UNDERSTANDING OPTIMIZATION FROM ACTIVATION AND GRADIENT SPECTRA: A MODDED-NANOGPT STUDY

Anonymous authors
Paper under double-blind review

ABSTRACT

We study the activation covariance and per-sample gradient SVD spectra throughout training in a modded-NanoGPT study. We show that the tail exponents of these spectra reveal differences in internal geometry that the training loss does not. We propose using these spectra as practical predictors of training optimization in two ways. First, we show that the tail exponent of the activation covariance spectrum at an early checkpoint predicts the terminal token efficiency. In controlled experiments in which the runs reach the same terminal loss, it achieves a Spearman correlation of 0.970, compared with 0.212 for the early test loss. Second, we show that along a cumulative chain of modded-NanoGPT interventions, the activation and gradient spectra together distinguish the changes in internal geometry that accompany gains in token efficiency, in throughput, or in both. For example, the Muon$\rightarrow$Untied transition improves token efficiency by $2.09\times$ while throughput rises by only 2.0%, a shift marked primarily by a change in the activation covariance tail. Both signals hold across architectures and for model sizes up to 3.57B parameters. We explain this behavior with an exactly solvable kernel model of next-token prediction on modular arithmetic. Our code is available here.

1 INTRODUCTION

The optimization of large language models is largely guided by neural scaling laws, which show that test loss evolves predictably with compute, data, and model size (Kaplan et al., 2020; Hoffmann et al., 2022). This steady decrease of the loss with scale means that increasing scale unlocks downstream capabilities (Brown et al., 2020; Rae et al., 2021; Ganguli et al., 2022). Because the training loss decreases just as smoothly, training and test loss curves have become the standard diagnostic of a training run. These curves, however, are used *post hoc* to guide empirical experimentation, which is prohibitively expensive at scale. This motivates better metrics that can predict the optimization dynamics *in vivo*. Such a metric should correlate strongly with how optimization reshapes the internal representations, so that its value at a given point in training encodes information about how these representations will evolve. The training and test losses fall short of this requirement, since they are consequences of optimization shaping the ever-evolving internal representations rather than measures of the representations themselves, and therefore reveal little about how the internal geometry changes during training (Li et al., 2025; Liu et al., 2025b).

To construct such metrics, we turn to spectral learning theories, which connect performance to the distribution of feature eigenvalues and the alignment of the learning task with those features (Bahri et al., 2024; Bordelon et al., 2020; Canatar et al., 2021). Inspired by this connection, we develop empirical measures of optimization from activation covariance and per-sample gradient singular-value-decomposition (SVD) spectra, which capture how the activation covariance is distributed across internal dimensions and how per-sample gradients concentrate across directions. These measures enable tracking of the internal geometry of representations and of the gradients that drive them throughout training, thereby offering a window into feature learning as it unfolds.

We demonstrate the utility of these measures in a study on modded-NanoGPT (Jordan, 2024b). We first show that the activation spectrum measured early in training ranks training configurations (e.g., batch size, learning rate, or optimizer) by the token cost they incur later in training. Applying this to

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

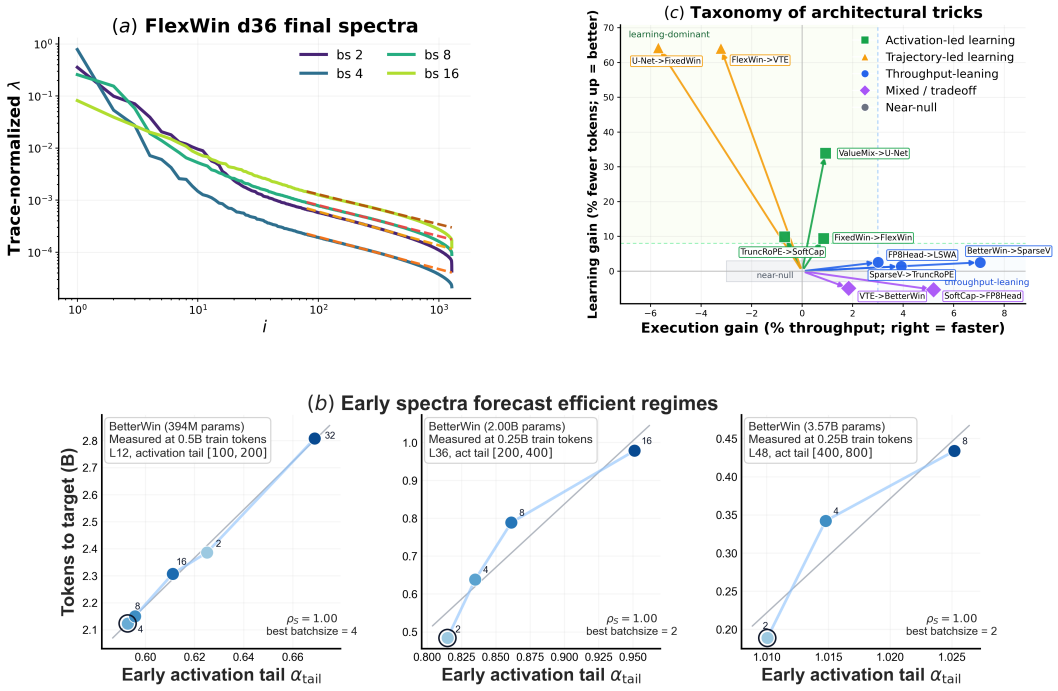


Figure 1: **Spectral diagnostics for training optimization.** Each panel compares training runs from modded-NanoGPT using trace-normalized activation covariance spectra and per-sample gradient SVD spectra, aligned either at the same loss or at the same early token count. **(a) Same loss, different internal geometry.** Final-layer activation spectra of FlexWin d36 runs aligned at the same target loss. Dashed lines show power-law fits $\lambda_k \propto k^{-\alpha_{\text{tail}}}$ over the tail window. **(b) Early spectra forecast token efficiency across model sizes.** Correlation between the early activation tail exponent, measured at the same token count for all runs, and the total tokens required to reach the target loss, for models of depth 12, 36, and 48 (d12, d36, d48). **(c) Taxonomy of interventions.** Consecutive architectural transitions, all trained at batch size bs8, classified by their spectral changes and the corresponding gains in token efficiency and throughput.

the batch size within a fixed architecture, we find that the tail slope of the activation spectrum, α_{tail}, measured at an early time ranks the later token cost with a Spearman correlation of $\bar{\rho}_{\text{within}} = 0.970$, computed within each model family and averaged over families. In contrast, the validation loss measured at the same early checkpoint reaches a correlation of only 0.212, indicating that it carries little information about which configuration will ultimately require fewer tokens. We show that this behavior holds for models with up to 3.57B parameters. We then turn to the cumulative chain of architectural and optimizer interventions in modded-NanoGPT, and use the spectra to classify each intervention by whether it improves token efficiency, throughput, or both. For example, for the Muon→Untied transition, we show that the activation spectrum is substantially modified while the per-sample gradient spectrum remains largely unchanged, accompanied by a $2.09\times$ improvement in token efficiency and only a 2.0% increase in throughput. One possible explanation for this effect is feature learning, in which training redistributes capacity toward task-relevant directions as representations are transformed through the network, enabling the model to achieve a given loss with fewer tokens.

We capture this mechanism in a toy model in which α_{tail} sets the learning rates vary across modes, and show that a flatter tail in the activation spectrum implies a rate of learning that decays more slowly with mode rank, so that weaker modes, those further down the spectrum, are learned earlier in training. This explains our empirical observation that runs with different batch sizes can have the same loss yet differ in their activation spectra, with a flatter tail predicting fewer tokens needed to reach a target loss. We also show, within this model, that the activation spectrum tracks the learned component of the target in each mode, whereas the per-sample gradient spectrum serves as a proxy for the residual component, so that the two spectra carry complementary information, which justifies their use as metrics for classifying optimization interventions.

Related work. Spectral methods have been widely used to study representation geometry during optimization, including analyses of the eigendecay and effective rank of activation covariance (Li et al., 2025; Chang et al., 2022; Garrido et al., 2023), the concentration of gradients in low-dimensional subspaces and outliers in the Hessian spectrum (Gur-Ari et al., 2018; Ghorbani et al., 2019; Papyan, 2019), power-law structure in stochastic-gradient covariance (Xie et al., 2023), heavy-tailed weight spectra and their dependence on the training regime (Mahoney & Martin, 2019), and spectral mechanisms of neural scaling laws (Bahri et al., 2024; Spigler et al., 2020; Bordelon et al., 2020). The effect of batch size on internal geometry and training efficiency has also been studied through the sharpness of the minima reached at different batch sizes (Keskar et al., 2017) and the gradient noise scale (McCandlish et al., 2018). Unlike these works, we use activation and gradient spectra not only to understand how internal representations coevolve with optimization, but also to forecast token efficiency early in training, to reveal differences in internal geometry between batch sizes that the loss does not show, and to identify how optimization interventions improve training. Additional related work is in Appendix A.

2 Framework

2.1 Spectral diagnostics

A standard decoder-only transformer (Vaswani et al., 2017) passes token embeddings through L residual blocks. Rather than relying only on the loss computed from the output logits, our spectral diagnostics probe the intermediate representations within these blocks. For a chosen layer l, let $h_i^{(l)} \in \mathbb{R}^d$ denote the hidden state at token position i, computed on the same set of test sequences for all checkpoints and runs. Let $H_l \in \mathbb{R}^{N\times d}$ contain the hidden states $h_i^{(l)}$, with N token positions and hidden width D. With sample mean $\bar{h}$, we compute, for a given layer,

$$\hat{\Sigma}_h = \frac{1}{N-1}\sum_{i=1}^{N}(h_i^{(l)} - \bar{h})(h_i^{(l)} - \bar{h})^\top \quad \text{and normalized eigenvalues } \tilde{\lambda}_j = \frac{\lambda_j}{\sum_k \lambda_k}, \tag{1}$$

where $\lambda_1 \geq \cdots \geq \lambda_d \geq 0$ are the eigenvalues of $\hat{\Sigma}_h$. The trace normalization ensures that our comparisons depend on the *shape* of the spectrum rather than its raw scale.

Inspired by theory that links the spectral decay to neural scaling laws (Bahri et al., 2024; Maloney et al., 2022; Bordelon et al., 2020; Canatar et al., 2021; Spigler et al., 2020), we quantify the power-law decay $\tilde{\lambda}_j \propto j^{-\alpha}$ of the activation spectrum to characterize the optimization dynamics. For a band I of mode indices, we define the exponent $\alpha(I) := -\mathrm{slope}_{j\in I}\Big((\log j, \log \tilde{\lambda}_j)\Big)$, obtained from a least-squares fit. We estimate the tail exponent $\alpha_{\mathrm{tail}} = \alpha^{\mathrm{act}}_{I^{(s)}}$ over the tail band $I^{(s)}$, which depends on the model scale s (d12, d36, d48). We select the fitting window that maximizes log-log linearity, that is, the rank range over which the normalized spectrum is closest to a straight line on log-log axes, as required for a power law. For d12 (with with $d = 768$), we scan the windows $I_r = [r, 2r]$ over $r = 20, 30, \ldots, 380$, and use the window with the highest mean R^2 of the power-law fit, averaged over architecture families, which yields $I^{(d12)} = [100, 200]$. We determine the window before examining token efficiency, so that its choice is independent of the prediction results. To control for scale, we then select $I^{(d36)} = [200, 400]$, and $I^{(d48)} = [400, 800]$, each of which is log-log linear with mean $R^2 = 0.999$. The starting rank of the window doubles from one scale to the next, consistent with larger models resolving more feature directions and thus placing the boundary between the learned head and the unresolved tail at higher rank. Appendix G.4 reports the full scan.

To complement the activation spectrum, we also probe the optimization dynamics of the model. For a given weight matrix $W \in \mathbb{R}^{d_{\mathrm{out}}\times d_{\mathrm{in}}}$ with $P = d_{\mathrm{out}} \cdot d_{\mathrm{in}}$ parameters, we compute per-sample gradients on the same set of test sequences; here a *sample* is one sequence not one token. Let $g_m = \mathrm{vec}(\nabla_W L_m(\theta)) \in \mathbb{R}^P$ denote the flattened gradient with respect to W of the autoregressive loss averaged over the tokens of sample m. Stacking the M gradients as rows gives the per-sample gradient matrix

$$G = [g_1 \ldots g_M]^\top \in \mathbb{R}^{M\times P}. \tag{2}$$

We then compute the singular value spectrum of G. Because G is defined for a single weight matrix W, its spectrum depends on which matrix is chosen as the probe. By default, we use the output

projection W_O of the attention layer in the final block; we examine other choices in Appendix C. The choice of matrix changes the degree of spectral concentration and the shape of the tail but not the qualitative conclusions. The two spectra play complementary roles. The activation spectrum describes how the activation covariance is distributed across internal dimensions, whereas the gradient spectrum describes how the per-sample gradients concentrate across directions. We compare runs aligned by terminal loss, by token count, and by fraction of the training schedule. We focus here mainly on the comparison at matched terminal loss, which isolates differences in internal geometry between runs that the loss cannot distinguish

2.2 Experimental Setup

Our experiments evaluate training interventions adopted from the modded-NanoGPT codebase (Jordan, 2024b). The interventions, specified in Table 1, are applied cumulatively, so that each variant keeps all modifications of the preceding variants and adds a new one. Details are in Appendix B.

Table 1: **Experimental blocks and model families.** We summarize the core training configurations here. Details of the cumulative changes are given in Appendix B.

Experimental block	Model families	Core setting
Variants 1–4	Baseline, RoPE, Muon, Untied	d12 GPT-2-small scale models (1k context), trained on the 100B-token split of FineWeb.
Variants 5–16	ValueMix, U-Net, FixedWin, FlexWin, VTE, BetterWin, SparseV, TruncRoPE, SoftCap, FP8Head, LSWA, AttnScale	d12 models trained on the 10B-token split of FineWeb. ValueMix and U-Net remain in the short-context regime; the 1k→65k context shift occurs at U-Net→FixedWin. All regimes match a target total batch budget of ≈0.5M tokens/step (batch tier $B = 8$, or bs8).
Larger-scale variants	FlexWin d36, BetterWin d36, SparseV d36; BetterWin d48	Runs testing consistency at up to 48 layers and 32k context, over several batch tiers.

We train each run until it reaches a target test loss of 3.2 on a fixed held-out set drawn from the FineWeb-10B test split (Penedo et al., 2024). We train the earlier, less data-efficient variants on the 100B-token FineWeb split and switch to the 10B-token split from the ValueMix variant onward. The two splits have nearly identical token-window covariance spectra (Fig. 5), so the switch does not confound the comparisons across the two splits.

Defining the batch size requires care because the context length changes along the intervention chain. We index runs by a batch tier B, equivalently the effective batch size, which fixes the number of *tokens* per optimizer step rather than the number of sequences. The d12 reference tier $B = 8$ corresponds to ≈0.5M tokens/step, realized as 512 sequences at 1k context or 8 sequences at 65k context. The other tiers scale this token budget in proportion to B. Defining the tiers by tokens per step makes them comparable across the 1k→65k shift. We evaluate $B \in \{1, 2, 4, 8, 16, 32\}$ for d12, $B \in \{2, 4, 8, 16\}$ for d36, and $B \in \{1, 2, 4, 8\}$ for d48. To ensure that our spectral comparisons reflect the effect of batch size rather than mistuned optimization, we select the learning rate for every family and tier before any analysis, using a sweep over the first 0.5B tokens pruned with ASHA (Li et al., 2020) (Appendix B). In other words, each comparison at matched target loss uses the tuned per-tier learning rates.

Let $T(B)$ be the total number of tokens a training run at batch tier B requires to reach the target loss. We measure the relative token cost, or token inefficiency, within each architectural family as

$$\epsilon_{\text{tok}}(B) = \frac{T(B)}{\min_{B'} T(B')}, \tag{3}$$

where the minimum is taken over the tiers B' swept within the same family. Thus $\epsilon_{\text{tok}} = 1$ marks the token-optimal tier for a given family, with larger values indicating proportionally more tokens. Unless noted otherwise, we use $T(B)$ to count the training tokens consumed up to the first checkpoint at which the target loss is reached. We treat this token count as a proxy for compute (given that the architecture is the same within each comparison). The probes contribute subdominantly to compute cost; for example, one joint activation and gradient probe on a single d12 layer at a single checkpoint takes 281 s, or 0.40% of the wall-clock time of the corresponding training run (Appendix D).

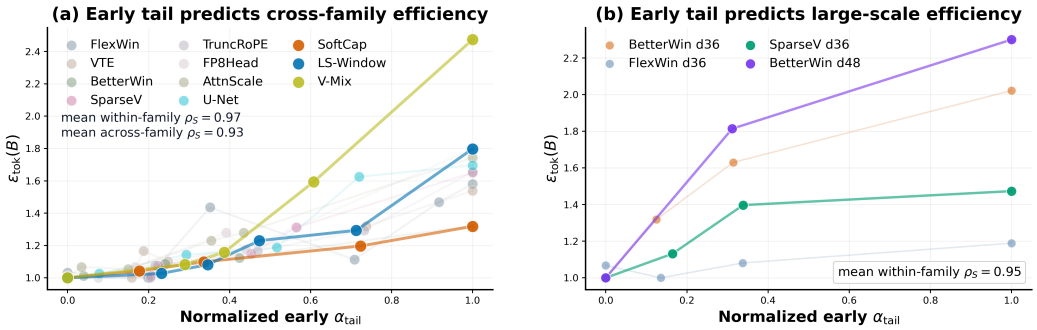


Figure 2: **Early tail exponent predicts token inefficiency across training configurations and model sizes.** Both panels show the early α_{tail} against $\epsilon_{\text{tok}}(B)$. An upward trend means that a flatter early tail identifies the more token-efficient tier. **(a)** For d12, the mean within-family Spearman correlation is $\rho_S = 0.97$, and the mean across-family correlation is $\rho_S = 0.93$. **(b)** The same comparison for d36 and d48, with a mean within-family correlation of $\rho_S = 0.95$.

3 Early Spectra Forecast Later Token Efficiency

We apply the measures of Section 2 to the effective batch size and show that the activation covariance spectrum early in training forecasts which batch tier will consume fewer tokens. First, we establish that batch size leaves distinct activation covariance spectra even at matched target loss (Appendix J). This suggests that the spectrum encodes information about optimization that the loss does not. We confirm this by showing that the early tail exponent α_{tail} ranks the batch tiers by their later token cost. Finally, we show that both observations hold from d12 to d36 and d48.[1]

Forecasting protocol. We use our framework to forecast the relative order of runs with respect to token inefficiency. Within each model family, we rank the batch tiers by their early α_{tail}, measured at the same token count across runs, and score this ranking against the later token inefficiency $\epsilon_{\text{tok}}(B)$ using the Spearman correlation ρ_S. We report the mean over families $\bar{\rho}_{\text{within}}$, and a correlation ρ_{across} over all runs pooled across families. Each configuration uses the learning rate we select as described in Section 2.2, so we compare configurations with tuned hyperparameters.[2] We assess ρ_S against the correlations obtained from three baselines at the same checkpoint, described below. Because ρ_S is a rank correlation, α_{tail} predicts the order of the tiers by token inefficiency, not the number of tokens each tier requires.

Early ranking accuracy. Following the protocol above, we find that at the early checkpoint of 0.5B tokens, the tail exponent ranks the batch tiers by their later token cost with $\bar{\rho}_{\text{within}} = 0.970$ within each family (Fig. 2a) and $\bar{\rho}_{\text{across}} = 0.930$ across families. The ordering persists under held-out calibration in a leave-one-family-out test, in which we choose the fitting window and all other calibrated settings using only the remaining families and then evaluate on the held-out family, obtaining $\rho_S = 0.945$ (95% CI $[0.857, 0.971]$). The confidence interval is computed by resampling families, so it reflects variation across architectures (Appendix G.6).

Comparison with baselines. We next test whether simpler diagnostics are as informative. We repeat the ranking with three diagnostics that require no spectral computation, namely the early test loss, the recent slope of the test loss, and the gradient norm, and score each in the same way (Table 7). At 0.5B tokens none is competitive: the tail exponent reaches $\rho_S = 0.970$, the early test loss only 0.212, and no baseline exceeds 0.367. By 1.0B tokens the loss-based predictors improve (early test loss 0.624), although the tail exponent still leads (0.911). This is expected: by then the faster runs have pulled ahead, so the current loss already indicates which runs will reach the target first. The spectral probe is therefore most useful early in training, before the loss can distinguish the runs.

Consistency across model scale. These results extend to the d36 and d48 (BetterWin 3.57B-parameter) models. Across these runs, the early tail exponent reaches $\bar{\rho}_{\text{within}} = 0.95$ (Fig. 2b), and

[1] We note that these observations concern training efficiency, not downstream capability. At the target loss, accuracy on seven downstream benchmarks is nearly the same across batch tiers and model families (Appendix F).

[2] We do not account for the cost of tuning.

Table 2: **Outcome and spectral response of each step of the d12 chain.** Step $a \to b$ adds the named variant to the cumulative chain; gains are relative to the preceding variant (Section 4.1) and shaded from loss (red) to gain (teal). The dashed rule marks the FineWeb 100B→10B split (Table 1).

Step	Variant	Token gain (%)	Throughput gain (%)	Outcome	Spectral response
1	Baseline	*reference trunk*			
1→2	RoPE	+1180.6	−2.5	T	A
2→3	Muon	+37.5	−2.8	T	A
3→4	Untied	+109.0	+2.0	J	A
4→5	ValueMix	+44.8	+26.5	J	A
5→6	U-Net	+33.9	+0.9	J	A
6→7	FixedWin	+64.3	−5.7	T	G
7→8	FlexWin	+9.4	+0.8	J	A
8→9	VTE	+64.1	−3.2	T	G
9→10	BetterWin	−4.9	+1.8	T	M
10→11	SparseV	+2.5	+7.0	J	E
11→12	TruncRoPE	+2.5	+3.0	J	E
12→13	SoftCap	+9.9	−0.7	T	A
13→14	FP8Head	−5.3	+5.2	T	M
14→15	LSWA	+1.4	+3.9	J	E
15→16	AttnScale	−0.1	−1.7	N	—

Outcome: J joint (both gains positive), T tradeoff (opposite signs), N near-null (both below 3% in magnitude). Spectral response: A activation-dominant, G gradient-dominant, E execution-dominant, M mixed; — unassigned.

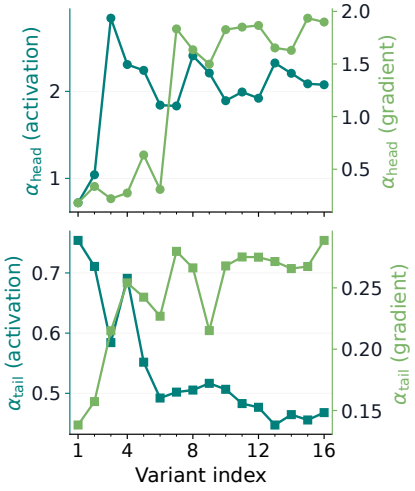


Figure 3: **Activation and gradient exponents along the intervention chain.** Final-layer activation (teal, left axes) and per-sample gradient (green, right axes) exponents α_{head} (top) and α_{tail} (bottom) for each variant. The two vary somewhat independently, e.g. U-Net→FixedWin (6→7) shifts mainly the gradient exponents, and RoPE→Muon (2→3) mainly the activation exponents.

the runs at matched terminal loss still yield distinct spectra at greater depth and context length. The log-log linearity extends over a window that moves to higher rank with depth, consistent with deeper models resolving more feature directions.

Robustness with respect to seed, probe layer, and optimizer. We run three control experiments. First, three seeds of the FlexWin configuration at $B = 16$ produce indistinguishable test-loss curves and nearly identical spectra, so seed noise does not account for the results (Appendix G). Second, we test how the result depends on the probed layer. In all three d36 families, the final layer (layer 35) gives the strongest positive correlation, whereas earlier layers give a weak signal or one whose sign varies across families. This supports our use of the final layer as the probe. Third, we test whether the result depends on the optimizer by training an LSWA variant with both Adam and Muon. Under each optimizer, the spectra remain distinct across batch tiers, so the effect is not specific to Muon, although Muon produces a sharper concentration in the leading eigenvalues (Appendix E).

4 Activation and Gradient Spectra Classify Interventions

Having established the spectra as predictors of the training efficiency, where we showed that, when varying the batch size, the early activation tail ranks batch tiers by their later token cost, we now keep the batch tier constant at $B = 8$ and instead vary the architecture and optimizer, applying the same activation and gradient measures along the cumulative chain of modded-NanoGPT interventions. We characterize each intervention by whether it improves token efficiency, throughput, or both, and by whether the accompanying change in training geometry lies mainly in the activation spectrum, which signals a change in the inner representations, or in the per-sample gradient spectrum, which signals a change in the optimization dynamics.

4.1 Intervention Outcome and Spectral Responses

For each transition $a \to b$ between consecutive variants at batch tier $B = 8$, we define

$$\text{TokGain}(a \to b) = \frac{T(a)}{T(b)} - 1, \qquad \text{ThrGain}(a \to b) = \frac{Q(b)}{Q(a)} - 1, \tag{4}$$

where $T(\cdot)$ is the number of training tokens needed to reach the target loss and $Q(\cdot)$ is the throughput in tokens per second. The wall-clock speedup reported by the modded-NanoGPT speedrun is then $(1 + \mathrm{TokGain})(1 + \mathrm{ThrGain})$, so our analysis separates the contributions of token efficiency and throughput to the speedup. Because the interventions are applied cumulatively, each gain measures the effect of adding one modification to the preceding variant, which may differ from the effect of that modification on its own. The Untied→ValueMix transition also switches from the 100B-token to the 10B-token FineWeb split.

Based on the results, we classify the transitions as follows:

Activation-dominant (A): The token gain is accompanied by a clear shift in the final-layer activation covariance spectrum, in particular in α_{head}. These interventions directly change the representational degrees of freedom or the readout geometry (e.g., RoPE, untied heads, FlexWin, SoftCap).

Gradient-dominant (G): The main change is in the per-sample gradient spectrum, or in the spectra at intermediate checkpoints, while the activation spectrum at the target loss changes little. These interventions typically change routing or the attention window (e.g., FixedWin, VTE).

Execution-dominant (E): These interventions mainly reduce the cost of computation, sometimes at the expense of token efficiency. They include masking, kernel optimizations, and optimizer tuning (e.g., SparseV, TruncRoPE).

Mixed (M): The intervention improves one of token efficiency and throughput while worsening the other by a comparable amount (e.g., a positive token gain with a negative throughput gain), so it does not fall clearly into a single category (e.g., FP8Head)

This classification is consistent with the mechanism of each intervention (Appendix B). Examining what each intervention modifies in the architecture or optimizer, we can trace why a given intervention produces a specific spectral response:

Representational shifts manifest as activation-dominant transitions. The clearest cases are transitions that directly expand or reshape the representational degrees of freedom of the model. In Muon→Untied, untying the token embedding from the LM head enables the model to learn the embedding and unembedding matrices separately. In TruncRoPE→SoftCap, bounding the logits with a $\tanh$ changes the geometry of the outputs seen by the loss. In both cases, the main signature we find is a clear shift in the final-layer activation covariance spectrum.

Changes to routing and the attention window manifest as gradient-dominant transitions. These transitions change how information moves through the network, with comparatively small changes to the final-layer readout. In U-Net→FixedWin, a sliding-window FlexAttention mask replaces full causal attention, which changes which tokens attend to one another but not the feature space. In FlexWin→VTE, a value-embedding pathway is added at each layer, which changes how token information is routed to the attention values. In both cases, the per-sample gradient and the activation spectra at intermediate checkpoints change more than the activation spectrum at the target loss.

Changes motivated by systems efficiency manifest as execution-dominant transitions. These transitions come late in the chain and mainly improve throughput, with comparatively small changes to the activation spectrum. BetterWin→SparseV replaces the dense value embeddings with sparse tables that are reused across layers, SparseV→TruncRoPE applies the rotary position embedding to only a subset of the dimensions of each attention head, and FP8Head→LSWA mainly improves throughput. In all three, the throughput gain is clearer than a change in spectral geometry.

This taxonomy relies on different spectral properties than our earlier batch-size analysis. Early prediction (Section 3) depends on the tail of the activation spectrum, which reflects the modes not yet learned, or unresolved mass. This taxonomy depends on changes in the head of the activation spectrum and in the concentration of the per-sample gradients, which track the balance between the learned and residual components of the target.

5 A Solvable Model: Why Early Tails Forecast Token Cost

We now explain the forecast of Section 3 in an exactly solvable model. Proofs are in Appendix H.

Task and model. Consider next-token prediction on arithmetic progressions modulo an odd integer c. The vocabulary is $\mathbb{Z}_c$. A starting token $a \in \mathbb{Z}_c$, drawn uniformly, determines the context $x(a) = (a, a + d, \ldots, a + (L - 1)d) \in \mathbb{Z}_c^L$ and the next token $y(a) = a + Ld$. The model outputs a

score vector $f(a) \in \mathbb{R}^c$ and is trained with the squared loss toward the target $Y(a)_t = s(t - y(a))$, $t \in \mathbb{Z}_c$. Here the mean-zero profile $s : \mathbb{Z}_c \to \mathbb{R}$ gives partial credit to tokens near $y(a)$. The centered one-hot target, which gives credit only to $y(a)$, is the special case in which s is concentrated at 0. Increasing a by one adds one to every token, so it is natural to use a learner with this symmetry built in. We take the kernel limit of such a learner, for instance a linear model on fixed features or an infinitely wide network. After n training examples (nL tokens), its scores f_n follow, in the gradient-flow limit of SGD (Appendix H.3),

$$\frac{d}{dn} f_n = -K(f_n - Y), \qquad f_0 = 0, \qquad (Kf)(a) = \frac{1}{c} \sum_{a' \in \mathbb{Z}_c} \kappa(a - a') f(a'), \tag{5}$$

where the kernel κ depends only on $a - a'$. The kernel stays fixed while the scores evolve. As the activation covariance, we take the covariance of the scores, $\Sigma(n) = \mathbb{E}_a[f_n(a) f_n(a)^\top]$, which equals that of the final hidden state when the unembedding is a fixed isometry.

Exact spectrum. Because κ depends only on $a - a'$, the operator K is diagonal in the discrete Fourier basis. We write ρ_k for its eigenvalue at frequency $k \in \{1, \dots, m\}$, where $m = (c-1)/2$. Since the target $Y(a)_t$ depends only on $t - a$, it is a sum of Fourier modes $e^{2\pi i k(t-a)/c}$. The mode of frequency k has energy $|\beta_k|^2 = c\,|\hat{s}(k)|^2$, where $\hat{s}(k)$ is the Fourier coefficient of the profile s at frequency k, and modes of different frequencies are orthogonal.

Proposition 1 (Exact). *Let $|\beta_k|^2$ be positive and let $|\beta_k|^2$ and ρ_k be nonincreasing in k. For $n > 0$, the eigenvalues of $\Sigma(n)$ are*

$$\Lambda_k(n) = |\beta_k|^2 \big(1 - e^{-\rho_k n}\big)^2, \qquad k = 1, \dots, m,$$

each of multiplicity two and nonincreasing in k, and the loss is $\mathcal{L}(n) = \frac{1}{2}\mathbb{E}_a \|f_n(a) - Y(a)\|^2 = \sum_k |\beta_k|^2 e^{-2\rho_k n}$.

A frequency that has been learned ($\rho_k n \gg 1$) contributes its target energy $|\beta_k|^2$. One that has not ($\rho_k n \ll 1$) contributes $|\beta_k|^2 \rho_k^2 n^2$. If $|\beta_k|^2 \propto k^{-p}$ and $\rho_k \propto k^{-q}$, the spectrum therefore has a head with exponent p, set by the data, and a tail with exponent

$$\alpha_{\text{tail}} = p + 2q, \tag{6}$$

which records how learning slows down with frequency. The tail can be measured before the high frequencies are learned. Once they are learned, the tail follows the data exponent p and no longer reflects the learning rates, which is why it must be measured early in training.

Batch size and rate budget. For SGD with batch size B and step size γ, the rates are $\rho_k = \gamma \lambda_k / B$, where λ_k are the eigenvalues of the network's kernel. Below the critical batch size, stability caps γ at order $B / \sum_k \lambda_k$ (McCandlish et al., 2018), so a tuned run spends a total rate $\Gamma = \sum_k \rho_k$ that does not depend on B. The batch size and the optimizer can change how this budget is spread across frequencies, but not its total. We therefore compare regimes b with the same budget Γ and rates $\rho_k(b) \propto k^{-q_b}$. For plain SGD the exponent q_b is fixed by the features; adaptive optimizers such as Adam and Muon reshape it in a way that can depend on the batch size.

Flat tails trade higher early loss for lower token cost. Under a fixed budget, a flatter rate profile speeds up the high frequencies by slowing down the low ones.

Theorem 2 (Informal version of Theorem 4). *Consider regimes with a common budget and exponents $q_1 < \cdots < q_K$, and a target whose energy $|\beta_k|^2$ decreases with k. At early times, the steepest profile has the lowest loss. For low enough target losses, however, the time to reach the target increases with q_b, and so does the early tail exponent. Hence at an early checkpoint, the tail exponent ranks the regimes by token cost with Spearman correlation $+1$, while the early loss ranks them with -1.*

Fig. 4 shows the mechanism. Early loss is dominated by the leading frequencies, which hold most of the target energy and which steep profiles learn sooner. A low target loss requires the high frequencies, which flat profiles learn sooner. The tail exponent reads the profile directly, so it identifies the eventual winner while the losses still point the other way. This is the split seen in Section 3, where the early tail exponent ranks batch tiers with $\bar{\rho}_{\text{within}} = 0.970$ and the early loss with 0.212. The inversion needs target energy concentrated at low frequencies, a signature of *smooth* targets. With one-hot targets, every frequency carries the same energy, and the early loss ranks the regimes correctly. Above the

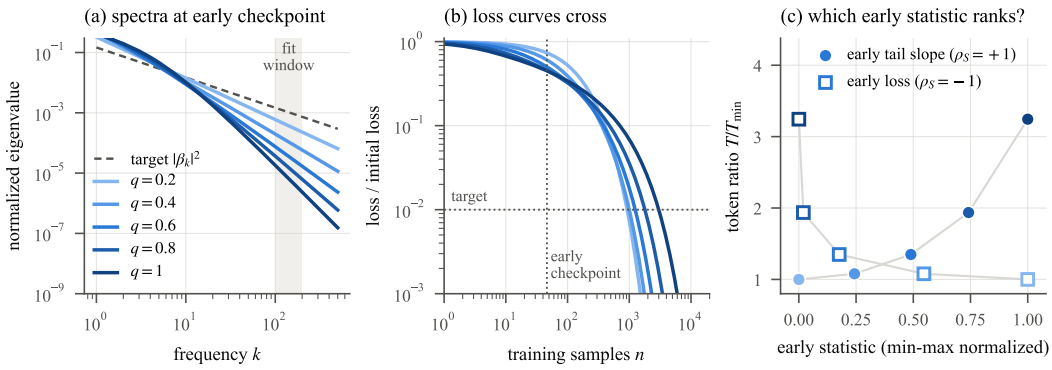


Figure 4: **Exact solutions of Eq.** (5) for $c = 1001$, $|\beta_k|^2 \propto k^{-1}$ ($p = 1$), and five regimes with the same budget and $q = 0.2, 0.4, \ldots, 1$ (light to dark). **(a)** Normalized eigenvalues of $\Sigma(n)$ early in training. The head follows the target spectrum (dashed); the tail steepens with q, with fitted exponents from 1.38 to 2.95 on the shaded window, close to $p + 2q$. **(b)** Loss relative to its initial value. Steep profiles lead early, and flat profiles overtake them before the target loss is reached. **(c)** Time to reach 1% of the initial loss against each early statistic. The early tail exponent ranks the regimes exactly (Spearman $+1$); the early loss ranks them in reverse (-1).

critical batch size, the budget itself shrinks, which moves the loss and tail in the same direction. A sweep that mixes both kinds of change therefore weakens the early-loss correlation but not the tail correlation (Appendix H.5).

We also train a two-layer transformer on the same modular-arithmetic task. Unlike the kernel model, networks trained on this task away from the kernel limit are known to represent tokens with a few Fourier modes of $\mathbb{Z}_c$ (Nanda et al., 2023; Gromov, 2023), so feature learning can be measured directly (Appendix I). As in the language-model experiments, the batch size changes the activation spectrum, and the early tail exponent ranks the runs by token cost. Here the spectral change corresponds to feature learning, whereby the activations move during training from a nearly uniform spread across Fourier modes to a few sharp peaks.

6 Conclusion

In this work, we take a first step towards a spectral approach to optimization. We use activation covariance and per-sample gradient spectra to diagnose training optimization. Across batch sweeps and the cumulative architectural intervention chain of modded-NanoGPT, we show that these measurements reveal distinct internal representations, predict token efficiency from the early activation-tail exponent, and disentangle gains in token efficiency from gains in throughput. We attribute this behavior to the fact that activation spectra encode the distribution of representation variance, while gradient spectra capture how per-sample loss gradients concentrate across directions. We then introduce a toy model that captures these signals. Because the early tail exponent ranks batch tiers before the loss does, our results suggest measuring spectra during training to adjust the batch size or learning rate, or to decide early which runs in a hyperparameter sweep to continue. However, a few limitations remain. The tail window is currently chosen by a scan at each model scale. A rule that selects it automatically, for example from the model width or from the spectrum itself, would allow the method to be applied to new models without calibration. Because the interventions are applied cumulatively, each measured effect includes interactions with the modifications before it, so our analysis does not isolate the effect of individual interventions, which future work could do by applying each intervention to a common baseline. Our experiments also use dense decoder-only models, so a natural next test is mixture-of-experts routing at production scale. Routing changes how information moves through the network without directly altering the final readout, so our classification predicts a gradient-dominant signature. We hope these results can open a path toward a more principled science of optimization at scale.

AI USE STATEMENT

AI assistance was used to edit and polish parts of the manuscript for readability and to create analysis and plotting scripts. The main pretraining experiments are written by human. AI was also used to check rigour of mathematical proofs. The original idea, empirical findings and the toy-model theory, and all conclusions drawn from them are the authors' own. We have reviewed all AI-assisted work and all AI-drafted text and figures were revised by the authors for accuracy. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

All code required to reproduce our experiments, including the spectral measurement pipeline and the analysis scripts that produce every figure, is available at the anonymized repository linked here. The measurement protocol and the definitions of the spectral summaries are given in Section 2; the architectural variants, efficiency accounting, and transition-outcome coordinates are specified in Sections 2.2 and 4.1. Appendix B documents the full architectural dimensions, the cumulative sequence of code modifications with per-variant attribution, model cards and parameter counts, and the training specifics, including sequence lengths, the FineWeb splits used by each variant, hardware, and the fixed held-out validation target that defines token efficiency. Appendix C gives the per-sample gradient computation and the choice of gradient matrix. Appendix G reports the ablations that characterize the stability of the reported correlations, covering Spearman correlation over training percentage, random-seed repetitions, and the layerwise scan. The toy-model construction and its optimization and parameterization interventions are described in Appendix I, and complete statements, explicit assumptions, and proofs for all theoretical claims are given in Appendix I.4. All language-model experiments use the public FineWeb corpora; no new dataset was collected or released.

REFERENCES

Kumar Krishna Agrawal, Arnab Kumar Mondal, Arna Ghosh, and Blake A. Richards. α-ReQ: Assessing representation quality in self-supervised learning by measuring eigenspectrum decay. In *Advances in Neural Information Processing Systems*, volume 35, pp. 17626–17638, 2022. doi: 10.52202/068431-1281. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/70596d70542c51c8d9b4e423f4bf2736-Abstract-Conference.html.

Yasaman Bahri, Ethan Dyer, Jared Kaplan, Jaehoon Lee, and Utkarsh Sharma. Explaining neural scaling laws. *Proceedings of the National Academy of Sciences*, 121(27):e2311878121, 2024. doi: 10.1073/pnas.2311878121. URL https://doi.org/10.1073/pnas.2311878121.

Adrien Bardes, Jean Ponce, and Yann LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. In *Proceedings of ICLR*, 2022. URL https://openreview.net/forum?id=xm6YD62D1Ub.

Jeremy Bernstein and Laker Newhouse. Old optimizer, new norm: An anthology. In *OPT 2024: Optimization for Machine Learning*, 2024. URL https://openreview.net/forum?id=ux18f5nOpD. NeurIPS 2024 workshop.

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

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.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, 2021. doi: 10.1038/s41467-021-23103-1. URL https://doi.org/10.1038/s41467-021-23103-1.

Tyler A. Chang, Zhuowen Tu, and Benjamin K. Bergen. The geometry of multilingual language model representations. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 119–136, 2022. doi: 10.18653/v1/2022.emnlp-main.9. URL https://aclanthology.org/2022.emnlp-main.9/.

Youngmin Cho and Lawrence K. Saul. Kernel methods for deep learning. In *Advances in Neural Information Processing Systems*, volume 22, 2009. URL https://papers.nips.cc/paper_files/paper/2009/hash/5751ec3e9a4feab575962e78e006250d-Abstract.html.

Kawin Ethayarajh. How contextual are contextualized word representations? comparing the geometry of BERT, ELMo, and GPT-2 embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pp. 55–65, 2019. doi: 10.18653/v1/D19-1006. URL https://aclanthology.org/D19-1006/.

Deep Ganguli, Danny Hernandez, Liane Lovitt, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova Dassarma, Dawn Drain, Nelson Elhage, et al. Predictability and surprise in large generative models. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, pp. 1747–1764, 2022. doi: 10.1145/3531146.3533229. URL https://doi.org/10.1145/3531146.3533229.

Quentin Garrido, Randall Balestriero, Laurent Najman, and Yann LeCun. Rankme: Assessing the downstream performance of pretrained self-supervised representations by their rank. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 10929–10974. PMLR, 2023. URL https://proceedings.mlr.press/v202/garrido23a.html.

Gemma Team. Gemma 2: Improving open language models at a practical size, 2024. URL https://arxiv.org/abs/2408.00118.

Behrooz Ghorbani, Shankar Krishnan, and Ying Xiao. An investigation into neural net optimization via hessian eigenvalue density. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2232–2241. PMLR, 2019. URL https://proceedings.mlr.press/v97/ghorbani19b.html.

Andrey Gromov. Grokking modular arithmetic, 2023. URL https://arxiv.org/abs/2301.02679.

Guy Gur-Ari, Daniel A. Roberts, and Ethan Dyer. Gradient descent happens in a tiny subspace, 2018. URL https://arxiv.org/abs/1812.04754.

Jianliang He, Leda Wang, Siyu Chen, and Zhuoran Yang. On the mechanism and dynamics of modular addition: Fourier features, lottery ticket, and grokking, 2026a. URL https://arxiv.org/abs/2602.16849.

Jianliang He, Leda Wang, Fengzhuo Zhang, Siyu Chen, and Zhuoran Yang. Neural networks provably learn spectral representations for group composition, 2026b. URL https://arxiv.org/abs/2606.02993.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Oriol Vinyals, Jack W. Rae, and Laurent Sifre. Training compute-optimal large language models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 30016–30030, 2022. doi: 10.52202/068431-2176. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/c1e2faff6f588870935f114ebe04a3e5-Abstract-Conference.html.

Prateek Jain, Sham M. Kakade, Rahul Kidambi, Praneeth Netrapalli, and Aaron Sidford. Parallelizing stochastic gradient descent for least squares regression: Mini-batching, averaging, and model misspecification. *Journal of Machine Learning Research*, 18(223):1–42, 2018. URL https://jmlr.org/papers/v18/16-595.html.

Keller Jordan. Muon: An optimizer for hidden layers in neural networks, 2024a. URL https://kellerjordan.github.io/posts/muon/.

Keller et al Jordan. modded-nanogpt: Speedrunning the nanogpt baseline, 2024b. URL https://github.com/KellerJordan/modded-nanogpt.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models, 2020. URL https://arxiv.org/abs/2001.08361.

KellerJordan/modded-nanogpt contributors. modded-nanogpt record 11: U-net pattern skip connections and double lr, 2024a. URL https://github.com/KellerJordan/modded-nanogpt/blob/bc3a0c2d640d0d73dedaef87eae26148d2e32afb/records/track_1_short/2024-11-10_UNetDoubleLr/c87bb826-797b-4f37-98c7-d3a5dad2de74.txt.

KellerJordan/modded-nanogpt contributors. modded-nanogpt record 12: 1024-ctx dense causal attention to 64k-ctx flexattention, 2024b. URL https://github.com/KellerJordan/modded-nanogpt/blob/bc3a0c2d640d0d73dedaef87eae26148d2e32afb/records/track_1_short/2024-11-19_FlexAttention/8384493d-dba9-4991-b16b-8696953f5e6d.txt.

KellerJordan/modded-nanogpt contributors. modded-nanogpt record 13: Attention window warmup, 2024c. URL https://github.com/KellerJordan/modded-nanogpt/blob/bc3a0c2d640d0d73dedaef87eae26148d2e32afb/records/track_1_short/2024-11-24_WindowWarmup/cf9e4571-c5fc-4323-abf3-a98d862ec6c8.txt.

KellerJordan/modded-nanogpt contributors. modded-nanogpt record 14: Value embeddings, 2024d. URL https://github.com/KellerJordan/modded-nanogpt/tree/bc3a0c2d640d0d73dedaef87eae26148d2e32afb/records/track_1_short/2024-12-04_ValueEmbed.

KellerJordan/modded-nanogpt contributors. modded-nanogpt record 16: Split value embeddings, block sliding window, separate block mask, 2024e. URL https://github.com/KellerJordan/modded-nanogpt/tree/bc3a0c2d640d0d73dedaef87eae26148d2e32afb/records/track_1_short/2024-12-10_MFUTweaks.

KellerJordan/modded-nanogpt contributors. modded-nanogpt record 17: Sparsify value embeddings, improve rotary embeddings, drop an attn layer, 2024f. URL https://github.com/KellerJordan/modded-nanogpt/tree/bc3a0c2d640d0d73dedaef87eae26148d2e32afb/records/track_1_short/2024-12-17_SparsifyEmbeds.

KellerJordan/modded-nanogpt contributors. modded-nanogpt record 1: llm.c baseline, 2024g. URL https://github.com/KellerJordan/modded-nanogpt/blob/bc3a0c2d640d0d73dedaef87eae26148d2e32afb/records/track_1_short/2024-10-13_llmc/main.log. Replacement link for the repository's record-1 log; the originally uploaded 2024-05-28_llmc path did not resolve.

KellerJordan/modded-nanogpt contributors. modded-nanogpt record 2: Tuned learning rate and rotary embeddings, 2024h. URL https://github.com/KellerJordan/modded-nanogpt/blob/bc3a0c2d640d0d73dedaef87eae26148d2e32afb/records/track_1_short/2024-06-06_AdamW/f66d43d7-e449-4029-8adf-e8537bab49ea.log.

KellerJordan/modded-nanogpt contributors. modded-nanogpt record 3: Introduced the muon optimizer, 2024i. URL https://github.com/KellerJordan/modded-nanogpt/tree/bc3a0c2d640d0d73dedaef87eae26148d2e32afb#world-record-history. The repository's world-record table lists record 3, but says no log is available; the originally uploaded 2024-10-04_Muon path did not resolve.

KellerJordan/modded-nanogpt contributors. modded-nanogpt record 8: Untied embedding and head, 2024j. URL https://github.com/KellerJordan/modded-nanogpt/blob/bc3a0c2d640d0d73dedaef87eae26148d2e32afb/records/track_1_short/2024-11-03_UntieEmbed/d6b50d71-f419-4d26-bb39-a60d55ae7a04.txt.

KellerJordan/modded-nanogpt contributors. modded-nanogpt record 9: Value and embedding skip connections, momentum warmup, logit softcap, 2024k. URL https://github.com/KellerJordan/modded-nanogpt/blob/bc3a0c2d640d0d73dedaef87eae26148d2e32afb/records/track_1_short/2024-11-06_ShortcutsTweaks/dd7304a6-cc43-4d5e-adb8-c070111464a1.txt.

KellerJordan/modded-nanogpt contributors. modded-nanogpt record 18: Lower logit softcap from 30 to 15, 2025a. URL https://github.com/KellerJordan/modded-nanogpt/blob/bc3a0c2d640d0d73dedaef87eae26148d2e32afb/records/track_1_short/2025-01-04_SoftCap/31d6c427-f1f7-4d8a-91be-a67b5dcd13fd.txt.

KellerJordan/modded-nanogpt contributors. modded-nanogpt record 19: Fp8 head, offset logits, lr decay to 0.1 instead of 0.0, 2025b. URL https://github.com/KellerJordan/modded-nanogpt/blob/bc3a0c2d640d0d73dedaef87eae26148d2e32afb/records/track_1_short/2025-01-13_Fp8LmHead/c51969c2-d04c-40a7-bcea-c092c3c2d11a.txt.

KellerJordan/modded-nanogpt contributors. modded-nanogpt record 20: Merged qkv weights, long-short attention, attention scale, lower adam epsilon, batched muon, 2025c. URL https://github.com/KellerJordan/modded-nanogpt/blob/bc3a0c2d640d0d73dedaef87eae26148d2e32afb/records/track_1_short/2025-01-16_Sub3Min/1d3bd93b-a69e-4118-aeb8-8184239d7566.txt.

Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. In *International Conference on Learning Representations*, 2017. URL https://arxiv.org/abs/1609.04836.

Liam Li, Kevin Jamieson, Afshin Rostamizadeh, Ekaterina Gonina, Jonathan Ben-tzur, Moritz Hardt, Benjamin Recht, and Ameet Talwalkar. A system for massively parallel hyperparameter tuning. In *Proceedings of Machine Learning and Systems*, volume 2, pp. 230–246, 2020. URL https://proceedings.mlsys.org/paper_files/paper/2020/hash/a06f20b349c6cf09a6b171c71b88bbfc-Abstract.html.

Melody Zixuan Li, Kumar Krishna Agrawal, Arna Ghosh, Komal Kumar Teru, Adam Santoro, Guillaume Lajoie, and Blake A. Richards. Tracing the representation geometry of language models from pretraining to post-training. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-1829. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/4eac651e6c06d6b374332cd14ab50f55-Abstract-Conference.html.

Andy Zeyi Liu, Elliot Paquette, and John Sous. Evolution of the spectral dimension of transformer activations. In *OPT 2025: Optimization for Machine Learning*, 2025a. URL https://openreview.net/forum?id=Va5is76bTP.

Yizhou Liu, Ziming Liu, and Jeff Gore. Superposition yields robust neural scaling. In *Advances in Neural Information Processing Systems*, volume 38, 2025b. doi: 10.52202/085713-5320. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/e97ac22927560eb2de6b658498cbc575-Abstract-Conference.html.

Ziming Liu, Ouail Kitouni, Niklas S. Nolte, Eric J. Michaud, Max Tegmark, and Mike Williams. Towards understanding grokking: An effective theory of representation learning. In *Advances in Neural Information Processing Systems*, volume 35, pp. 34651–34663, 2022. doi: 10.52202/068431-2511. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/dfc310e81992d2e4cedc09ac47eff13e-Abstract-Conference.html.

Michael W. Mahoney and Charles H. Martin. Traditional and heavy tailed self regularization in neural network models. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 4284–4293. PMLR, 2019. URL https://proceedings.mlr.press/v97/mahoney19a.html.

Alexander Maloney, Daniel A. Roberts, and James Sully. A solvable model of neural scaling laws, 2022. URL https://arxiv.org/abs/2210.16859.

James Martens and Roger Grosse. Optimizing neural networks with Kronecker-factored approximate curvature. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 2408–2417. PMLR, 2015. URL https://proceedings.mlr.press/v37/martens15.html.

Sam McCandlish, Jared Kaplan, Dario Amodei, and OpenAI Dota Team. An empirical model of large-batch training. *arXiv preprint arXiv:1812.06162*, 2018. doi: 10.48550/arXiv.1812.06162. URL https://arxiv.org/abs/1812.06162.

William Merrill, Nikolaos Tsilivis, and Aman Shukla. A tale of two circuits: Grokking as competition of sparse and dense subnetworks, 2023. URL https://arxiv.org/abs/2303.11873. Workshop on Understanding Foundation Models at ICLR 2023.

Paulius Micikevicius, Dusan Stosic, Neil Burgess, Marius Cornea, Pradeep Dubey, Richard Grisenthwaite, Sangwon Ha, Alexander Heinecke, Patrick Judd, John Kamalu, Naveen Mellempudi, Stuart Oberman, Mohammad Shoeybi, Michael Siu, and Hao Wu. FP8 formats for deep learning, 2022. URL https://arxiv.org/abs/2209.05433.

Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. Progress measures for grokking via mechanistic interpretability. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=9XFSbDPmdW.

Pascal Jr. Tikeng Notsawo, Hattie Zhou, Mohammad Pezeshki, Irina Rish, and Guillaume Dumas. Predicting grokking long before it happens: A look into the loss landscape of models which grok, 2023. URL https://arxiv.org/abs/2306.13253.

Vardan Papyan. Measurements of three-level hierarchical structure in the outliers in the spectrum of deepnet hessians. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5012–5021. PMLR, 2019. URL https://proceedings.mlr.press/v97/papyan19a.html.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben Allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. The FineWeb datasets: Decanting the web for the finest text data at scale. In *Advances in Neural Information Processing Systems*, volume 37, pp. 30811–30849, 2024. doi: 10.52202/079017-0970. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/370df50ccfdf8bde18f8f9c2d9151bda-Abstract-Datasets_and_Benchmarks_Track.html.

Alethea Power, Yuri Burda, Harri Edwards, Igor Babuschkin, and Vedant Misra. Grokking: Generalization beyond overfitting on small algorithmic datasets, 2022. URL https://arxiv.org/abs/2201.02177.

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

Jack W Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. Scaling language models: Methods, analysis & insights from training Gopher. *arXiv preprint arXiv:2112.11446*, 2021. doi: 10.48550/arXiv.2112.11446. URL https://arxiv.org/abs/2112.11446.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems*, volume 20, 2007. URL https://papers.nips.cc/paper_files/paper/2007/hash/013a006f03dbc5392effeb8f18fda755-Abstract.html.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, Lecture Notes in Computer Science, pp. 234–241. Springer, 2015. doi: 10.1007/978-3-319-24574-4_28. URL https://doi.org/10.1007/978-3-319-24574-4_28.

Andrew M. Saxe, James L. McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. In *International Conference on Learning Representations*, 2014. URL https://arxiv.org/abs/1312.6120.

Utkarsh Sharma and Jared Kaplan. Scaling laws from the data manifold dimension. *Journal of Machine Learning Research*, 23(9):1–34, 2022. URL https://www.jmlr.org/papers/v23/20-1111.html.

Stefano Spigler, Mario Geiger, and Matthieu Wyart. Asymptotic learning curves of kernel methods: Empirical data versus teacher–student paradigm. *Journal of Statistical Mechanics: Theory and Experiment*, 2020(12):124001, 2020. doi: 10.1088/1742-5468/abc61d. URL https://doi.org/10.1088/1742-5468/abc61d.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024. doi: 10.1016/j.neucom.2023.127063. URL https://doi.org/10.1016/j.neucom.2023.127063.

William Timkey and Marten van Schijndel. All bark and no bite: Rogue dimensions in transformer language models obscure representational quality. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 4527–4546, 2021. doi: 10.18653/v1/2021.emnlp-main.372. URL https://aclanthology.org/2021.emnlp-main.372/.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017. URL https://papers.neurips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Zhichao Wang, Denny Wu, and Zhou Fan. Nonlinear spiked covariance matrices and signal propagation in deep neural networks. In *Proceedings of the Thirty Seventh Conference on Learning Theory*, volume 247 of *Proceedings of Machine Learning Research*, pp. 4891–4957. PMLR, 2024. URL https://proceedings.mlr.press/v247/wang24b.html.

Zeke Xie, Qian-Yuan Tang, Mingming Sun, and Ping Li. On the overlooked structure of stochastic gradients. In *Proceedings of NeurIPS*, volume 36, pp. 66257–66276, 2023. doi: 10.52202/075280-2892. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/d0b2eda0386f477ab14d7e181e16c899-Abstract-Conference.html.

Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stephane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 12310–12320. PMLR, 2021. URL https://proceedings.mlr.press/v139/zbontar21a.html.

APPENDIX / SUPPLEMENTAL MATERIAL

This appendix reorganizes the experimental details, controls, and extended plots behind the main paper. It first expands the related work, then documents the model and training setup, the spectral measurement protocol, optimizer controls, predictive ablations, toy-model evidence, and finally the larger figure atlases.

Appendix contents.

1. Additional related work
2. Model training setup
3. Per-sample gradient computation
4. Computational cost of the spectral probes

5. Muon/Adam comparison
6. Downstream behavior at matched test loss
7. Predictive ablations
8. The solvable model in full
9. Toy model
10. Additional plots

A Additional Related Work

The main paper uses spectra as operational diagnostics, so the closest prior work spans representation geometry, optimization spectra, spectral learning theory, and simplified models of algorithmic feature learning.

Representation spectra and effective-rank summaries. RankMe (Garrido et al., 2023) motivates the entropy effective rank as a compact unsupervised summary of representation spread, while α-ReQ (Agrawal et al., 2022) uses eigenspectrum decay as a representation-quality signal in self-supervised learning. Covariance and dimensionality summaries also appear implicitly in redundancy-reduction objectives such as Barlow Twins and VICReg (Zbontar et al., 2021; Bardes et al., 2022), where representation collapse or excessive concentration is treated as a training pathology. In language models, anisotropy and low-dimensional dominant directions have long been observed in contextual embeddings (Ethayarajh, 2019; Timkey & van Schijndel, 2021); these results motivate reporting the whole spectrum, or at least several spectral summaries, rather than relying only on raw cosine geometry. More recent work uses covariance eigendecay and geometry phases to study representation changes during pretraining and post-training (Li et al., 2025; Chang et al., 2022). The most directly related activation-spectrum precursor is *Evolution of the Spectral Dimension of transformer Activations* (Liu et al., 2025a), which studies heavy-tailed activation spectra and spectral-exponent evolution across layers and training. Our use is complementary: we test whether early and matched-loss spectra predict batch efficiency and distinguish architectural interventions.

Optimization-side spectra and heavy-tail views. Optimization spectra provide a complementary lens on the update path. Prior work shows that gradient descent can concentrate in a low-dimensional subspace (Gur-Ari et al., 2018), stochastic-gradient covariance can exhibit power-law structure (Xie et al., 2023), and Hessian spectra contain informative outliers tied to data and class structure (Ghorbani et al., 2019; Papyan, 2019). Heavy-tailed random-matrix analyses likewise connect spectral decay exponents to optimization regimes and batch-size effects (Mahoney & Martin, 2019), while scaling-law theory connects spectral structure to learning curves and generalization (Bahri et al., 2024; Sharma & Kaplan, 2022). We therefore treat activation and gradient spectra as related but non-interchangeable measurements: one describes represented variance, the other describes update concentration.

Spectral learning theory and random-feature models. The toy analysis starts from linearized gradient flow because, in kernel and random-feature regimes, eigenvalues of the feature covariance or kernel operator directly control learning rates across target components. Random Fourier features provide a controlled finite-dimensional approximation to shift-invariant kernels (Rahimi & Recht, 2007), and later learning-curve analyses make the dependence on kernel spectra and task alignment explicit (Bordelon et al., 2020; Canatar et al., 2021; Spigler et al., 2020). Recent nonlinear spiked-covariance theory studies related signal-propagation and feature-selection questions (Wang et al., 2024). This line of work motivates the appendix's first toy step: before studying transformers, isolate how a spectrum over Fourier modes shapes which task components are learned early, late, or not at all.

Fourier mechanisms in grokking and modular arithmetic. Modular arithmetic is a natural second toy setting because the cyclic group has an explicit Fourier basis. The original grokking experiments showed delayed generalization on small algorithmic datasets, including modular tasks (Power et al., 2022). Subsequent work connected these dynamics to structured representation learning and phase diagrams (Liu et al., 2022), reverse-engineered modular-addition transformers into Fourier-space circuits with continuous progress measures (Nanda et al., 2023), and studied interpretable two-layer

solutions for modular arithmetic (Gromov, 2023). Recent analyses of two-layer modular-addition networks further emphasize single-frequency Fourier features, phase alignment, and gradient-flow competition among frequencies (He et al., 2026a). More generally, He et al. (2026b) study two-layer networks for finite-group composition. For projected gradient flow under a small-logit approximation to the risk, they prove that each neuron almost surely selects a single nontrivial irreducible representation and its conjugate, with rank-one alignment of the surviving Fourier coefficients across layers. Other grokking studies frame delayed generalization through competing subnetworks or early spectral signatures of the learning curve (Merrill et al., 2023; Notsawo et al., 2023). Our toy analysis uses these explicit coordinates to distinguish modal progress, concentration on a known teacher support, and measured Fourier concentration in transformer representations.

B Model Training Setup

This section elaborates on the architectural tricks, model cards, dataset choices, and learning-rate selection protocol used by the language-model experiments. The suite follows a cumulative intervention path: each later label inherits the implementation choices of the previous label unless the row states a new change.

B.1 Architectural tricks and variants

Table 3 integrates the intervention descriptions, external provenance, and mechanism-level grouping used throughout the paper. The table is intentionally descriptive rather than code-level: it names what changes in the model or optimizer and how that change should be interpreted in the spectral analysis.

B.2 Model card and parameter counts

Table 4 gives the canonical tier notation used throughout the paper: d12, d36, and d48 refer to 12-, 36-, and 48-layer transformer profiles, respectively. Table 5 then summarizes the model families and parameter counts used in the paper. Counts include registered trainable parameters and exclude buffers, masks, cached rotary tables, and other runtime state. All counts use the padded training vocabulary size of 50,304 tokens. Optimizer-only controls, such as the LSWA Adam comparison, therefore share the same parameter count as the corresponding architectural family.

The scaled full-VTE profile is larger than the half-VTE BetterWin profile, but the completed scale-robustness analysis reported here uses BetterWin and sparse-family follow-ups. We therefore reserve the "largest run" statement for the completed BetterWin d48 experiment rather than for every model profile that can be instantiated by the architecture.

B.3 Training specifics

The short-context prefix variants use sequence length $1{,}024$ and the 100B-token FineWeb sample for training; the long-context variants uses sequence length $65{,}536$ for the 12-layer runs and the 10B-token split for training. The d36 and d48 robustness runs use the long-context family with sequence length $32{,}768$. All 12-layer runs are trained on a single H200; the d36/d48 follow-up uses a single B200. Unless stated otherwise, the main d12 comparisons target test loss 3.2 on a fixed held-out validation set drawn from the FineWeb-10B validation split, which is reused across the 100B/10B training-data bridge. Token efficiency is measured by the number of training tokens consumed before the first checkpoint meeting that target.

FineWeb is a cleaned English CommonCrawl corpus drawn from 96 dumps spanning 2013 through April 2024 and containing approximately 15 trillion GPT-2 tokens (Penedo et al., 2024). The two sampled splits are used for practical reasons: less token-efficient early variants are run on the 100B-token sample, while later variants use the 10B-token split. To verify that this switch does not introduce a spectral confound, we compare $1{,}024$ windows of $1{,}024$ tokens from the local FineWeb-10B shard against the Hugging Face FineWeb sample-100BT split. As shown in Fig. 5, the covariance spectra nearly overlap: the Jensen–Shannon divergence between the two splits is 4.31×10^{-4}, well below the run-to-run variability observed in the same spectral diagnostic. This supports treating the two data samples on a common spectral scale while explicitly noting the protocol difference.

Stage	Group	Incremental change	Provenance / reference
Baseline	Reference trunk	GPT-2-small-style decoder with learned absolute positions, tied token embedding and LM head, dense causal attention, RMS normalization, and squared-ReLU MLPs.	GPT-2 (Radford et al., 2019); modded-NanoGPT record (KellerJordan/modded-nanogpt contributors, 2024g).
RoPE	Positional control	Replaces learned positions with rotary embeddings and explicit query/key/value projections; normalizes query/key states before rotation.	RoFormer (Su et al., 2024); record (KellerJordan/modded-nanogpt contributors, 2024h).
Muon	Optimizer	Keeps the RoPE trunk but changes hidden-layer matrix updates to Muon; embeddings, LM head, and scalar/control parameters remain being trained with Adam.	Muon (Jordan, 2024a); record (KellerJordan/modded-nanogpt contributors, 2024i).
Untied	Parameterization	Unties the token embedding and LM head and assigns them separate optimizer groups.	Record (KellerJordan/modded-nanogpt contributors, 2024j).
ValueMix	Value pathway	Adds cross-layer value mixing through learned interpolation between the current value tensor and previous-layer value state.	Record (KellerJordan/modded-nanogpt contributors, 2024k).
U-Net	Depth routing	Adds encoder–decoder-style skip connections across the 12-layer stack.	U-Net (Ronneberger et al., 2015); record (KellerJordan/modded-nanogpt contributors, 2024a).
FixedWin	Attention geometry	Replaces dense causal attention with document-aware local FlexAttention over a fixed horizon.	Record (KellerJordan/modded-nanogpt contributors, 2024b).
FlexWin	Attention geometry	Warms the local attention horizon during training instead of using one fixed window throughout.	Record (KellerJordan/modded-nanogpt contributors, 2024c).
VTE	Value pathway	Adds a dedicated value-token embedding pathway injected into the value stream.	Record (KellerJordan/modded-nanogpt contributors, 2024d).
BetterWin	Attention/value geometry	Refines the VTE/windowed trunk with split value embeddings, block sliding windows, and separate block masks.	Record (KellerJordan/modded-nanogpt contributors, 2024e).
SparseV	Value sparsity	Replaces dense per-layer value-token embeddings with a sparse reusable embedding pattern.	Record (KellerJordan/modded-nanogpt contributors, 2024f).
TruncRoPE	Positional shaping	Applies rotary phase information to only a subset of each head dimension.	Record (KellerJordan/modded-nanogpt contributors, 2024f).
SoftCap	Output shaping	Applies bounded logit soft-capping before the loss.	Gemma 2 (Gemma Team, 2024); record (KellerJordan/modded-nanogpt contributors, 2025a).
FP8Head	Output/system path	Moves LM-head computation to an FP8 matrix-multiply path with explicit scaling.	FP8 formats (Micikevicius et al., 2022); record (KellerJordan/modded-nanogpt contributors, 2025b).
LSWA	Attention geometry	Uses paired long and short attention masks assigned across layers.	Gemma 2 (Gemma Team, 2024); record (KellerJordan/modded-nanogpt contributors, 2025c).
AttnScale/ SubLR	Schedule controls	Rescales attention scores or selected parameter learning rates while keeping the late trunk fixed.	Record family (KellerJordan/modded-nanogpt contributors, 2025c).

Table 3: **Cumulative intervention catalog with mechanism groupings.** The 16 named variants form a single incremental chain: each row inherits every prior row's changes and describes only its own delta. The **Group** column encodes the mechanism-level axis used in the spectral taxonomy of Section 4: positional control, optimizer, parameterization, value pathway, depth routing, attention geometry, and output/system path; allowing each transition's spectral signature to be read back against its architectural role. Provenance and contributor attributions follow the modded-NanoGPT record chain.

Spectral measurement protocol. Spectra are extracted from deterministic prefixes of a held-out validation shard and reused across checkpoints and runs within each family. From FixedWin onward in the main d12 trunk, we use $T = 65{,}536$, 512 validation sequences for activation covariance, and 512 validation sequences for per-sample gradient SVD, with covariance batch size=2 and gradient batch size=2; activation covariance therefore uses $N = 512 \times 65{,}536$ token-position rows. Before FixedWin, the short-context prefix uses the equivalent activation budget $32{,}768 \times 1{,}024$ and 512 gradient samples. Short-context targets are formed by shifting within a sequence and masking the last position, whereas the long-context trunk reads length-$(T + 1)$ chunks so that each input token has a true next-token target.

Learning-rate selection. Learning rates are selected separately for each model family and effective batch tier. The reference tier matches the tier-8 modded-NanoGPT token batch: 8 sequences for 65,536-token runs and 512 sequences for the 1,024-token prefix runs, with reference rates taken

Tier	Layers	Heads / width	Parameter range	Context length	Dataset split	Representative runs
d12 prefix	12	6 / 768	123.57M–162.20M	1,024	FineWeb-100B	Baseline, RoPE, Muon, Untied.
d12 trunk	12	6 / 768	162.20M–625.80M	65,536	FineWeb-10B	ValueMix through AttnScale.
d36 support	36	20 / 1280	836.57M–2.00B	32,768	FineWeb-10B	FlexWin, BetterWin, SparseV, AttnScale.
d48 support	48	25 / 1600	1.87B–3.57B	32,768	FineWeb-10B	BetterWin and SparseV scale follow-ups.

Table 4: **Canonical model-tier notation.** The paper names scale primarily by layer count: d12, d36, and d48 denote the 12-, 36-, and 48-layer profiles. Parameter ranges vary because architectural families differ in value-token embeddings, sparse value pathways, and output parameterization.

Family	Representative variants	Shape	Parameters	Training role
Baseline learned-position tied	Baseline	12 layers / 6 heads / width 768	124.35M	Short-context reference.
RoPE tied	RoPE, Muon	12 / 6 / 768	123.57M	Positional and optimizer prefix.
Untied RoPE	Untied	12 / 6 / 768	162.20M	Output-parameterization control.
ValueMix	ValueMix	12 / 6 / 768	162.20M	Cross-layer value mixing.
U-Net/fixed/flex	U-Net, FixedWin, FlexWin	12 / 6 / 768	162.20M	Main local-attention trunk before VTE.
Full VTE	VTE	12 / 6 / 768	625.80M	Dense value-token embedding pathway.
Half VTE / BetterWin	BetterWin	12 / 6 / 768	394.00M	Split/reused value-token embedding pathway.
Sparse family	SparseV, TruncRoPE, SoftCap, FP8Head, LSWA, LSWA-Adam, SubLR	12 / 6 / 768	275.74M	Late sparse value and output-shaping trunk.
Scaled U-Net/flex	FlexWin d36	36 / 20 / 1280	836.57M	Scale robustness for the local-attention trunk.
Scaled BetterWin	BetterWin d36 / d48	36 / 20 / 1280 or 48 / 25 / 1600	2.00B / 3.57B	Largest half-VTE robustness runs.
Scaled sparse family	SparseV d36 / d48, AttnScale d36	36 / 20 / 1280 or 48 / 25 / 1600	1.02B / 1.87B	Larger sparse-trunk support.

Table 5: **Model-card summary with parameter counts.** The 12-layer experiments use GPT-2-small-scale width unless otherwise noted. The d36 and d48 follow-ups use the larger long-context profiles. Scaled sparse-family counts use the runtime-resolved sparse attention pattern used in training. The largest completed robustness run reported in the paper is BetterWin d48 at 3.57B parameters.

from the corresponding modded-NanoGPT configuration. Concretely, the tied AdamW prefix uses 6×10^{-4}; the Muon prefix uses embedding and hidden-matrix rates $(3.6 \times 10^{-3},\ 3.6 \times 10^{-4})$; the untied prefix uses $(0.3,\ 0.002,\ 0.02)$ for embedding, LM-head, and Muon; the long-context trunk uses $(0.6,\ 0.008,\ 0.04,\ 0.04)$ for embedding, LM-head, hidden-matrix, and scalar/control parameters. The LSWA Adam control keeps the same embedding, head, and scalar rates but sets the Adam matrix rate to 0.004.

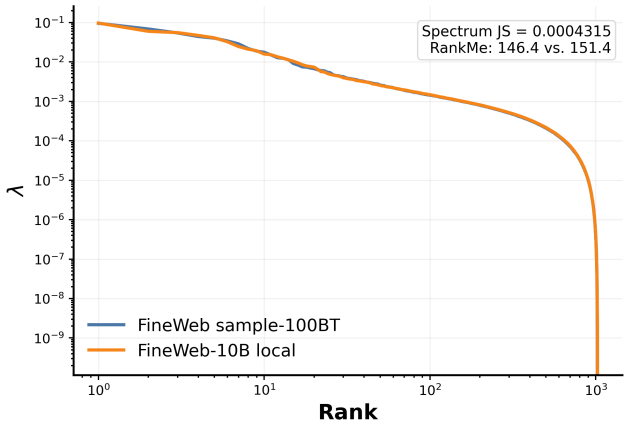


Figure 5: **FineWeb-10B and FineWeb-100B samples have nearly identical token-window spectra.** The plot compares trace-normalized covariance spectra from matched 1,024-token windows. The very small spectral Jensen–Shannon divergence supports treating the data switch as a minor spectral confound relative to the batch and architecture effects analyzed in the paper.

To scale to a target tier B from reference B_0, we form scaling centers $\sqrt{B/B_0}$ and B/B_0 and evaluate each at local multipliers $\{0.5,\ 1/\sqrt{2},\ 1,\ \sqrt{2},\ 2\}$, deduplicating coincident candidates. Each multiplier is applied uniformly across all optimizer groups, preserving relative rates between parameter classes while varying the global step size.

All d12 variants sweep over $\{1, 2, 4, 8, 16, 32\}$; d36 uses $\{2, 4, 8, 16\}$ and d48 uses $\{1, 2, 4, 8\}$, subject to hardware constraints. Candidates are filtered by a synchronized successive-halving procedure (Li et al., 2020): all candidates train to 50M tokens, survivors extend to 150M, and remaining candidates extend to 500M, with test loss evaluated every 20M tokens. Promotion retains the top half at each rung, always keeping at least one survivor per family/tier pair. Power-law extrapolations to 1B tokens are recorded as diagnostics but do not drive promotion, which is based on observed test loss at each rung. Final candidates were selected under the top-half rule throughout; a top-third rule used in early pilots was abandoned before the main sweep. The surviving configuration for each family and batch tier is then used for the final constant-loss spectral runs—a necessary step, since matched-loss spectral comparisons are only meaningful when each tier trains under its own well-tuned schedule.

C Per-Sample Gradient Computation

For a hidden activation matrix $H \in \mathbb{R}^{N\times d}$ collected from a fixed measurement pool, we compute the centered covariance $C = (N-1)^{-1}(H-\bar{H})^\top(H-\bar{H})$ and analyze its eigenvalue spectrum after trace normalization. RankMe is the entropy effective rank,

$$\mathrm{RankMe}(C) = \exp\left(-\sum_j p_j \log p_j\right), \qquad p_j = \lambda_j / \sum_k \lambda_k, \tag{7}$$

where λ_j are the covariance eigenvalues. We also fit band-restricted power-law slopes $\lambda_j \propto j^{-\alpha}$ over head or tail rank windows; these exponents summarize whether variance is concentrated in leading modes or spread through a heavy tail.

For gradient spectra, choose a trainable weight matrix $W \in \mathbb{R}^{d_{\mathrm{out}}\times d_{\mathrm{in}}}$ and compute one loss gradient per sample from a fixed held-out pool of validation sequences, where one *sample* means one validation sequence. Writing $g_m = \mathrm{vec}(\nabla_W \ell_m)$ for the gradient of the sample-mean autoregressive loss on sample m, we stack the rows into

$$G = \begin{bmatrix} g_1^\top \\ \cdots \\ g_M^\top \end{bmatrix} \in \mathbb{R}^{M\times d_{\mathrm{out}} d_{\mathrm{in}}}, \tag{8}$$

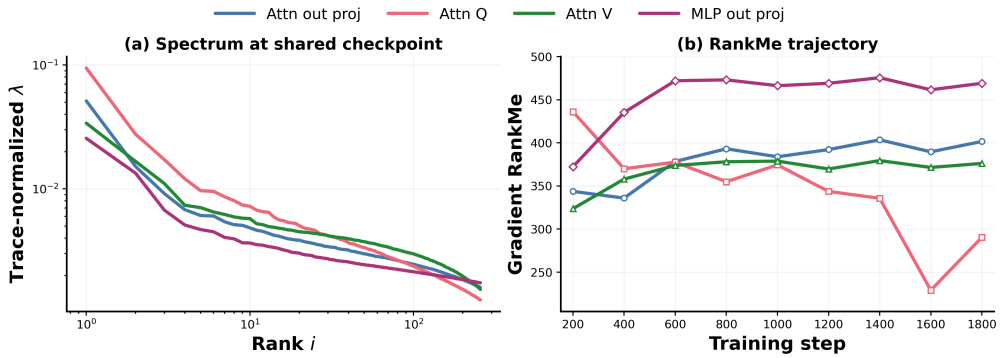


Figure 6: **Gradient spectra depend on the probed weight matrix.** Representative matrix-level comparison from the d12 BetterWin bs8 run at layer 11. The query projection, value projection, attention-output projection, and MLP-output projection produce different concentration levels and RankMe trajectories. This is why gradient spectra are interpreted as tensor-specific complements to activation spectra rather than architecture-level summaries.

and analyze the singular values of the uncentered matrix G. Its second-moment eigenvalues are $\sigma_j(G)^2/M$, so their fitted slope is twice the singular-value slope on the same positive rank window. This matrix is a tensor-specific view of sample-gradient concentration, not a global invariant of the architecture. In the main text we standardize on the deepest saved attention-output projection because it is broadly available and directly measures attention writeback into the residual stream.

Fig. 6 shows the matrix-choice dependence. Each weight matrix exposes a different stage of the attention computation. Query and key gradients reflect how the model is revising its *attention routing*, which positions or feature directions a token should attend to or be attended by, upstream of any content movement. Value gradients reflect how the model adjusts *what content gets aggregated* once routing is fixed: a large gradient here means the read-out per attended position is changing, not the selection pattern itself. Attention-output gradients capture something distinct from both: W_O is the matrix that projects the concatenated multi-head outputs back into the residual stream, making it the unique write gate through which all attended content must pass before influencing downstream computation. Its gradient therefore summarizes the attention layer's net contribution to the residual stream after routing and content selection have both been applied. MLP-output gradients play the analogous role for the feed-forward pathway.

We use W_O of the final attention block as our primary gradient probe for two reasons. First, it is the natural complement to the activation covariance spectrum: activation spectra answer *what has been learned* (the state of the residual stream), while W_O gradients answer *how the attention layer is currently updating that stream*, giving the dual-view its interpretive coherence. Second, W_O is the most stable probe across the intervention chain. Unlike W_V, whose shape and semantics shift when value pathways are restructured (ValueMix, VTE, BetterWin), and unlike W_Q/W_K, whose routing role changes with attention geometry (FixedWin, FlexWin, LSWA), W_O retains the same architectural role, attention writeback to the residual stream, across all 16 variants. This invariance is what allows gradient spectra to serve as a consistent comparative diagnostic along the full intervention chain. As Fig. 6 confirms, the qualitative diagnostic conclusions are robust to this choice; the different probes differ in concentration level and tail shape but not in their implied taxonomic groupings.

D Computational Cost of the Spectral Probes

Because the proposed use is operational regime selection, the cost of the diagnostic relative to the training it is meant to inform is a central quantity. Both probes are computed offline from saved checkpoints and therefore add no training-loop or inference overhead.

On one representative d12 checkpoint, using the same NVIDIA H200 hardware as the training run, a complete activation-plus-gradient spectrum computation for a single layer required 280.93 seconds

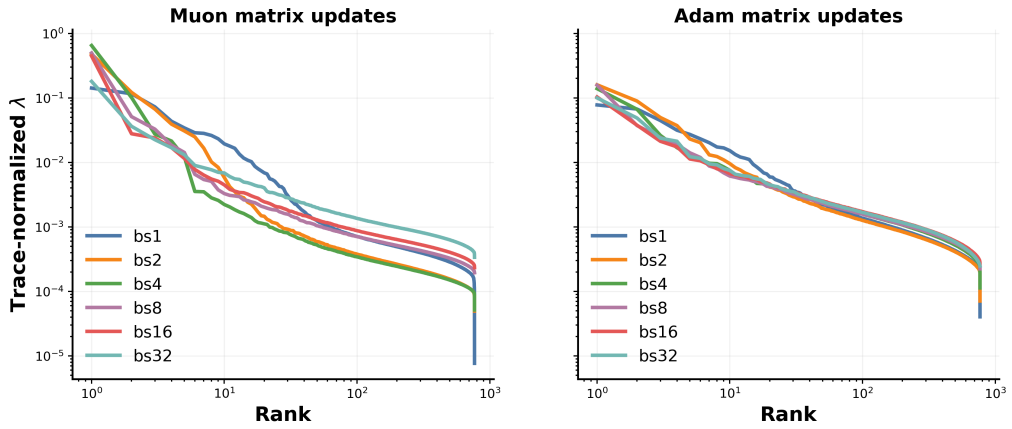


Figure 7: **Batch-dependent activation spectra appear under both Muon and Adam matrix updates.** Each panel shows final layer-11 activation covariance spectra for LSWA across effective batch tiers. The Adam variant still shows batch-dependent spectral separation, so the hidden-regime effect is not intrinsic to Muon. The separation is stronger and more sharply structured in the Muon runs, consistent with Muon changing the geometry of hidden-layer matrix updates.

(0.078 GPU-hours). The corresponding training run required 19.40 hours, so the full joint probe cost 0.402% of training wall time.

The activation-only measurement required for early batch-tier selection is a strict subset of this computation. Per-sample gradient spectra are an optional, higher-cost diagnostic used for the architectural analysis of Section 4; they are not needed for tier ranking. The implementation is also narrower than a full-model spectral decomposition. The activation probe collects hidden states from one fixed layer, forms the $d \times d$ centered covariance of Section 2, and eigendecomposes that single matrix. The gradient probe extracts per-sample gradients for one fixed projection matrix, W_O of the final attention block, and computes its singular spectrum, rather than materializing per-sample gradients for every model parameter.

We report wall-clock cost only. The profiling runs did not separately record peak incremental memory for each stage, so we do not infer a memory figure from the training logs. Until stage-level profiling is available for activation collection, covariance eigendecomposition, per-sample gradient extraction, and gradient SVD, we describe the method as a sparse-checkpoint diagnostic rather than an online training controller.

E MUON/ADAM COMPARISON

Most later variants use a mixed optimizer: Adam-style updates for embeddings, the LM head, and scalar/control parameters, and Muon updates for hidden-layer matrix parameters. The batch-size phenomenon studied in Section 3, point 1 is therefore not automatically a property of Muon alone. To isolate this, we train the LSWA variant under both the standard mixed-Muon setup and an Adam matrix-update variant, keeping all other hyperparameters fixed.

Fig. 7 shows that the Adam control preserves the key qualitative result: equal architecture and different batch tiers still produce measurably different activation spectra. Muon amplifies this separation, particularly in the leading modes, but its existence under Adam confirms that effective batch size is a latent determinant of representation geometry, not merely an artifact of the matrix optimizer choice.

The exact polar update gives one way to interpret the stronger batch dependence. For a matrix gradient $D_W = U_r \Sigma_r V_r^\top$, its compact polar factor $Q(D_W) = U_r V_r^\top$ gives each retained singular direction unit magnitude. This solves the operator-norm linear subproblem in Theorem 30 and makes changes in singular directions relevant to the update. Practical Muon also uses momentum, approximate orthogonalization, and scaling, so the theorem isolates a component of its geometry.

This interpretation requires care near rank changes. Even if a perturbation vanishes, its polar direction need not vanish: $Q(\mathrm{diag}(1,\epsilon)) = I_2$ for every $\epsilon > 0$, whereas $Q(\mathrm{diag}(1,0)) = \mathrm{diag}(1,0)$. Appendix I.4.5 gives a continuity statement with rank and singular-gap conditions. Adam's coordinatewise rescaling can also change singular values and singular directions; it supplies no invariant spectral baseline.

The empirical conclusion is that batch-dependent activation separation occurs under both matrix optimizers, with stronger separation in the displayed Muon control. This motivates calibrating batch tiers within each optimizer recipe. A general comparison of batch sensitivity would require controlling the stochastic gradient distribution and both optimizers' state dynamics.

F Downstream Behavior at Matched Test Loss

Section 3 shows that batch tiers reaching the same test loss can carry visibly different activation spectra. To test whether that hidden geometry corresponds to a difference in capability, we evaluated 30 d12 checkpoints from five families, all matched at test loss 3.2, on seven downstream benchmarks: ARC-Easy, ARC-Challenge, HellaSwag, LAMBADA, OpenBookQA, PIQA, and WinoGrande. Table 6 reports the mean and standard deviation across each family's batch tiers for two representative tasks and for the seven-task aggregate.

Table 6: **Downstream accuracy at matched test loss is flat across batch tiers.** Mean $\pm$ standard deviation (accuracy, %) over the d12 batch tiers of each family, 30 checkpoints in total, all at test loss 3.2. The aggregate averages ARC-Easy, ARC-Challenge, HellaSwag, LAMBADA, OpenBookQA, PIQA, and WinoGrande.

Family	HellaSwag	PIQA	Seven-task aggregate
BetterWin	34.36 ± 0.60	64.72 ± 0.75	38.07 ± 0.56
FixedWin	34.75 ± 0.59	64.39 ± 0.44	37.74 ± 0.22
FlexWin	34.34 ± 0.47	65.83 ± 0.64	37.79 ± 0.40
U-Net	35.47 ± 0.44	66.02 ± 0.96	39.41 ± 0.32
ValueMix	35.14 ± 0.47	66.00 ± 0.26	39.69 ± 0.29

Downstream accuracy is broadly stable across batch tiers once test loss is matched: no family's tier-to-tier standard deviation of the aggregate exceeds 0.6 points, whereas family means span 37.7 to 39.7. The spectral separation between tiers therefore does not signal an unobserved capability difference at the matched checkpoint. We read this as a sharpening of the main claim rather than a weakening of it: the spectra diagnose *how efficiently* a comparable performance level is reached, which is exactly the quantity the early-tail forecast of Section 3 targets. Accordingly, matched-loss spectral differences in this paper are presented as differences in internal geometry and training efficiency, not as established differences in downstream behavior.

G Predictive Ablations

This section tests how robust the early-prediction story is to training progress, random seed, and probe layer. These ablations are intentionally narrower than the main result: they ask whether the diagnostic survives plausible measurement choices rather than claiming that one scalar is universally optimal.

G.1 Spearman correlation over training percentage

Fig. 8 asks when the early-prediction signal becomes visible across the d12 variant chain. Each panel fixes one d12 model family and plots the Spearman correlation between a local activation-spectrum exponent and the final token-efficiency ranking of that family's batch tiers. The three curves correspond to power-law fits over progressively deeper rank windows of the trace-normalized activation covariance spectrum: ranks 10–40 probe the upper modes just past the leading head, ranks 40–90 probe an intermediate band, and ranks 100–200 correspond to the informative tail window used by our main early-prediction analysis (Section 3).

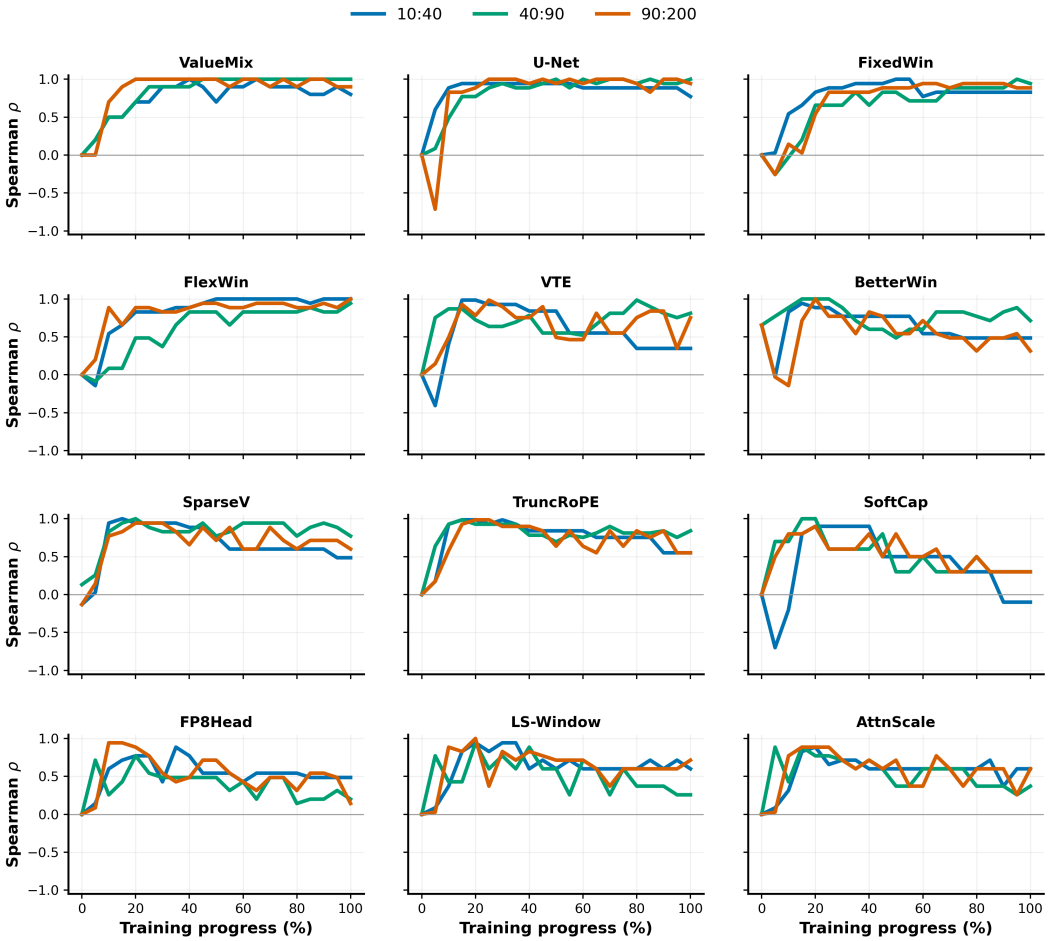


Figure 8: **Early-prediction strength evolves with training progress for d12 variants.** Each small panel shows one d12 model family. The colored lines report Spearman correlation between final token efficiency and local activation-spectrum exponents fit over rank windows 10–40, 40–90, and 90–200. The deeper-tail window 90–200 typically reaches a high positive correlation by about 20% of training, often saturating near $\rho_S = 1$, while shallower windows are more variant-dependent.

A unifying observation across all panels is that the deepest-window correlation *peaks near roughly* 20% *of training progress*, regardless of variant. This holds even for the variants where the signal is otherwise weak or noisy: FP8Head, LSWA, and AttnScale all attain their highest ρ_S near the same early window before drifting downward over the rest of training. The 20% checkpoint is therefore not a feature of "easy" variants—it is a property of the diagnostic itself, and a practical one: once a variant has produced its peak early-tail signal, additional training does not improve and may erode the diagnostic. This is consistent with the mechanistic picture in Section 5, where early tail shape reflects which slow modes are still being recruited; once recruitment completes, that informative window closes. Beyond this shared peak time, the panels separate into three regimes that map cleanly onto the architectural groups in Table 3 and the taxonomy of Section 4.

Clean monotone signal in single-mechanism variants. In ValueMix, U-Net, FixedWin, FlexWin, SparseV, and TruncRoPE, the deepest-window correlation rises smoothly, reaches its maximum around 20% progress, and *holds* near that maximum for the remainder of training. These variants share a structural property: each introduces a single, uniformly-applied change—one new value-mixing rule, one skip pattern, one window size, one sparsity pattern, or one rotary truncation—so the activation tail develops along a single timescale and the early shape stably ranks batch tiers throughout training.

Delayed-onset inversion in multi-component or amplitude-dependent variants. In VTE, BetterWin, and SoftCap, the deepest-window correlation begins strongly negative and climbs to positive values during training, again peaking near 20% progress. Each of these variants introduces a structural mechanism whose effect on the tail emerges with delay rather than from initialization. VTE injects a dedicated value-token embedding pathway that must itself be learned before its spectral signature stabilizes; BetterWin simultaneously layers split value embeddings, block sliding windows, and separate block masks, requiring the three components to differentiate across blocks; and SoftCap's bounded $\tanh$ on logits is amplitude-dependent, behaving nearly linearly until logits enter the saturation regime later in training. In all three cases the early tail does not yet reflect the intervention's eventual geometry, producing transient anti-correlation that resolves as the new component specializes. Theorem 24 gives a controlled example of nonlinear feature-amplitude growth, but does not derive these variant-specific delays. Notably, in several of these panels the shallow 10–40 window reaches a high correlation *before* the deeper window does, consistent with the mode-dependent recruitment in Theorem 16.

Peak-then-decay in execution-dominant variants. FP8Head, LSWA, and AttnScale also peak around 20% progress, but their correlations decay back toward $\rho_S \approx 0.5$ over the rest of training rather than holding. These are precisely the variants classified as execution-dominant in Section 4: FP8Head changes the LM-head's numerical precision rather than its geometry; LSWA assigns different mask types to different layers, so a single-layer probe sees only one mask regime; and AttnScale rescales attention scores or selected learning rates while keeping the late-trunk geometry fixed. None of these interventions reshape the late-trunk feature geometry that the layer-11 probe measures, so the small early-tail signal that does exist is gradually overwritten by execution-side noise as training progresses. Under this view the decay is consistent with, rather than contrary to, our framework: the diagnostic correctly reports low *stable* discrimination when the underlying token-efficiency gap is small, while still recovering the canonical 20%-peak behavior shared with the rest of the chain.

Together, the three regimes turn Fig. 8 from a robustness check into structural validation of the Section 4 taxonomy: every variant shares the same early-prediction time horizon, but only those whose architectural change reshapes late-trunk feature geometry retain a stable diagnostic afterward. The 20%-checkpoint rule for measuring α_{tail} is therefore well-founded across the entire intervention chain, while persistence of the signal beyond that point is itself a useful secondary indicator of whether the intervention is learning-side or execution-side.

G.2 Random seed experiments

Fig. 9 provides a focused robustness check for FlexWin tier-16 configuration, comparing the original seed against two additional seeds. All three panels support the same conclusion. The test-loss curves are essentially indistinguishable throughout training, confirming that the three runs reach the same loss trajectory rather than merely converging at a shared late checkpoint. At the common step-3000 checkpoint, both the activation covariance spectra and the gradient SVD spectra stack so tightly across seeds that the curves are nearly coincident—seed-to-seed variation is negligible relative to the scale of cross-tier separation visible in the main figures. This is the relevant comparison: the batch-tier differences reported throughout the paper are not an artifact of a single lucky or unlucky initialization, but reflect a structural property of the training regime that is stable across independent runs.

G.3 Layerwise ablation

In this ablation we ask whether the final-layer activation probe used in the main early-prediction analysis is a principled measurement choice, or whether an earlier probe layer would serve equally well. For each d36 run, we select the saved checkpoint at 0.21B training tokens and refit the tail exponent over ranks $[200, 400]$, matching the predictive window used in the main figures.

Fig. 10 shows the Spearman correlation between this early tail exponent and tokens-to-target as a function of probe layer, for the three available d36 families. The pattern is consistent: early and middle layers carry weak or sign-inconsistent signal, while the final stored probe layer (L35) shows the strongest positive correlation in all three families. This late-layer advantage is visible directly in the heatmap, correlations tend to become more positive moving from L00 toward L35, and supports

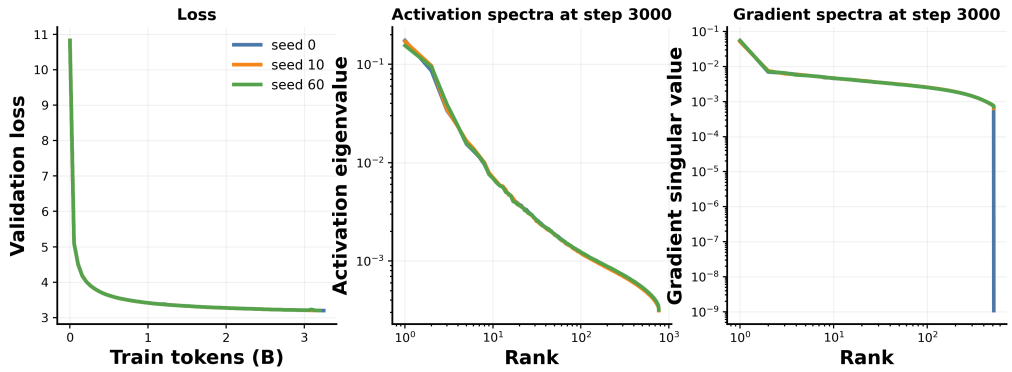


Figure 9: **FlexWin tier-16 spectra are stable across random seeds.** The left panel shows test-loss curves; the middle and right panels show activation and gradient spectra at the shared step-3000 checkpoint. The small seed-to-seed variation supports treating the batch-tier separation as larger than ordinary seed noise in this setting.

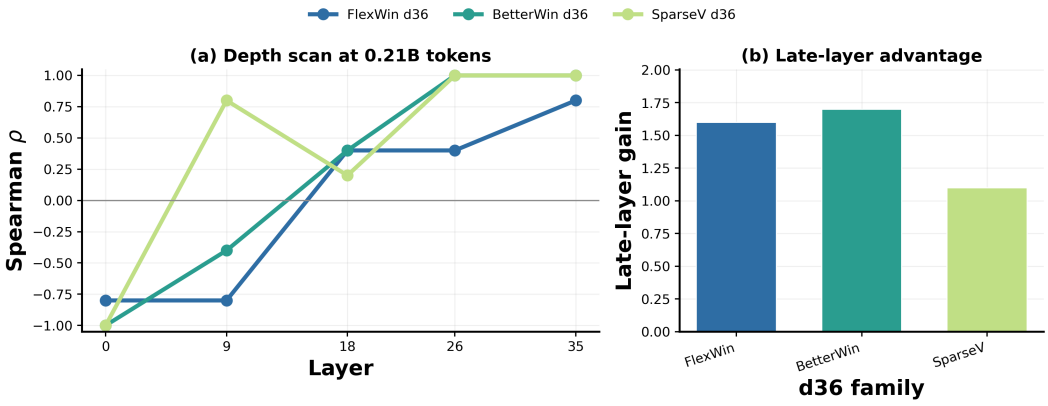


Figure 10: **Late-layer probes carry the clearest early-prediction signal in the d36 support runs.** (a) Spearman correlation between the early activation tail exponent and tokens-to-target proxy across batch tiers, computed at the saved checkpoint closest to 0.25B training tokens. Each line is one d36 family and each x-position is a stored probe layer. The tail exponent is fit over ranks 200–400 of the activation covariance spectrum. (b) Difference between the final-layer correlation and the average correlation of the two earliest stored layers, $\rho(L35) - (\rho(L0) + \rho(L9))/2$. Positive values indicate that the late probe gives a stronger monotonic ordering of batch-tier token cost than early-layer probes.

the convention of probing the deepest available activation layer in the main experiments. We treat this as supporting evidence rather than a universal layer-selection rule: the evidence is limited to three d36 families and does not cover every d12 variant or every architectural regime in the chain.

G.4 Tail-window selection

Power-law decay in neural-network spectra, and the link between spectral heavy tails and scaling behavior, are well established (Mahoney & Martin, 2019; Xie et al., 2023; Bahri et al., 2024; Maloney et al., 2022), but prior work does not prescribe a standard rank interval for fitting activation-covariance tails. We therefore fix the window by a criterion that never consults token-efficiency outcomes.

At the 0.5B-token checkpoint we scan inclusive one-octave windows $I_r = [r, 2r]$ for $r = 20, 30, \ldots, 380$. The upper end of this range is set by the data rather than by choice: I_r is defined only when the $2r$-th eigenvalue exists, so the d12 width of $d = 768$ admits $r \leq 384$, and $r = 380$ is the largest grid point satisfying it. For each candidate we fit $\log \tilde{\lambda}_j$ against $\log j$ over I_r for

every calibration run, and score the window by the family-balanced mean coefficient of determination R^2, so that families contribute equally regardless of how many batch tiers each contains. We then select the single common window maximizing this score and freeze it before any efficiency outcome is examined. Requiring one common window across families, rather than a per-family choice, is what makes the resulting exponent comparable across architectures.

Applied to all d12 calibration spectra, this procedure selects exactly $I^{(d12)} = [100, 200]$, with family-balanced mean $R^2 = 0.99936$. Under a stricter leave-one-family-out protocol, in which the scan runs on the remaining families and the held-out family contributes nothing to the selection, the criterion returns either $[90, 180]$ or $[100, 200]$, recovering essentially the same region without access to the held-out family.

The admissible range of I_r scales with hidden width, so the larger tiers are handled differently and we state the distinction explicitly rather than implying a single scan produced all three windows. At $d = 1280$ the d36 tier admits $r \leq 640$; we verify the bank entry $I^{(d36)} = [200, 400]$ directly rather than re-running the selection, obtaining mean $R^2 = 0.99870$ and within-family Spearman correlation $\rho_S = 0.95$ against token efficiency. At $d = 1600$ the d48 tier admits $r \leq 800$, but the d48 support runs do not provide enough independent families for a family-balanced scan to be meaningful; we therefore take $I^{(d48)} = [400, 800]$ as the next entry and similarly $R^2 = 0.99850$.

Two properties of the estimator are worth stating explicitly. First, trace normalization cannot affect α_{tail}: rescaling all eigenvalues by a common factor adds a constant in log space and leaves the fitted slope unchanged. Second, the probe layer is not tuned per family. We use the deepest stored transformer layer for every architecture, a fixed rule supported by the layerwise scan above rather than selected per run.

G.5 Comparison with simpler early predictors

A high correlation is informative only relative to what simpler measurements already achieve. We therefore compare α_{tail} against four diagnostics that are available at the same checkpoint and require no spectral computation: the early test loss itself, the recent test-loss slope, the gradient norm, and the activation effective rank (RankMe) (Garrido et al., 2023). Each is substituted for α_{tail} as the quantity used to rank batch tiers, and scored by Spearman correlation against realized token efficiency.

We report both a within-family and an across-family column. The within-family column is the primary comparison: it ranks tiers inside a fixed architecture and requires no normalization, since Spearman correlation is invariant to monotone transformations. The across-family column pools the per-family min-max normalized values, and the normalization exists only to place distinct architectures on a common axis for that pooled summary; it plays no role in within-family ranking or in tier selection.

Table 7: **Early activation-tail exponent against simpler diagnostics available at the same checkpoint.** Spearman correlation ρ_S with realized token efficiency at the 0.5B- and 1.0B-token checkpoints. A dash indicates the diagnostic was not evaluated at that checkpoint.

	0.5B tokens		1.0B tokens	
Diagnostic	Within-family	Across-family	Within-family	Across-family
Activation-tail exponent α_{tail}	**0.970**	**0.930**	**0.911**	**0.893**
Early test loss	0.212	0.155	0.624	0.549
Recent test-loss slope	0.367	0.347	0.421	0.234
Gradient norm	−0.011	−0.095	0.463	0.403

At the 0.5B-token checkpoint the separation is large. The activation-tail exponent ranks tiers almost perfectly within family, while early test loss and its slope carry weak signal and the gradient norm and effective rank carry essentially none. This is the regime the diagnostic is designed for: a decision taken after 0.5B tokens is what makes early ranking operationally useful. By 1.0B tokens the loss-based predictors improve substantially, which is expected once runs have begun to separate on the very quantity being predicted, but they remain well below α_{tail}, and the ordering of the two checkpoints shows that the advantage is largest exactly where it matters. Notably α_{tail} itself declines slightly

from 0.5B to 1.0B, consistent with the peak-then-plateau behavior in Fig. 8: the informative window closes as tail recruitment completes.

G.6 Uncertainty and held-out evaluation

Each architecture family contributes only a handful of batch tiers, so the point correlations above should not be read without resampling-based uncertainty. Confidence intervals use family-level bootstrap resampling, taking the family rather than the individual run as the resampling unit, which avoids treating tiers within a family as independent observations. The permutation null is formed by permuting token-efficiency outcomes within families, preserving family structure while destroying the association being tested; we use $10{,}000$ within-family permutations.

Under a leave-one-family-out protocol, in which the fitting window and every other calibrated choice are set using only the remaining families, α_{tail} attains held-out $\rho_S = 0.945$, against a within-family point estimate of 0.970 with 95% family-bootstrap confidence interval $[0.857, 0.971]$. The held-out value falls inside this interval, indicating that the within-family estimate is not an artifact of family-specific calibration.

H The Solvable Model in Full

This appendix develops the model of Section 5 in full. We motivate it with the next-token prediction task (Appendix H.1), define it precisely (Appendix H.2), explain how it arises from training with mini-batch SGD (Appendix H.3), show what the tail exponent measures (Appendix H.4), and state the formal theorem (Appendix H.5). Proofs are in Appendix H.6.

H.1 Motivation: continuing an arithmetic progression

Consider next-token prediction on arithmetic progressions modulo c. Fix an odd integer $c \geq 3$, a context length $L \geq 1$ and a step $d \in \mathbb{Z}_c$, where the vocabulary $\mathbb{Z}_c = \{0, 1, \ldots, c-1\}$ carries addition modulo c. A context is determined by its starting token $a \in \mathbb{Z}_c$:

$$x(a) = \big(a,\ a+d,\ \ldots,\ a+(L-1)d\big) \in \mathbb{Z}_c^L, \qquad y(a) = a + Ld. \tag{9}$$

A model reads $x(a)$ and outputs one score for each token of the vocabulary. We train the scores with the squared loss against a target score vector that may also give partial credit to tokens near the correct one: for a fixed profile $s : \mathbb{Z}_c \to \mathbb{R}$,

$$Y(a)_t = s\big(t - y(a)\big), \qquad t \in \mathbb{Z}_c. \tag{10}$$

The one-hot target is $s = \mathbf{1}_{\{0\}}$. We subtract the mean of s, which shifts all scores equally and does not change the predicted token. Since the context is determined by a, any model is a function of a, and learning the task means regressing the function $Y : \mathbb{Z}_c \to \mathbb{R}^c$. The task has one symmetry: increasing a by one adds one to every token of the context and cyclically shifts the target, $Y(a+1) = \Pi Y(a)$, where Π shifts the vocabulary by one place. It is natural to consider learners with this symmetry built in, which treat all starting tokens alike. For a kernel method this means that the kernel depends only on $a - a'$, and the effective model below assumes exactly this.

H.2 The effective model: kernel gradient flow on $\mathbb{Z}_c$

Let $m = (c-1)/2$ and let a be uniform on $\mathbb{Z}_c$.

Definition 3 (Effective model)**.** *A regime b is a rate profile $\rho_1(b) \geq \cdots \geq \rho_m(b) > 0$. It defines the circulant kernel and the kernel operator*

$$\kappa_b(\delta) = 2\sum_{k=1}^{m} \rho_k(b)\cos(2\pi k\delta/c), \qquad (K_b f)(a) = \frac{1}{c}\sum_{a' \in \mathbb{Z}_c} \kappa_b(a-a')\, f(a'),$$

with K_b acting on each coordinate of a function $f : \mathbb{Z}_c \to \mathbb{R}^c$. The model's scores $f_n : \mathbb{Z}_c \to \mathbb{R}^c$ follow the gradient flow (5) *of the squared loss with $K = K_b$, in continuous time $n \geq 0$. We record the following quantities:*

- *the loss* $\mathcal{L}_b(n) = \frac{1}{2}\mathbb{E}_a\|f_n(a) - Y(a)\|^2$ *and the time to target* $T_\varepsilon(b) = \inf\{n \geq 0 : \mathcal{L}_b(n) \leq \varepsilon\}$*;*
- *the activation covariance* $\Sigma_b(n) = \mathbb{E}_a\left[f_n(a)f_n(a)^\top\right]$ *and its eigenvalues* $\Lambda_1(b,n) \geq \Lambda_2(b,n) \geq \cdots$*, each of multiplicity two;*
- *for a window* $I \subset \{1,\dots,m\}$ *of ranks, the tail exponent* $\widehat{\alpha}_I(b,n)$*, equal to minus the least-squares slope of* $\log \Lambda_k(b,n)$ *against* $\log k$ *over* $k \in I$*.*

The regimes we compare have the same budget $\Gamma = \sum_{k=1}^m \rho_k(b)$*, so their kernels have the same trace* $\kappa_b(0) = 2\Gamma$*. They have power-law profiles*

$$\rho_k(b) = \frac{\Gamma\, k^{-q_b}}{Z_m(q_b)}, \qquad Z_m(q) = \sum_{k=1}^m k^{-q}, \tag{11}$$

with a rate exponent $q_b \geq 0$*.*

The kernel κ_b depends only on $a - a'$, so K_b is diagonalized by the discrete Fourier basis, and the flow (5) splits into independent frequencies. The target is diagonalized by the same basis because of its symmetry: its frequency-k component points along a fixed Fourier vector of the vocabulary, with squared size set by the Fourier transform $\hat{s}(k) = c^{-1}\sum_{\delta\in\mathbb{Z}_c} s(\delta)e^{-2\pi ik\delta/c}$ of the profile.

Its exact solution is Proposition 1: with $|\beta_k|^2 = c\,|\hat{s}(k)|^2$, the eigenvalues of $\Sigma_b(n)$ are $\Lambda_k(b,n) = |\beta_k|^2(1 - e^{-\rho_k(b)n})^2$, each of multiplicity two, and $\mathcal{L}_b(n) = \sum_{k=1}^m |\beta_k|^2 e^{-2\rho_k(b)n}$.

Frequency k is therefore learned at rate $\rho_k(b)$: the fraction of its target energy still missing at time n is $e^{-2\rho_k(b)n}$. The one-hot target has $|\beta_k|^2 = 1/c$ at every frequency, while a profile that decays with distance from the correct token puts more energy at low frequencies. Fig. 4 in the main text plots the exact solution for $c = 1001$, a profile with $|\beta_k|^2 \propto k^{-1}$, budget $\Gamma = 1$ and five rate exponents.

H.3 Where the effective model comes from

A linear model with fixed features. Let the scores be $h_W(a) = W\phi(a) \in \mathbb{R}^c$, with a fixed feature map $\phi : \mathbb{Z}_c \to \mathbb{R}^N$ and a trainable matrix $W \in \mathbb{R}^{c\times N}$. Gradient flow on $\frac{1}{2}\mathbb{E}_a\|h_W(a) - Y(a)\|^2$ from $W = 0$ moves the scores exactly by (5), with kernel $\phi(a)^\top\phi(a')$, provided that this inner product depends only on $a - a'$. The weighted Fourier features

$$\phi(a) = \Big(\sqrt{2\rho_k}\cos(2\pi ka/c),\ \sqrt{2\rho_k}\sin(2\pi ka/c)\Big)_{k=1}^m$$

give exactly the kernel κ_b of Definition 3. An infinitely wide two-layer network in the kernel regime also reduces to (5). For example, random ReLU features of the context, with each token embedded as a point on the unit circle, have a circulant kernel whose rates decay like k^{-4} (Appendix H.6).

Mini-batch SGD and the budget. Now train W by mini-batch SGD with batch size B and step size γ, drawing starting tokens uniformly. The expected iterate follows full-batch gradient descent, which shrinks the frequency-k error by the factor $1 - \gamma\lambda_k$ at each step, where λ_k is the rate of the features' kernel at frequency k. After n training examples, that is n/B steps and nL context tokens, the error has shrunk by $(1 - \gamma\lambda_k)^{n/B} = \exp\{-\rho_k n(1 + O(\gamma\lambda_k))\}$ with

$$\rho_k = \gamma\lambda_k/B. \tag{12}$$

The effective model, with time n counting training examples, is therefore the gradient-flow limit of the mean SGD dynamics. The batch size enters in two places. The first is the per-example rate (12). The second is stability: mini-batch SGD on least squares converges only for step sizes of order $B/(\operatorname{tr} H + (B-1)\|H\|)$, where H is the feature covariance (Jain et al., 2018). Here $\operatorname{tr} H = 2\sum_k \lambda_k$ for mean-zero features. A step size tuned to a fixed fraction of this limit spends a budget $\Gamma = \sum_k \rho_k$ that is independent of B below the critical batch size $B^* = \operatorname{tr} H/\|H\|$ (McCandlish et al., 2018), and that decays like B^*/B above it. This is why we compare regimes at equal budget. The shape of the profile is a separate matter. For plain SGD it is fixed by the features; adaptive optimizers such as Adam and Muon reshape it in a way that can depend on the batch size. We take q_b as given and ask what the spectrum reveals about it.

Activations. In a language model, the unembedding maps the final hidden state to the scores. If this map is a fixed isometry, the covariance of the final hidden state has the same nonzero eigenvalues as the covariance of the scores. We therefore use $\Sigma_b(n)$ as the activation covariance of the effective model.

H.4 What the tail exponent measures

By Proposition 1, a frequency that has been learned ($\rho_k n \gg 1$) contributes its target energy $|\beta_k|^2$ to the spectrum. One that has not ($\rho_k n \ll 1$) contributes $|\beta_k|^2 \rho_k^2 n^2$. The spectrum therefore has a head that follows the target spectrum and a tail that multiplies it by the squared rate profile. The two meet at a knee near $\rho_k n \approx 1$, which moves to deeper frequencies as training proceeds (Fig. 4a). If $|\beta_k|^2 \propto k^{-p}$, the head decays with exponent p and the tail with exponent

$$\alpha_{\mathrm{tail}}(b) = p + 2q_b. \tag{13}$$

The data fix p. The additional $2q_b$ records how quickly learning slows down with depth, and it can be read before the deep frequencies are learned. On a common fitting window the target contributes the same slope to every regime, so differences between regimes isolate $2q_b$ even when the target spectrum is not an exact power law (Proposition 6). The readout expires once the window is learned, because every fitted slope then relaxes toward the common head value. This is why the tail must be read early.

H.5 Flat tails trade early loss for token cost

Under a fixed budget, a flatter profile is not free: it speeds up the deep frequencies by slowing down the leading ones. Which regime looks better therefore depends on when one looks.

Theorem 4 (Early tails rank token cost; early loss inverts it)**.** *Let regimes $b = 1, \dots, K$ share the budget Γ and have rate exponents $0 \le q_1 < \cdots < q_K$ in* (11)*. Let the target energies $|\beta_k|^2$ be positive, nonincreasing in k and not all equal. Then:*

- *(i)* Trade-off. *For $b < b'$ there is a frequency $k_\times \in (1, m)$ such that regime b learns every frequency above $k_\times$ faster than regime b', and every frequency below it more slowly.*
- *(ii)* Early loss favors steep profiles. *There is $\bar{n} > 0$ such that $\mathcal{L}_1(n) > \cdots > \mathcal{L}_K(n)$ for $0 < n < \bar{n}$.*
- *(iii)* Deep targets favor flat profiles. *There is $\bar{\varepsilon} > 0$ such that $T_\varepsilon(1) < \cdots < T_\varepsilon(K)$ for $0 < \varepsilon < \bar{\varepsilon}$.*
- *(iv)* The early tail ranks correctly. *For every window I there is $n_I > 0$ such that $\widehat{\alpha}_I(1, n) < \cdots < \widehat{\alpha}_I(K, n)$ for $0 < n < n_I$.*

Consequently, at any early time $n < \min(\bar{n}, n_I)$ and for any deep target $\varepsilon < \bar{\varepsilon}$, the early tail exponent ranks the regimes by time to target with Spearman correlation $+1$, while early loss ranks them with Spearman correlation -1.

Fig. 4 shows the mechanism. Early loss is dominated by the leading frequencies, which carry most of the target energy and which steep profiles learn first (panel b, before the curves cross). Reaching a deep target requires the tail frequencies, which flat profiles learn first. The tail exponent reads the profile directly, so it identifies the eventual winner while the losses still point the other way (panel c). Section 3 finds the same split in the language models: the early tail exponent ranks batch tiers with $\bar{\rho}_{\mathrm{within}} = 0.970$, early loss with only 0.212.

Two boundary cases mark the scope of the theorem. First, the inversion needs target energy concentrated in a head. If every frequency carries the same energy, as for one-hot targets, the flattest profile has the lowest loss at every n, so early loss ranks the regimes correctly (Proposition 8). Second, suppose the regimes share a shape and differ only in budget, as above the critical batch size. Then the slower regime has both the higher loss and the steeper fitted slope at every n (Remark 9), so both statistics rank correctly. Real batch sweeps can mix the two kinds of change. They push early loss in opposite directions but push the tail exponent in the same direction, so the model predicts a strong tail correlation alongside a weak early-loss correlation.

Scope. Theorem 4 is a statement about the effective model of Definition 3. Its connection to training uses three modelling steps: the gradient-flow limit of the mean SGD dynamics, which omits SGD

fluctuations; a budget motivated by the stability limit; and a rate exponent q_b taken as given rather than derived from the batch size and optimizer. The target must also be deep enough: the larger p, the more of the loss sits in the leading frequencies and the later the loss curves cross.

H.6 PROOFS

This subsection proves the statements of Section 5 and Appendix H. Throughout, $c \geq 3$ is odd, $m = (c-1)/2$, $\omega = e^{2\pi i/c}$, and a is uniform on $\mathbb{Z}_c$. We use two Fourier bases: one for functions of the starting token a, and one for vectors over the vocabulary. For functions of a we use the characters

$$\chi_r(a) = \omega^{ra}, \qquad r \in \mathbb{Z}_c,$$

which are orthonormal, $\mathbb{E}_a\big[\chi_r(a)\overline{\chi_{r'}(a)}\big] = \mathbf{1}_{r=r'}$. For vectors over the vocabulary we use

$$\psi_r = \frac{1}{\sqrt{c}} \sum_{t \in \mathbb{Z}_c} \omega^{-rt} e_t, \qquad r \in \mathbb{Z}_c,$$

where e_t is the standard basis of $\mathbb{R}^c$; these vectors form an orthonormal basis of $\mathbb{C}^c$. The profile s has mean zero, so $\hat{s}(0) = 0$.

Lemma 5 (Solution of the effective model)**.** *Let $\kappa : \mathbb{Z}_c \to \mathbb{R}$ be even, with Fourier coefficients*

$$\rho_r = \frac{1}{c} \sum_{\delta \in \mathbb{Z}_c} \kappa(\delta)\, \omega^{-r\delta}, \qquad r \in \mathbb{Z}_c.$$

For the kernel κ_b of Definition 3, $\rho_0 = 0$ and $\rho_r = \rho_k(b)$ for $r \in \{k, c-k\}$.

(i) The target has the expansion

$$Y(a) = \sum_{r \neq 0} \beta_r\, \psi_r\, \chi_r(a), \qquad \beta_r = \sqrt{c}\, \omega^{rLd}\, \hat{s}(-r),$$

so that $|\beta_r|^2 = c\,|\hat{s}(r)|^2 = |\beta_{c-r}|^2$.

(ii) The solution of (5) *is*

$$f_n(a) = \sum_{r \neq 0} \beta_r \big(1 - e^{-\rho_r n}\big)\, \psi_r\, \chi_r(a).$$

(iii) The scores have mean zero, and their covariance and loss are

$$\Sigma(n) = \sum_{r \neq 0} |\beta_r|^2 \big(1 - e^{-\rho_r n}\big)^2\, \psi_r \psi_r^*, \qquad \mathcal{L}(n) = \frac{1}{2} \sum_{r \neq 0} |\beta_r|^2 e^{-2\rho_r n}.$$

Proof. Part (i). We start by computing the Fourier coefficients of the target in the variable a. Because $Y(a)_t$ depends only on $t - a$, each coefficient turns out to be a multiple of a single Fourier vector over the vocabulary. Expanding each coordinate of Y in characters gives

$$Y(a) = \sum_{r \in \mathbb{Z}_c} v_r\, \chi_r(a), \qquad v_r = \mathbb{E}_a\big[Y(a)\, \overline{\chi_r(a)}\big] \in \mathbb{C}^c.$$

We compute the t-th entry of v_r. By (10), $Y(a)_t = s(t - a - Ld)$, so

$$(v_r)_t = \frac{1}{c} \sum_{a \in \mathbb{Z}_c} s(t - a - Ld)\, \omega^{-ra}.$$

Substitute $\delta = t - a - Ld$, so that $a = t - Ld - \delta$ and $\omega^{-ra} = \omega^{-rt}\omega^{rLd}\omega^{r\delta}$. As a runs over $\mathbb{Z}_c$, so does δ, and therefore

$$(v_r)_t = \omega^{rLd}\, \omega^{-rt}\, \frac{1}{c} \sum_{\delta \in \mathbb{Z}_c} s(\delta)\, \omega^{r\delta} = \omega^{rLd}\, \hat{s}(-r)\, \omega^{-rt}.$$

The entry depends on t only through the factor ω^{-rt}, which is $\sqrt{c}$ times the t-th entry of ψ_r. Hence $v_r = \sqrt{c}\, \omega^{rLd} \hat{s}(-r)\, \psi_r = \beta_r \psi_r$. The term $r = 0$ vanishes because $\hat{s}(0) = 0$. Finally, s is real,

so $\hat{s}(-r) = \overline{\hat{s}(r)}$; and frequencies are taken modulo c, so $\hat{s}(-r) = \hat{s}(c-r)$. Together these give $|\beta_r|^2 = c\,|\hat{s}(r)|^2 = |\beta_{c-r}|^2$.

Part (ii). The point is that each character is an eigenfunction of K, so the flow splits into one scalar linear equation per frequency. First, since κ depends only on $a - a'$, the substitution $\delta = a - a'$ gives

$$(K\chi_r)(a) = \frac{1}{c}\sum_{a' \in \mathbb{Z}_c} \kappa(a-a')\,\omega^{ra'} = \omega^{ra}\,\frac{1}{c}\sum_{\delta \in \mathbb{Z}_c} \kappa(\delta)\,\omega^{-r\delta} = \rho_r\,\chi_r(a).$$

Now expand the solution in characters, $f_n = \sum_r u_r(n)\chi_r$ with $u_r(n) \in \mathbb{C}^c$. The operator K multiplies the coefficient of χ_r by ρ_r, and by part (i) the target's coefficient is $\beta_r\psi_r$. The flow (5) therefore becomes, for each r,

$$\frac{d}{dn}u_r(n) = -\rho_r\big(u_r(n) - \beta_r\psi_r\big), \qquad u_r(0) = 0.$$

This linear equation has the solution

$$u_r(n) = \beta_r\big(1 - e^{-\rho_r n}\big)\psi_r,$$

which is the claim. For $r = 0$ it gives $u_0 \equiv 0$, because $\beta_0 = 0$.

Part (iii). Every quantity here is quadratic in f_n, and orthonormality of the characters turns averages over a into sums over frequencies. The mean of f_n over a is its $r = 0$ coefficient, $u_0(n) = 0$. For the covariance, note that f_n is real, so $f_n f_n^\top = f_n f_n^*$. Expanding both factors in characters and averaging over a, orthonormality removes every cross term:

$$\Sigma(n) = \mathbb{E}_a\big[f_n(a)f_n(a)^*\big] = \sum_{r,r'} u_r(n)\,u_{r'}(n)^*\;\mathbb{E}_a\big[\chi_r(a)\overline{\chi_{r'}(a)}\big] = \sum_r u_r(n)\,u_r(n)^*.$$

Substituting part (ii) gives the formula for $\Sigma(n)$. The same computation applies to $f_n - Y$, whose coefficients are $u_r(n) - \beta_r\psi_r = -\beta_r e^{-\rho_r n}\psi_r$. Since $\|\psi_r\| = 1$, it gives

$$\mathbb{E}_a\|f_n(a) - Y(a)\|^2 = \sum_{r\neq 0}\big\|u_r(n) - \beta_r\psi_r\big\|^2 = \sum_{r\neq 0}|\beta_r|^2 e^{-2\rho_r n},$$

and halving gives $\mathcal{L}(n)$. □

Proof of Proposition 1. The key point is that the frequencies r and $c - r$ have the same coefficient, so together they give one eigenvalue of multiplicity two. By Lemma 5(iii), $\Sigma(n)$ is already written in the orthonormal basis $\{\psi_r\}$, and its eigenvalue on ψ_r is $|\beta_r|^2(1 - e^{-\rho_r n})^2$. Because κ is real and even, its Fourier coefficients are real and satisfy $\rho_r = \rho_{c-r}$. By Lemma 5(i), also $|\beta_r| = |\beta_{c-r}|$. So the terms $r = k$ and $r = c - k$ share the eigenvalue

$$\Lambda_k(n) = |\beta_k|^2\big(1 - e^{-\rho_k n}\big)^2,$$

which therefore has multiplicity two. Its eigenspace is spanned by ψ_k and $\psi_{c-k} = \overline{\psi_k}$, or equivalently by the real orthonormal vectors $\sqrt{2}\,\mathrm{Re}\,\psi_k$ and $\sqrt{2}\,\mathrm{Im}\,\psi_k$. The terms $r = k$ and $r = c - k$ of the loss are also equal, so

$$\mathcal{L}(n) = \frac{1}{2}\sum_{r\neq 0}|\beta_r|^2 e^{-2\rho_r n} = \frac{1}{2}\cdot 2\sum_{k=1}^{m}|\beta_k|^2 e^{-2\rho_k n} = \sum_{k=1}^{m}|\beta_k|^2 e^{-2\rho_k n}.$$

Finally, Λ_k is the product of $|\beta_k|^2$, which is positive and nonincreasing in k by assumption, and $(1 - e^{-\rho_k n})^2$, which is nonincreasing because ρ_k is. So Λ_k is nonincreasing in k, and the frequency index is also the rank of the eigenvalue. □

The linear model. Suppose the scores are $h_W(a) = W\phi(a)$, with fixed features ϕ and a trainable matrix W. Gradient flow on $\frac{1}{2}\mathbb{E}_a\|h_W(a) - Y(a)\|^2$ is

$$\frac{dW}{dt} = -\mathbb{E}_a\big[(h_W(a) - Y(a))\,\phi(a)^\top\big].$$

Multiplying on the right by $\phi(a')$ shows how the scores at a' move:

$$\frac{d}{dt}h_W(a') = -\mathbb{E}_a\big[(h_W(a) - Y(a))\,\phi(a)^\top\phi(a')\big].$$

If $\phi(a)^\top\phi(a') = \kappa(a - a')$, the right side is $-(K(h_W - Y))(a')$, so the scores follow (5) with kernel κ. This kernel is even because the inner product is symmetric. For the weighted Fourier features of Appendix H.3, the identity $\cos x\cos y + \sin x\sin y = \cos(x - y)$ gives $\phi(a)^\top\phi(a') = \kappa_b(a - a')$.

A two-layer network. Embed the token t as $\mathrm{emb}(t) = (\cos(2\pi t/c), \sin(2\pi t/c))$, and let $z(a) \in \mathbb{R}^{2L}$ concatenate the embeddings of the tokens of $x(a)$. Take $\phi_i(a) = N^{-1/2}\sigma(g_i^\top z(a) + b_i)$ for $i = 1, \dots, N$, with σ the ReLU, $g_i \sim \mathcal{N}(0, I_{2L})$ and $b_i \sim \mathcal{N}(0, \sigma_b^2)$, all independent. Training only W is the random-features model, and as $N \to \infty$ its kernel converges to $\mathbb{E}[\sigma(g^\top z(a) + b)\sigma(g^\top z(a') + b)]$. Since $z(a)^\top z(a') = L\cos(2\pi(a - a')/c)$ and $\|z(a)\|^2 = L$, this limit is the ReLU arc-cosine kernel (Cho & Saul, 2009) of the augmented inputs (z, σ_b). It depends only on the angle $\theta = 2\pi(a - a')/c$, through $\cos\psi = (L\cos\theta + \sigma_b^2)/(L + \sigma_b^2)$. The arc-cosine kernel is $\propto \sin\psi + (\pi - \psi)\cos\psi$, which has a $|\psi|^3$ singularity at $\psi = 0$ and is smooth elsewhere, so its Fourier coefficients decay like k^{-4}. Numerically, for $c = 1001$, $L = 4$ and $\sigma_b^2 = 1$, the fitted decay exponent over $10 \le k \le 100$ is 4.01. Without biases, the rates of the odd frequencies $k \ge 3$ vanish.

Mini-batch SGD and the budget. Mini-batch SGD with batch size B and step size γ has an expected iterate that follows full-batch gradient descent. In function space, one step applies $h \mapsto h - \gamma K_\phi(h - Y)$, where K_ϕ has rates λ_r in the sense of Lemma 5. After n examples, that is n/B steps, the frequency-r error has been multiplied by $(1 - \gamma\lambda_r)^{n/B} = \exp\{-\rho_r n(1 + O(\gamma\lambda_r))\}$ with $\rho_r = \gamma\lambda_r/B$. Equation (5) with these rates is its exponential form.

The feature covariance $H = \mathbb{E}_a[\phi\phi^\top]$ has nonzero eigenvalues λ_r, so $\operatorname{tr} H = \mathbb{E}_a\|\phi(a)\|^2 = \kappa(0) = \sum_r \lambda_r$ and $\|H\| = \max_r \lambda_r$. Jain et al. (2018) show that the largest step size for which mini-batch SGD on least squares converges is of order $B/(R^2 + (B-1)\|H\|)$, where R^2 is the smallest constant with $\mathbb{E}[\|\phi\|^2\phi\phi^\top] \preceq R^2 H$. Here $\|\phi(a)\|^2 = \kappa(0)$ for every a, so $R^2 = \operatorname{tr} H$ exactly. For mean-zero features $\lambda_0 = 0$, and

$$\Gamma = \sum_{k=1}^{m} \rho_k = \frac{\gamma}{2B}\operatorname{tr} H \lesssim \frac{\operatorname{tr} H}{2(\operatorname{tr} H + (B-1)\|H\|)}.$$

Consider a step size at a fixed fraction of the stability limit. For $B \ll B^* = \operatorname{tr} H/\|H\|$ the budget is then a constant independent of B, and for $B \gg B^*$ it decays like $B^*/(2B)$. A preconditioner P for which $\phi(a)^\top P\phi(a')$ depends only on $a - a'$ is absorbed by replacing ϕ with $P^{1/2}\phi$. It changes the shape of the rate profile but leaves this bound on the budget unchanged.

The spectrum and the fitted slope. For a positive sequence v and a window $I = \{R_1, \dots, R_2\}$ with $1 \le R_1 < R_2 \le m$, let $\widehat{\alpha}_I[v]$ be minus the least-squares slope of $\log v_k$ against $\log k$ over $k \in I$; thus $\widehat{\alpha}_I(b, n) = \widehat{\alpha}_I[\Lambda(b, n)]$. The function

$$g(\xi) = \frac{\xi}{e^\xi - 1}, \qquad \xi > 0,$$

measures how far a frequency is from being learned. It decreases strictly from $g(0^+) = 1$, for a frequency that has not started to be learned, to 0 as $\xi \to \infty$.

Proposition 6 (Tail exponent of the effective model)**.** *Fix a regime b with profile* (11) *and a time $n > 0$, and write $\xi_k = \rho_k(b)n$ for the progress of frequency k.*

(i) Learned and unlearned frequencies. *If $\xi_k \ge h$ for some $h > 0$, then*

$$(1 - e^{-h})^2\,|\beta_k|^2 \le \Lambda_k(b, n) \le |\beta_k|^2.$$

If $\xi_k \le \epsilon$ for some $\epsilon \le 1$, then

$$(1 - \epsilon/2)^2\,|\beta_k|^2\xi_k^2 \le \Lambda_k(b, n) \le |\beta_k|^2\xi_k^2.$$

(ii) Fitted exponent. *The fitted exponent splits into a target term, common to all regimes, and a learning term:*

$$\widehat{\alpha}_I(b, n) = \widehat{\alpha}_I\big[|\beta|^2\big] + 2q_b\,\bar{g}_b \qquad \text{for some } \bar{g}_b \in \big[g(\xi_{R_1}),\, g(\xi_{R_2})\big].$$

In particular, if $|\beta_k|^2 \propto k^{-p}$, then

$$p + 2q_b\,g(\xi_{R_1}) \le \widehat{\alpha}_I(b, n) \le p + 2q_b\,g(\xi_{R_2}) \le p + 2q_b.$$

Proof. Part (i). By Proposition 1, $\Lambda_k(b,n) = |\beta_k|^2(1-e^{-\xi_k})^2$, so both claims are bounds on the single function $\xi \mapsto 1-e^{-\xi}$, which increases from 0 to 1. If $\xi_k \geq h$, then $1-e^{-h} \leq 1-e^{-\xi_k} \leq 1$; squaring and multiplying by $|\beta_k|^2$ gives the first claim. For small ξ we use the elementary bounds

$$\xi - \frac{\xi^2}{2} \leq 1 - e^{-\xi} \leq \xi \qquad (\xi \geq 0),$$

which follow by integrating $1-u \leq e^{-u} \leq 1$ over $u \in [0,\xi]$. If $\xi_k \leq \epsilon \leq 1$, the lower bound is at least $\xi_k(1-\epsilon/2) \geq 0$. Squaring and multiplying by $|\beta_k|^2$ gives the second claim.

Part (ii). We take logarithms, which turns the product $|\beta_k|^2 \cdot (1-e^{-\xi_k})^2$ into a sum. A least-squares slope is linear in the data, so the fitted exponent splits into the same two pieces. The first piece depends only on the target. For the second we compute the local slope and then use the fact that a least-squares slope is an average of local slopes. Taking logarithms in Proposition 1,

$$\log \Lambda_k(b,n) = \log|\beta_k|^2 + \log\bigl(1-e^{-\xi_k}\bigr)^2,$$

and so

$$\widehat{\alpha}_I(b,n) = \widehat{\alpha}_I\bigl[|\beta|^2\bigr] + \widehat{\alpha}_I\bigl[(1-e^{-\xi})^2\bigr].$$

The first term is the same for every regime, so it remains to bound the second. Treat the rank as a continuous variable and write $u = \log k$. The profile (11) then gives

$$\xi(u) = \frac{\Gamma n}{Z_m(q_b)}\, e^{-q_b u}, \qquad \frac{d\xi}{du} = -q_b\, \xi.$$

The local slope of the learning factor on log–log axes is therefore

$$-\frac{d}{du}\log\bigl(1-e^{-\xi(u)}\bigr)^2 = -\frac{2e^{-\xi}}{1-e^{-\xi}}\,\frac{d\xi}{du} = 2q_b\,\frac{\xi}{e^{\xi}-1} = 2q_b\, g\bigl(\xi(u)\bigr).$$

Next, for points (u_i, y_i) the least-squares slope can be written as a weighted average of secant slopes (see the proof of Proposition 19):

$$\frac{\sum_{i<j}(u_j-u_i)(y_j-y_i)}{\sum_{i<j}(u_j-u_i)^2} = \sum_{i<j} w_{ij}\,\frac{y_j-y_i}{u_j-u_i}, \qquad w_{ij} = \frac{(u_j-u_i)^2}{\sum_{i'<j'}(u_{j'}-u_{i'})^2}.$$

The weights are positive and sum to one. Apply this with $u_i = \log k_i$ and $y_i = \log(1-e^{-\xi_{k_i}})^2$ for $k_i \in I$. By the mean value theorem, minus each secant slope equals the local slope $2q_b\, g(\xi(u))$ at some point of $[\log R_1, \log R_2]$. So the second term, which is minus the fitted slope, is a weighted average of such values. It therefore lies between the smallest and largest values of $2q_b\, g(\xi(u))$ on this interval. Since $\xi(u)$ decreases in u and g decreases in ξ, the function $g(\xi(u))$ increases in u. Its extreme values are thus attained at the ends of the window, namely $2q_b\, g(\xi_{R_1})$ and $2q_b\, g(\xi_{R_2})$. This proves the splitting. For the power law, $\log|\beta_k|^2$ is exactly linear in $\log k$ with slope $-p$, so its fitted exponent is p. The last inequality uses $g \leq 1$. □

Ordering lemma. Each power-law profile spreads the budget according to a probability distribution on the frequencies. For $q \geq 0$ let

$$\pi_q(k) = \frac{k^{-q}}{Z_m(q)}, \qquad F_q(j) = \sum_{k\leq j}\pi_q(k), \qquad k,j \in \{1,\dots,m\},$$

and write $\mathbb{E}_q$ for expectation under π_q. Then $\rho_k(b) = \Gamma\,\pi_{q_b}(k)$. The lemma below makes precise the statement that a larger exponent q concentrates this distribution on low frequencies.

Lemma 7. *Let $0 \leq q < q'$ and $m \geq 2$.*

(i) $F_{q'}(j) > F_q(j)$ *for* $1 \leq j < m$.

(ii) If f is nonincreasing on $\{1,\dots,m\}$ and not constant, then $\mathbb{E}_{q'} f > \mathbb{E}_q f$.

(iii) $\pi_q(1) < \pi_{q'}(1)$ *and* $\pi_q(m) > \pi_{q'}(m)$.

Proof. Part (i). Observe that $\pi_{q'}$ is π_q reweighted by a decreasing function, and that such a reweighting moves mass toward small k. Precisely,

$$\pi_{q'}(k) = \frac{\pi_q(k)\, w(k)}{\mathbb{E}_q w}, \qquad w(k) = k^{-(q'-q)},$$

and the weight w is strictly decreasing. Fix $j < m$ and split the frequencies into $\{k \le j\}$ and $\{k > j\}$. Every weight in the first set is at least $w(j)$, and every weight in the second is at most $w(j+1) < w(j)$, so

$$\mathbb{E}_q[w \mid k \le j] \ge w(j) > w(j+1) \ge \mathbb{E}_q[w \mid k > j].$$

The overall average $\mathbb{E}_q w$ is a combination of these two conditional averages with positive weights $F_q(j)$ and $1 - F_q(j)$, so it lies strictly below the first: $\mathbb{E}_q[w \mid k \le j] > \mathbb{E}_q w$. Therefore

$$F_{q'}(j) = \frac{\mathbb{E}_q\big[w\, \mathbf{1}_{k \le j}\big]}{\mathbb{E}_q w} = F_q(j)\, \frac{\mathbb{E}_q[w \mid k \le j]}{\mathbb{E}_q w} > F_q(j).$$

Part (ii). We rewrite the expectation of f in terms of the distribution function. Part (i) says that $\pi_{q'}$ puts more mass on every initial segment $\{1, \dots, j\}$, which is where f is largest. Writing $f(k) = f(m) + \sum_{j=k}^{m-1} \big(f(j) - f(j+1)\big)$ and exchanging the order of summation gives

$$\mathbb{E}_q f = \sum_{k=1}^{m} f(k)\, \pi_q(k) = f(m) + \sum_{j=1}^{m-1} F_q(j) \big(f(j) - f(j+1)\big).$$

Since f is nonincreasing, every difference $f(j) - f(j+1)$ is nonnegative, and since f is not constant at least one is positive. By part (i), replacing q by q' strictly increases every $F(j)$ with $j < m$. Hence $\mathbb{E}_{q'} f > \mathbb{E}_q f$.

Part (iii). Here we express the masses at the two ends as reciprocals of sums whose terms move monotonically with q. Dividing numerator and denominator by 1^{-q} and by m^{-q}, respectively,

$$\pi_q(1) = \frac{1}{\sum_{k=1}^{m} k^{-q}}, \qquad \pi_q(m) = \frac{1}{\sum_{k=1}^{m} (m/k)^q}.$$

Each term k^{-q} is nonincreasing in q, strictly for $k \ge 2$, so the first denominator strictly decreases and $\pi_q(1)$ strictly increases with q. Each term $(m/k)^q$ is nondecreasing in q, strictly for $k < m$, so the second denominator strictly increases and $\pi_q(m)$ strictly decreases with q. □

Proof of Theorem 4. *Part (i): the trade-off.* Two power laws with the same total must cross exactly once. For $b < b'$, the ratio of the two rate profiles is

$$\frac{\rho_k(b)}{\rho_k(b')} = \frac{\pi_{q_b}(k)}{\pi_{q_{b'}}(k)} = \frac{Z_m(q_{b'})}{Z_m(q_b)}\; k^{\,q_{b'} - q_b}.$$

Viewed as a function of a real variable $k \in [1, m]$, this ratio is continuous and strictly increasing, because $q_{b'} > q_b$. By Lemma 7(iii), it is below one at $k = 1$ and above one at $k = m$: the flatter regime is slower on the first frequency and faster on the last. So it equals one at exactly one point $k_\times \in (1, m)$. Regime b is slower below $k_\times$ and faster above it.

Part (ii): early loss favors steep profiles. The reason is that all losses start at the same value, so at early times they are ordered by their initial slopes. The initial slope is the budget times the average target energy under π_{q_b}, and a steeper profile spends more of its budget where the energy is. By Proposition 1, every regime starts from $\mathcal{L}_b(0) = \sum_k |\beta_k|^2$, with initial slope

$$\mathcal{L}_b'(0) = -2 \sum_{k=1}^{m} |\beta_k|^2 \rho_k(b) = -2\Gamma \sum_{k=1}^{m} |\beta_k|^2\, \pi_{q_b}(k) = -2\Gamma\, \mathbb{E}_{q_b}\big[|\beta|^2\big].$$

The function $k \mapsto |\beta_k|^2$ is nonincreasing and not constant, so by Lemma 7(ii) the average $\mathbb{E}_{q_b}[|\beta|^2]$ strictly increases with b. The initial slopes are therefore strictly ordered:

$$\mathcal{L}_1'(0) > \mathcal{L}_2'(0) > \cdots > \mathcal{L}_K'(0).$$

The curves start at the same point and are smooth, so a strictly steeper initial descent means a strictly smaller loss for a short time. There are finitely many regimes, so there is a common $\bar{n} > 0$ with $\mathcal{L}_1(n) > \cdots > \mathcal{L}_K(n)$ for $0 < n < \bar{n}$.

Part (iii): deep targets favor flat profiles. What drives this is that at late times each loss is dominated by its slowest frequency, the last one, and the flatter regime learns that frequency faster. So its loss eventually falls below that of the steeper regime, which orders the times to reach small targets. Fix $b < b'$ and write $S = \sum_k |\beta_k|^2$. The rates decrease in k, so every term of $\mathcal{L}_b(n)$ is at most $|\beta_k|^2 e^{-2\rho_m(b)n}$. Keeping only the last term of $\mathcal{L}_{b'}(n)$ gives a lower bound for it. Together,

$$\mathcal{L}_b(n) \le S\, e^{-2\rho_m(b)n}, \qquad \mathcal{L}_{b'}(n) \ge |\beta_m|^2 e^{-2\rho_m(b')n}.$$

By Lemma 7(iii), $\rho_m(b) > \rho_m(b')$. Dividing the two bounds,

$$\frac{\mathcal{L}_b(n)}{\mathcal{L}_{b'}(n)} \le \frac{S}{|\beta_m|^2}\, e^{-2(\rho_m(b)-\rho_m(b'))n} \longrightarrow 0 \qquad (n \to \infty),$$

so there is a time $N_{bb'}$ with $\mathcal{L}_b(n) < \mathcal{L}_{b'}(n)$ for all $n \ge N_{bb'}$. To translate this into times to target, note that each loss decreases continuously and strictly from $\mathcal{L}_b(0)$ to zero. So $T_\varepsilon(b)$ is the unique time at which $\mathcal{L}_b$ equals ε. Take $\varepsilon < \mathcal{L}_{b'}(N_{bb'})$. Then regime b' reaches ε only after time $N_{bb'}$, and at that moment regime b is already below ε:

$$\mathcal{L}_b\big(T_\varepsilon(b')\big) < \mathcal{L}_{b'}\big(T_\varepsilon(b')\big) = \varepsilon.$$

Hence $T_\varepsilon(b) < T_\varepsilon(b')$. Taking $\bar{\varepsilon}$ to be the smallest of the values $\mathcal{L}_{b'}(N_{bb'})$ over the finitely many pairs $b < b'$ proves part (iii).

Part (iv): the early tail ranks correctly. Recall that Proposition 6 writes each fitted exponent as the common target term plus $2q_b$ times a factor in $(0, 1]$. At early times the window has not been learned, so this factor is close to one, and the fitted exponents are ordered like the q_b. For regime b, the bound $\bar{g}_b \le 1$ gives

$$\widehat{\alpha}_I(b, n) \le \widehat{\alpha}_I\big[|\beta|^2\big] + 2q_b,$$

and for regime $b+1$, the bound $\bar{g}_{b+1} \ge g(\xi_{R_1})$ gives

$$\widehat{\alpha}_I(b+1, n) \ge \widehat{\alpha}_I\big[|\beta|^2\big] + 2q_{b+1}\, g\big(\rho_{R_1}(b+1)\, n\big).$$

Since $g(\xi) \to 1$ as $\xi \to 0$ and $q_b/q_{b+1} < 1$, we can choose $n_I > 0$ so small that

$$g\big(\rho_{R_1}(b+1)\, n\big) > \frac{q_b}{q_{b+1}} \qquad \text{for all } b < K \text{ and } 0 < n < n_I.$$

For such n, the lower bound for regime $b+1$ exceeds the upper bound for regime b, so $\widehat{\alpha}_I(b, n) < \widehat{\alpha}_I(b+1, n)$.

The consequence. Take an early time $n < \min(\bar{n}, n_I)$ and a deep target $\varepsilon < \bar{\varepsilon}$. By parts (iii) and (iv), the time to target and the early tail exponent are both strictly increasing in b. They rank the regimes identically, so their Spearman correlation is $+1$. By part (ii), early loss is strictly decreasing in b, so it ranks the regimes in exactly the reverse order, and its Spearman correlation with the time to target is -1. □

Proposition 8 (Flat target energy)**.** *If $|\beta_k|^2$ is constant in k and $0 \le q < q'$, then the regime with exponent q has strictly smaller loss at every $n > 0$. It therefore reaches every target first, and early loss ranks the regimes correctly.*

Proof. With equal energies, the loss is a sum of one convex function evaluated at the rates. Such a sum is smallest when a fixed budget is spread evenly, and a steeper profile spreads it less evenly. When every frequency carries the same energy β^2, Proposition 1 gives

$$\mathcal{L}(n) = \beta^2 \sum_{k=1}^{m} \varphi(\rho_k), \qquad \varphi(x) = e^{-2nx},$$

and φ is strictly convex for $n > 0$. "Less evenly" has a precise meaning here, majorization. Both rate vectors are sorted in decreasing order and sum to Γ, and by Lemma 7(i) their partial sums satisfy

$$\sum_{k \le j} \rho_k(q') = \Gamma F_{q'}(j) > \Gamma F_q(j) = \sum_{k \le j} \rho_k(q) \qquad (1 \le j < m).$$

So $\rho(q')$ strictly majorizes $\rho(q)$, and Karamata's inequality for the strictly convex φ gives

$$\sum_{k=1}^{m} e^{-2n\rho_k(q')} > \sum_{k=1}^{m} e^{-2n\rho_k(q)}.$$

Thus the steeper regime has strictly larger loss at every $n > 0$. The flatter regime therefore reaches every target first, and early loss already ranks the regimes correctly. □

Remark 9 (Budget changes at a fixed shape)**.** *Fix a shape $q > 0$ and compare two budgets $\Gamma < \Gamma'$ at the same time $n > 0$. Scaling the budget scales every rate by $\Gamma'/\Gamma > 1$, so every frequency is learned faster and the loss is strictly smaller under Γ'. For the tail exponent, return to the proof of Proposition 6(ii). The fitted exponent is a weighted average of the local slope $2q\, g(\xi(u))$, with weights that depend only on the window and not on Γ. Raising the budget raises $\xi(u)$ at every u, and since g is strictly decreasing, it lowers every local slope. So the fitted exponent is strictly smaller under Γ'. The two statistics therefore agree: the regime with the larger budget has the lower early loss and the flatter early tail, and both rank it first.*

I Toy Model

The toy analysis connects the empirical diagnostics to quantities that can be defined and checked directly. The linear Fourier model isolates information about modal progress that scalar loss can hide. The factor model identifies conditions under which hidden-feature energy moves toward a fixed task support. The modular-arithmetic transformer supplies measured spectra and Fourier profiles, with its own training objective and probe distribution.

I.1 Linearized and diagonal Fourier models

The linear model evolves coefficients in a fixed Fourier basis and observes their energies through an explicit orthogonal map. Its power-law specialization relates a finite-window slope to the relative rates at which modes are learned. The diagonal two-layer model instead learns hidden-feature amplitudes, with a positive balanced initialization that makes the energy calculation exact. These are complementary constructions, not successive approximations in a proof of full transformer dynamics. Their definitions and proofs are given in Appendix I.4.

I.2 Optimization and parameterization interventions

The empirical chain adds RoPE, Muon, and an untied readout to the baseline. Appendix I.4.5 analyzes a component property of each: relative-position dependence of attention scores, the geometry of an exact polar step, and the output-subspace constraint of a fixed-embedding linear readout. These properties motivate the joint spectral measurements in Section 4. The cumulative recipes can also differ in optimization details, so a measured transition does not isolate the effect of its named modification.

I.3 Two-layer transformer and feature probes

The transformer toy has two layers, four heads, hidden width 128, context length 64, vocabulary size 1,024, and a training dataset of 32,000 sequences. We use last-token representations and the cumulative chain Baseline→RoPE→Muon→Untied. The batch labels $B \in \{32, 64, 128, 256, 512\}$ count sequences per optimizer step, so an update consumes $64B$ tokens. These labels differ from the token-batch tiers used for the language models in Section 2.2.

Endpoints and progress. The selected Muon sweep in Fig. 11(a) and the replays in Fig. 11(b)–(c) use a validation cross-entropy target of 2.5. The Muon sweep combines local-grid learning-rate selections for $B = 32, 64, 128, 256$ with an earlier successive-halving run at $B = 512$, whose saved trajectory is evaluated at the 2.5 target. Thus these plots compare selected training recipes rather than evaluate a complete prospective selection procedure.

Panel (a) computes the ordinary least-squares slope magnitude on ranks $10, \ldots, 39$ of the trace-normalized activation covariance spectrum. The implementation uses an exclusive upper endpoint

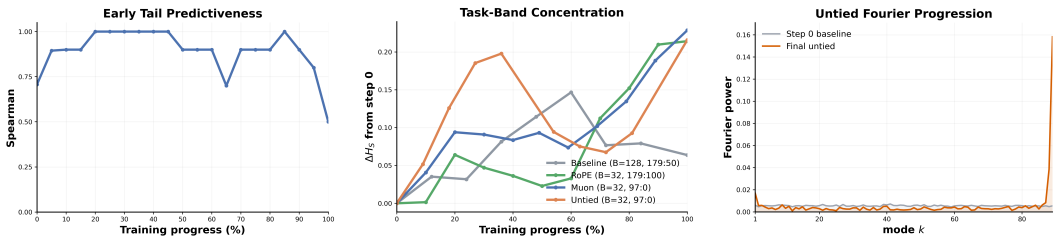


Figure 11: **Toy simulations connect spectral diagnostics to Fourier structure.** (a) In the Muon batch sweep, the activation slope on ranks 10:39 has absolute Spearman correlation $|\rho_S| = 1$ with eventual token cost around 20% progress; the signed correlation is negative. (b) Selected replays track the change in Fourier peak concentration, ΔH_{peak}, across the baseline, RoPE, Muon, and untied stages. The legend gives batch size and probe cycle length/offset. (c) The same untied run has a diffuse initial Fourier profile and a final profile concentrated on a few modes. Progress in (a)–(b) is normalized by each run's final step. Probe definitions and configurations are given in Appendix I.3.

of 40. At each displayed fraction u, it uses the saved spectral checkpoint nearest u times that run's final step and correlates the slope with total tokens to target across the five batch sizes. The plotted value is $|\rho_S|$. At $u = 0.20$, the signed correlation is -1, with selected steps $500, 500, 250, 250, 250$ for increasing batch size. The association therefore has the opposite sign from the LLM result in Section 3. It shows that the local spectrum contains ranking information in this toy sweep, without testing the common teacher, anchor, or unresolved-window assumptions of Corollary 18.

Panel (b) uses checkpoint progress $100\, s/s_{\text{final}}$, rounded to the nearest integer percentage. In both panels, the normalization uses the eventual endpoint, so progress is retrospective. The fixed-token forecasting protocol of Section 3 does not use this normalization.

The implemented Fourier statistic. A probe configuration (c, o) specifies a cycle length and a vocabulary offset, not an interval of Fourier ranks. Clean probes generate a single modular component with that cycle and offset. The matched-distribution probe retains the training generator's mixture and nuisance settings while forcing an anchor component with the specified (c, o). Its phase labels refer to that component.

For a fixed checkpoint and probe, let $h_i \in \mathbb{R}^{128}$ be the representation at the last input token and $z_i \in \mathbb{Z}_c$ its phase label. Form the conditional sample means and center them uniformly over phase bins:

$$n_z = \#\{i : z_i = z\}, \qquad \bar{h}_z = \frac{\sum_{i:z_i=z} h_i}{\max\{n_z, 1\}}, \qquad H_z = \bar{h}_z - \frac{1}{c}\sum_{a=0}^{c-1} \bar{h}_a.$$

Empty bins are assigned zero before centering. Fourier transformation acts along the phase axis, after this conditional averaging:

$$F_k = \sum_{z=0}^{c-1} H_z e^{-2\pi i k z/c}, \qquad E_k = \|F_k\|_2^2, \qquad P_k = \frac{E_k}{\sum_{j=0}^{c-1} E_j}. \tag{14}$$

The stored profile is set to zero when total energy is numerically zero. Otherwise the denominator includes all Fourier frequencies. The positive-frequency statistic is

$$H_{\text{peak}} = \max_{1 \le k \le \lfloor c/2 \rfloor} P_k, \qquad \Delta H_{\text{peak}}(s) = H_{\text{peak}}(s) - H_{\text{peak}}(0). \tag{15}$$

Conjugate pairs are not summed and the positive half is not renormalized. For the real-valued profiles and odd cycles used here, the positive-frequency mass is one half, up to numerical error. This is the Fourier energy of conditional mean representations, not the mean Fourier energy of individual sequences.

Panel (b) plots (15) for the configurations in Table 8. Because the batch and probe choices differ, the curves describe within-run changes rather than a controlled ordering of concentration rates across recipes. The maximizing frequency can also change between checkpoints. In contrast, the theoretical

Table 8: **Selected replays in Fig. 11(b).** The endpoint is the saved final step at the validation target of 2.5. The probe descriptor is cycle length and vocabulary offset (c, o).

Stage	B	(c, o)	Probe construction	Final step
Baseline	128	$(179, 50)$	Matched-distribution	4,780
RoPE	32	$(179, 100)$	Clean	10,870
Muon	32	$(97, 0)$	Clean	2,550
Untied	32	$(97, 0)$	Clean	1,270

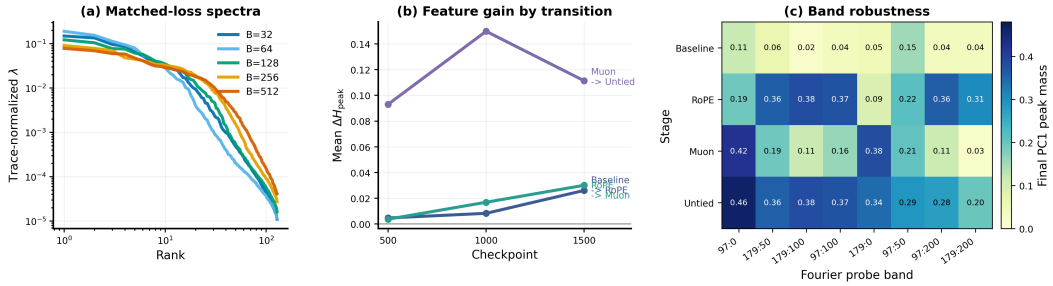


Figure 12: **Additional modular-arithmetic spectra and Fourier probes.** (a) Untied activation spectra across sequence batch sizes $B \in \{32, 64, 128, 256, 512\}$. Saved checkpoints selected near a nominal loss target span losses 1.99–3.91, so they are not an exact matched-loss comparison. (b) Mean changes in Fourier peak concentration between consecutive recipes at three optimizer steps, using probe $(c, o) = (97, 200)$ and the available pairs described in the text. (c) Final PC1 peak mass across eight clean cycle/offset configurations at $B = 32$. The relative ordering depends on the probe. These profiles describe Fourier concentration without identifying a fixed task support.

H_S sums hidden energy over a support fixed by the teacher; the peak statistic does not test its monotonicity.

Panel (c) uses the same Untied run listed in the table at steps 0 and 1,270, but a separate clean probe with $(c, o) = (179, 200)$, 2,048 sequences, and probe seed 0. It displays P_k for $1 \le k \le 89$. The final profile is more peaked than the initial profile. Establishing that the peaks are useful for prediction would require a task-support argument or an intervention on the represented modes; the profile alone measures Fourier concentration.

Additional batch and probe comparisons. Fig. 12(a) retains the Untied batch-sweep spectra. The stored selection uses a nominal test-loss target of 3.796612 and chooses available checkpoints. The selected losses are 3.686, 3.907, 3.157, 2.505, and 1.990 for increasing B. Because checkpoint spacing leaves substantial differences, this panel documents batch-dependent spectra but is not an exact matched-loss control.

Panel (b) uses the matched-distribution probe $(c, o) = (97, 200)$ and shows the mean difference in H_{peak} between consecutive recipes at steps 500, 1,000, and 1,500. These are differences between recipes at a checkpoint, unlike the initialization-subtracted changes in Fig. 11(b). The available pair counts are $(3, 3, 3)$ for Baseline→RoPE, $(3, 3, 2)$ for RoPE→Muon, and $(2, 1, 1)$ for Muon→Untied. The last transition has the largest reported peak-concentration change at each displayed checkpoint, but the differing pair counts and cumulative recipes limit a causal interpretation.

Panel (c) compares eight clean probes at $B = 32$. If $H = U\Sigma V^\top$ is the singular value decomposition of the centered phase profile, PC1 peak mass is the largest normalized positive-frequency power of the first left singular vector $U_{:,1}$, using the full-frequency denominator as in (14). Scaling by the first singular value leaves this normalized statistic unchanged. The metric concerns the phase structure of one principal component, whereas H_{peak} pools energy across all hidden coordinates. The recipe ordering varies across probe configurations; these descriptors change the cycle and offset, not the width of a Fourier support. The displayed configuration order follows the stored probe ranking and carries no independent frequency ordering.

I.4 Formal toy-model theory and proofs

The theory addresses the two empirical uses of spectra: ranking learning regimes within a family and distinguishing changes in representation from changes in gradient geometry. We first construct a linear model in which loss can hide differences in modal progress. A power-law specialization connects those differences to a fitted spectral slope. We then derive the relation between layer inputs and actual sample gradients, before introducing trainable feature amplitudes and the three intervention results.

The linear and factor models use squared loss and have different observation maps and initializations. Their time variable is deterministic gradient-flow time. The regime label b specifies a learning-rate profile; its dependence on batch size is an assumption of the model, not a consequence derived from stochastic training. The transformer measurements in Appendix I.3 provide empirical comparisons with these constructions.

I.4.1 Linear Fourier model and information hidden by loss

Fix the cyclic task of Section 5, with phase $z \in \mathbb{Z}_c$, step d, offset o, and context length L. On the space $\mathcal{H}$ of complex functions on $\mathbb{Z}_c$, use uniform phase measure and

$$\langle f, g\rangle = \frac{1}{c}\sum_{z\in\mathbb{Z}_c} f(z)\overline{g(z)}, \qquad (\mathcal{S}_q f)(z) = f(z+q \bmod c), \qquad \chi_r(z) = e^{2\pi i r z/c}.$$

The characters, indexed by $0 \le r < c$, form a deterministic orthonormal basis. The scalar token label $y(z)$ motivates these coordinates; the squared-loss teacher introduced below is specified separately.

Lemma 10 (The clean label is a shifted phase readout)**.** *Let $u_0(z) = o + z$ for representatives $z \in \{0, \dots, c-1\}$. Then $y = \mathcal{S}_{Ld} u_0$ and*

$$\mathcal{S}_{Ld}\chi_r = e^{2\pi i r L d/c}\chi_r.$$

Proof. Substitution gives $(\mathcal{S}_{Ld}u_0)(z) = o + ((z + Ld) \bmod c) = y(z)$. Periodicity gives $\chi_r(z + q \bmod c) = e^{2\pi i r q/c}\chi_r(z)$, including $d = 0$. The finite geometric sum also gives $c^{-1}\sum_{z=0}^{c-1}\chi_r(z)\overline{\chi_s(z)} = \mathbf{1}_{r=s}$, proving the stated orthonormality. The label identity concerns this particular readout; an arbitrary learned predictor need not be a shift operator. □

Dynamics and observation map. Let $\mathcal{K} : \mathcal{H} \to \mathcal{H}$ be time independent, self-adjoint, positive semidefinite, and commuting with every $\mathcal{S}_q$. For a fixed teacher $f^\star = \sum_r \beta_r\chi_r$ and predictor $f_t = \sum_r a_r(t)\chi_r$, consider

$$\partial_t f_t = \mathcal{K}(f^\star - f_t), \qquad \mathcal{L}(t) = \tfrac{1}{2}\|f_t - f^\star\|^2_{\mathcal{H}}. \tag{16}$$

To observe the coefficients separately, choose fixed orthonormal vectors $e_0, \dots, e_{c-1}$ in $\mathbb{C}^{D_{\mathrm{obs}}}$, where $D_{\mathrm{obs}} \ge c$, and define

$$h^{\mathrm{lin}}_t(z) = \sum_r a_r(t) e_r \chi_r(z), \qquad v^{\mathrm{upd}}_t(z) = \sum_r \kappa_r(\beta_r - a_r(t)) e_r \chi_r(z). \tag{17}$$

Here κ_r are the eigenvalues identified below. The first field is an imposed observation of learned coefficients; the second is an observation of their rates of change. Neither field is assumed to equal a transformer hidden state or a per-sample parameter gradient. Write $M_h = \mathbb{E}[h^{\mathrm{lin}}(h^{\mathrm{lin}})^*]$ and $M_{\mathrm{upd}} = \mathbb{E}[v^{\mathrm{upd}}(v^{\mathrm{upd}})^*]$, where $*$ denotes conjugate transpose.

Theorem 11 (Modewise dynamics and second moments)**.** *Under the preceding assumptions, $\mathcal{K}$ is a convolution operator and $\mathcal{K}\chi_r = \kappa_r\chi_r$ with $\kappa_r \ge 0$. Equation* (16) *has the unique global solution*

$$\dot a_r = \kappa_r(\beta_r - a_r), \qquad a_r(t) = \beta_r + (a_r(0) - \beta_r)e^{-\kappa_r t}.$$

Its loss and the moments of the fields in (17) *are*

$$\mathcal{L}(t) = \tfrac{1}{2}\sum_{r=0}^{c-1}|a_r(t) - \beta_r|^2,$$

$$M_h(t) = \sum_{r=0}^{c-1}|a_r(t)|^2 e_r e_r^*, \qquad M_{\mathrm{upd}}(t) = \sum_{r=0}^{c-1}\kappa_r^2|\beta_r - a_r(t)|^2 e_r e_r^*.$$

Proof. Let $K(z, z')$ be the matrix of $\mathcal{K}$ in the point-evaluation basis. Commutation with shifts gives $K(z+s, z'+s) = K(z, z')$, so $K(z, z') = k(z - z')$ for $k(\delta) = K(\delta, 0)$. With the convention $(\mathcal{K}f)(z) = \sum_{z'} k(z - z')f(z')$,

$$(\mathcal{K}\chi_r)(z) = \chi_r(z) \sum_{\delta \in \mathbb{Z}_c} k(\delta) e^{-2\pi i r\delta/c}.$$

Thus each character is an eigenvector. Self-adjointness makes its eigenvalue real, and positive semidefiniteness gives $\kappa_r = \langle \mathcal{K}\chi_r, \chi_r \rangle \geq 0$. Expanding (16) in this basis gives the scalar differential equations. The exponential solution is unique for each mode and defined for all $t \geq 0$, including $\kappa_r = 0$. Parseval's identity gives the loss.

Character orthogonality gives

$$\mathbb{E}[h_t^{\mathrm{lin}}(h_t^{\mathrm{lin}})^*] = \sum_{r,s} a_r \overline{a_s} e_r e_s^* \mathbb{E}[\chi_r \overline{\chi_s}] = \sum_r |a_r|^2 e_r e_r^*.$$

Replacing a_r by $\kappa_r(\beta_r - a_r)$ proves the second moment formula. Orthogonality of the e_r identifies the displayed coefficients as eigenvalues, with additional zeros if $D_{\mathrm{obs}} > c$. □

Proposition 12 (Centering removes the constant mode)**.** *For the fields in* (17)*,*

$$\mathbb{E}h_t^{\mathrm{lin}} = a_0(t)e_0, \qquad \mathbb{E}v_t^{\mathrm{upd}} = \kappa_0(\beta_0 - a_0(t))e_0.$$

Their centered covariances are

$$\Sigma_h(t) = \sum_{r=1}^{c-1} |a_r(t)|^2 e_r e_r^*, \qquad \Sigma_{\mathrm{upd}}(t) = \sum_{r=1}^{c-1} \kappa_r^2 |\beta_r - a_r(t)|^2 e_r e_r^*.$$

Proof. Uniform phase averaging gives $\mathbb{E}\chi_r = \mathbf{1}_{r=0}$. The outer products of the displayed means are exactly the $r = 0$ terms of the corresponding second moments. Subtracting them gives the covariances. □

For the rest of the linear-model analysis, take an active set of m nonconstant modes, $1 \leq m \leq c-1$, with positive rates, and initialize all coefficients at zero. The teacher vanishes outside this set. Relabel its modes $1, \ldots, m$ in a fixed order. Then $a_r(t) = \beta_r(1 - e^{-\kappa_r t})$ and the activation energies are $|a_r(t)|^2$. Mode labels need not be sorted eigenvalue ranks; the power-law family below establishes when the two agree.

Proposition 13 (Matched loss does not identify spectral state)**.** *Take $m \geq 2$ and $\beta_r = 1$ on the active set. Regime* A *has rate $\kappa > 0$ in every mode, while regime* B *has $\kappa_1^{\mathrm{B}} > \cdots > \kappa_m^{\mathrm{B}} > 0$. For every $\ell \in (0, m/2)$ there are unique times $t_{\mathrm{A}}, t_{\mathrm{B}} > 0$ with loss ℓ. Nevertheless, the leading-mode fractions*

$$P_j(t) = \frac{|a_1^j(t)|^2}{\sum_{r=1}^m |a_r^j(t)|^2}$$

satisfy $P_{\mathrm{A}}(t_{\mathrm{A}}) = 1/m$ and $P_{\mathrm{B}}(t_{\mathrm{B}}) > 1/m$.

Proof. For either regime,

$$\mathcal{L}_j(t) = \tfrac{1}{2} \sum_r e^{-2\kappa_r^j t}, \qquad \mathcal{L}_j'(t) = -\sum_r \kappa_r^j e^{-2\kappa_r^j t} < 0.$$

The loss decreases continuously from $m/2$ to zero, giving the unique crossing of each stated level. At any positive finite time, regime A has equal positive amplitudes. In regime B, $a_1^{\mathrm{B}} > a_r^{\mathrm{B}} > 0$ for every $r > 1$, so $\sum_r |a_r^{\mathrm{B}}|^2 < m|a_1^{\mathrm{B}}|^2$. The two fractions follow. □

The preceding comparison aligns runs at their loss crossings. The next result uses a common observation time and shows that equal early loss can also hide different future efficiency.

Proposition 14 (Equal loss at a common time can conceal different future loss)**.** *Take $m \geq 2$, unit teacher coefficients, and a regime with positive rates $\kappa_1, \ldots, \kappa_m$ that are not all equal. Fix $t_0 > 0$ and define*

$$z_r = e^{-2\kappa_r t_0}, \qquad \bar{z} = \frac{1}{m}\sum_r z_r, \qquad \kappa_{\mathrm{iso}} = -\frac{\log \bar{z}}{2t_0}.$$

Compare this heterogeneous regime with one having rate κ_{iso} in every mode. Their losses agree at t_0, but

$$\mathcal{L}_{\mathrm{het}}(t) > \mathcal{L}_{\mathrm{iso}}(t) \qquad (t > t_0).$$

Consequently, the isotropic regime reaches every target $0 < \varepsilon < \mathcal{L}(t_0)$ strictly earlier. At t_0, its trace-normalized activation spectrum is uniform, while that of the heterogeneous regime is nonuniform.

Proof. Since $0 < z_r < 1$, the isotropic rate is positive and $\mathcal{L}_{\mathrm{het}}(t_0) = m\bar{z}/2 = \mathcal{L}_{\mathrm{iso}}(t_0)$. For $t > t_0$, strict convexity of $x \mapsto x^{t/t_0}$ and the unequal z_r give

$$\mathcal{L}_{\mathrm{het}}(t) = \tfrac{1}{2}\sum_r z_r^{t/t_0} > \tfrac{m}{2}\bar{z}^{t/t_0} = \mathcal{L}_{\mathrm{iso}}(t).$$

Both losses decrease continuously to zero. At the isotropic crossing of any $0 < \varepsilon < \mathcal{L}(t_0)$, the heterogeneous loss is still larger than ε, so it crosses later. At t_0 the activation energies are $(1-\sqrt{z_r})^2$; these are unequal in the heterogeneous regime and equal and positive in the isotropic regime. □

This is an existence result about information hidden by loss. It does not establish a universal ranking rule from spectral concentration, or the empirical correlation advantage over early loss in Table 7.

Proposition 15 (Fast-band concentration and loss-crossing time)**.** *Take unit teacher coefficients, $1 \leq k < m$, rate $\bar{\kappa} > 0$ on modes $1, \ldots, k$, and a common rate $\kappa_s \in (0, \bar{\kappa})$ on the remaining modes. Fix $t_0 > 0$ and $0 < \varepsilon < m/2$. The fast-band fraction*

$$C^{(k)}(t_0; \kappa_s) = \frac{k(1-e^{-\bar{\kappa}t_0})^2}{k(1-e^{-\bar{\kappa}t_0})^2 + (m-k)(1-e^{-\kappa_s t_0})^2}$$

and the unique loss-crossing time $\tau_\varepsilon(\kappa_s)$ both decrease strictly with κ_s. Thus, within this family, a larger fast-band fraction orders a later loss crossing.

Proof. Set $A = k(1-e^{-\bar{\kappa}t_0})^2 > 0$ and $D = (m-k)(1-e^{-\kappa_s t_0})^2 > 0$. Since $D' = 2(m-k)t_0 e^{-\kappa_s t_0}(1-e^{-\kappa_s t_0}) > 0$, we have $\partial_{\kappa_s} C^{(k)} = -AD'/(A+D)^2 < 0$. The loss decreases continuously from $m/2$ to zero, so its crossing is the unique positive root of

$$F(t, \kappa_s) = \tfrac{1}{2}\{ke^{-2\bar{\kappa}t} + (m-k)e^{-2\kappa_s t}\} - \varepsilon.$$

At that root, $\partial_t F = -k\bar{\kappa}e^{-2\bar{\kappa}t} - (m-k)\kappa_s e^{-2\kappa_s t} < 0$ and $\partial_{\kappa_s} F = -(m-k)te^{-2\kappa_s t} < 0$. The implicit function theorem gives $d\tau_\varepsilon/d\kappa_s = -\partial_{\kappa_s} F/\partial_t F < 0$. □

Here early loss also decreases strictly with κ_s and therefore gives the same ordering. Moreover, interpreting t_0 as an early forecast requires $t_0 < \tau_\varepsilon$ for the regimes compared. Unlike Proposition 14, this family does not separate the predictive information in loss from that in the spectrum.

I.4.2 A POWER-LAW SPECTRUM AND FINITE-WINDOW SLOPES

Keep the zero-initialized linear model and fix a common teacher on m active modes. For a finite nonempty set of regimes $b \in \mathcal{B}$, assume

$$|\beta_r|^2 = Cr^{-p}, \qquad C > 0,\ p > 0, \tag{18}$$

$$\kappa_r(b) = \eta_b r^{-q_b}, \qquad \eta_b > 0,\ q_b \geq 0. \tag{19}$$

Teacher phases and mode order are shared across regimes. The activation energies are

$$\Lambda_r(b,t) = Cr^{-p}(1-e^{-\eta_b t r^{-q_b}})^2. \tag{20}$$

Write $x_{r,b}(t) = \eta_b t r^{-q_b}$. For $q_b > 0$, the crossover scale is

$$r_*(b,t) = (\eta_b t)^{1/q_b}. \tag{21}$$

The exponent of the unresolved-tail envelope is

$$\alpha_{\mathrm{tail}}^{\mathrm{model}}(b) = p + 2q_b. \tag{22}$$

We distinguish this envelope exponent from the slope estimated on a finite rank window.

Theorem 16 (Finite head, tail, and crossover bounds)**.** *Under* (18)–(19)*, fix b and $t > 0$. The $\Lambda_r(b,t)$ are the positive centered activation eigenvalues in strictly decreasing rank order. For $0 < \epsilon \le 1 \le h$:*

(i) If $x_{r,b}(t) \ge h$, then

$$(1 - e^{-h})^2 C r^{-p} \le \Lambda_r(b,t) \le C r^{-p}.$$

(ii) If $0 < x_{r,b}(t) \le \epsilon$, then

$$(1 - \epsilon/2)^2 C \eta_b^2 t^2 r^{-(p+2q_b)} \le \Lambda_r(b,t) \le C \eta_b^2 t^2 r^{-(p+2q_b)}.$$

(iii) If $\epsilon \le x_{r,b}(t) \le h$, then

$$(1 - e^{-\epsilon})^2 C r^{-p} \le \Lambda_r(b,t) \le (1 - e^{-h})^2 C r^{-p}.$$

For $q_b > 0$, the head and tail conditions are respectively $r \le r_* h^{-1/q_b}$ and $r \ge r_* \epsilon^{-1/q_b}$, intersected with $\{1, \dots, m\}$; either set can be empty. For $q_b = 0$, $\Lambda_r = C(1 - e^{-\eta_b t})^2 r^{-p}$ and there is no rank-dependent crossover.

Proof. The solution of Theorem 11 and Proposition 12 give (20). For $t > 0$, the first factor Cr^{-p} is positive and strictly decreasing, while the second is positive and nonincreasing. Their product is strictly decreasing, so the mode index is also the eigenvalue rank in this family.

Monotonicity of $1 - e^{-x}$ gives (i) and (iii). For $0 < x \le \epsilon \le 1$, integrating $1 - u \le e^{-u} \le 1$ over $[0, x]$ yields

$$x - x^2/2 \le 1 - e^{-x} \le x.$$

All terms are nonnegative, so $(1 - \epsilon/2)^2 x^2 \le (1 - e^{-x})^2 \le x^2$. Substitute $x = \eta_b t r^{-q_b}$ to obtain (ii). Solving the inequalities for r gives the crossover intervals when $q_b > 0$. When $q_b = 0$, the exponential factor is independent of rank. □

Completion of a relevant band. For $1 \le R \le m$ and $0 < \delta < 1$, define the time to learn every mode through R to relative coefficient error δ:

$$\tau_{R,\delta}(b) = \inf\{t \ge 0 : |a_r(b,t) - \beta_r| \le \delta |\beta_r| \text{ for all } 1 \le r \le R\}.$$

This endpoint differs from a crossing of the total scalar loss, which weights errors by the teacher energies.

Theorem 17 (Band-completion time)**.** *For the power-law family, the set in the definition of $\tau_{R,\delta}(b)$ is the closed ray $[\tau_{R,\delta}(b), \infty)$, where*

$$\tau_{R,\delta}(b) = \frac{\log(1/\delta)}{\eta_b} R^{q_b} = \frac{\log(1/\delta)}{\eta_b} R^{(\alpha_{\mathrm{tail}}^{\mathrm{model}}(b) - p)/2}. \tag{23}$$

The infimum is finite and attained, including $q_b = 0$.

Proof. Every active teacher coefficient is nonzero, and its residual is $-\beta_r e^{-\kappa_r(b)t}$. Thus the relative-error condition is equivalent to

$$e^{-\kappa_r(b)t} \le \delta \quad \Longleftrightarrow \quad t \ge \frac{\log(1/\delta)}{\kappa_r(b)}.$$

Taking absolute values makes this equivalence independent of the sign or complex phase of β_r. Intersecting the R rays takes the maximum of their endpoints. Since $q_b \ge 0$, the smallest rate is at $r = R$, giving (23). □

The exponent alone cannot order times because the prefactor η_b can vary. The next result removes this ambiguity by calibrating progress at one mode.

Corollary 18 (Ordering after matching head progress). *Fix $1 \le r_h < R \le m$, a common observation time $t_0 > 0$, and $\xi \in (0,1)$. Suppose $a_{r_h}(b,t_0) = \xi\beta_{r_h}$ for every regime. Then*

$$\tau_{R,\delta}(b) = \frac{\log(1/\delta)}{-\log(1-\xi)} t_0 \left(\frac{R}{r_h}\right)^{q_b}. \tag{24}$$

A smaller q_b, equivalently a smaller $\alpha_{\mathrm{tail}}^{\mathrm{model}}(b)$, gives a strictly smaller completion time. The sets minimizing these three quantities coincide, including ties.

Proof. The common anchor gives $1 - e^{-\eta_b t_0 r_h^{-q_b}} = \xi$, hence $\eta_b = -t_0^{-1} r_h^{q_b} \log(1-\xi)$. Substitute into (23). Its remaining prefactor is positive and common, while $R/r_h > 1$. The completion time and envelope exponent are therefore strictly increasing functions of q_b. □

An interior anchor does not order all scalar loss targets: smaller q_b raises rates above r_h but lowers them below it. If $r_h = 1$, it does order total loss. Indeed, the rates then have a common κ_1 and $\kappa_r = \kappa_1 r^{-q_b}$, so smaller q_b increases every rate with $r > 1$. The loss $\frac{1}{2}\sum_r |\beta_r|^2 e^{-2\kappa_r t}$ is consequently smaller at every $t > 0$ when the exponents differ and $m \ge 2$.

The fitted slope. For positive eigenvalues on a fixed window $I = \{R_1, \dots, R_2\}$ with $R_1 < R_2$, put $u_r = \log r$ and $\bar{u} = |I|^{-1}\sum_{r\in I} u_r$. The estimator used in the paper is

$$\widehat{\alpha}_I(b,t) = -\frac{\sum_{r\in I}(u_r - \bar{u})\log \Lambda_r(b,t)}{\sum_{r\in I}(u_r - \bar{u})^2}. \tag{25}$$

Trace normalization adds a constant to the logarithms and leaves this estimator unchanged.

Proposition 19 (Finite-window slope error). *For the power-law family at $t > 0$, let $g(x) = x/(e^x - 1)$. Then*

$$p + 2q_b g(x_{R_1,b}(t)) \le \widehat{\alpha}_I(b,t) \le p + 2q_b g(x_{R_2,b}(t)) \le p + 2q_b.$$

If $x_{R_1,b}(t) \le \epsilon \le 1$, then

$$0 \le \alpha_{\mathrm{tail}}^{\mathrm{model}}(b) - \widehat{\alpha}_I(b,t) \le 2q_b\epsilon.$$

For $q_b > 0$, the sufficient tail-window condition is $R_1 \ge r_*(b,t)\epsilon^{-1/q_b}$. A finite model need not have a window satisfying it. For $q_b = 0$, every fitted slope equals p.

Proof. Extend (20) to positive real ranks and differentiate with respect to $u = \log r$. With $x = \eta_b t e^{-q_b u}$,

$$-\frac{d}{du}\log \Lambda_{e^u}(b,t) = p + 2q_b \frac{x}{e^x - 1}.$$

The derivative of g has numerator $e^x(1-x) - 1$, which is zero at $x = 0$ and has derivative $-xe^x < 0$ for $x > 0$. Hence g decreases strictly, with $0 < g(x) < 1$. Since x is nonincreasing with rank, the negative logarithmic derivative lies between the stated endpoint bounds.

To pass from derivatives to the fitted slope, write $y_i = \log \Lambda_i$. The exact identity

$$\frac{\sum_i (u_i - \bar{u})y_i}{\sum_i (u_i - \bar{u})^2} = \frac{\sum_{i<j}(u_j - u_i)(y_j - y_i)}{\sum_{i<j}(u_j - u_i)^2}$$

follows by expanding the pairwise sums; each equals $|I|$ times the corresponding centered sum. The right side is a positive weighted average of secant slopes. The mean value theorem bounds each secant by the derivative extrema, proving the first inequality after negation.

Finally, $1 - e^{-x} \le x$ gives $g(x) \ge e^{-x} \ge 1 - x$, while $e^x - 1 \ge x$ gives $g(x) \le 1$. If $x_{R_1,b}(t) \le \epsilon$, all local slopes lie between $p + 2q_b(1-\epsilon)$ and $p + 2q_b$. The same holds for their fitted slope. Solving the condition for R_1 gives the stated window bound. □

This result explains why the location of the fitting window matters. A window that enters the learned head estimates a slope closer to p than to $p + 2q_b$. High log-log R^2 alone cannot locate the unresolved tail, since the learned head can also follow a power law. The empirical window procedure in Appendix G.4 assesses fit stability; it does not establish a known teacher spectrum or the common-anchor condition.

Corollary 20 (A sufficient margin with approximate head matching)**.** *Consider two regimes with common teacher and observation time t_0, and $R > r_h$. Set $\xi_b = a_{r_h}(b, t_0)/\beta_{r_h} \in (0, 1)$ and $s_b = -\log(1 - \xi_b)$. Suppose estimates $\widehat{\alpha}_b$ obey $|\widehat{\alpha}_b - \alpha_{\mathrm{tail}}^{\mathrm{model}}(b)| \le e_b$ and $|\log(s_1/s_2)| \le h$. Then*

$$\widehat{\alpha}_2 - \widehat{\alpha}_1 > e_1 + e_2 + \frac{2h}{\log(R/r_h)} \quad \Longrightarrow \quad \tau_{R,\delta}(1) < \tau_{R,\delta}(2).$$

Proof. Using each regime's own anchor in (23) gives

$$\log \frac{\tau_{R,\delta}(2)}{\tau_{R,\delta}(1)} = \log \frac{s_1}{s_2} + \frac{\alpha_{\mathrm{tail}}^{\mathrm{model}}(2) - \alpha_{\mathrm{tail}}^{\mathrm{model}}(1)}{2} \log \frac{R}{r_h}.$$

The right side is at least $-h + (\widehat{\alpha}_2 - \widehat{\alpha}_1 - e_1 - e_2)\log(R/r_h)/2$, which is positive under the stated condition. □

Proposition 19 bounds the deterministic fitting error in this construction. Sampling error in an empirical spectrum would require an additional bound. Likewise, an ordering in gradient-flow time transfers to tokens if a common increasing clock relates the two, for example $t = \nu N_{\mathrm{tok}}$ with the same $\nu > 0$ for all regimes. For a small-step update with step size γ_b and K_b tokens per update, a per-update drift proportional to γ_b becomes a per-token drift proportional to γ_b/K_b. Thus differing batches and schedules require an explicit calibration before analytic time can be read as token cost.

I.4.3 Constructed update fields and actual sample gradients

The linear observation map separates learned energy $A_r = |a_r|^2$ from the rate-weighted residual energy $U_r = |\kappa_r(\beta_r - a_r)|^2$. The following crossover gives a simple example of how their orderings differ.

Proposition 21 (Mode crossovers in the linear update field)**.** *Take the zero-initialized linear model with $m \ge 2$, unit active teacher coefficients, and $\kappa_1 > \cdots > \kappa_m > 0$. For every $i < j$, $A_i(t) > A_j(t)$ at every finite $t > 0$, while U_i and U_j cross once at*

$$t_{ij} = \frac{\log(\kappa_i/\kappa_j)}{\kappa_i - \kappa_j} > 0.$$

Before this time $U_i > U_j$, and afterward $U_i < U_j$. For $t > \max_{i<j} t_{ij}$, the activation order is $A_1 > \cdots > A_m$ and the update-field order is $U_m > \cdots > U_1$.

Proof. The energies are $A_r = (1 - e^{-\kappa_r t})^2$ and $U_r = \kappa_r^2 e^{-2\kappa_r t}$. The activation ordering follows from the strict ordering of the rates. For finite t all update energies are positive, and

$$\log \frac{U_i(t)}{U_j(t)} = 2\log(\kappa_i/\kappa_j) - 2(\kappa_i - \kappa_j)t.$$

This strictly decreasing affine function is positive at zero and vanishes exactly at t_{ij}. Taking the maximum of the finitely many crossing times gives the full reversed ordering. □

The vector field in this proposition is defined from coefficient velocities. To connect the empirical diagnostic to parameter gradients, we instead use the chain rule at the probed matrix. This is the same layerwise factorization used in curvature methods such as K-FAC (Martens & Grosse, 2015); the identities below require no independence approximation.

Proposition 22 (A sample-gradient moment depends on inputs and errors)**.** *For a single affine application $s = Wz$ and differentiable scalar loss ℓ, let $\delta = \nabla_s \ell$ and use columnwise vectorization. Then*

$$\nabla_W \ell = \delta z^\top, \qquad g = \mathrm{vec}(\nabla_W \ell) = z \otimes \delta.$$

Whenever the second moment exists,

$$\mathbb{E}[gg^\top] = \mathbb{E}[(zz^\top) \otimes (\delta\delta^\top)]. \tag{26}$$

If the same matrix is applied at positions $t = 1, \ldots, L$, let δ_t be the derivative of the full sample loss with respect to $s_t = Wz_t$. Then

$$g = \sum_t z_t \otimes \delta_t, \qquad \mathbb{E}[gg^\top] = \sum_{t,s} \mathbb{E}[(z_t z_s^\top) \otimes (\delta_t \delta_s^\top)]. \tag{27}$$

Proof. The differential is $d\ell = \delta^\top (dW) z = \langle \delta z^\top, dW \rangle_F$. Vectorizing the outer product gives $z \otimes \delta$, whose outer product is $(zz^\top) \otimes (\delta\delta^\top)$. For a shared parameter, the chain rule sums these contributions over its applications. Expanding the outer product of the sum retains every pair of positions and gives (27). □

For the sequence-mean loss used in Section 2, the averaging factor is already included in δ_t. The input z_t to W_O is the concatenated attention output; it is distinct from the residual-stream activation whose covariance is measured separately. Equation (26) also depends on how inputs and errors co-vary. Factoring its expectation into $\mathbb{E}[zz^\top] \otimes \mathbb{E}[\delta\delta^\top]$ would require an additional assumption.

Proposition 23 (Equal activation covariance and loss do not fix gradient spectra)**.** *Let x_1, x_2 be independent uniform signs, take $y = 0$, $f_w(x) = w^\top x$, and $\ell_w(x) = \frac{1}{2}(w^\top x)^2$. Use x as the representation and compare $w^{(1)} = (1,0)^\top$ with $w^{(2)} = (1,1)^\top/\sqrt{2}$. Both have representation covariance I_2 and population loss $1/2$, but their actual sample-gradient second moments are*

$$\mathbb{E}[g_1 g_1^\top] = I_2, \qquad \mathbb{E}[g_2 g_2^\top] = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}.$$

Their eigenvalues are respectively $(1,1)$ and $(2,0)$.

Proof. Since $\mathbb{E}x = 0$ and $\mathbb{E}[xx^\top] = I_2$, the population loss is $\|w\|^2/2 = 1/2$ for both choices. The sample gradient is $g = (w^\top x)x$. For $w^{(1)}$, $g_1 = (1, x_1 x_2)^\top$, whose second moment is I_2. For $w^{(2)}$, $g_2 = (1 + x_1 x_2)(1,1)^\top/\sqrt{2}$. Since $\mathbb{E}(1 + x_1 x_2)^2 = 2$, its second moment is the all-ones matrix. The eigenvalues follow directly. □

This example establishes complementarity for the uncentered sample-gradient moment actually probed in the paper. It does not order learning speed by gradient concentration. For the finite sample matrix $G \in \mathbb{R}^{M \times P}$, its nonzero singular values satisfy

$$\lambda_j(G^\top G/M) = \sigma_j(G)^2/M. \tag{28}$$

Thus a slope fitted to these second-moment eigenvalues is twice the slope fitted to the singular values on the same positive rank window. Neither the constructed M_{upd} nor an optimizer's matrix update should be substituted for G. The observed-label second moment also need not equal the Fisher matrix formed using labels sampled from the model.

I.4.4 Two-layer Fourier factor model

The linear observation map makes mode learning explicit but does not derive hidden-feature learning. We now give the hidden coordinates trainable amplitudes, using a diagonal factor model of the type studied in deep linear learning dynamics (Saxe et al., 2014). Let $\phi_1, \ldots, \phi_m$ be real, mean-zero, orthonormal Fourier basis functions under uniform phase measure, with $1 \le m \le c - 1$. Define

$$f_t^{\mathrm{fac}}(z) = \sum_r u_r(t) v_r(t) \phi_r(z), \qquad f^{\mathrm{fac},\star}(z) = \sum_r \beta_r \phi_r(z), \qquad \beta_r \ge 0.$$

The hidden coordinates are $h_r^{\mathrm{fac}}(z) = u_r \phi_r(z)$ and the population squared loss is

$$\mathcal{J}(u, v) = \tfrac{1}{2} \sum_r (u_r v_r - \beta_r)^2.$$

This model uses Euclidean gradient flow on both factors, initialized with $u_r(0) = v_r(0) > 0$. Its teacher and initialization are separate from those of the linear model. In particular, setting both factors to zero would leave them at zero.

Theorem 24 (Balanced factor dynamics and hidden-feature energy)**.** *Under the preceding assumptions, a unique solution exists for every $t \ge 0$, with $u_r(t) = v_r(t) > 0$. The product $m_r = u_r v_r = u_r^2$ equals the variance of the hidden coordinate h_r^{fac}, and*

$$\dot{m}_r = 2 m_r (\beta_r - m_r).$$

For $\beta_r > 0$ and any $m_r(0) > 0$,

$$m_r(t) = \frac{\beta_r}{1 + (\beta_r/m_r(0) - 1) e^{-2\beta_r t}};$$

for $\beta_r = 0$,

$$m_r(t) = \frac{m_r(0)}{1 + 2m_r(0)t}.$$

Proof. Differentiating the loss gives $\dot{u}_r = -(u_r v_r - \beta_r)v_r$ and $\dot{v}_r = -(u_r v_r - \beta_r)u_r$. The polynomial vector field is locally Lipschitz. On its maximal solution, $d_r = u_r - v_r$ satisfies $\dot{d}_r = (u_r v_r - \beta_r)d_r$ with $d_r(0) = 0$, so $u_r = v_r =: s_r$. Hence $\dot{s}_r = s_r(\beta_r - s_r^2)$.

Uniqueness and the sign of this scalar vector field imply $s_r(t) \le \max\{s_r(0), \sqrt{\beta_r}\}$. Positivity on each finite interval follows from

$$s_r(t) = s_r(0) \exp\left(\int_0^t (\beta_r - s_r(v)^2)\, dv \right) > 0.$$

These bounds exclude finite-time escape and extend the solution globally. Now $m_r = s_r^2$ gives the asserted differential equation. Set $z_r = 1/m_r$ to obtain $\dot{z}_r = 2 - 2\beta_r z_r$. For $\beta_r > 0$,

$$z_r(t) = \frac{1}{\beta_r} + \left(\frac{1}{m_r(0)} - \frac{1}{\beta_r} \right) e^{-2\beta_r t}.$$

This is a positive convex combination of $1/m_r(0)$ and $1/\beta_r$. Taking reciprocals gives the formula, including equilibrium and above-target initializations. When $\beta_r = 0$, integration gives $z_r(t) = 1/m_r(0) + 2t$.

Finally, mean-zero orthonormality gives $\mathbb{E}h_r^{\mathrm{fac}} = 0$ and $\mathbb{E}[h_r^{\mathrm{fac}} h_s^{\mathrm{fac}}] = u_r u_s \mathbf{1}_{r=s}$. The hidden covariance is diagonal with entries $u_r^2 = m_r$. Its energy is therefore m_r, whereas the linear observation map had energy $|a_r|^2$. □

Corollary 25 (Increasing concentration on a fixed task support)**.** *In the factor model, let S be a nonempty proper subset of $\{1, \dots, m\}$. Assume $\beta_r > 0$ and $0 < m_r(0) < \beta_r$ for $r \in S$, and $\beta_r = 0$ for $r \notin S$. Then*

$$H_S(t) = \frac{\sum_{r \in S} m_r(t)}{\sum_{r=1}^m m_r(t)}$$

is defined and strictly increasing at every finite $t \ge 0$, and converges to one as $t \to \infty$.

Proof. On S, the solution formula gives $0 < m_r(t) < \beta_r$ at finite times, hence $\dot{m}_r > 0$. Off S, $m_r(t) > 0$ and $\dot{m}_r = -2m_r^2 < 0$. Thus $A = \sum_{r \in S} m_r$ and $B_{\mathrm{off}} = \sum_{r \notin S} m_r$ satisfy $A, B_{\mathrm{off}} > 0$, $A' > 0$, and $B'_{\mathrm{off}} < 0$. Differentiation gives

$$H'_S(t) = \frac{A' B_{\mathrm{off}} - A B'_{\mathrm{off}}}{(A + B_{\mathrm{off}})^2} > 0.$$

The explicit solutions converge to β_r on S and zero off S, proving the limit. □

The support is fixed by the teacher. A peak selected from each measured spectrum is a different statistic, and above-target initialization can make supported modes decrease. Concentration across supported and unsupported modes also differs from concentration within the supported set: useful modes can acquire a larger share of total energy while that share spreads across more useful directions. This is why task-support concentration and unresolved-tail flattening need not conflict. Sorted covariance eigenvalues alone do not identify which directions belong to S.

I.4.5 Geometric properties of the three interventions

The following results concern individual components: relative-position attention scores, fixed-embedding linear readouts, and exact polar updates. They help interpret what a recipe changes without assigning its observed gain to a single cause.

RoPE and relative positions. Let $X = (X_i)_{i\in\mathbb{Z}}$ be a bi-infinite sequence with $X_i \in \mathbb{R}^{d_{\text{in}}}$, and define $(\mathsf{S}_\tau X)_i = X_{i-\tau}$. For fixed projections $W_q, W_k \in \mathbb{R}^{D_q\times d_{\text{in}}}$, set $q_i = W_q X_i$ and $k_i = W_k X_i$. Suppose $R_i \in \mathbb{R}^{D_q\times D_q}$ are orthogonal, $R_0 = I$, and $R_{i+j} = R_i R_j$. Standard rotary blocks satisfy this representation law (Su et al., 2024). The score is

$$s_{ij}^{\text{rot}}(X) = \langle R_i q_i(X), R_j k_j(X)\rangle.$$

Theorem 26 (Relative-offset dependence of rotary scores)**.** *Under these assumptions,*

$$s_{ij}^{\text{rot}}(X) = q_i(X)^\top R_{j-i} k_j(X), \qquad s_{i+\tau,j+\tau}^{\text{rot}}(\mathsf{S}_\tau X) = s_{ij}^{\text{rot}}(X)$$

for every sequence, pair of indices, and shift τ.

Proof. The group law and orthogonality imply $R_i^\top = R_{-i}$ and $R_i^\top R_j = R_{j-i}$, giving the first equality. The shifted content satisfies $q_{i+\tau}(\mathsf{S}_\tau X) = q_i(X)$ and $k_{j+\tau}(\mathsf{S}_\tau X) = k_j(X)$, while the relative offset remains $j - i$. This gives the second equality. □

Proposition 27 (Absolute scores and a necessary equivariance condition)**.** *Let W_q and W_k have full row rank and let $p_i \in \mathbb{R}^{d_{\text{in}}}$ be an absolute positional table. If*

$$s_{ij}^{\text{abs}}(X) = \langle W_q(X_i + p_i), W_k(X_j + p_j)\rangle$$

satisfies $s_{i+\tau,j+\tau}^{\text{abs}}(\mathsf{S}_\tau X) = s_{ij}^{\text{abs}}(X)$ for every sequence, pair of indices, and shift, then both $W_q p_i$ and $W_k p_i$ are constant in i. A nonconstant projected table therefore violates this universal equivariance for some contents and indices.

Proof. Fix $i \neq j$ and τ, and vary $X_i = x$ and $X_j = y$ independently. Expanding the score identity gives

$$0 = \langle W_q x, W_k(p_{j+\tau} - p_j)\rangle + \langle W_q(p_{i+\tau} - p_i), W_k y\rangle + c_{ij\tau},$$

where the last term is independent of x, y. Setting both to zero gives $c_{ij\tau} = 0$. Set $y = 0$ and vary x. Surjectivity of W_q then forces $W_k(p_{j+\tau} - p_j) = 0$. For every j, τ one can choose $i \neq j$, so $W_k p_j$ is constant. Reversing the roles and using surjectivity of W_k proves constancy of $W_q p_i$. □

These are statements about shifts of sequence positions and content-conditioned scores. The cyclic task shifts $\mathcal{S}_q$ act on latent phase instead. Identifying the two group actions would require an additional argument; the score result alone does not prove equivariance or faster learning for a finite, masked transformer.

Untying a readout with the embedding fixed. Fix $E \in \mathbb{R}^{D\times V}$ of full row rank, with $1 \le D < V$. Compare the linear token-to-logit classes

$$\mathcal{F}_{\text{tied}}(E) = \{E^\top A E : A \in \mathbb{R}^{D\times D}\}, \qquad \mathcal{F}_{\text{untied}}(E) = \{W^\top A E : A \in \mathbb{R}^{D\times D},\ W \in \mathbb{R}^{D\times V}\}.$$

The embedding is held fixed in both classes.

Theorem 28 (An untied linear readout removes an output-subspace constraint)**.** *For the fixed embedding above:*

(i) $\mathcal{F}_{\text{untied}}(E) = \{C_{\text{out}} E : C_{\text{out}} \in \mathbb{R}^{V\times D}\}$.

(ii) $\mathcal{F}_{\text{tied}}(E) \subsetneq \mathcal{F}_{\text{untied}}(E)$.

(iii) Every tied map has image in $\operatorname{col}(E^\top)$ *and kernel containing* $\ker E$.

(iv) Every untied map still has kernel containing $\ker E$*, but its image can be any prescribed subspace of* $\mathbb{R}^V$ *of dimension at most* D.

Proof. Writing $C_{\text{out}} = W^\top A$ gives one inclusion in (i); choosing $A = I_D$ and $W = C_{\text{out}}^\top$ gives the other. Every tied map is untied by choosing $W = E$. To show strictness, choose $u \notin \operatorname{col}(E^\top)$ and $z \in \mathbb{R}^D \setminus \{0\}$. Full row rank gives $z^\top E \neq 0$, so the untied map $uz^\top E$ has image exactly $\operatorname{span}\{u\}$ and cannot be tied. This proves (ii).

The left factor $E^\top$ places the image of each tied map in $\operatorname{col}(E^\top)$, and the right factor E gives the kernel condition for both classes. Finally, $R_E = E^\top(EE^\top)^{-1}$ is a right inverse. Since $C_{\text{out}} = (C_{\text{out}}E)R_E$, the column spaces of C_{out} and $C_{\text{out}}E$ coincide. Choosing the columns of C_{out} to span any subspace of dimension at most D proves (iv), including the zero subspace. □

Corollary 29 (An output-subspace lower bound)**.** *Let P_{out} be the orthogonal projector onto* $\mathrm{col}(E^\top)$. *For any target $A_\star \in \mathbb{R}^{V\times V}$,*

$$\inf_{A_{\mathrm{eff}}\in\mathcal{F}_{\mathrm{tied}}(E)} \|A_{\mathrm{eff}} - A_\star\|_F \geq \|(I - P_{\mathrm{out}})A_\star\|_F.$$

If $A_\star = C_{\mathrm{out}}E$ for the same fixed E, the untied class realizes it exactly.

Proof. For every tied map, $P_{\mathrm{out}}A_{\mathrm{eff}} = A_{\mathrm{eff}}$. Orthogonality of the projected column spaces gives

$$\|A_{\mathrm{eff}} - A_\star\|_F^2 = \|P_{\mathrm{out}}(A_{\mathrm{eff}} - A_\star)\|_F^2 + \|(I - P_{\mathrm{out}})A_\star\|_F^2.$$

Discard the first term, take square roots, and then take the infimum. The final assertion follows from Theorem 28(i). □

This bound compares linear maps at a fixed embedding. It does not hold with the same fixed projector after optimizing the embedding, and it does not quantify cross-entropy or training speed. Its role is to identify a representational constraint that tying can impose.

Exact polar descent and the Muon connection. For a nonzero matrix $D_W \in \mathbb{R}^{d_1\times d_2}$ with compact SVD $D_W = U_r\Sigma_r V_r^\top$ and rank r, define

$$Q(D_W) = U_r V_r^\top = D_W\big((D_W^\top D_W)^{1/2}\big)^\dagger.$$

Zero singular directions remain zero; this definition includes no arbitrary orthogonal completion. The optimization identity below is the operator-norm steepest-descent property discussed by Bernstein & Newhouse (2024).

Theorem 30 (Exact polar descent in an operator-norm ball)**.** *For the nonzero matrix above, $Q(aD_W) = Q(D_W)$ for every scalar $a > 0$, and*

$$\max_{\|M\|_{\mathrm{op}}\leq 1} \langle D_W, M\rangle_F = \|D_W\|_*,$$

with the maximum attained by $M = Q(D_W)$. Equivalently, $-\eta Q(D_W)$ minimizes $\langle D_W, \Delta W\rangle_F$ over $\|\Delta W\|_{\mathrm{op}} \leq \eta$.

If $D_W = \nabla F(W)$ for an objective with L_{sm}-Lipschitz gradient in Frobenius norm, $L_{\mathrm{sm}} > 0$, then $W^+ = W - \eta Q(D_W)$ satisfies

$$F(W^+) \leq F(W) - \eta\|D_W\|_* + \frac{L_{\mathrm{sm}}\eta^2 r}{2}.$$

In particular, $0 < \eta < 2\|D_W\|_*/(L_{\mathrm{sm}}r)$ gives strict descent.

Proof. Positive scaling changes the singular values but leaves the compact polar factor unchanged. For any feasible M, write $D_W = \sum_{i=1}^r \sigma_i u_i v_i^\top$ and use $|u_i^\top M v_i| \leq 1$ to obtain

$$\langle D_W, M\rangle_F = \sum_i \sigma_i u_i^\top M v_i \leq \sum_i \sigma_i = \|D_W\|_*.$$

The matrix $Q(D_W)$ has operator norm one and maps v_i to u_i, attaining equality. Substituting $M = -\Delta W/\eta$ proves the equivalent minimum. Uniqueness is not needed.

For any increment Z, integrate the gradient along the segment from W to $W + Z$:

$$\begin{aligned} F(W+Z) - F(W) - \langle \nabla F(W), Z\rangle_F &= \int_0^1 \langle \nabla F(W+sZ) - \nabla F(W), Z\rangle_F \, ds \\ &\leq \tfrac{1}{2}L_{\mathrm{sm}}\|Z\|_F^2. \end{aligned}$$

Take $Z = -\eta Q(D_W)$. Its squared Frobenius norm is $\eta^2 r$ and its inner product with the gradient is $-\eta\|D_W\|_*$. The stated inequality and strict step-size condition follow. □

Practical Muon applies an approximate orthogonalization to a matrix momentum state, with additional scaling. The theorem isolates the exact polar geometry; its descent conclusion requires the actual gradient and does not automatically apply to those implementation choices. The matrix D_W is also different from the stacked per-sample matrix G of Section 2.

Rank changes and polar continuity. Small perturbations of a gradient need not yield small perturbations of its compact polar factor when rank changes. For example,

$$D_\epsilon = \mathrm{diag}(1, \epsilon) \longrightarrow D_0 = \mathrm{diag}(1, 0), \qquad Q(D_\epsilon) = I_2 \quad (\epsilon > 0), \qquad Q(D_0) = \mathrm{diag}(1, 0).$$

The polar factors stay distance one apart in operator norm. Thus decay of noise singular values alone does not justify convergence to the population polar factor.

Continuity does hold along a convergent sequence $D_n \to D$ of the same nonzero rank if all retained singular values are bounded below by a common $\gamma > 0$. To see this, the spectra of $D_n^\top D_n$ and $D^\top D$ lie in $\{0\} \cup [\gamma^2, M]$ for a finite M. Choose a continuous function f on $[0, M]$ with $f(0) = 0$ and $f(x) = x^{-1/2}$ on $[\gamma^2, M]$. Then $Q(D_n) = D_n f(D_n^\top D_n)$ and likewise for D. Uniform polynomial approximation of f, the spectral norm bound on its approximation error, and continuity of matrix polynomials imply $Q(D_n) \to Q(D)$. This conditional statement supplies the qualification used in Appendix E; the required rank condition has not been established for the empirical optimizer trajectories.

J Additional Plots

This section collects supporting plots that audit the claims of the main paper but were too large to include there. RankMe is the entropy effective rank defined in Section C, used here as a compact trajectory summary alongside tail exponents and raw spectra.

J.1 Loss atlases

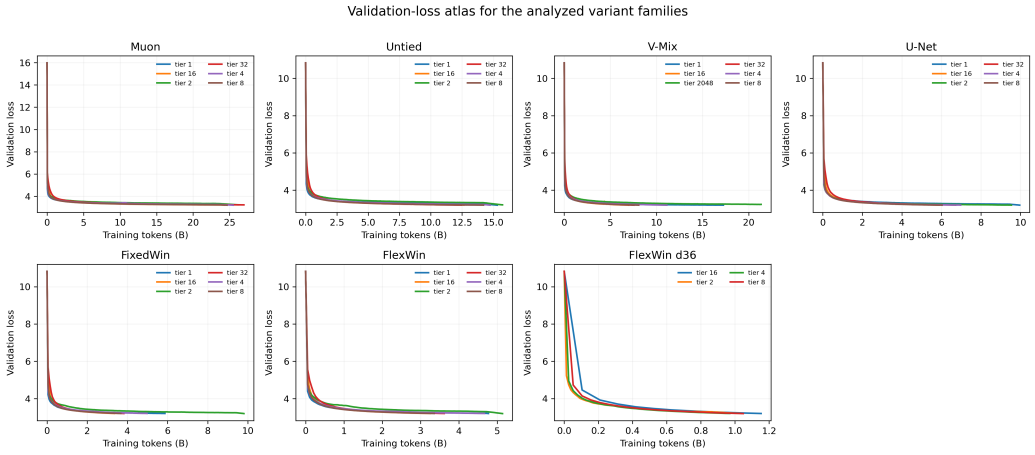
Fig. 13 shows the loss-curve evidence behind every matched-loss comparison in the main paper. Panel (a) collects test-loss trajectories for families with stored validation checkpoints; panel (b) collects train-loss trajectories for the late-trunk and d36 support runs where validation checkpoints were unavailable. Two features matter. First, every batch tier within every family reaches the target loss, confirming that matched-loss comparisons are not selecting only the easiest tiers. Second, the spread between tiers in tokens-to-target is visible directly: smaller batches sit on shallower learning curves and require substantially more tokens than the family-local efficient regime, which sits near the modal $B = 8$ tier in most variants.

J.2 Spectral atlases for individual models

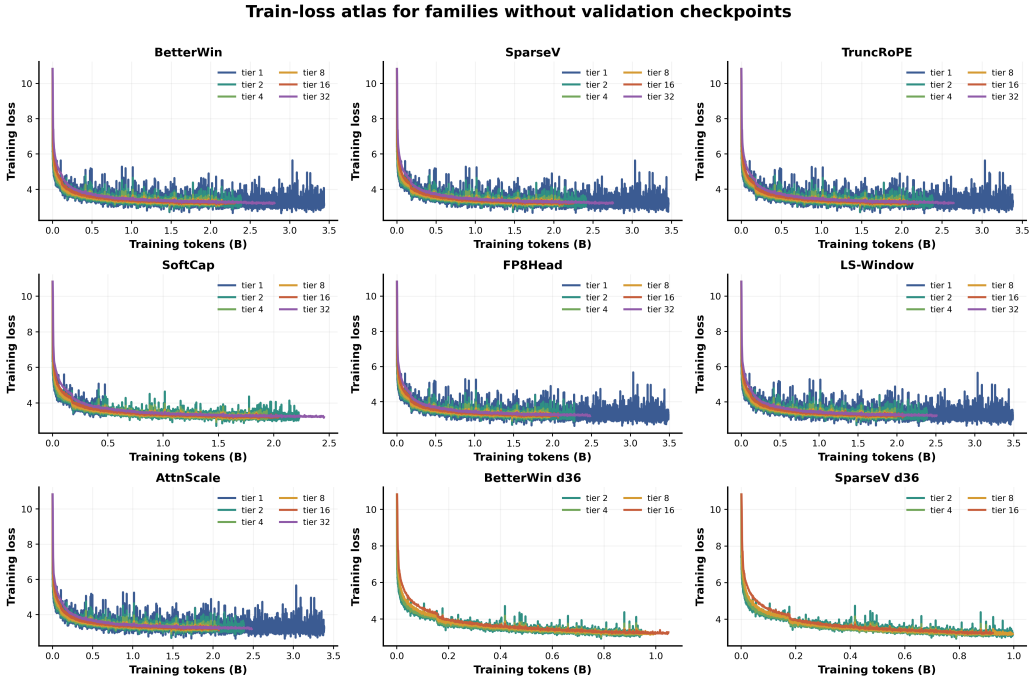
Figs. 14–16 provide the per-variant raw evidence underlying the transition-level taxonomy of Section 4. Each row shows one variant in three views: trace-normalized activation covariance spectra, gradient SVD spectra, and summary trajectories of RankMe and α_{tail} over training. The legacy prefix atlas (Fig. 14) contains the largest geometric shifts in the chain, spanning both the early gradient-dominant and activation-dominant transitions of Table 2. The matched-trunk atlases (Figs. 15–16) show the smaller incremental changes from ValueMix through AttnScale, where the transition labels are best inferred from the joint activation–gradient summaries rather than from any single raw spectrum alone. The tier-2 atlas (Fig. 17) applies the same format to the larger-scale support runs and shows that the qualitative spectral signatures persist at scale.

J.3 Weight-matrix spectra

Activation and gradient spectra describe the data-side and update-side of training. Fig. 18 adds a third view: the spectra of the trained weight matrices themselves, comparing the layer-11 attention-output projection W_O against the layer-11 MLP-output projection at the final checkpoint, with head exponents shown at step 1600 and at the end of training. The MLP-output projection shows clearer cross-variant divergence than W_O in both raw spectrum shape and head-exponent trajectories. The asymmetry is consistent with the gradient-probe stability argument in Appendix C: W_O retains the same architectural role across the full chain, while the feed-forward writeback path is touched by several trunk-side interventions (ValueMix, U-Net, BetterWin, SparseV). Either probe is informative, but they answer different questions and should not be expected to coincide.



(a) Test-loss atlas for families with validation checkpoints.



(b) Train-loss atlas for d36 support families without validation checkpoints.

Figure 13: **Loss-curve atlases.** Validation-aligned and train-loss-only evidence are separated to keep the support runs distinct from the main protocol. Within every family, all batch tiers reach the target loss, and the tokens-to-target spread across tiers is visible directly in the curves.

J.4 Training dynamics and phase visibility

Prior geometry work has reported a collapse–expansion–compression phase sequence in representation rank during training. Fig. 19 plots RankMe trajectories on normalized training progress for FixedWin, SparseV, and LSWA across all batch tiers. The classical phase shape is most clearly visible in intermediate tiers; very small and very large batches often produce monotone or muted trajectories without a sharply localized expansion peak. We therefore treat phase-like dynamics as secondary qualitative evidence rather than a universal training signature, since their visibility depends on both batch tier and variant.

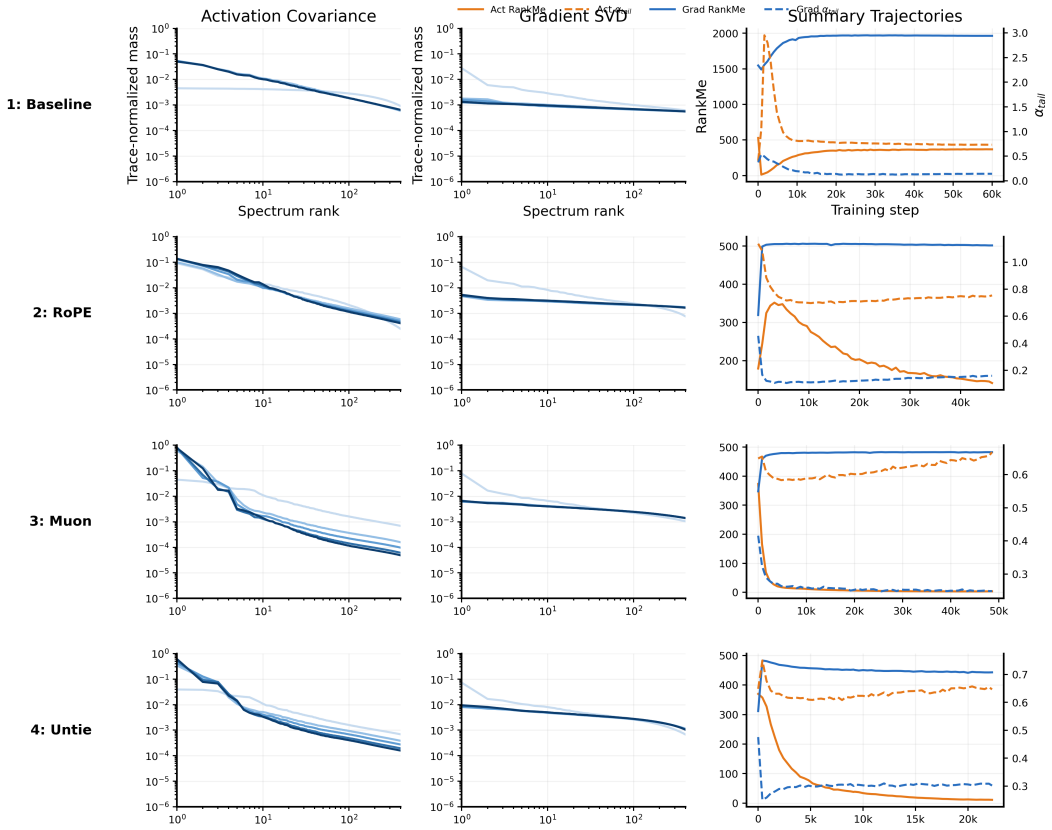


Figure 14: **Spectral atlas for the legacy prefix variants.** Rows show Baseline, RoPE, Muon, and Untied. Each consecutive variant produces visibly distinct activation and gradient spectra, dominating the activation-dominant rows of Table 2.

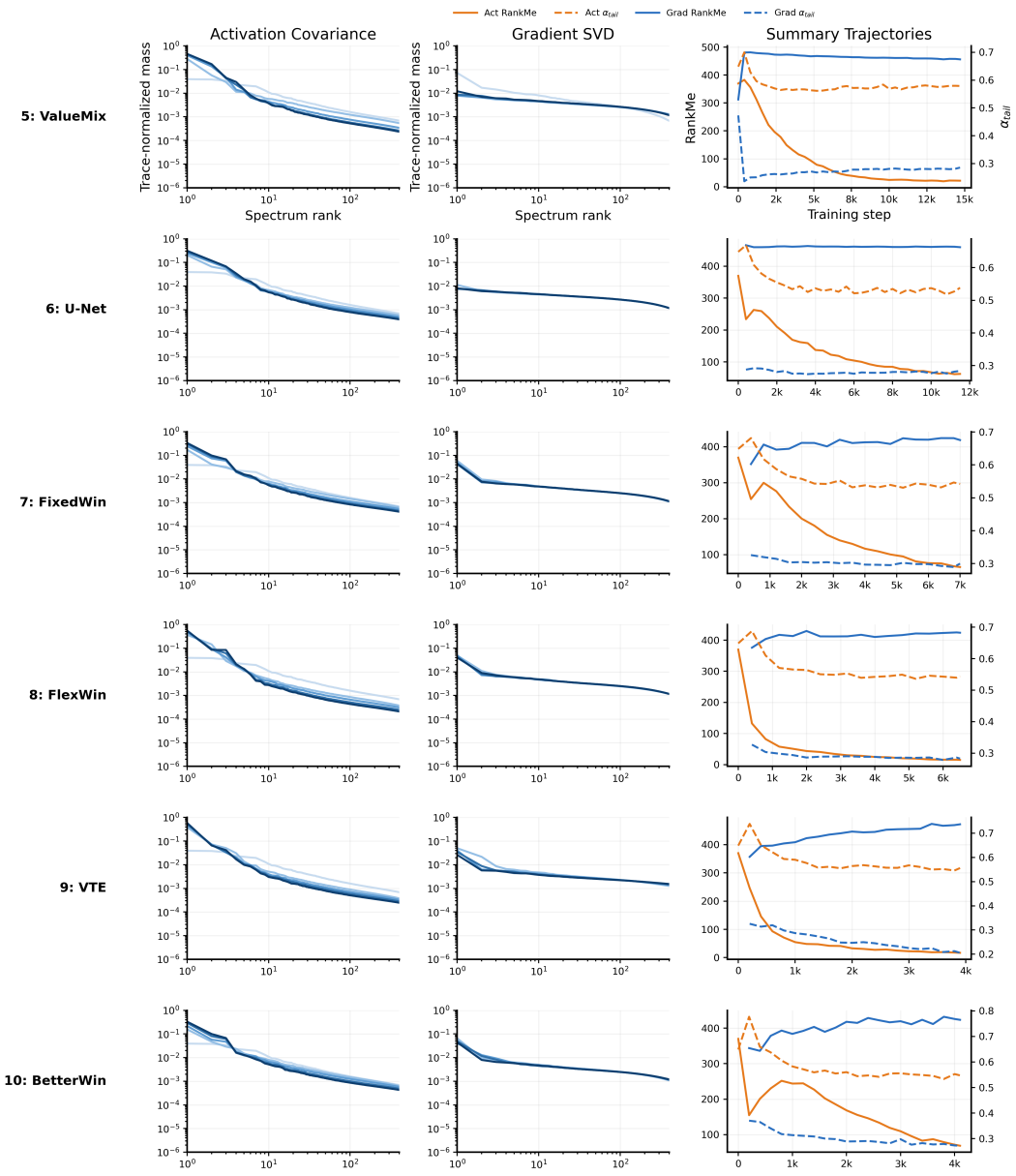


Figure 15: **Spectral atlas for the first half of the matched trunk.** Rows show ValueMix, U-Net, FixedWin, FlexWin, VTE, and BetterWin. Per-row spectral differences are smaller than across the legacy prefix; the taxonomic split is best read off the joint activation–gradient summary trajectories.

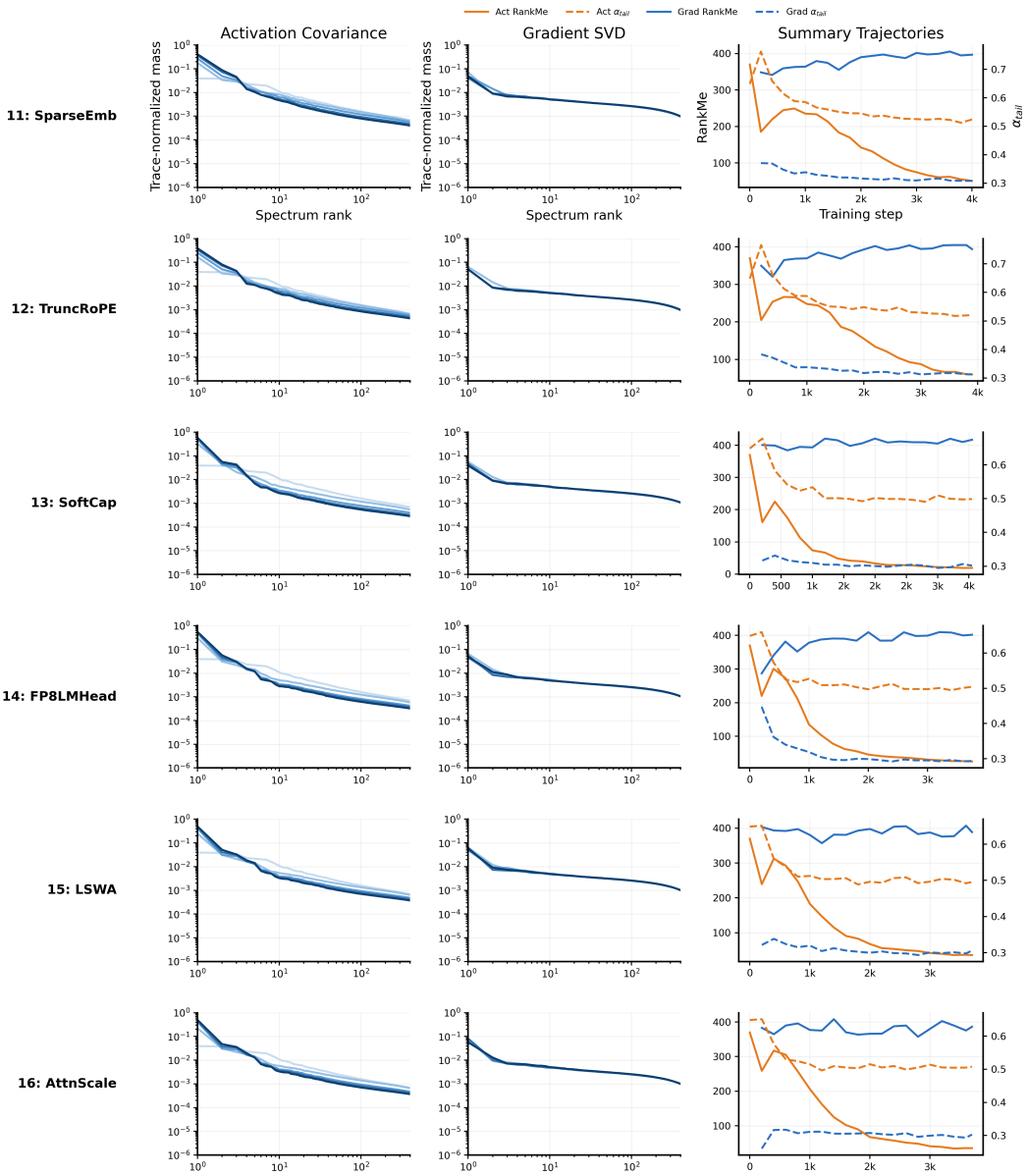


Figure 16: **Spectral atlas for the second half of the matched trunk.** Rows show SparseV, TruncRoPE, SoftCap, FP8Head, LSWA, and AttnScale. The execution-dominant variants (FP8Head, LSWA, AttnScale) show smaller activation-side shifts than the earlier trunk variants, consistent with the Section 4 taxonomy.

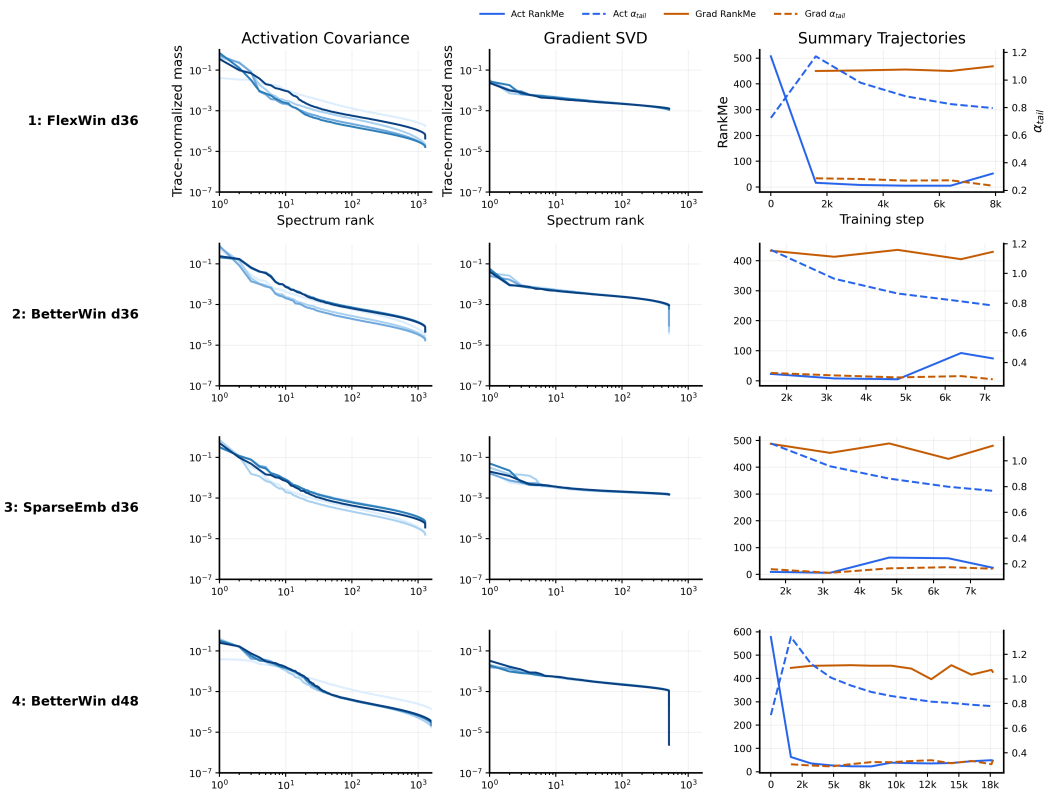
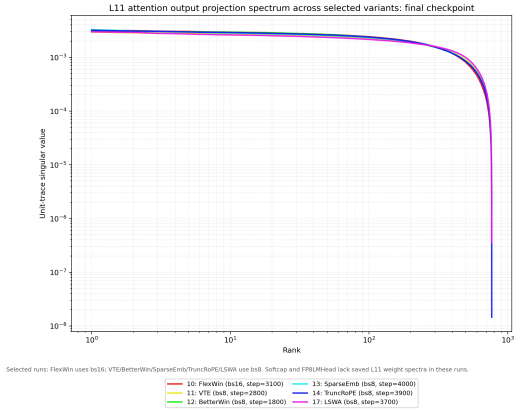
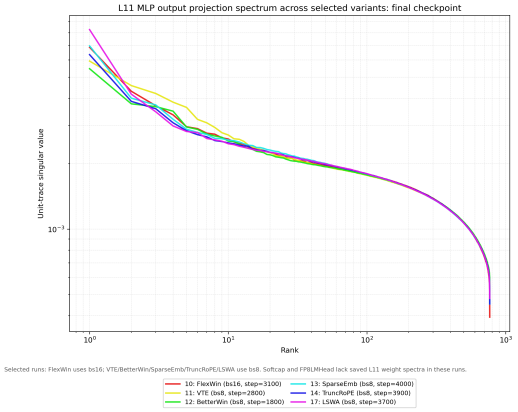


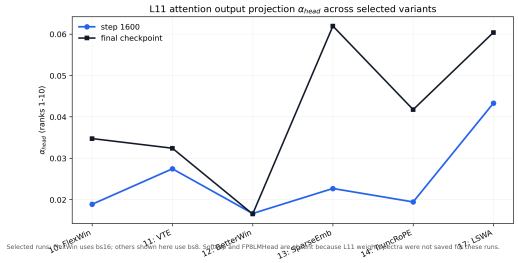
Figure 17: **Tier-2 spectral atlas for the d36/d48 scale follow-up.** Activation covariance, gradient spectra, RankMe, and tail-exponent trajectories for FlexWin d36, BetterWin d36, SparseV d36, and BetterWin d48. The qualitative spectral signatures match the corresponding d12 variants, supporting the cross-scale claim of Section 3



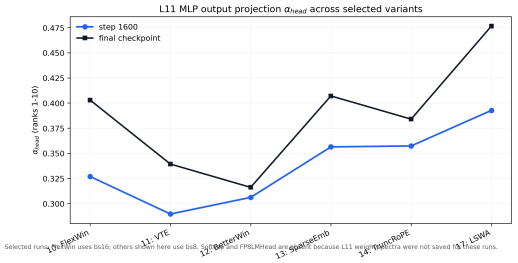
(a) Layer-11 attention-output weight spectrum.



(b) Layer-11 MLP-output weight spectrum.

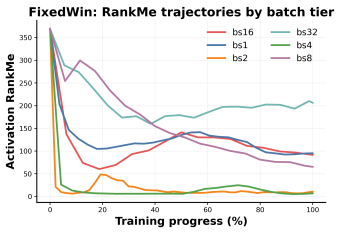


(c) Attention-output head exponent.

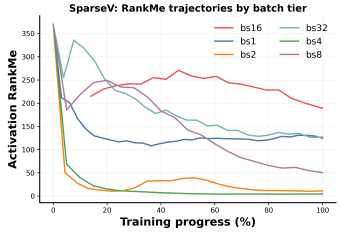


(d) MLP-output head exponent.

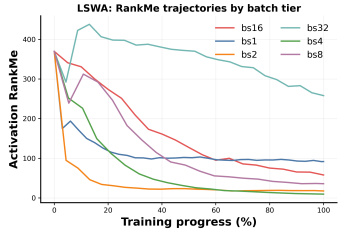
Figure 18: **Weight spectra are informative but tensor-dependent.** Layer-11 attention-output and MLP-output projections at the final checkpoint, plus their head exponents at step 1600 and at the end of training. The MLP-output projection shows clearer parameter-side divergence across variants, consistent with W_O's stable architectural role and the additional cross-variant variance accumulated by the feed-forward writeback path.



(a) FixedWin.



(b) SparseV.



(c) LSWA.

Figure 19: **Phase-like RankMe trajectories are batch-regime dependent.** Collapse–expansion–compression behavior is not uniform across batch size or variant; the phase sequence reported in prior geometry work appears most clearly in intermediate tiers, so we treat it as qualitative support rather than a universal law.