabla_{\vartheta_S} L$, and $H_{SP} = \nabla^2_{\vartheta_S \vartheta_P} L$. The joint trajectory obeys gradient flow in both groups, whereas the frozen-prefix trajectory fixes $\vartheta_P^F(t) = \vartheta_P^0$ and follows gradient flow only in S.

Proof of Proposition 3.2. At the branch, the two suffix trajectories share position and velocity:

$$\vartheta_S^F(0) = \vartheta_S^J(0), \qquad \dot{\vartheta}_S^F(0) = \dot{\vartheta}_S^J(0) = -g_S.$$

Differentiating their gradient-flow equations gives

$$\ddot{\vartheta}_S^F(0) - \ddot{\vartheta}_S^J(0) = -H_{SP} g_P.$$

Hence

$$\vartheta_S^F(t) - \vartheta_S^J(t) = -\frac{t^2}{2} H_{SP} g_P + o(t^2).$$

Expanding the loss at the common frozen prefix and using $\nabla_{\vartheta_S} L = g_S$ yields

$$L(\vartheta_P^0, \vartheta_S^J(t)) - L(\vartheta_P^0, \vartheta_S^F(t)) = \frac{t^2}{2} \langle g_S, H_{SP} g_P \rangle + o(t^2),$$

which is equation 12. □

The interaction has a direct finite-width interpretation. Let J_b be the Jacobian with respect to the trainable coordinates of block b, and let $M = \nabla_F^2 L$. Then

$$\chi_{S,P} = \left\langle \sum_{b \in S} J_b g_b, M \sum_{p \in P} J_p g_p \right\rangle + \sum_{b \in S, p \in P} \left\langle q, (D_{\vartheta_p} J_b[g_p])[g_b] \right\rangle. \tag{37}$$

The first term couples prefix and retained-layer updates through the shared output error. Let $\delta W_b(g_b)$ be the matrix perturbation induced by coordinate direction g_b. Using equation 20, the second term becomes

$$(D_{\vartheta_p} J_b[g_p])[g_b] = (D_{\vartheta_p} \mathcal{R}_b[g_p])[\delta W_b(g_b) A_b] + \mathcal{R}_b[\delta W_b(g_b) D_{\vartheta_p} A_b[g_p]]. \tag{38}$$

Thus compensation is governed by the same residual, activation, and routing changes that drive functional redistribution. The proposition is local and concerns Euclidean gradient flow in the trained coordinates; Panel C3 tests its finite-step consequence under AdamW.

A.7 Conditional Early Functional Concentration

Proposition A.8 (Conditional early functional concentration)**.** *Let θ_t collect the trainable parameters and $L^\star = \inf_\theta L(\theta)$. Suppose that after t_0, the loss is $\beta_\mathcal{D}$-smooth, satisfies the PL condition with constant $\mu_\mathcal{D}$, and the step size obeys $\eta \le 1/\beta_\mathcal{D}$. Assume also that*

$$\sup_{\xi \in [\theta_t, \theta_{t+1}]} \|M_t^{1/2} D_\theta F(\xi)\|_{\mathrm{op}} \le B_\mathcal{D}.$$

Then per-step functional movement decays geometrically as in equation 40, and its remaining sum is bounded by the corresponding geometric tail.

Proof. Let $\Delta_t = L_t - L^\star$. Smoothness gives $\|\nabla L_t\|_2^2 \le 2\beta_\mathcal{D} \Delta_t$, while the PL condition and $\eta \le 1/\beta_\mathcal{D}$ give $\Delta_{t+1} \le (1 - \eta\mu_\mathcal{D})\Delta_t$. The finite-width mean-value identity and routed-Jacobian bound give

$$\|M_t^{1/2}(F_{t+1} - F_t)\|_2 \le B_\mathcal{D} \|\theta_{t+1} - \theta_t\|_2 = B_\mathcal{D} \eta \|\nabla L_t\|_2. \tag{39}$$

Combining these inequalities gives

$$\|M_t^{1/2}(F_{t+1} - F_t)\|_2 \le B_\mathcal{D} \eta \sqrt{2\beta_\mathcal{D}(L_{t_0} - L^\star)} (1 - \eta\mu_\mathcal{D})^{(t - t_0)/2}. \tag{40}$$

Summing the right-hand side bounds later movement by the first post-burn-in bound divided by $1 - \sqrt{1 - \eta\mu_\mathcal{D}}$. □

This is a sufficient conditional result; our experiments do not estimate t_0, $\mu_\mathcal{D}$, $\beta_\mathcal{D}$, or $B_\mathcal{D}$, and do not test an exact geometric law. It also applies to a complete block sweep whenever that sweep satisfies the same descent and routed-Jacobian bounds.

A.8 Population Coverage and Boundary Recovery

Fix a checkpoint and let $x(z) = (\widehat{u}_0(z), \ldots, \widehat{u}_{L-1}(z))^\top$ for an independently sampled sequence $z \sim \mathcal{D}$. Then $G = \mathbb{E}[xx^\top]$ and $\widehat{G} = n^{-1}\sum_{i=1}^n x(z_i)x(z_i)^\top$. Assume x is sub-Gaussian relative to G, $\|x\|_2 \le B_x$, and $\mathbf{1}^\top G \mathbf{1} \ge \nu > 0$. Assume also that, throughout the stated operator-norm neighborhood, every numerical pseudoinverse retains the same rank and every retained eigenvalue is at least $\alpha > 0$. Let $r_{\text{eff}} = \operatorname{tr}(G)/\|G\|_{\text{op}}$, and define

$$c^\star = \max\{c : C_G(S_c) \ge \gamma\}, \qquad m_{\mathcal{D}} = \min\{C_G(S_{c^\star}) - \gamma, \gamma - C_G(S_{c^\star+1})\},$$

where the second term is omitted when no shorter candidate exists.

For $\delta \in (0,1)$, intrinsic-dimension matrix Bernstein gives, with probability at least $1-\delta$ and a universal constant $C > 0$,

$$\|\widehat{G} - G\|_{\text{op}} \le C B_x^2 \left[\sqrt{\frac{r_{\text{eff}} + \log(L/\delta)}{n}} + \frac{r_{\text{eff}} + \log(L/\delta)}{n}\right] =: \eta_n. \tag{41}$$

On the ball $\|\widetilde{G} - G\|_{\text{op}} \le \min\{\alpha, \nu/L\}/2$, the thresholded pseudoinverse, numerator, and denominator in equation 14 are Lipschitz. Let $\Gamma_{\mathcal{D}}$ be a common Lipschitz constant over the L suffixes. Principal submatrices satisfy $\|\widehat{G}_{SS} - G_{SS}\|_{\text{op}} \le \eta_n$, so

$$\max_c |C_{\widehat{G}}(S_c) - C_G(S_c)| \le \Gamma_{\mathcal{D}}\eta_n =: \varepsilon_n. \tag{42}$$

If $\varepsilon_n < m_{\mathcal{D}}$, the selected suffix remains above γ and the next shorter suffix remains below it, proving exact boundary recovery. Writing the current matrix as G_t, a later matrix G_{t+h} replaces η_n by $\eta_n + \|G_{t+h} - G_t\|_{\text{op}}$. An empirical certificate follows by replacing the population margin and retained eigenvalue with lower confidence bounds obtained from equation 41.

B Complete Experimental Results

SuffixXX retains the shortest output-nearest layer set reaching an XX% functional-coverage target. Table A.2 gives the task-level Qwen3-8B $K = L$ results summarized in the main text. Table A.3 reports GPT-2 Medium and Large under $K = 5$ and $K = L$; Table A.4 reports Qwen3-8B with $K = 5$; and Table A.5 reports LLaMA2-7B with $K = L = 32$. Retained is the output-nearest depth, t_{switch} is the input optimizer step at which broad training ends, and every PFLOP total includes the broad measurement phase. One sweep visits every available layer once; when $K < L$, it contains several groups of τ optimizer steps, explaining why t_{switch} ranges from 6 to 264.

For GPT-2 Medium and Large on WebNLG, E2E, and DART with $K = 5$, increasing coverage from 80% to 90% to 95% gives lower or equal validation loss whenever the selected suffix changes, supporting coverage as an allocation rule rather than an accuracy predictor. The large LLaMA2-7B IMDB difference is treated as convergence sensitivity of the all-layer baseline under the fixed budget, not evidence that freezing layers improves accuracy; we apply the same caution to the comparatively low Yelp results.

B.1 Controlled-Test Quantities

In Panel A of Table 1, each fraction compares one four-layer group with the same quantity measured for the all-layer update. Parameter fraction uses adapter-update norm; functional norm fraction uses the norm of the induced logit displacement; functional projection fraction retains only its component along the all-layer displacement; and loss-recovery fraction uses the corresponding reduction in task loss. Functional and loss efficiency divide the respective functional or loss fraction by parameter fraction. Thus the panel compares how far parameters move, how much the model output changes, how much of that change agrees with the complete update, and how much loss it recovers under a common normalization.

B.2 Compensation Intervention

Panel C3 uses one GPT-2 Large/E2E run with a single branch point after 50 shared all-layer steps. Run A continues all 36 layers for another 100 steps. Run B freezes the preceding 32 or 28 layers

and continues training only the last four or eight on the identical batches and random stream. Run C is an evaluation control: it combines the frozen prefix at the branch with the retained-layer update learned by run A. All losses are measured on the same 16 held-out batches. Thus $L_C - L_A$ is the loss gap caused by removing the later prefix update, while $L_C - L_B$ isolates the benefit of allowing the retained layers to adjust to that removal. The ratio $(L_C - L_B)/(L_C - L_A)$ is 10.6% and 21.6% for the last-four and last-eight sets; their update norms are respectively 36.2% and 20.3% larger under frozen-prefix continuation, consistent with partial compensatory movement.

B.3 What the Selected Depths Show

A fixed output-side boundary is a simple realization of the training principle, while profiling adapts that boundary to measured functional variation. Tuning a fixed boundary separately for each dataset requires a cutoff sweep; exact selection has up to L candidate runs, which the one-run profile is designed to avoid. On Qwen3-8B with $K = L$, Suffix95 ranges from 12 layers on IMDB and 13 on HellaSwag to all 36 on MNLI. These choices reduce complete-run training PFLOPs by 41.6% and 40.2% on the first two datasets, while yielding no compute reduction on MNLI, where the 28-layer Suffix80 run already costs 0.6 points. A rule retaining the final half of the model would use six layers beyond the IMDB profile, yet fall below MNLI's 28-layer Suffix80 choice and therefore below its measured 80% functional-coverage level. The 80%, 90%, and 95% rows show performance across progressively shallower data-dependent choices.

The same separation appears on LLaMA2-7B, where Suffix95 ranges from 19 to 32 layers, and across the GPT-2 settings. Performance is summarized across tasks rather than through selected single-task gains: Table 2 reports all seven paired medians directly, while compute savings follow from selected depth and t_{switch}. Coverage is a resource decision, not an accuracy ranking, and full depth remains the fallback when shortening is unsupported.

B.4 Coverage-Based Profiler

Each profiling sweep is a short experiment on the current residual. We index sweeps by k; with active-layer budget K, each sweep trains groups of at most K layers for τ steps, and $K = L$ trains all layers jointly. Let $\mathcal{M}_\ell$ contain layer ℓ's LoRA modules. For an independent context z_j, the layer's first-order loss effect over selected prediction positions is

$$u_{\ell,k}[j] = \sum_{m \in \mathcal{M}_\ell} \left\langle \nabla_{h_{m,j}} L,\, h^+_{m,j} - h^-_{m,j} \right\rangle, \tag{43}$$

where $h^-_{m,j}$ and $h^+_{m,j}$ are module outputs before and after the update. The theoretical signal $\mathcal{S}_{\mathcal{D},t}(\ell)$ measures the correction available in loss-weighted logit space, without parameter scale or learning geometry. The vector $u_{\ell,k}$ measures the effect of a completed optimizer update and includes learning strength, parameterization, and model changes accumulated during the sweep. It is the empirical guide used to choose layers; $\mathcal{S}_{\mathcal{D},t}(\ell)$ is the corresponding theoretical quantity.

We rescale each vector using its observed relation to loss reduction and denote the result by $\widehat{u}_{\ell,k}$. Their running overlap matrix $\bar{G}_k$ records change size on the diagonal and shared directions off the diagonal. For suffix S_c, let $v_c = (\bar{G}_k)_{S_c,:}\mathbf{1}$ and compute $C(S_c)$ using equation 14. This is the share of combined realized change reproduced by the suffix after accounting for shared directions. The profiler selects the shortest suffix reaching γ. Appendix A.8 gives a sufficient separation condition for recovering this observed boundary; the theoretical projection boundary is defined separately.

B.5 Implementation and Reproducibility Details

What is measured. Profiling uses ordinary minibatches from the training split and adds no transformer forward passes for alternative update choices. Layers are visited in shuffled, non-repeating groups. On the final minibatch of each profiling sweep, all available LoRA gradients are temporarily exposed, while the optimizer update is applied only to the scheduled active group. Forward and backward hooks record the input x to each LoRA branch and the output adjoint h. For Qwen3-8B and LLaMA2-7B, the profiled modules are `q_proj` and `v_proj` in every transformer layer.

For a LoRA branch with factors A and B, we form the rank-r features $z = xA^\top$ and their $r \times r$ covariance $\Sigma = z^\top z/n$. If (λ_i, e_i) are its eigenpairs and $\widetilde{g}_i$ are the corresponding components of the

Table A.2: Complete Qwen3-8B results with $K = L = 36$; all costs include the broad measurement phase.

Model	Data	K	Run	Retained	t_{switch}	Acc. (%)	Bwd. PF	Bwd. ↓	Train PF	Train ↓	Post Mem. (GB)	Mem. ↓
Qwen3-8B	IMDB	36	All	36	–	**82.2237**	62.9572	–	95.7481	–	53.6540	–
		36	Suffix95	12	6	81.3026	23.1600	63.21%	55.9509	41.56%	33.2397	38.05%
		36	Suffix90	6	6	80.6711	13.2107	79.02%	46.0016	51.96%	28.0960	47.63%
		36	Suffix80	4	6	81.6053	9.8943	84.28%	42.6851	55.42%	26.3814	50.83%
	MNLI	36	All	36	–	**80.0000**	22.3987	–	34.0726	–	37.4079	–
		36	Suffix95	36	36	**80.0000**	22.3987	0.00%	34.0726	0.00%	37.4178	−0.03%
		36	Suffix90	36	36	**80.0000**	22.3987	0.00%	34.0726	0.00%	37.4178	−0.03%
		36	Suffix80	28	36	79.3947	17.8564	20.28%	29.5303	13.33%	33.4820	10.49%
	DBpedia	36	All	36	–	78.6184	161.7151	–	245.9896	–	53.6721	–
		36	Suffix95	18	15	81.2500	84.6149	47.68%	168.8894	31.34%	38.3156	28.61%
		36	Suffix90	15	15	**81.5132**	71.7648	55.62%	156.0393	36.57%	35.7562	33.38%
		36	Suffix80	5	15	79.0132	28.9314	82.11%	113.2059	53.98%	27.2233	49.28%
	AG News	36	All	36	–	**92.8947**	29.8791	–	45.4517	–	34.7122	–
		36	Suffix95	14	9	91.5132	12.5523	57.99%	28.1249	38.12%	25.2202	27.34%
		36	Suffix90	11	9	91.2237	10.1895	65.90%	25.7621	43.32%	23.9248	31.08%
		36	Suffix80	3	9	90.2368	3.8889	86.98%	19.4615	57.18%	20.4691	41.03%
	Yelp	36	All	36	–	57.8684	130.1526	–	197.9495	–	53.7337	–
		36	Suffix95	16	18	**58.3289**	61.8594	52.47%	129.6563	34.50%	36.6070	31.87%
		36	Suffix90	13	18	57.3684	51.6154	60.34%	119.4123	39.68%	34.0468	36.64%
		36	Suffix80	6	18	57.0395	27.7128	78.71%	95.5097	51.75%	28.0731	47.76%
	BoolQA	36	All	36	–	**83.1804**	35.4800	–	53.9660	–	53.6883	–
		36	Suffix95	23	9	82.9358	23.4351	33.95%	41.9212	22.32%	42.5446	20.76%
		36	Suffix90	18	9	83.0887	18.8025	47.01%	37.2885	30.90%	38.2779	28.70%
		36	Suffix80	14	9	82.2630	15.0964	57.45%	33.5824	37.77%	34.8722	35.05%
	RTE	36	All	36	–	87.0036	23.8183	–	36.2304	–	39.0241	–
		36	Suffix95	30	18	**88.0866**	20.1507	15.40%	32.5628	10.12%	35.9519	7.87%
		36	Suffix90	29	21	81.9495	19.5649	17.86%	31.9770	11.74%	35.4210	9.23%
		36	Suffix80	25	21	85.5596	17.1343	28.06%	29.5465	18.45%	33.2986	14.67%
	WiC	36	All	36	–	71.3166	9.3760	–	14.2631	–	22.8652	–
		36	Suffix95	28	6	**73.5110**	7.4041	21.03%	12.2911	13.83%	21.5471	5.76%
		36	Suffix90	27	6	72.4138	7.1576	23.66%	12.0446	15.55%	21.3774	6.51%
		36	Suffix80	25	6	69.5925	6.6646	28.92%	11.5517	19.01%	21.0380	7.99%
	GSM8K	36	All	36	–	80.970	150.919	–	229.561	–	30.742	–
		36	Suffix95	20	15	82.790	87.006	42.3%	165.649	27.8%	25.272	17.8%
		36	Suffix90	15	15	81.122	67.034	55.6%	145.676	36.5%	23.546	23.4%
		36	Suffix80	12	15	**83.397**	55.050	63.5%	133.693	41.8%	22.510	26.8%
	SVAMP	36	All	36	–	**81.000**	5.442	–	8.279	–	19.363	–
		36	Suffix95	14	6	77.000	2.330	57.2%	5.167	37.6%	17.378	10.3%
		36	Suffix90	12	6	72.667	2.048	62.4%	4.884	41.0%	17.197	11.2%
		36	Suffix80	5	6	71.667	1.057	80.6%	3.894	53.0%	16.560	14.5%
	MathQA	36	All	36	–	66.466	710.971	–	1081.426	–	32.905	–
		36	Suffix95	29	6	**67.471**	578.653	18.6%	949.109	12.2%	30.170	8.3%
		36	Suffix90	26	6	66.131	521.946	26.6%	892.401	17.5%	28.994	11.9%
		36	Suffix80	23	6	65.059	465.238	34.6%	835.693	22.7%	27.817	15.5%
	HellaSwag	36	All	36	–	89.5041	1056.439	–	1606.888	–	28.565	–
		36	Suffix95	13	6	**92.541**	410.337	61.2%	960.786	40.2%	21.778	23.8%
		36	Suffix90	12	6	84.993	382.245	63.8%	932.695	42.0%	21.482	24.8%
		36	Suffix80	6	6	82.981	213.697	79.8%	764.146	52.4%	19.707	31.0%

B gradient, a scale-normalized local diagnostic sums $\|\widetilde{g}_i\|_2^2/\lambda_i$, with LoRA scaling and the factor $1/2$ included as in equation 3. Under $z \mapsto z/c$, both λ_i and $\|\widetilde{g}_i\|_2^2$ scale by $1/c^2$, so their ratio and ranking are unchanged. Equivalently, writing $z = U\Lambda^{1/2}V^\top$ reduces the quotient to the backward task signal projected onto the unit feature directions in U. This reproduces the scale invariance of Proposition 3.1 within the local rank-r LoRA feature space. The computation stays in that space; for the large models $r = 4$, so each eigendecomposition is 4×4.

The direction $d_{\ell,k}$ is represented by one first-order loss-change value per token in the accounting minibatch. It is obtained locally from the LoRA input, output adjoint, and a unit negative-B-gradient direction. After the optimizer step, $y_{\ell,k}$ is computed in the same token space from the exact change in the effective LoRA factors. The profiler uses the scale-normalized diagnostic only to fit a nonnegative empirical slope to realized benefit over observed sweeps. This calibration does not establish that the token-level Gram geometry matches the theoretical loss-weighted logit geometry. When a layer has no positive fitted slope, it uses the median positive slope among eligible layers; if none exists, the fallback is zero.

Table A.3: Complete GPT-2 functional-suffix results; all costs include the broad measurement phase.

Model	Data	K	Run	Retained	t_{switch}	Val. loss	PPL	Bwd. PF	Bwd. ↓	Train PF	Train ↓	Post Mem. (GB)	Mem. ↓
A. Memory-capped training: $K = 5$													
GPT-2-M	WebNLG	5	LISA	24	–	2.4476	11.5609	22.9015	–	36.8775	–	16.5694	–
		5	Suffix95	17	45	2.4639	11.7502	17.0062	25.74%	30.9822	15.99%	13.0684	21.13%
		5	Suffix90	14	45	2.5011	12.1963	14.5473	36.48%	28.5233	22.65%	11.5679	30.19%
		5	Suffix80	11	45	2.5697	13.0614	12.0227	47.50%	25.9987	29.50%	10.0674	39.24%
	E2E	5	LISA	24	–	2.6153	13.6710	53.3583	–	85.9690	–	16.5694	–
		5	Suffix95	17	75	2.6164	13.6867	39.5788	25.82%	72.1895	16.03%	13.0684	21.13%
		5	Suffix90	14	75	2.6321	13.9034	33.8585	36.55%	66.4692	22.68%	11.5679	30.19%
		5	Suffix80	10	75	2.6660	14.3819	26.1457	51.00%	58.7564	31.65%	9.5673	42.26%
	DART	5	LISA	24	–	2.6016	13.4856	79.4852	–	128.0693	–	16.5694	–
		5	Suffix95	19	45	2.6021	13.4918	64.9199	18.32%	113.5041	11.37%	14.0687	15.09%
		5	Suffix90	17	45	2.6126	13.6344	58.9954	25.78%	107.5796	16.00%	13.0684	21.13%
		5	Suffix80	15	45	2.6248	13.8016	53.3110	32.93%	101.8952	20.44%	12.0681	27.17%
GPT-2-L	WebNLG	5	LISA	36	–	2.3602	10.5935	49.8702	–	80.1133	–	29.5615	–
		5	Suffix95	29	264	2.3584	10.5744	41.5280	16.73%	71.7711	10.41%	25.1567	14.90%
		5	Suffix90	27	264	2.3604	10.5947	38.8944	22.01%	69.1375	13.70%	23.8980	19.16%
		5	Suffix80	21	264	2.3760	10.7619	31.6180	36.60%	61.8611	22.78%	20.1259	31.92%
	E2E	5	LISA	36	–	2.5575	12.9037	116.6992	–	187.2664	–	29.5615	–
		5	Suffix95	30	192	2.5536	12.8530	98.9102	15.24%	169.4773	9.50%	25.7885	12.76%
		5	Suffix90	26	192	2.5544	12.8641	87.0227	25.43%	157.5899	15.85%	23.2702	21.28%
		5	Suffix80	17	192	2.5855	13.2696	60.4156	48.23%	130.9828	30.06%	17.6125	40.42%
	DART	5	LISA	36	–	2.5287	12.5366	173.9999	–	279.1325	–	29.5615	–
		5	Suffix95	32	144	2.5271	12.5174	156.2156	10.22%	261.3483	6.37%	27.0462	8.51%
		5	Suffix90	32	144	2.5271	12.5174	156.2156	10.22%	261.3483	6.37%	27.0462	8.51%
		5	Suffix80	32	144	2.5271	12.5174	156.2156	10.22%	261.3483	6.37%	27.0462	8.51%
B. Joint all-layer training: $K = L$													
GPT-2-M	WebNLG	24	All	24	–	2.2537	9.5228	26.0596	–	40.0356	–	16.8663	–
		24	Suffix95	19	9	2.2783	9.7599	21.0364	19.28%	35.0124	12.55%	14.2874	15.29%
	E2E	24	All	24	–	2.4962	12.1365	60.8058	–	93.4165	–	16.8663	–
		24	Suffix95	21	24	2.5012	12.1969	53.7752	11.56%	86.3859	7.53%	15.3190	9.17%
	DART	24	All	24	–	2.4581	11.6821	90.5899	–	139.1740	–	16.8663	–
		24	Suffix95	19	9	2.4770	11.9058	73.1028	19.30%	121.6870	12.56%	14.2874	15.29%
GPT-2-L	WebNLG	36	All	36	–	2.1472	8.5608	58.1252	–	88.3683	–	30.1621	–
		36	Suffix95	29	15	2.1629	8.6965	47.3209	18.59%	77.5640	12.23%	25.6235	15.05%
	E2E	36	All	36	–	2.4237	11.2873	135.6254	–	206.1926	–	30.1621	–
		36	Suffix95	32	9	2.4270	11.3250	121.1839	10.65%	191.7511	7.00%	27.5686	8.60%
	DART	36	All	36	–	2.3743	10.7440	202.0579	–	307.1905	–	30.1621	–
		36	Suffix95	31	15	2.3816	10.8224	175.1666	13.31%	280.2992	8.75%	26.9202	10.75%

Algorithm 1 Broad-to-focused functional scheduling.

Require: Layers $\{0, \dots, L-1\}$, active-layer budget K, interval τ, coverage target γ, and transition step $t_{\text{switch}} \geq 0$ at a completed-sweep boundary

Initialize $k \leftarrow 0$ and $\bar{G}_0 \leftarrow 0$
while current optimizer step $< t_{\text{switch}}$ **do**
  $k \leftarrow k + 1$
  Shuffle layers into non-overlapping groups of size at most K
  Train each group for τ steps
  Measure $u_{\ell,k}$ using equation 43
  Rescale by observed loss reduction to obtain $\widehat{u}_{\ell,k}$
  $G_k[i,j] \leftarrow \langle \widehat{u}_{i,k}, \widehat{u}_{j,k} \rangle$
  $\bar{G}_k \leftarrow ((k-1)\bar{G}_{k-1} + G_k)/k$
end while
for $c = 0, \dots, L-1$ **do**
  Compute $C(S_c)$ using equation 14
end for
$c^\star \leftarrow \max\{c : C(S_c) \geq \gamma\}$
Freeze layers below $c^\star$ and continue with at most K active layers in $S_{c^\star}$

Geometry and numerical conventions. The token-level vectors are used only to form an $L \times L$ Gram matrix, which is at most 36×36 for Qwen3-8B and 32×32 for LLaMA2-7B. Gram matrices and coverage are computed in double precision. Coverage uses `torch.linalg.pinv` with the

Table A.4: Complete Qwen3-8B functional-suffix results with $K = 5$.

Model	Data	K	Run	Retained	t_{switch}	Acc. (%)	Bwd. PF	Bwd. ↓	Train PF	Train ↓	Post Mem. (GB)	Mem. ↓
Qwen3-8B	IMDB	5	LISA	36	–	81.7368	55.1776	–	87.9684	–	51.4820	–
		5	Suffix95	16	72	**82.0263**	28.0203	49.22%	60.8112	30.87%	35.8102	30.44%
		5	Suffix90	7	72	81.1842	18.2874	66.86%	51.0783	41.94%	28.7742	44.11%
		5	Suffix80	1	72	81.2368	11.1572	79.78%	43.9481	50.04%	23.8058	53.76%
	MNLI	5	LISA	36	–	82.9474	19.3549	–	31.0288	–	35.7142	–
		5	Suffix95	18	48	85.1316	10.6884	44.78%	22.3623	27.93%	28.0438	21.48%
		5	Suffix90	17	48	78.0132	10.0876	47.88%	21.7615	29.87%	27.1408	24.01%
		5	Suffix80	10	48	**85.3684**	7.1802	62.90%	18.8541	39.24%	24.4296	31.60%
	DBpedia	5	LISA	36	–	81.6184	138.9744	–	223.2489	–	51.4976	–
		5	Suffix95	19	48	83.5132	78.3441	43.63%	162.6186	27.16%	34.2496	33.49%
		5	Suffix90	19	48	83.5132	78.3441	43.63%	162.6186	27.16%	34.2496	33.49%
		5	Suffix80	17	48	**91.3158**	69.0791	50.29%	153.3536	31.31%	35.8097	30.46%
	AG News	5	LISA	36	–	89.5132	25.8315	–	41.4041	–	31.6395	–
		5	Suffix95	24	48	**90.5789**	17.9559	30.49%	33.5285	19.02%	27.2830	13.77%
		5	Suffix90	18	48	87.2500	14.0947	45.44%	29.6673	28.35%	26.4881	16.28%
		5	Suffix80	12	48	88.8816	10.0088	61.25%	25.5814	38.22%	22.7929	27.96%
	Yelp	5	LISA	36	–	57.3421	112.4923	–	180.2892	–	51.5258	–
		5	Suffix95	26	96	**58.1053**	79.1034	29.68%	146.9003	18.52%	43.7493	15.09%
		5	Suffix90	21	96	55.5263	66.9157	40.52%	134.7126	25.28%	39.8243	22.71%
		5	Suffix80	17	96	42.2763	58.5016	48.00%	126.2985	29.95%	36.6620	28.85%
	RTE	5	LISA	36	–	87.0036	20.8899	–	33.3021	–	36.2379	–
		5	Suffix95	29	72	**92.4188**	17.1848	17.74%	29.5970	11.13%	33.8259	6.66%
		5	Suffix90	21	72	86.6426	12.8735	38.37%	25.2857	24.07%	29.4640	18.69%
		5	Suffix80	9	72	86.6426	8.1178	61.14%	20.5300	38.35%	24.5735	32.19%
	WiC	5	LISA	36	–	65.6740	8.2240	–	13.1111	–	21.9775	–
		5	Suffix95	31	48	**73.9812**	6.9302	15.73%	11.8173	9.87%	21.5219	2.07%
		5	Suffix90	27	48	72.2571	6.3218	23.13%	11.2088	14.51%	21.0588	4.18%
		5	Suffix80	19	48	67.7116	4.8215	41.37%	9.7085	25.95%	19.8218	9.81%

Hermitian option and the library's default tolerance and is clipped to $[0, 1]$ because finite-precision pseudoinversion can otherwise produce slight overshoot. It is defined as one when aggregate denominator energy is at most 10^{-12}. Feature-covariance eigenvalues are floored at machine precision times the larger of the maximum eigenvalue and one. In the initialization diagnostic, no eigenvalue reaches this floor, and signal rankings agree across three disjoint eight-sequence panels with Kendall's $\tau \geq 0.98$. Functional signal is invariant to uniform activation rescaling, whereas $D_{\mathcal{D}}(b)$ and $\bar{\sigma}^2_{\mathcal{D}}(b) = D_{\mathcal{D}}(b)/(2\mathcal{S}_{\mathcal{D}}(b))$ retain the task-weighted feature scale removed from it.

Large-model training. Qwen3-8B and LLaMA2-7B are trained in bfloat16 with LoRA rank 4, scaling 8, dropout 0.1, and no bias adaptation. LoRA is applied to `q_proj` and `v_proj`. Each benchmark task is fine-tuned independently from the same initial checkpoint; the experiments are not a continual sequence. We use learning rate 10^{-3}, a constant schedule, no warmup, maximum source length 512, maximum target length 50, and effective batch size 8. Classification experiments use two epochs and reasoning experiments use three epochs. The accounting interval is three optimizer steps per active group, and the maximum profiling budget is twelve sweeps. Gradient checkpointing is disabled so that freezing the prefix shortens the realized backward graph. Dataset configurations, prompts, random seeds, and exact checkpoint identifiers are included with the supplementary code.

B.6 Efficiency Accounting

Backward and training PFLOPs count the dominant model operations over the complete run, including the broad measurement phase. One multiply-add counts as two FLOPs. The calculation includes attention and MLP projections, attention products, the vocabulary projection, propagation through all blocks downstream of the earliest active layer, and gradients for active LoRA factors. Measurement performs no additional transformer forward passes. We use PFLOPs as the primary efficiency measure because they are stable across hardware and runtime conditions, with synchronized wall-clock time as a systems check.

The reported Post Mem. quantity is peak allocated GPU memory after t_{switch}, when the excluded prefix and its optimizer state have been released. Whole-run peak memory remains determined by the broad phase, which visits the complete model depth. The implementation additionally records

Table A.5: Complete LLaMA2-7B functional-suffix results with $K = L = 32$.

Model	Data	K	Run	Retained	t_{switch}	Acc. (%)	Bwd. PF	Bwd. ↓	Train PF	Train ↓	Post Mem. (GB)	Mem. ↓
LLaMA2-7B	IMDB	32	All	32	–	57.4737	58.3391	–	87.7854	–	37.6737	–
		32	Suffix95	27	18	73.6184	49.6286	14.93%	79.0748	9.92%	34.1145	9.45%
		32	Suffix90	26	18	73.6184	47.8864	17.92%	77.3327	11.91%	33.3760	11.41%
		32	Suffix80	21	18	**73.8158**	39.1759	32.85%	68.6221	21.83%	29.6852	21.20%
	MNLI	32	All	32	–	**82.1974**	23.9006	–	35.9661	–	28.0947	–
		32	Suffix95	23	15	79.9342	17.3769	27.30%	29.4424	18.14%	24.0313	14.46%
		32	Suffix90	21	15	80.4868	15.9272	33.36%	27.9926	22.17%	23.1172	17.72%
		32	Suffix80	11	15	70.4079	8.6786	63.69%	20.7440	42.32%	18.5700	33.90%
	DBpedia	32	All	32	–	98.3684	167.0681	–	251.4046	–	37.6494	–
		32	Suffix95	21	9	98.6447	110.3533	33.95%	194.6899	22.56%	29.6215	21.32%
		32	Suffix90	19	9	98.3947	100.0416	40.12%	184.3781	26.66%	28.1484	25.24%
		32	Suffix80	10	9	**98.8947**	53.6386	67.89%	137.9752	45.12%	21.5218	42.84%
	AG News	32	All	32	–	90.6316	32.2536	–	48.5357	–	28.1394	–
		32	Suffix95	26	9	91.4211	26.3203	18.40%	42.6025	12.22%	25.4024	9.73%
		32	Suffix90	23	9	90.6579	23.3537	27.59%	39.6359	18.34%	24.0341	14.59%
		32	Suffix80	21	9	**92.1974**	21.3760	33.73%	37.6581	22.41%	23.1219	17.83%
	Yelp	32	All	32	–	59.0395	126.8329	–	190.8522	–	37.6985	–
		32	Suffix95	20	12	58.3421	80.1720	36.79%	144.1913	24.45%	28.9477	23.21%
		32	Suffix90	19	12	**59.6842**	76.2836	39.86%	140.3028	26.49%	28.2085	25.17%
		32	Suffix80	15	12	58.4737	60.7299	52.12%	124.7492	34.64%	25.2569	33.00%
	BoolQA	32	All	32	–	**75.0765**	36.2962	–	54.6177	–	37.6394	–
		32	Suffix95	21	6	73.3945	24.0982	33.61%	42.4197	22.33%	29.5652	21.45%
		32	Suffix90	19	6	72.2018	21.8804	39.72%	40.2019	26.39%	28.0988	25.35%
		32	Suffix80	14	6	62.8440	16.3358	54.99%	34.6574	36.55%	24.4328	35.09%
	RTE	32	All	32	–	76.5343	25.0985	–	37.7682	–	30.2072	–
		32	Suffix95	19	6	**79.7834**	15.1073	39.81%	27.7771	26.45%	23.5008	22.20%
		32	Suffix90	8	6	68.9531	6.6532	73.49%	19.3230	48.84%	17.8312	40.97%
		32	Suffix80	4	6	68.9531	3.5790	85.74%	16.2488	56.98%	15.7694	47.80%
	WiC	32	All	32	–	50.1567	9.9920	–	15.0363	–	18.6078	–
		32	Suffix95	32	36	50.1567	9.9920	0.00%	15.0363	0.00%	18.7260	−0.64%
		32	Suffix90	32	36	50.1567	9.9920	0.00%	15.0363	0.00%	18.7260	−0.64%
		32	Suffix80	32	36	50.1567	9.9920	0.00%	15.0363	0.00%	18.7260	−0.64%

phase-specific allocated and reserved memory, broad-phase time, suffix-training time, and transition overhead.

C Limitations and Statistical Scope

Coverage is not an accuracy predictor. Coverage measures how much of the profiled loss-change pattern a suffix can represent. Downstream accuracy can vary across the 80%, 90%, and 95% targets. HellaSwag and SVAMP show accuracy loss under aggressive shortening, while gains on other tasks may reflect regularization or convergence noise. We therefore use coverage as a resource-control variable and retain full depth whenever the target requires it.

Conditional theory and empirical certificates. The boundary-recovery result requires independent profiling sequences, a retained spectral gap, and an empirical error smaller than the population margin. We do not estimate $m_{\mathcal{D}}$ or ε_n, so the experiments demonstrate stable observed boundaries rather than certify the theorem's finite-sample condition. Likewise, equation 40 is a sufficient consequence of local smoothness, a PL condition, and bounded routing; the experiments show a large early-to-late decline but do not estimate these constants or establish geometric decay. The fixed-mode relation in equation 9 is isolated by construction in Panel B, whereas the full nonlinear comparison in Proposition A.4 would require measuring its trajectory-wide factor at matched loss.

Scope. The profiler uses a first-order, loss-local measurement and may become stale when later feature learning changes routes substantially. The input t_{switch} and median fallback are practical controls. The cumulative profile in Figure 4 smooths transient changes because later checkpoints inherit early displacement; it establishes stability of accumulated functional depth rather than an unaveraged single-sweep boundary. The interval profile is the noisier non-cumulative check. The large-model diagnostic uses LoRA rank 4; sensitivity to rank and calibration is not evaluated, although no measured eigenvalue reaches the numerical floor. The bootstrap intervals in Figure 4 quantify variation across datasets, not uncertainty across training seeds. Our evidence covers GPT-2, Qwen3-

8B, and LLaMA2-7B on the reported tasks; other architectures, adapter parameterizations, longer horizons, and distribution shifts may yield different profiles.